EIR (Registro de Identidade de Equipamento)

Visão Geral

O HSS inclui um EIR (Registro de Identidade de Equipamento) embutido que fornece verificação de identidade de equipamentos para dispositivos móveis. O EIR valida números IMEI (Identidade Internacional de Equipamento Móvel) para determinar se o equipamento móvel está autorizado, foi roubado ou está sob observação antes de permitir o acesso à rede.

Principais Capacidades

- **Interface S13**: Verificação de identidade do equipamento via protocolo Diameter
- **Validação de IMEI**: Verificar a identidade do equipamento usando IMEI/IMEISV
- **Correspondência Flexível**: Correspondência de padrões baseada em regex para IMEI, IMEISV e IMSI
- **Classificação em Três Níveis**: Suporte a lista branca, lista negra e lista cinza
- **Avaliação Ordenada**: As regras são avaliadas em ordem de precedência ascendente, e uma nova regra pode ser inserida em qualquer posição
- **Políticas Configuráveis**: Comportamento personalizável para equipamentos desconhecidos
- **REST API**: Operações completas de CRUD para gerenciamento de regras EIR

Arquitetura

Interface Diameter

Interface	ID da Aplicação	Par	Propósito
S13	16.777.252	MME/SGSN	Verificação de identidade do equipamento

Banco de Dados de Regras de Equipamento

O EIR utiliza um sistema de correspondência baseado em regras flexível:

EIR_RULE		
uuid	id	PK
string	action	
string	regex	
int	precedence	
timestamp	inserted_at	
timestamp	updated_at	

Ações de Regra:

- `whitelist` - Permitir equipamento
- `blacklist` - Bloquear equipamento
- `greylist` - Monitorar equipamento

Padrões Regex: Correspondência contra IMEI, IMEISV ou IMSI

Valores de Status do Equipamento

Status	Código	Significado	Ação na Rede
Whitelist	0	Equipamento aprovado	Permitir acesso à rede
Blacklist	1	Equipamento roubado/bloqueado	Negar acesso à rede
Greylist	2	Equipamento sob observação	Permitir com monitoramento

Interface S13

Operações Suportadas

Solicitação de Verificação de Identidade do Equipamento (ECR) / Resposta de Verificação de Identidade do Equipamento (ECA)

Direção: MME/SGSN → HSS (EIR)

Gatilho: MME verifica a identidade do equipamento durante o anexo ou atualização da área de rastreamento

AVPs de Solicitação:

- Session-Id
- Origin-Host, Origin-Realm
- Destination-Realm
- Auth-Session-State
- Terminal-Information
  - IMEI (14 dígitos conforme transportado no AVP IMEI S13; concatenado com Software-Version forma o IMEISV de 16 dígitos. O comprimento não é validado)
  - Software-Version (2 dígitos, opcional)
- User-Name (IMSI, opcional)

- Vendor-Specific-Application-Id

Ações do EIR:

1. Extrair IMEI, Software-Version (se presente) e IMSI (se presente)
2. Se IMSI fornecido:
   - Validar se o assinante existe e está habilitado
   - Atualizar o estado do assinante com informações da última vez vista (`last_seen_realm`, `last_received_diameter_message_type`, `last_seen_imei`, `last_seen_imei_software_version`, `last_seen_imei_timestamp`). A identidade do equipamento é registrada independentemente do status do equipamento retornado.
3. Tentar busca de equipamento em ordem de prioridade:
   - **Correspondência IMEISV** (IMEI + Software-Version concatenados)
   - **Correspondência IMEI** (somente IMEI)
   - **Correspondência IMSI** (se fornecido na solicitação)
   - **Política de equipamento desconhecido** (comportamento padrão configurado)
4. Retornar status do equipamento

AVPs de Resposta:

- Session-Id (ecoado da solicitação)
- Result-Code: 2001 (sucesso)
- Equipment-Status: 0 (whitelist) / 1 (blacklist) / 2 (greylist)

Respostas de Erro:

- Experimental-Result: 5422 (equipamento/assinante não encontrado)
- Experimental-Result: 5012 (erro geral)

Lógica de Correspondência de Equipamentos

Ordem de Prioridade

O EIR utiliza uma estratégia de busca em cascata para maximizar a flexibilidade de correspondência:

```
1. IMEISV (IMEI + Software-Version)
   ↓ (se não houver correspondência)
2. IMEI apenas
   ↓ (se não houver correspondência)
3. IMSI (se fornecido na solicitação)
   ↓ (se não houver correspondência)
4. Política de Equipamento Desconhecido
```

Algoritmo de Correspondência

Passo 1: Correspondência IMEISV

- Concatenar IMEI + Software-Version: `"35979139461611" + "08" = "3597913946161108"`
- Testar contra as regras do EIR em ordem de precedência ascendente
- Retornar ação ("whitelist", "blacklist", "greylist") da primeira regra correspondente

Passo 2: Correspondência IMEI (fallback)

- Usar apenas IMEI: `"35979139461611"`
- Testar contra as regras do EIR em ordem de precedência ascendente
- Retornar ação da primeira regra correspondente

Passo 3: Correspondência IMSI (fallback se IMSI fornecido)

- Usar IMSI da solicitação: `"999999876543210"`
- Testar contra as regras do EIR em ordem de precedência ascendente

- Retornar ação da primeira regra correspondente
- **Caso de uso**: Bloquear todos os equipamentos para um assinante específico

Passo 4: Política de Equipamento Desconhecido (fallback final)

- Configuração: `eir_unknown_equipment_behaviour`
- Opções:
  - `:whitelist` - Permitir equipamentos desconhecidos (permissivo)
  - `:blacklist` - Bloquear equipamentos desconhecidos (restritivo)
  - `:greylist` - Observar equipamentos desconhecidos (moderado)
  - `:reject_unknown_equipment` - Retornar erro 5422 (estrito)

Precedência de Regras

Cada regra possui uma `precedence` inteira de 1 a 65535. Dentro de cada etapa de correspondência, as regras são testadas em ordem de precedência ascendente e a primeira correspondência vence, portanto, números mais baixos são avaliados primeiro.

- Criar uma regra sem precedência a adiciona após a precedência mais alta atual.
- Criar uma regra com uma precedência explícita a insere nessa posição. Cada regra existente com uma precedência igual ou superior se move para baixo em uma unidade, independentemente de o valor solicitado já estar em uso, portanto, lacunas nunca são preenchidas. Com as regras `a`, `b`, `c` em 1, 2, 3, criar `d` com precedência 1 dá `d`, `a`, `b`, `c` em 1, 2, 3, 4.
- Atualizar a precedência de uma regra a move para essa posição e desloca apenas as regras entre suas posições antiga e nova, portanto, o restante da lista mantém sua ordem.
- Enviar a precedência atual de uma regra de volta, por exemplo, um GET seguido por um PUT do mesmo corpo, não muda nada.
- Deletar uma regra deixa uma lacuna na numeração. Lacunas são inofensivas porque apenas a ordem relativa importa.
- A precedência ordena regras dentro de uma etapa de correspondência. Não altera a ordem das etapas IMEISV, IMEI, IMSI acima.

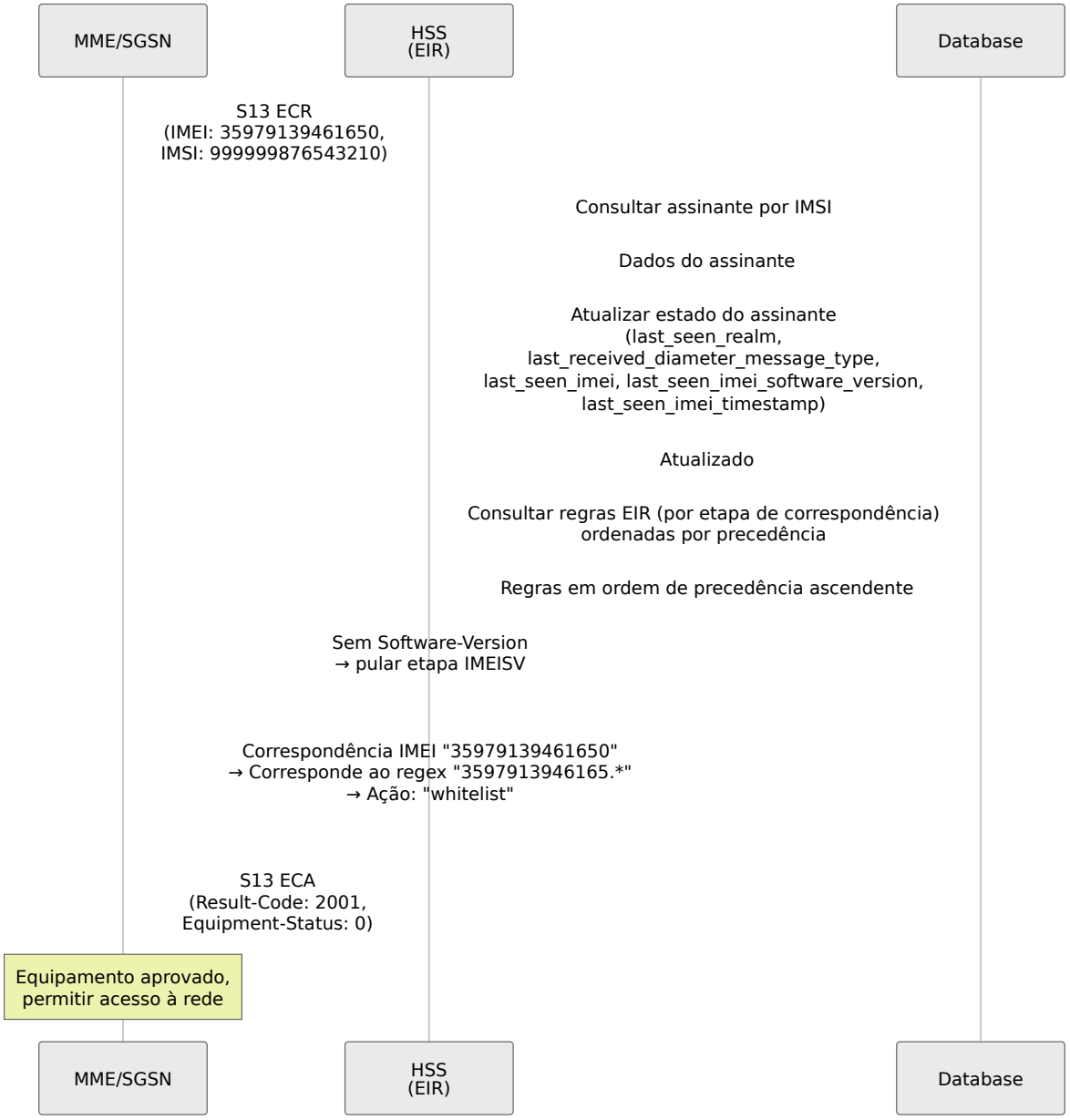
Regras que existiam antes da introdução da precedência são numeradas de 1 a N na ordem de criação quando o banco de dados é migrado. Se alguma dessas regras se sobrepuser, revise sua precedência com `GET /api/eir/rule` após a atualização.

Exemplos de Padrões Regex

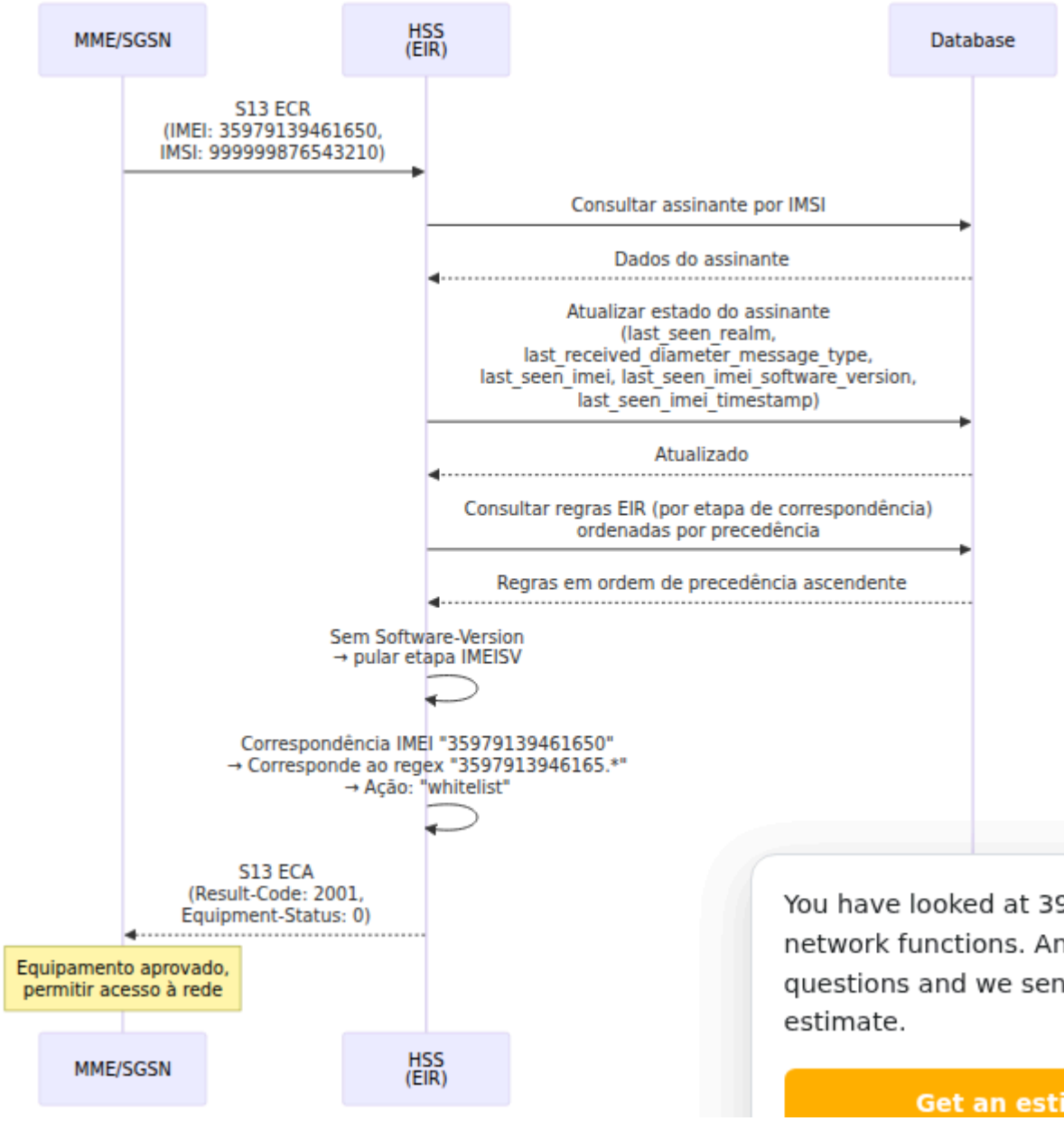
Padrão	Correspondências	Caso de Uso
`"35979139461650"`	IMEI exato	Lista branca/preta de dispositivo único
`"3597913946165.*"`	Coringa de prefixo IMEI	Faixa de fabricante/modelo
`"3597913946161108"`	IMEISV exato	Dispositivo específico com versão de software
`"999999876543210"`	IMSI	Bloquear todos os equipamentos para o assinante
`"359791.*"`	Coringa TAC	Alocação de todo tipo de dispositivo

Fluxos de Mensagens Comuns

Fluxo 1: Verificação de Equipamento - IMEI Conhecido na Lista Branca

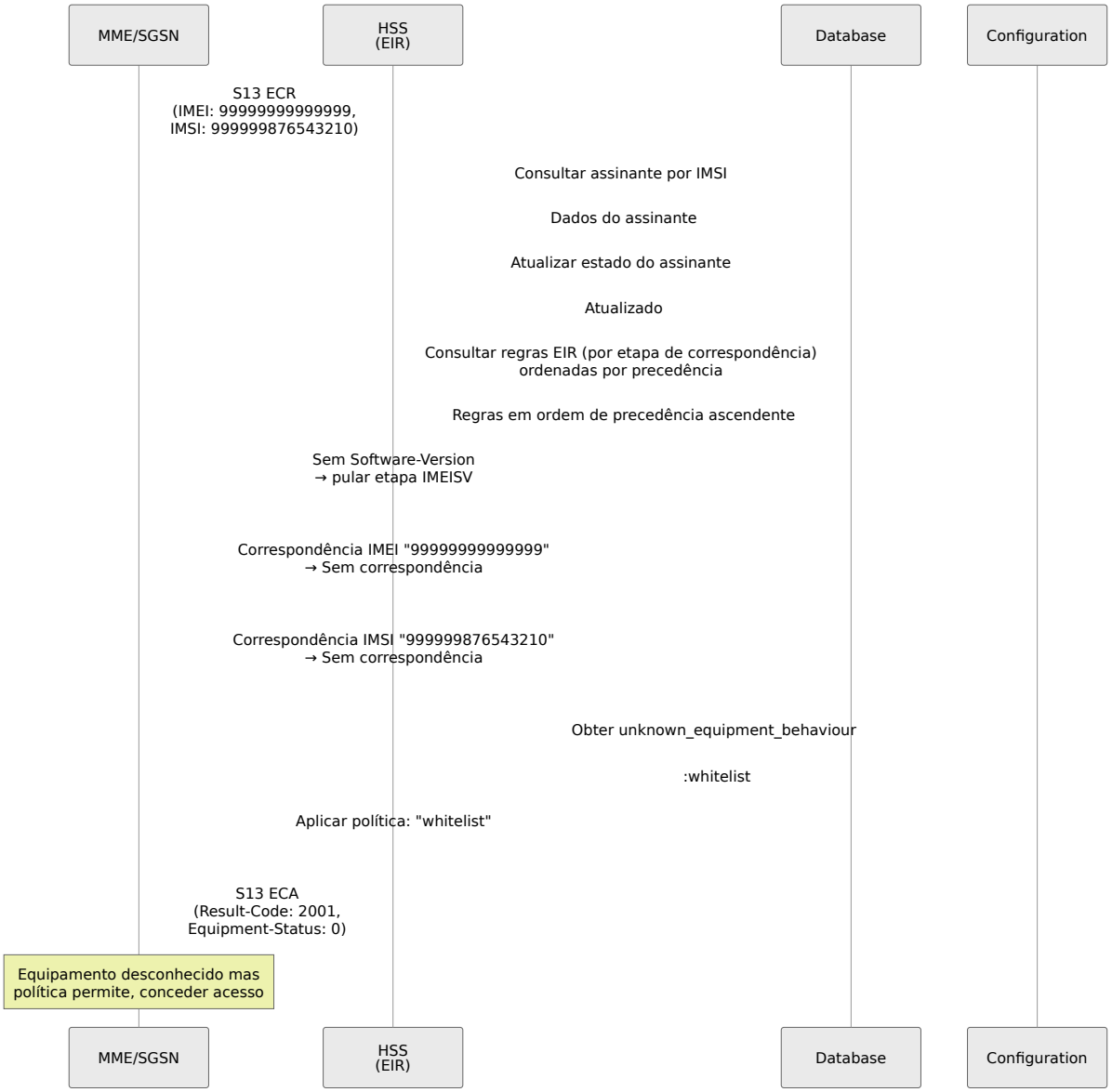


Fluxo 2: Verificação de Equipamento - IMEI na Lista Negra (Dispositivo Roubado)

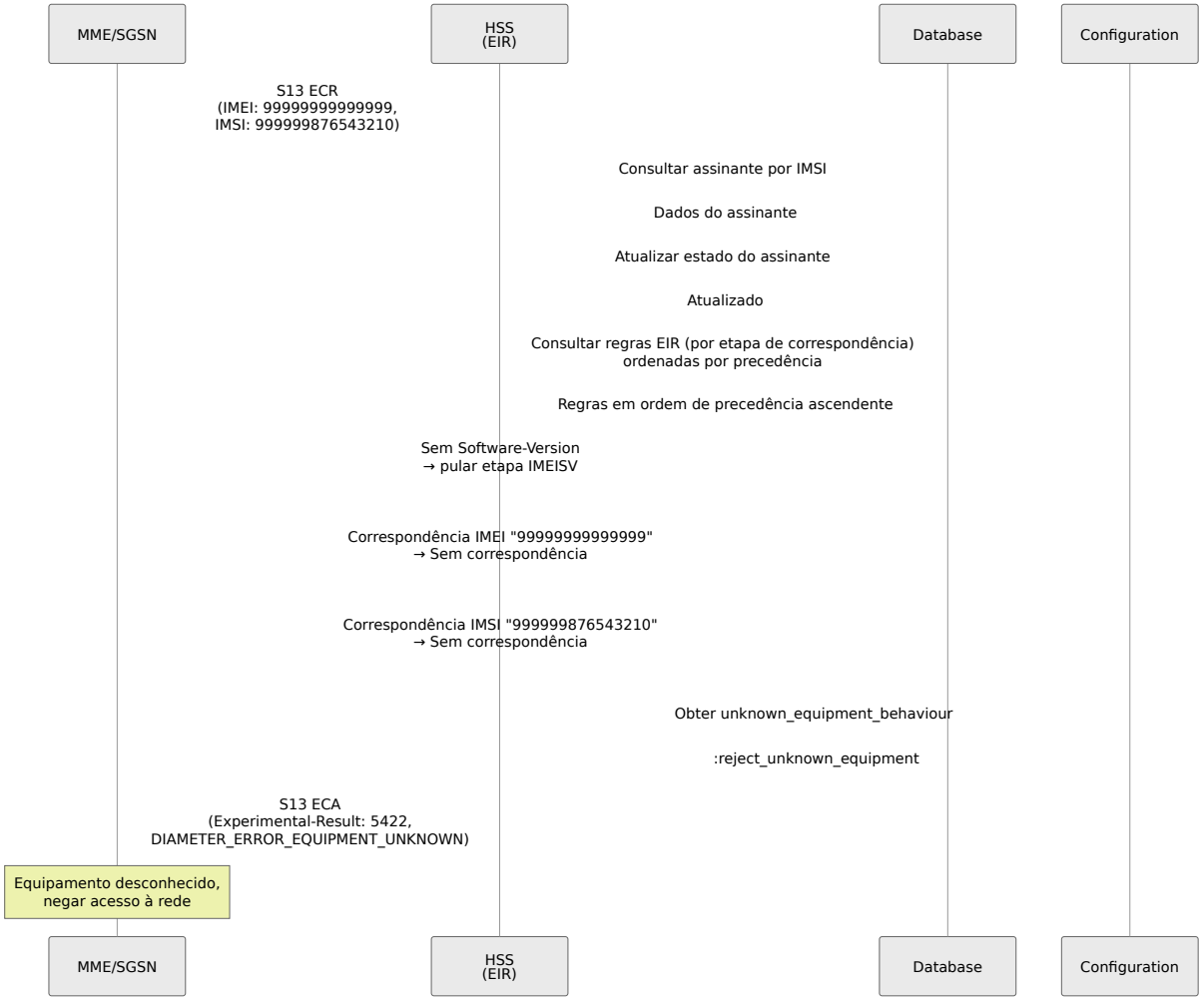


You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima

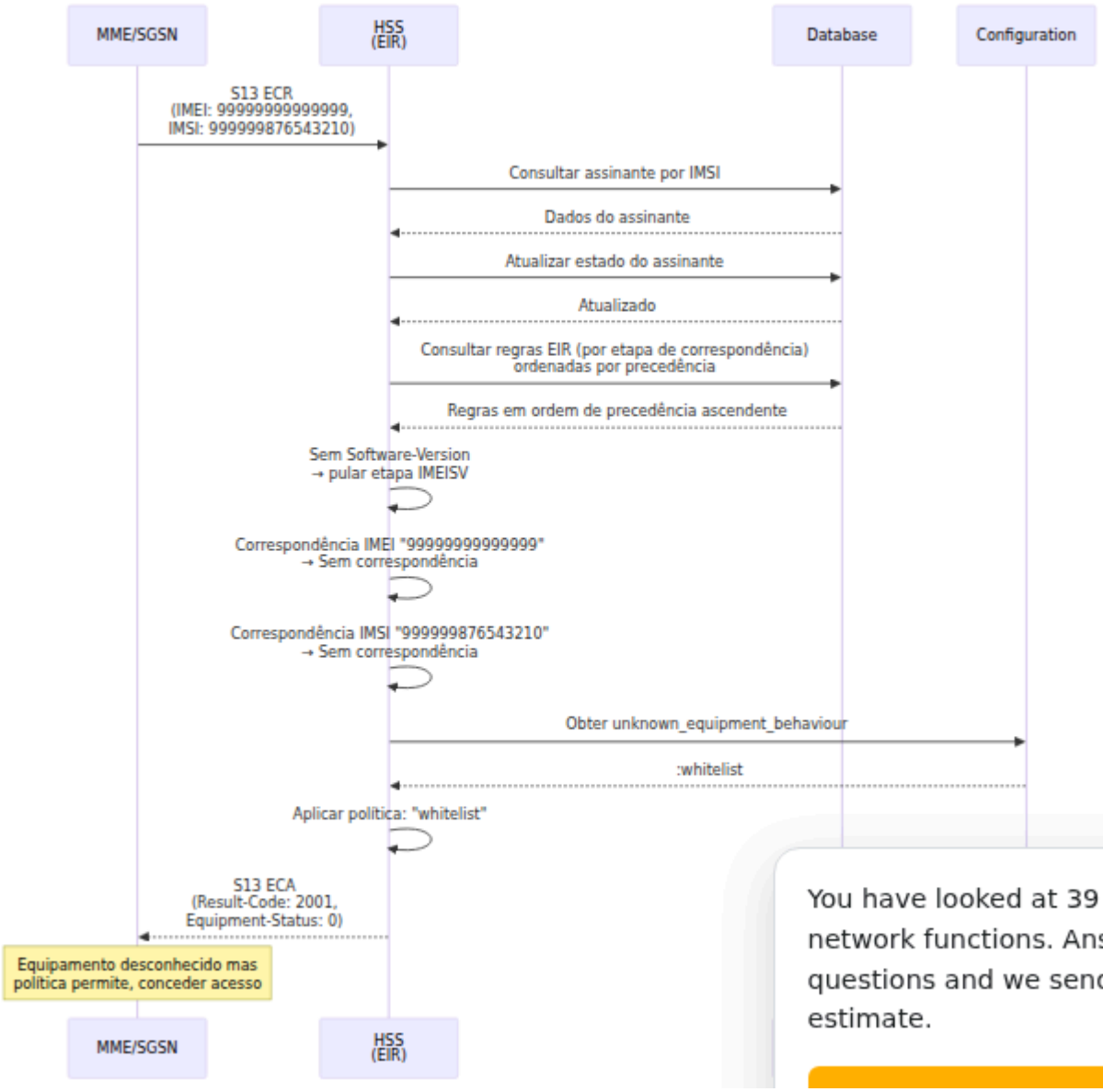
Fluxo 3: Verificação de Equipamento - Equipamento Desconhecido (Política de Lista Branca)



Fluxo 4: Verificação de Equipamento - Equipamento Desconhecido (Política de Rejeição)

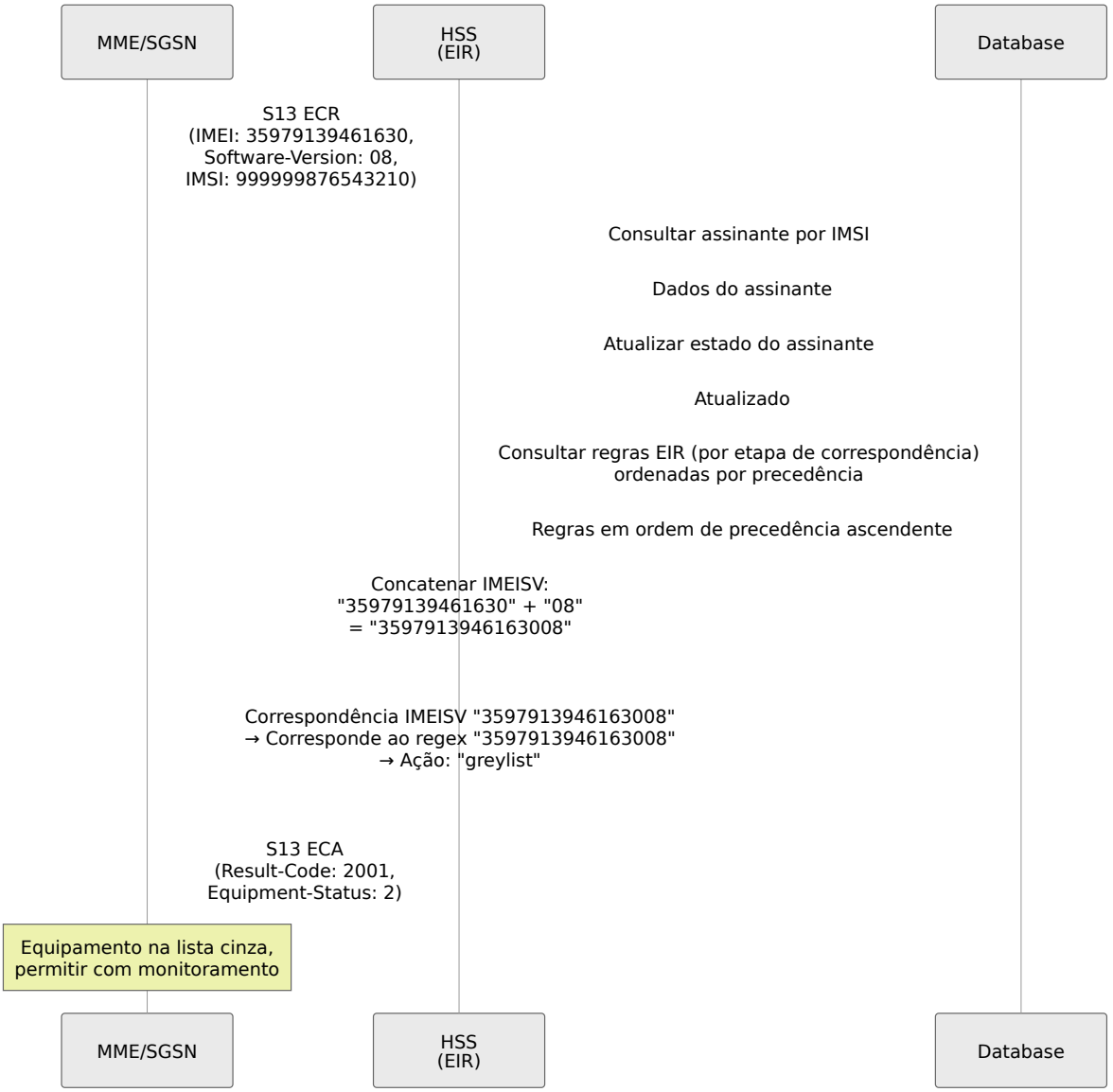


Fluxo 5: Verificação de Equipamento - Assinante Desconhecido ou Desabilitado

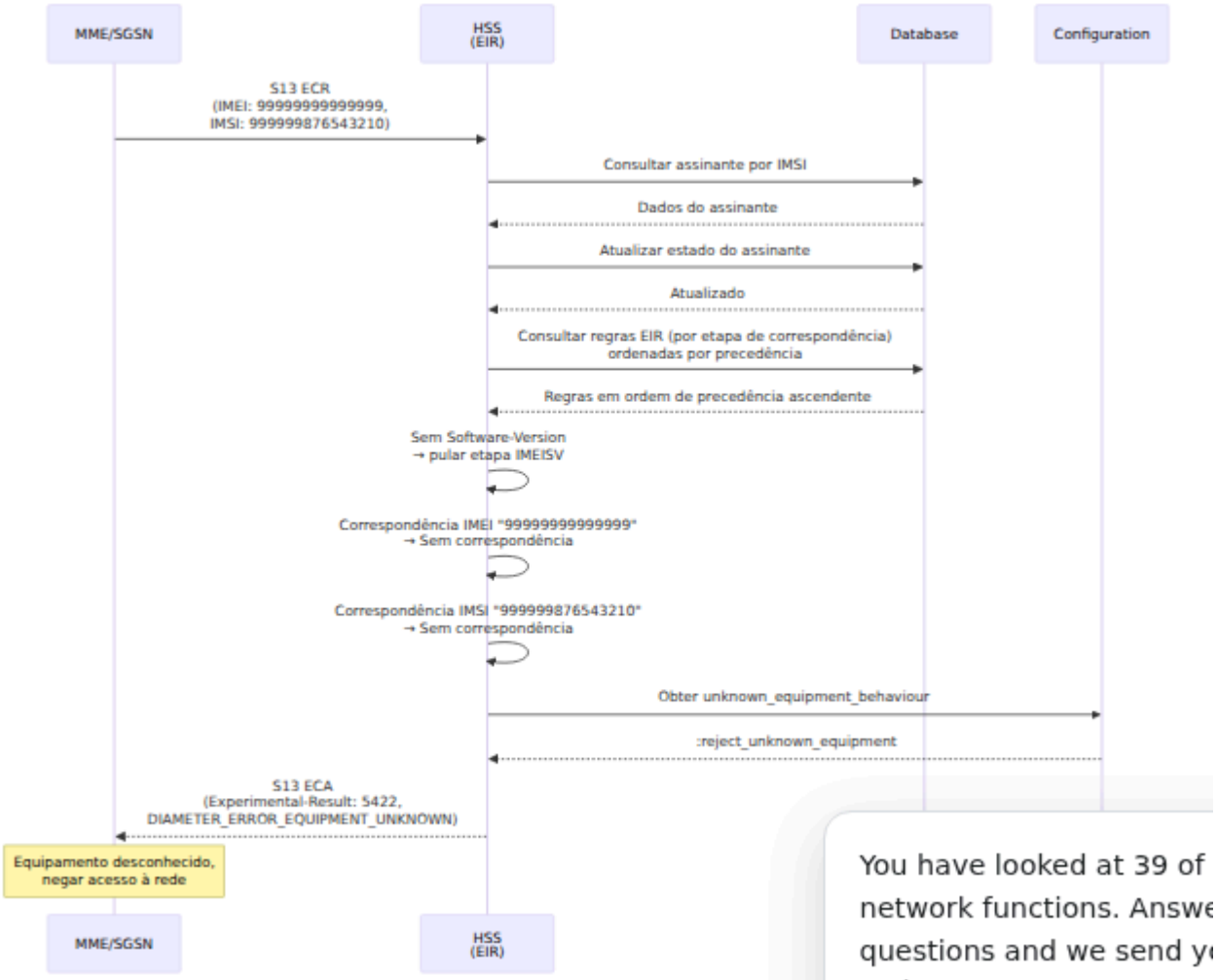


You have looked at 39 of
network functions. Answe
questions and we send y
estimate.

Fluxo 6: Verificação de Equipamento - Correspondência IMEISV



Fluxo 7: Verificação de Equipamento - Bloqueio de IMSI



REST API

Gerenciamento de Regras EIR

Caminho base: `/api/eir/rule`

Os corpos das solicitações são objetos JSON planos com os campos da regra no nível superior (veja Formato da Solicitação). As respostas usam o envelope padrão descrito na Referência da API. Cada regra contém `id`, `action`, `regex` e `precedence`.

Listar Todas as Regras EIR

Solicitação:

```
GET /api/eir/rule
```

Resposta (HTTP 200):

```
{
  "status": "success",
  "response": [
    {
      "id": "e7f6a5b4-c3d2-4e1f-8a9b-0c1d2e3f4a5b",
      "action": "whitelist",
      "regex": "3597913946165.*",
      "precedence": 2
    },
    {
      "id": "f8a7b6c5-d4e3-4f2a-9b0c-1d2e3f4a5b6c",
      "action": "blacklist",
      "regex": "35979139461640",
      "precedence": 1
    }
  ]
}
```

A lista é ordenada por `id`, não por precedência. Classifique por `precedence` para ver a ordem de avaliação.

Obter Regra EIR Específica

Solicitação:

```
GET /api/eir/rule/{id}
```

Resposta (HTTP 200):

```
{
  "status": "success",
  "response": {
    "id": "e7f6a5b4-c3d2-4e1f-8a9b-0c1d2e3f4a5b",
    "action": "whitelist",
    "regex": "3597913946165.*",
    "precedence": 2
  }
}
```

Criar Regra EIR

Solicitação:

```
POST /api/eir/rule
Content-Type: application/json

{
  "action": "blacklist",
  "regex": "35979139461640",
  "precedence": 1
}
```

Validação:

- `action`: Obrigatório, deve ser "whitelist", "blacklist" ou "greylist"
- `regex`: Obrigatório, deve ser um padrão regex válido
- `precedence`: Inteiro opcional de 1 a 65535. Omitir para adicionar a regra após a precedência mais alta atual. Fornecer um valor insere a regra nessa posição e move cada regra com uma precedência igual ou superior para baixo em uma unidade. Veja Precedência de Regras.

Resposta (HTTP 201):

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "blacklist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```

Resposta de Erro (HTTP 422):

```
{
  "status": "error",
  "response": {
    "invalid_fields": {
      "precedence": "must be greater than or equal to 1"
    }
  }
}
```

Atualizar Regra EIR (Parcial)

Solicitação:

```
PATCH /api/eir/rule/{id}
Content-Type: application/json

{
  "action": "greylist"
}
```

Resposta (HTTP 200):

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "greylist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```


Alterar `precedence` move a regra para essa posição e desloca apenas as regras entre suas posições antiga e nova. Omitir `precedence`, ou enviar seu valor atual, mantém a ordenação inalterada. Não envie `null`: isso limpa a precedência da regra e a regra é então avaliada após cada regra numerada.

Atualizar Regra EIR (PUT)

Solicitação:

```
PUT /api/eir/rule/{id}
Content-Type: application/json

{
  "action": "whitelist",
  "regex": "359791394616.*",
  "precedence": 3
}
```

Resposta (HTTP 200):


```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "whitelist",
    "regex": "359791394616.*",
    "precedence": 3
  }
}
```

Um PUT aplica os campos presentes no corpo, como um PATCH; campos omitidos mantêm seus valores atuais. Um PUT para um id que não existe cria a regra com esse id, seguindo a mesma colocação de precedência que um POST.

Deletar Regra EIR

Solicitação:

```
DELETE /api/eir/rule/{id}
```

Resposta (HTTP 200):

```
{
  "status": "success",
  "response": ""
}
```

Deletar uma regra deixa uma lacuna na numeração de precedência. As regras restantes mantêm sua ordem relativa.

Configuração

Configuração do Serviço Diameter

Aplicação S13 (`config/runtime.exs`):

```
%{
  application_name: :s13,
  application_dictionary: :diameter_gen_3gpp_s13,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_252}
  ]
}
```

Comportamento de Equipamento Desconhecido

Configure o comportamento padrão para equipamentos que não correspondem a nenhuma regra em `config/runtime.exs`:

Exemplo:

```
config :hss, :eir,
  unknown_equipment_behaviour: :whitelist
```

Valores Válidos:

- `:whitelist` - Permitir equipamentos desconhecidos (padrão, permissivo)
- `:blacklist` - Bloquear equipamentos desconhecidos (restritivo)
- `:greylist` - Monitorar equipamentos desconhecidos (moderado)
- `:reject_unknown_equipment` - Retornar erro Diameter 5422 (estrito)

Casos de Uso:

- **Desenvolvimento/Teste**: `:whitelist` - Permitir todos os dispositivos
- **Produção (permissivo)**: `:whitelist` - Apenas bloquear dispositivos conhecidos como ruins
- **Produção (moderado)**: `:greylist` - Registrar dispositivos desconhecidos para revisão
- **Produção (estrito)**: `:reject_unknown_equipment` - Apenas permitir dispositivos registrados

Tratamento de Erros

Código de Resultado	Tipo	Significado	
2001	Sucesso	DIAMETER_SUCCESS	Ve ec co
5422	Experimental	DIAMETER_ERROR_EQUIPMENT_UNKNOWN	As er ec de re
5012	Experimental	DIAMETER_UNABLE_TO_COMPLY	Er pr

Casos de Uso

1. Gerenciamento de Dispositivo Roubado

Cenário: Dispositivo reportado como roubado

Ação:

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "35979139461640"  # IMEI exato
}
```

Resultado: Dispositivo negado acesso à rede na próxima conexão

2. Lista Branca de Fabricante

Cenário: Pré-aprovar toda a faixa de modelo de dispositivo

Ação:

```
POST /api/eir/rule
{
  "action": "whitelist",
  "regex": "359791394.*"  # TAC para fabricante/modelo
}
```

Resultado: Todos os dispositivos na faixa TAC aprovados

3. Bloqueio de Equipamento do Assinante

Cenário: Bloquear todos os equipamentos para assinante específico (bloqueio de SIM)

Ação:

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "999999876543210"  # IMSI
}
```

Resultado: Qualquer equipamento usado com este SIM é bloqueado

4. Lista Cinza de Equipamento de Teste

Cenário: Monitorar equipamentos de teste em produção

Ação:

```
POST /api/eir/rule
{
  "action": "greylist",
  "regex": "35979139.*"  # Faixa TAC de equipamento de teste
}
```

Resultado: Equipamento permitido, mas sinalizado para monitoramento

5. Controle de Versão de Software

Cenário: Bloquear versão de firmware vulnerável específica

Ação:

```
POST /api/eir/rule
{
  "action": "blacklist",
  "regex": "359791394616.*05"  # Faixa IMEI + Versão de Software
05
}
```

Resultado: Apenas dispositivos com a versão de software "05" na faixa IMEI são bloqueados

Detalhes da Implementação

Componentes Internos

A funcionalidade do EIR é implementada usando vários módulos internos:

- **Manipulador de Protocolo S13** - Processamento de mensagens ECR/ECA
- **Motor de Correspondência de Equipamento** - Correspondência de IMEI/IMEISV/IMSI baseada em regex em ordem de precedência
- **Colocação de Precedência** - Lógica de inserção em posição que desloca regras EIR vizinhas quando uma regra é criada em, ou movida para, uma

precedência explícita. Regras de cobrança compartilham apenas a verificação de limites de 1 a 65535 e não são renumeradas
- **Banco de Dados de Regras EIR** - Armazenamento e busca de padrões
- **Controlador da API REST** - Endpoints de gerenciamento de regras

Função de Consulta de Status do Equipamento

A consulta de status do equipamento segue esta lógica em cascata:

1. **Correspondência IMEISV**: Verifique IMEI + Software-Version concatenados
2. **Correspondência IMEI**: Verifique apenas o IMEI
3. **Correspondência IMSI**: Verifique IMSI (se fornecido)
4. **Equipamento Desconhecido**: Aplique a política padrão configurada

Resultados Possíveis:

- `whitelist` - Equipamento permitido
- `blacklist` - Equipamento bloqueado
- `greylist` - Equipamento sob observação
- `reject_unknown_equipment` - Rejeição estrita

Considerações de Segurança

Privacidade do IMEI

Os números IMEI são identificadores sensíveis. O EIR:

- Não registra valores IMEI em texto simples por padrão
- Serve a API REST sem autenticação, a menos que a autenticação de chave compartilhada opcional esteja configurada (`config :api_ex, api: %{auth: %{key: "..."}}`, apresentada pelos clientes no cabeçalho `x-api-key`). Ative-a sempre que `/api/eir/rule` for acessível de redes não confiáveis

Ordenação de Regras

As regras são avaliadas em ordem crescente de `precedence` dentro de cada etapa de correspondência, e a primeira correspondência vence. Quando os padrões se sobrepõem, dê à regra específica a menor precedência:

```
Precedence 1: regex "35979139461640" action "blacklist"
(específica)
Precedence 2: regex "359791.*" action "whitelist"  (ampla)
```

Criar a regra específica com `"precedence": 1` enquanto a regra ampla já possui precedência 1 move automaticamente a regra ampla para 2. Veja Precedência de Regras para o comportamento completo de colocação.

Limitação de Taxa

Considere implementar limitação de taxa em:

- Solicitações ECR S13 de pares não confiáveis
- Modificações de regras EIR da API REST
- Consultas de busca de IMEI para prevenir ataques de enumeração

Documentação Relacionada

- Protocolos Diameter - Especificação do protocolo S13
- Referência da API - Documentação completa da API
- Arquitetura - Arquitetura geral do HSS
- Guia de Operações - Procedimentos operacionais

Apêndice: Estrutura do IMEI

Formato do IMEI (15 dígitos)

```
35 9791 394616 1
│  │    │      └─ Dígito de verificação (algoritmo de Luhn)
│  │    └─ Número de Série (6 dígitos)
│  └─ FAC (Código de Montagem Final, 4 dígitos)
└─ TAC (Código de Alocação de Tipo, 8 dígitos no total incluindo
RBI)
   │  └─ RBI (Identificador do Corpo Relator, 2 dígitos)
   └─ Fabricante/Modelo (6 dígitos)
```

Formato do IMEISV (16 dígitos)

```
35 9791 394616 1 08
│  │    │      │ └─ Versão do Software (2 dígitos)
└─ IMEI (15 dígitos)
```

Padrões de Exemplo

IMEI/IMEISV	Padrão	Correspondências
359791394616108	`3597913946161.*`	Todos os dispositivos com TAC+FAC+Serial 359791394616*
359791394616140	`35979139461614.`	Todos os dígitos de verificação para o Serial 359791394616141-9
35979139461640	`35979139461640`	Correspondência exata do IMEI
3597913946163008	`3597913946163008`	Correspondência exata do IMEISV (IMEI + SV)

PCRF (Função de Regras de Políticas e Cobrança)

Visão Geral

O HSS inclui um PCRF (Função de Regras de Políticas e Cobrança) embutido que fornece controle de políticas e regras de cobrança para sessões de dados móveis. O PCRF controla políticas de Qualidade de Serviço (QoS), alocação de largura de banda e regras de cobrança para portadoras padrão e dedicadas em redes LTE.

Principais Capacidades

- **Interface Gx**: Controle de políticas para PGW/PCEF (Gateway de Rede de Dados por Pacote / Função de Aplicação de Políticas e Cobrança)
- **Interface Rx**: Autorização e QoS para fluxos de mídia IMS (Subsistema Multimídia IP)
- **Interface Sy**: Controle de limite de gastos do assinante contra um OCS externo - veja Interface Sy
- **Gerenciamento Dinâmico de Políticas**: Atualizações de políticas em tempo real via Re-Auth Requests (RAR)
- **Suporte a VoLTE**: Criação de portadora dedicada para chamadas de voz com QoS garantido
- **Regras de Cobrança**: Definir comportamento de cobrança e perfis de velocidade usando Traffic Flow Templates (TFTs)
- **REST API**: Controle programático da aplicação de políticas e gerenciamento de regras

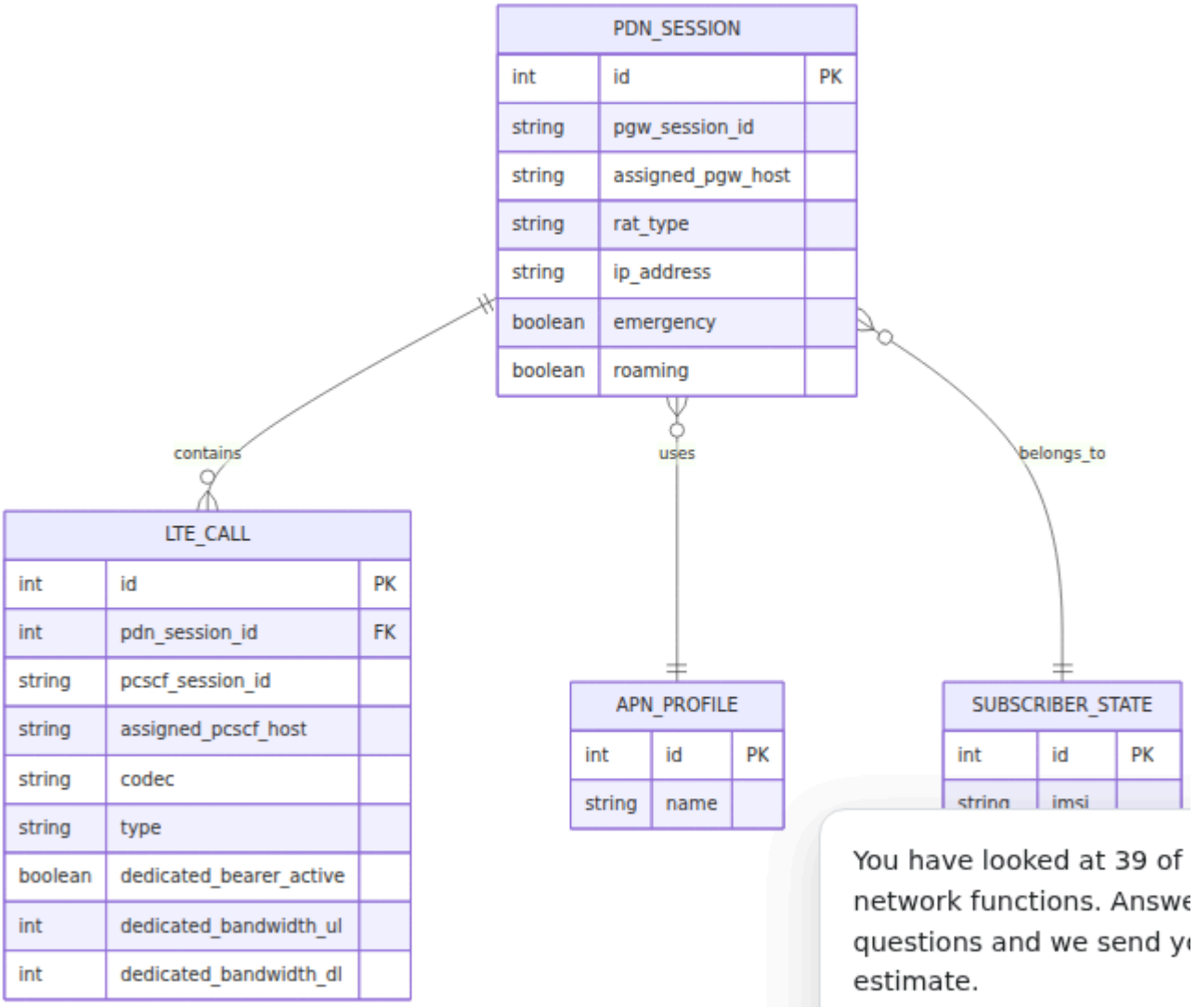
Arquitetura

Interfaces Diameter

Interface	ID da Aplicação	Par	Propósito
Gx	16.777.238	PGW (PCEF)	Gerenciamento de sessão PDN, aplicação de QoS, regras de cobrança
Rx	16.777.236	P-CSCF (AF)	Autorização de mídia IMS, reserva de largura de banda
Sy	16.777.302	OCS	Status do limite de gastos do assinante, política baseada em gastos - veja Interface Sy

Gerenciamento de Estado da Sessão

O PCRF mantém o estado da sessão para conexões PDN ativas e chamadas VoLTE:



Interface Gx

Operações Suportadas

1. Solicitação de Controle de Crédito - Inicial (CCR-I)

Gatilho: PGW cria nova conexão PDN para o assinante

AVPs de Solicitação:

- Session-Id
- Origin-Host, Origin-Realm
- Subscription-Id (contém IMSI)
- Called-Station-Id (nome do APN)

- IP-CAN-Type (tipo de Rede de Acesso à Conectividade IP)
- RAT-Type (Tecnologia de Acesso Rádio)
- Framed-IP-Address (endereço IP do UE)

Ações do PCRF:

1. Procurar assinante pelo IMSI
2. Recuperar perfil APN e configuração de QoS
3. Criar entrada de rastreamento de sessão
4. Construir políticas de QoS a partir do perfil APN

AVPs de Resposta:

- Result-Code: 2001 (DIAMETER_SUCCESS)
- QoS-Information (limites de largura de banda agregados do APN)
- Default-EPS-Bearer-QoS (QCI, ARP, prioridade)
- Bearer-Control-Mode
- Online / Offline (método de cobrança padrão em nível de sessão - veja Cobrança Online/Offline em Nível de Sessão)

2. Solicitação de Controle de Crédito - Atualização (CCR-U)

Gatilho: PGW relata mudanças na sessão (atualização de localização, mudança de RAT, etc.)

Ações do PCRF:

1. Localizar sessão existente pelo ID da sessão
2. Atualizar parâmetros da sessão (tipo de RAT, localização, etc.)
3. Retornar políticas atualizadas se necessário

Resposta: Result-Code 2001 com atualizações de política opcionais

3. Solicitação de Controle de Crédito - Terminar (CCR-T)

Gatilho: PGW termina a conexão PDN

Ações do PCRF:

1. Localizar sessão pelo ID da sessão
2. Excluir sessão e registros de chamadas associados
3. Confirmar a terminação

Resposta: Result-Code 2001

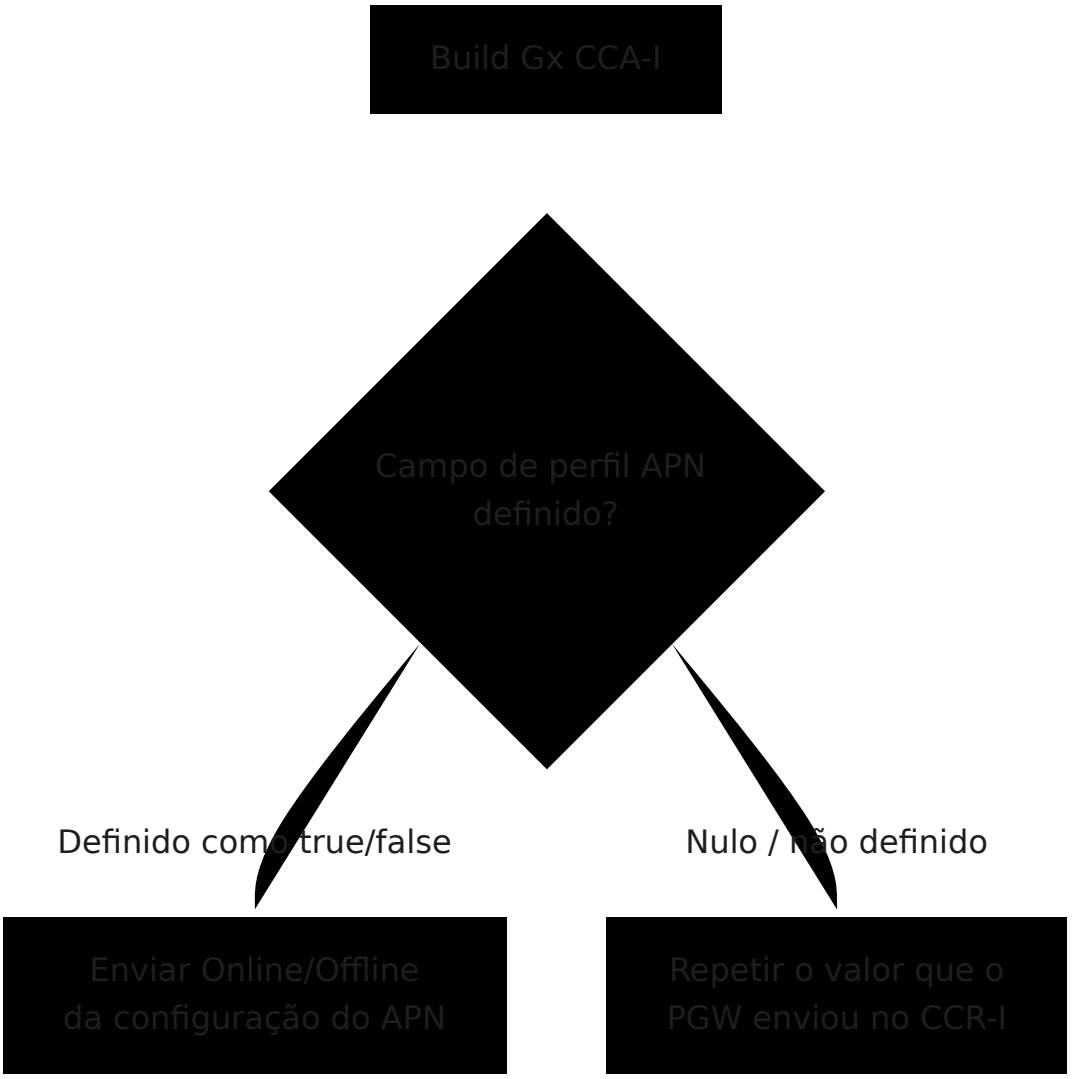
Cobrança Online/Offline em Nível de Sessão

Os AVPs `Online` (AVP 1009) e `Offline` (AVP 1008) enviados no nível superior de um Gx CCA-I definem o **método de cobrança padrão para toda a sessão IP-CAN** (a portadora padrão), conforme definido em 3GPP TS 29.212. Isso é distinto do `Online`/`Offline` por regra carregado *dentro* de uma Charging-Rule-Definition, que governa apenas o tráfego correspondido por essa regra específica.

Por padrão, o PCRF **repete** quaisquer valores que o PGW/PCEF anunciou no CCR-I. O perfil de QoS do APN pode, em vez disso, **fixar** o método de cobrança em nível de sessão, substituindo o eco. Isso é controlado por dois campos opcionais no perfil de QoS do APN:

Campo	Tipo	AVP	Efeito
`online_charging_enabled`	Booleano	`Online` (1009)	`true` → `ENABLE_ONLINE (1)`, `false` → `DISABLE_ONLINE (0)`
`offline_charging_enabled`	Booleano	`Offline` (1008)	`true` → `ENABLE_OFFLINE (1)`, `false` → `DISABLE_OFFLINE (0)`

Cada campo é resolvido independentemente:



Valores dos AVPs de Método de Cobrança (TS 29.212):

AVP	Código	Valor	Nome	Significado
`Online`	1009	0	DISABLE_ONLINE	Cobrança online (OCS / Gy) não aplicada à sessão
`Online`	1009	1	ENABLE_ONLINE	Cobrança online aplicada - crédito deve ser concedido antes que o tráfego flua
`Offline`	1008	0	DISABLE_OFFLINE	Cobrança offline (CDR / Gz) não aplicada à sessão
`Offline`	1008	1	ENABLE_OFFLINE	Cobrança offline aplicada - uso registrado para faturamento

Como funciona: Quando `online_charging_enabled`/`offline_charging_enabled` são deixados não definidos (o padrão para cada perfil existente), o comportamento não muda - o PCRF reflete o pedido do PGW, portanto, a postura de cobrança existente da rede é preservada. Definir qualquer um dos campos força esse modo de cobrança em cada CCR-I para qualquer APN usando o perfil, independentemente do que o PGW anuncia.

Caso de uso: Fixar a cobrança online para um APN pré-pago (para que o PGW deve consultar o OCS mesmo que tenha solicitado offline), ou forçar a cobrança offline em um APN pós-pago/IoT para garantir a geração de CDR. Como a configuração reside no perfil de QoS do APN, ela se aplica uniformemente a cada APN que referencia esse perfil.

> **Nota:** Isso controla apenas o *padrão da sessão*. A cobrança por tráfego ainda é governada pelos campos `online_charging_enabled`/`offline_charging_enabled` em Regras de

> Cobrança individuais, que aparecem dentro dos AVPs de Charging-Rule-Definition.

4. Solicitação de Reautenticação (RAR)

Direção: PCRF → PGW (HSS inicia)

Gatilho:

- Configuração de chamada IMS (Rx AAR aciona Gx RAR)
- Desmontagem de chamada IMS (Rx STR aciona Gx RAR)
- Reautenticação manual via REST API

AVPs RAR:

- Session-Id (ID da sessão PGW)
- Auth-Application-Id: 16.777.238
- Re-Auth-Request-Type (0 = Apenas Autorizar)
- Charging-Rule-Install/Remove
- QoS-Information (para portadoras dedicadas)

Ações do PGW: Criar/modificar/excluir portadoras dedicadas com base nas regras de cobrança

Regras de Cobrança e Traffic Flow Templates

O PCRF suporta a definição de regras de cobrança com Traffic Flow Templates (TFTs) para controlar:

- **Cobrança específica de serviço** - Tarifas diferentes para vídeo, jogos, redes sociais, etc.
- **Perfis de velocidade** - Limitar ou priorizar tráfego que corresponda a padrões específicos
- **Políticas baseadas em uso** - Aplicar diferentes QoS com base no tipo de tráfego ou destino

As regras de cobrança podem ser:

- Instaladas dinamicamente via Gx RAR com base na detecção de aplicativos
- Pré-definidas e acionadas por condições específicas (hora do dia, localização, cota)
- Associadas a TFTs usando regras de filtro de pacotes (5-tuple: protocolo, IP de origem/destino, porta de origem/destino)

Casos de Uso Comuns:

- **Zero-rating** - Acesso ilimitado a serviços específicos (Spotify, WhatsApp, Facebook) sem consumir a cota de dados
- **Acesso pós-cota** - Permitir portal de autoatendimento e sites de suporte mesmo após o assinante esgotar a cota de dados
- **Velocidade em camadas** - Alta velocidade para serviços premium, limitada para conteúdo padrão
- **Políticas baseadas em tempo** - Streaming ilimitado fora do pico, priorização em horários de pico
- **Políticas de roaming** - Cobrança diferente para uso de dados internacional vs nacional
- **SLAs empresariais** - QoS garantido para aplicativos críticos para os negócios

Estrutura da Política de QoS

QoS da Portadora Padrão (do perfil APN):

```
{
  "QoS-Class-Identifier": 9,          // QCI (9 = portadora
padrão)
  "APN-Aggregate-Max-Bitrate-UL": 50000,  // kbps
  "APN-Aggregate-Max-Bitrate-DL": 100000, // kbps
  "Allocation-Retention-Priority": {
    "Priority-Level": 8,
    "Pre-emption-Capability": 1,      // Pode preemptar
    "Pre-emption-Vulnerability": 1    // Pode ser preemptado
  }
}
```

QoS da Portadora Dedicada (para VoLTE):

```
{
  "QoS-Class-Identifier": 1,          // QCI 1 = Voz
Conversacional
  "Max-Requested-Bandwidth-UL": 128000,   // bps
  "Max-Requested-Bandwidth-DL": 128000,   // bps
  "Guaranteed-Bitrate-UL": 128000,
  "Guaranteed-Bitrate-DL": 128000
}
```

Interface Rx

Operações Suportadas

1. Solicitação AA (AAR) / Resposta AA (AAA)

Gatilho: P-CSCF solicita autorização para sessão de mídia IMS (configuração de chamada VoLTE)

AVPs de Solicitação:

- Session-Id (identificador da sessão P-CSCF)
- Subscription-Id (IMSI ou SIP URI)
- Media-Component-Description
  - Media-Type (áudio, vídeo)
  - Max-Requested-Bandwidth-UL/DL
  - Codec-Data
  - Flow-Description (filtros de pacotes 5-tuple)
- AF-Application-Identifier

Ações do PCRF:

1. Procurar assinante pelo IMSI ou SIP URI
2. Encontrar sessão IMS ativa
3. Extrair parâmetros de mídia (codec, largura de banda, regras de fluxo)

4. Criar entrada de rastreamento de chamada
5. **Acionar Gx RAR para PGW** para criar portadora dedicada
6. Aguardar resposta Gx RAA
7. Retornar Rx AAA com resultado da autorização

AVPs de Resposta:

- Result-Code: 2001 (sucesso) ou 5063 (serviço não autorizado)

Chamadas de emergência (SOS), incluindo chamadas não autenticadas / sem SIM

Chamadas de emergência (`urn:service:sos`) são autorizadas mesmo quando o chamador não é um assinante conhecido (um dispositivo sem SIM ou um assinante em roaming sem credenciais registradas). Isso é necessário para que as chamadas de emergência sempre obtenham a QoS correta (TS 23.167).

Como o PCRF reconhece e autoriza uma AAR de emergência:

1. **Detecção de emergência.** A solicitação é tratada como emergência se **ou** o `Subscription-Id` for um SIP URI contendo `sos`, **ou** o `Framed-IP-Address` da AAR resolver-se para uma sessão PDN que já foi criada como uma PDN de emergência pelo Gx CCR-I (o lado Gx sinaliza emergência pelo APN / `Called-Station-Id` contendo `sos`). O segundo caminho é o que permite que um **convite** de emergência **anônimo** (sem identidade utilizável) ainda seja autorizado.
2. **Resolução do assinante.** Procurado pelo `Subscription-Id`, caso contrário, pelo `Framed-IP-Address`. Uma PDN **órfã de emergência** (criada pelo Gx para um assinante desconhecido) não tem assinante (`subscriber_state_id: nil`); a AAR então segue o caminho órfão em vez do caminho de assinante normal.
3. **Perfil APN SOS.** A QoS da portadora dedicada é retirada do perfil APN SOS (veja "Perfil APN SOS padrão embutido" abaixo).
4. Um Gx RAR instala a portadora de voz dedicada (QCI 1) na PDN (órfã), e o AAA retorna `2001`.

> Nota: um `Framed-IP-Address` só gera autorização se corresponder a uma PDN que o Gx já criou **como emergência**. Uma solicitação não emergencial de um UE não registrado ainda é rejeitada no P-CSCF (nunca chega ao PCRF como uma AAR de emergência).

Perfil APN SOS padrão embutido. A autorização de emergência não depende mais de um operador ter pré-provisionado um perfil APN `sos` no banco de dados. O PCRF resolve o perfil APN SOS como:

- um perfil APN definido pelo operador cujo identificador APN é `sos`, se existir; caso contrário
- um **padrão embutido** de perfil APN SOS fornecido pelo OmniHSS.

O padrão embutido é um perfil transitório (sem linha no DB, `apn_profile_id: nil` na PDN órfã - a coluna é anulável) com QoS de emergência:

Parâmetro	Valor
QCI da portadora SOS padrão	5
QCI da portadora de voz dedicada	1 (prioridade ARP 2)
Prioridade de Alocação-Retenção	1
Capacidade de Preempção	habilitada
Vulnerabilidade à Preempção	desabilitada
APN-AMBR UL/DL	10000 kbps

Para substituir esses valores (ou para definir regras de cobrança por operador via um perfil PCRF), crie um perfil APN `sos` no DB - ele terá precedência sobre o padrão embutido.

2. Solicitação de Término de Sessão (STR) / Resposta de Término de Sessão (STA)

Gatilho: P-CSCF termina a sessão IMS (desligamento da chamada)

Ações do PCRF:

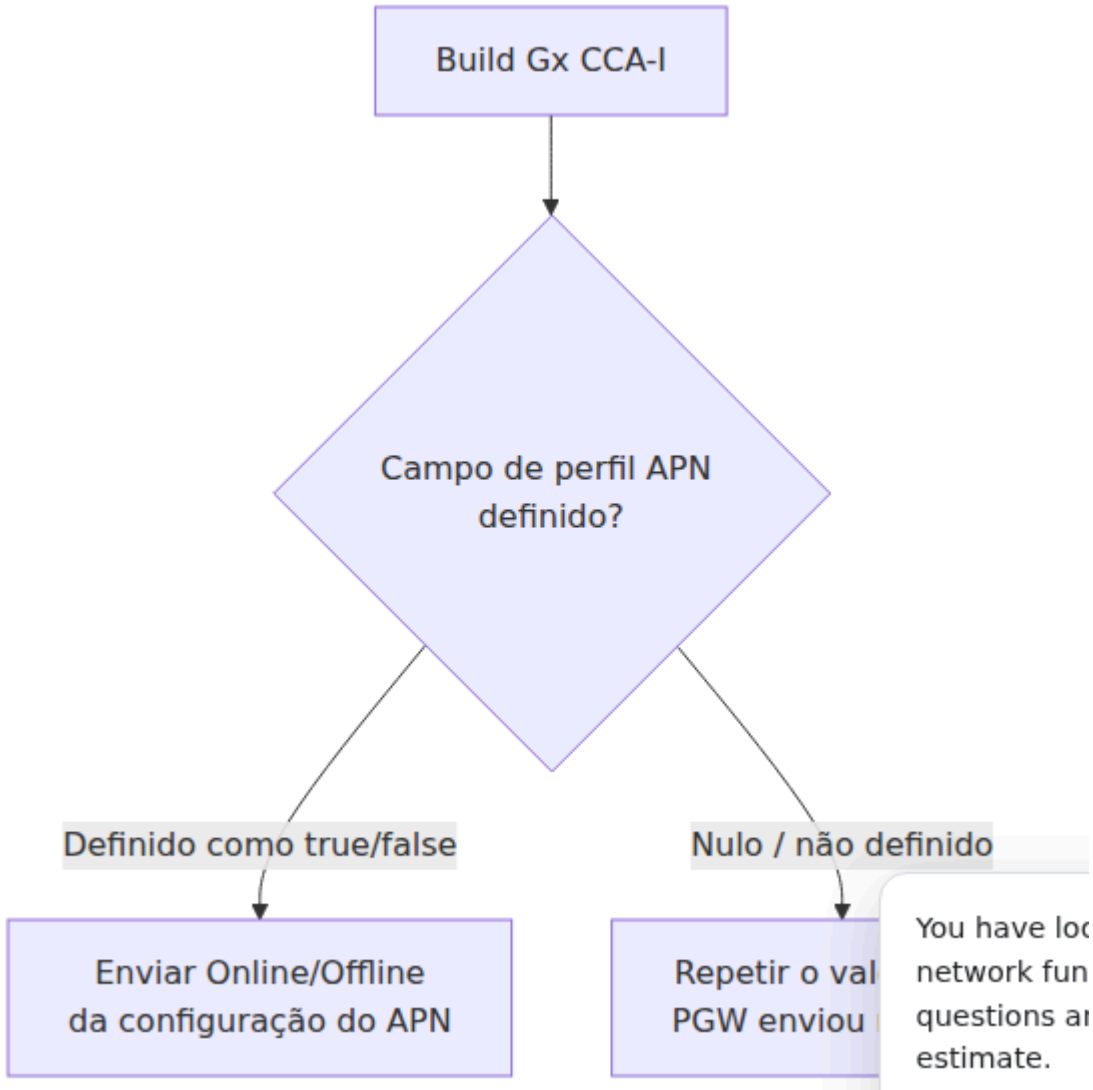
1. Localizar sessão de chamada pelo ID da sessão P-CSCF
2. **Acionar Gx RAR para PGW** para remover a portadora dedicada

3. Excluir entrada de rastreamento de chamada
4. Retornar confirmação STA

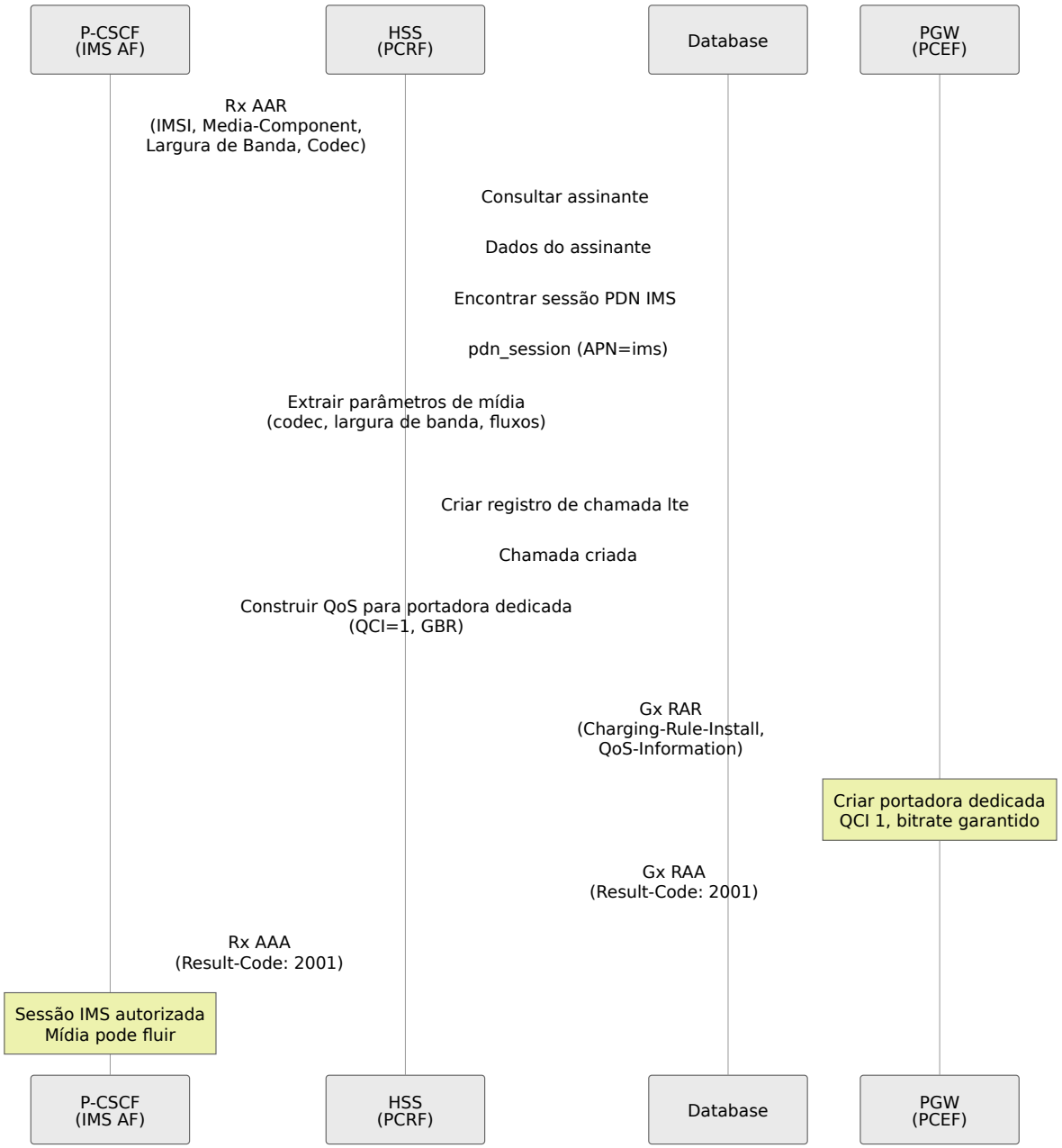
Resposta: Result-Code 2001

Fluxos de Mensagens Comuns

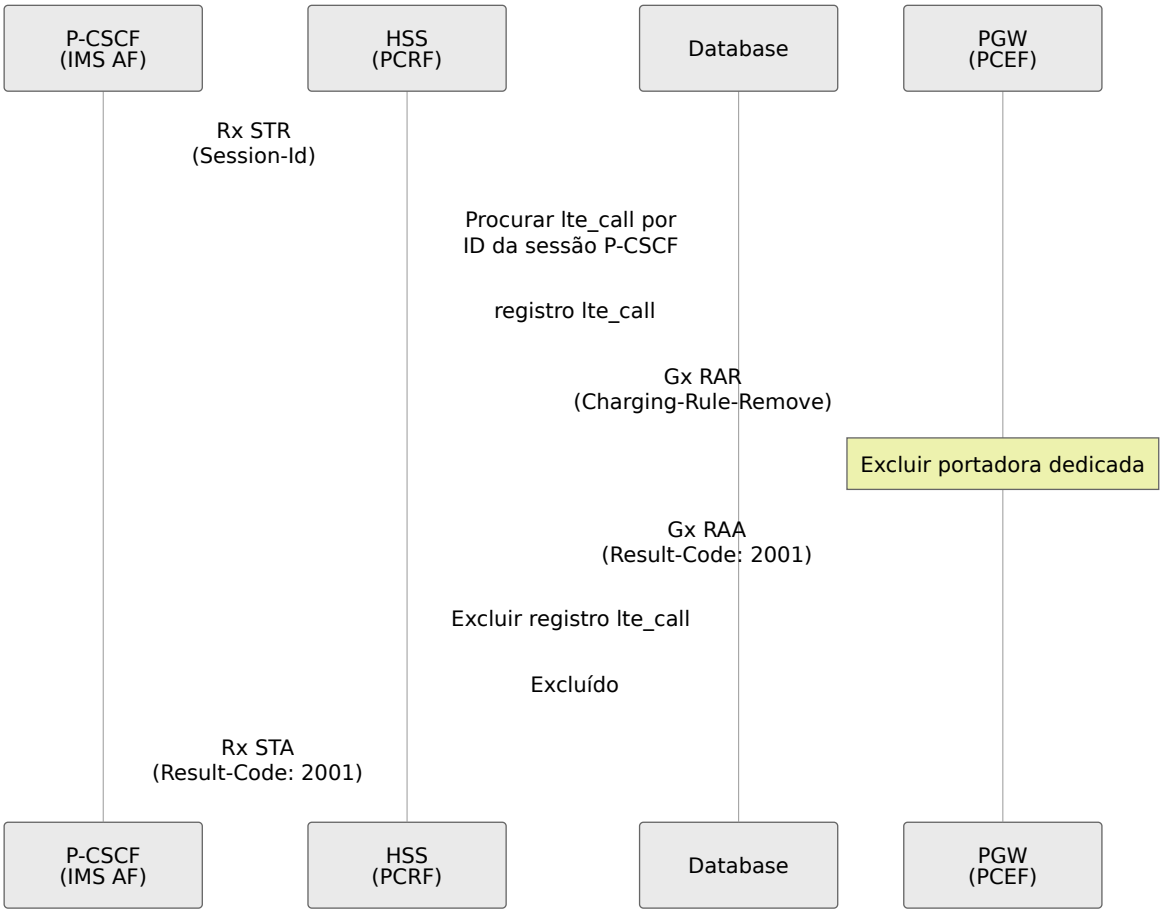
Fluxo 1: Estabelecimento da Sessão PDN



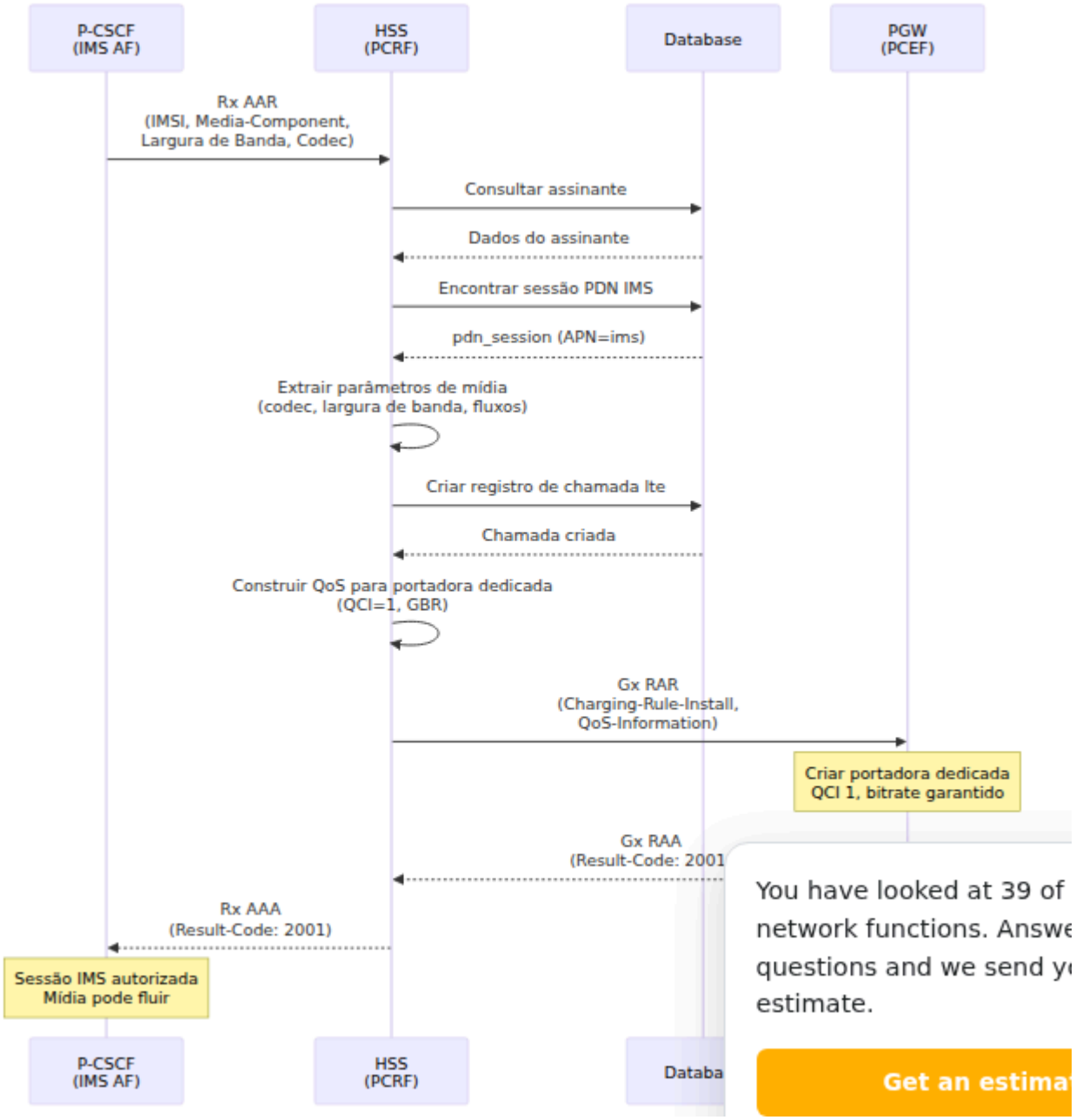
Fluxo 2: Configuração de Chamada VoLTE (Rx AAR → Gx RAR)



Fluxo 3: Desmontagem de Chamada VoLTE (Rx STR → Gx RAR)

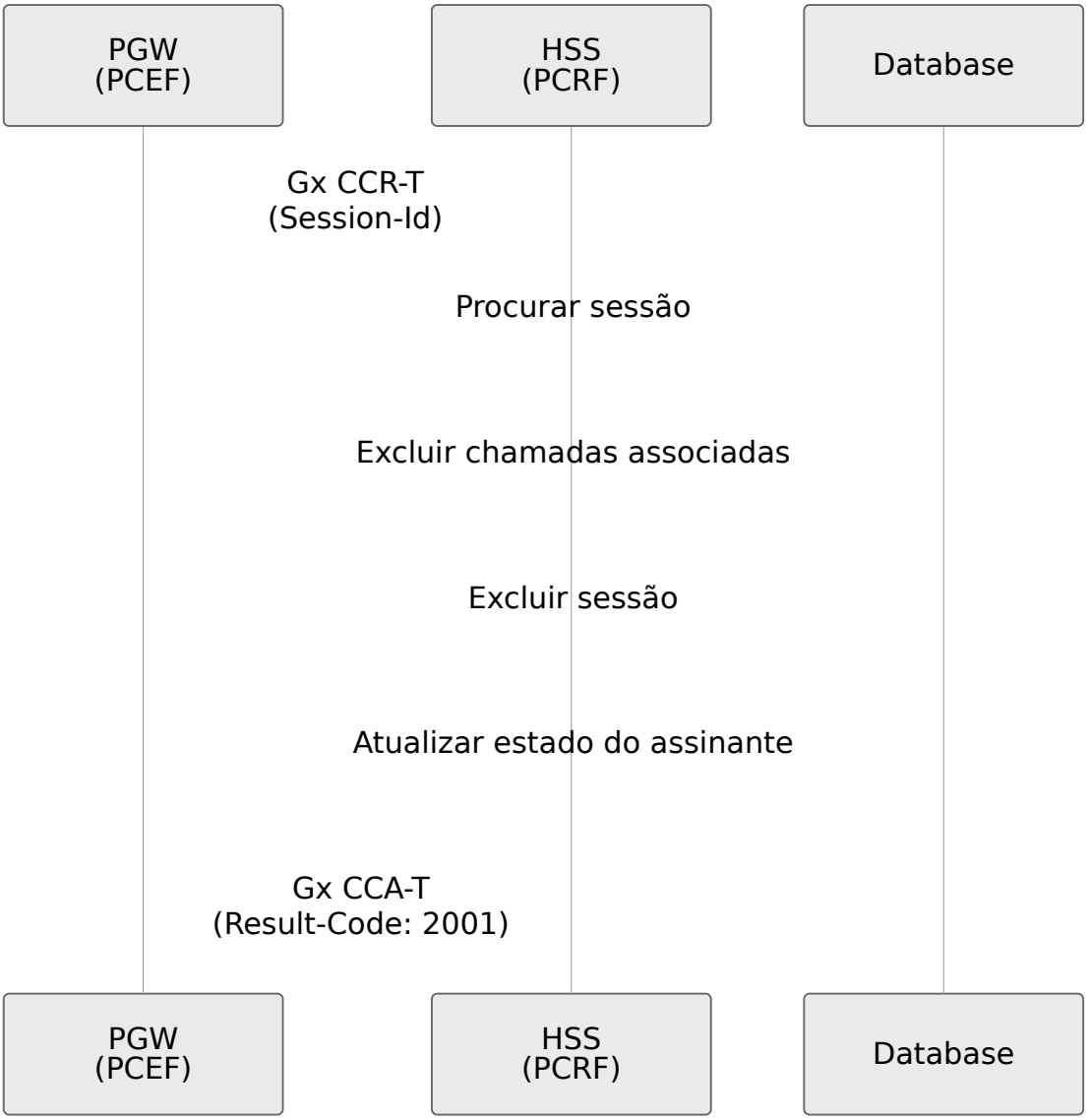


Fluxo 4: Atualização da Sessão PDN

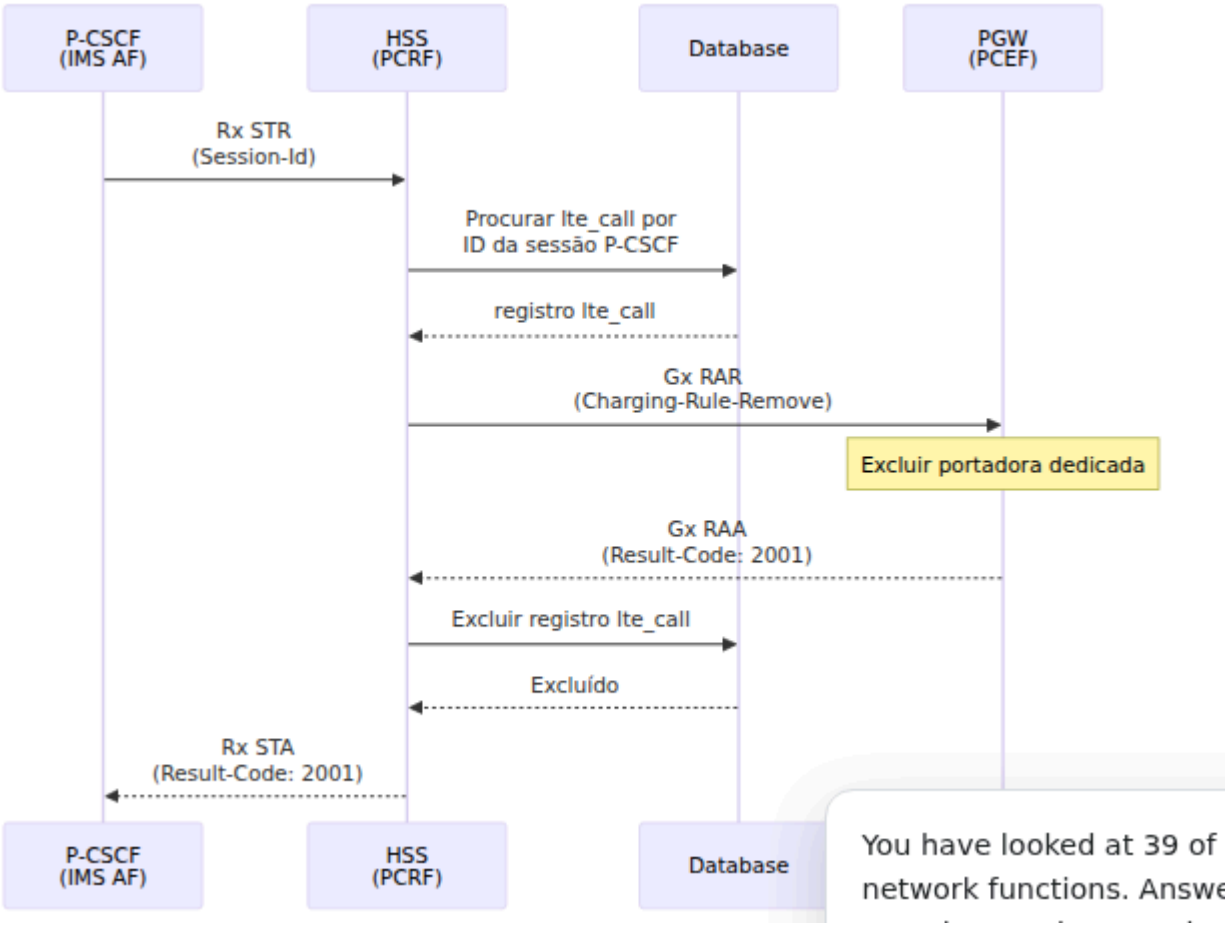


You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima

Fluxo 5: Término da Sessão PDN



Fluxo 6: Reautenticação Manual via REST API



REST API

Endpoint de Reautenticação PCRF

Endpoint: `POST /api/operation/pcrf_re_auth`

Propósito: Acionar manualmente a Solicitação de Reautenticação Gx para atualizar políticas

Quando Usar: Este endpoint manual é tipicamente usado para solução de problemas ou forçar a atualização de políticas em assinantes específicos. Para atualizações de políticas de rotina (mudança de perfis de QoS do APN), o sistema aciona automaticamente reautenticações para todas as sessões afetadas - nenhuma ação manual é necessária.

Corpo da Solicitação:

```
{
  "imsi": "999999876543210",
  "apn": "ims"
}
```

Resposta de Sucesso (HTTP 200):

```
{
  "data": "Requisição de Reautenticação Gx para 999999876543210
enviada para pgw.epc.mnc999.mcc999.3gppnetwork.org, Result-Code:
2001"
}
```

Resposta de Erro (HTTP 400):

```
{
  "error": "Não foi possível enviar a Reautenticação para
999999876543210 no APN ims, nenhuma Sessão PDN ativa encontrada"
}
```

API de Configuração de Políticas

O PCRF recupera políticas de QoS das configurações de APN armazenadas no banco de dados. Essas políticas podem ser criadas e gerenciadas via REST API.

Aplicação Automática de Políticas: Quando você atualiza um perfil de QoS do APN (por exemplo, altera limites de largura de banda ou QCI), o sistema envia automaticamente Solicitações de Reautenticação Gx (RAR) para todos os PGWs com sessões PDN ativas usando esse APN. Isso garante que as mudanças de políticas sejam aplicadas imediatamente a todos os assinantes conectados sem intervenção manual.

Arquitetura de Políticas

As políticas são definidas através de uma estrutura de três camadas:

```
Identificador APN → Perfil QoS do APN → Perfil APN
     ↓                    ↓                  ↓
  "internet"       QCI, AMBR, ARP     Liga ambos
```

1. Criar Identificador APN

Defina o nome do APN e o suporte à versão IP.

Endpoint: `POST /api/apn/identifier`

Corpo da Solicitação:

```
{
  "apn_identifier": {
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

Opções de Versão IP:

- `"ipv4"` - Apenas IPv4
- `"ipv6"` - Apenas IPv6
- `"ipv4v6"` - Pilha dupla (tanto IPv4 quanto IPv6)
- `"ipv4_or_ipv6"` - A rede decide (ou IPv4 ou IPv6)

Resposta (HTTP 201):

```
{
  "data": {
    "id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn": "internet",
    "ip_version": "ipv4v6"
  }
}
```

Validação:

- `apn`: Obrigatório, 1-254 caracteres, único
- `ip_version`: Obrigatório, deve ser uma das quatro opções acima

Listar Identificadores APN: `GET /api/apn/identifier`

2. Criar Perfil QoS do APN

Defina os parâmetros de QoS (largura de banda, QCI, prioridade).

Endpoint: `POST /api/apn/qos_profile`

Corpo da Solicitação:

```
{
  "apn_qos_profile": {
    "name": "Melhor Esforço Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "online_charging_enabled": false,
    "offline_charging_enabled": true
  }
}
```

Parâmetros de QoS:

Campo	Tipo	Faixa	Descriçã
`name`	string	1-254 chars	Nome do perfil (únic
`qci`	inteiro	1-254	Identificado de Classe d QoS (1-4 = GBR, 5-9 = Não-GBR)
`allocation_retention_priority`	inteiro	1-15	Nível de AR (1 = maior prioridade)
`apn_ambr_dl_kbps`	inteiro	1-4.294.967.293	Taxa Máxim Agregada d Bits do APN Downlink (kbps)
`apn_ambr_ul_kbps`	inteiro	1-4.294.967.293	Taxa Máxim Agregada d Bits do APN Uplink (kbp
`pre_emption_capability`	booleano	true/false	Pode preemptar portadoras de prioridad inferior
`pre_emption_vulnerability`	booleano	true/false	Pode ser preemptad por portadoras

Campo	Tipo	Faixa	Descriçã
			de prioridad superior
`online_charging_enabled`	booleano	true/false/null	AVP `Online` em nível de sessão para o Gx CCA-I. `null` (padrão) repete o valor do CC I do PGW. Veja Cobrança Online/Offlir em Nível de Sessão
`offline_charging_enabled`	booleano	true/false/null	AVP `Offlin` em nível de sessão para o Gx CCA-I. `null` (padrão) repete o valor do CC I do PGW. Veja Cobrança Online/Offlir em Nível de Sessão

Valores Comuns de QCI:

- `1` - Voz Conversacional (VoLTE) - GBR, 100ms de orçamento de atraso

- `2` - Vídeo Conversacional - GBR, 150ms de orçamento de atraso
- `5` - Sinalização IMS - Não-GBR, 100ms de orçamento de atraso
- `9` - Portadora Padrão (Internet) - Não-GBR, 300ms de orçamento de atraso

Resposta (HTTP 201):

```
{
  "data": {
    "id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793",
    "name": "Melhor Esforço Internet",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "online_charging_enabled": false,
    "offline_charging_enabled": true
  }
}
```

Listar Perfis QoS: `GET /api/apn/qos_profile`

3. Criar Perfil APN

Vincule o identificador APN a um perfil de QoS.

Endpoint: `POST /api/apn/profile`

Corpo da Solicitação:

```
{
  "apn_profile": {
    "name": "Perfil APN Internet",
    "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793"
  }
}
```

Campos:

- `name`: Nome do perfil (único), usado para referência
- `apn_identifier_id`: ID do Criar Identificador APN
- `apn_qos_profile_id`: ID do Criar Perfil QoS do APN

Resposta (HTTP 201):

```
{
  "data": {
    "id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "name": "Perfil APN Internet",
    "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
    "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793"
  }
}
```

Restrições:

- `apn_identifier_id` e `apn_qos_profile_id` devem referenciar registros existentes
- Cada combinação de identificador APN e perfil de QoS deve ser única

Listar Perfis APN: `GET /api/apn/profile`

Exemplo Completo de Configuração de Políticas

Passo 1: Criar Política APN IMS (VoLTE)

```
# 1. Criar Identificador APN
curl -X POST https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn_identifier": {
      "apn": "ims",
      "ip_version": "ipv4v6"
    }
  }'
# Resposta: {"data": {"id": "2c3e4f5a-6b7c-4d8e-9f0a-
1b2c3d4e5f6a", ...}}

# 2. Criar Perfil QoS (Sinalização IMS)
curl -X POST https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_qos_profile": {
      "name": "QoS Sinalização IMS",
      "qci": 5,
      "allocation_retention_priority": 2,
      "apn_ambr_dl_kbps": 5000,
      "apn_ambr_ul_kbps": 5000,
      "pre_emption_capability": true,
      "pre_emption_vulnerability": false
    }
  }'
# Resposta: {"data": {"id": "7b44a3e0-27cc-4b21-8d7a-
3f19e5c2b8d4", ...}}

# 3. Criar Perfil APN
curl -X POST https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profile": {
      "name": "Perfil APN IMS",
      "apn_identifier_id": "2c3e4f5a-6b7c-4d8e-9f0a-1b2c3d4e5f6a",
      "apn_qos_profile_id": "7b44a3e0-27cc-4b21-8d7a-3f19e5c2b8d4"
    }
  }'
# Resposta: {"data": {"id": "91c2f3a4-5d6e-4f7a-8b9c-
0d1e2f3a4b5c", ...}}
```

Passo 2: Atribuir ao Assinante

Uma vez criado, o perfil APN é atribuído a assinantes via perfis EPC. Veja Referência da API para vincular perfis APN a assinantes.

Atualização e Exclusão de Políticas

Atualizar Perfil QoS:

```
PATCH /api/apn/qos_profile/{id}
PUT /api/apn/qos_profile/{id}
```

Exemplo - Aumentar Largura de Banda para Todos os Usuários:

```
# Atualizar o perfil de QoS para aumentar a largura de banda
curl -X PATCH
https://hss.example.com:8443/api/apn/qos_profile/8d8ac610-566d-
4ef0-9c22-186b2a5ed793 \
  -H "Content-Type: application/json" \
  -d '{
    "apn_qos_profile": {
      "apn_ambr_dl_kbps": 150000,
      "apn_ambr_ul_kbps": 75000
    }
  }'
```

O que Acontece Automaticamente:

1. O perfil de QoS é atualizado no banco de dados
2. O sistema identifica todas as sessões PDN ativas usando APNs vinculados a esse perfil de QoS
3. Para cada sessão ativa, um Gx RAR é enviado para o PGW correspondente
4. Os PGWs atualizam a QoS da portadora para refletir os novos limites de largura de banda
5. Todos os assinantes conectados imediatamente recebem a política atualizada

Cenário de Exemplo: Se 100 assinantes estão atualmente conectados no APN "internet" usando este perfil de QoS, todos os 100 terão seus limites de largura de banda atualizados para 150 Mbps de download / 75 Mbps de upload dentro de segundos após a conclusão da chamada da API.

Nota: Quando você atualiza um perfil de QoS do APN, o sistema **aciona automaticamente a reautenticação** para todas as sessões PDN ativas usando esse APN, aplicando as novas políticas imediatamente aos assinantes conectados. Nenhuma reautenticação manual é necessária.

Excluir Recursos:

```
DELETE /api/apn/identifier/{id}
DELETE /api/apn/qos_profile/{id}
DELETE /api/apn/profile/{id}
```

Restrições de Exclusão:

- Não é possível excluir identificadores APN ou perfis de QoS referenciados por perfis APN
- Não é possível excluir perfis APN atribuídos a assinantes ativos

Modelos de Políticas

Internet de Alta Velocidade (100 Mbps de download / 50 Mbps de upload):

```
{
  "apn_qos_profile": {
    "name": "Internet de Alta Velocidade",
    "qci": 9,
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 100000,
    "apn_ambr_ul_kbps": 50000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

Internet Premium (500 Mbps de download / 100 Mbps de upload):

```
{
  "apn_qos_profile": {
    "name": "Internet Premium",
    "qci": 8,
    "allocation_retention_priority": 5,
    "apn_ambr_dl_kbps": 500000,
    "apn_ambr_ul_kbps": 100000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

IoT/M2M (Baixa Largura de Banda):

```
{
  "apn_qos_profile": {
    "name": "IoT M2M",
    "qci": 9,
    "allocation_retention_priority": 10,
    "apn_ambr_dl_kbps": 1024,
    "apn_ambr_ul_kbps": 512,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true
  }
}
```

Serviços de Emergência (Maior Prioridade):

```
{
  "apn_qos_profile": {
    "name": "Perfil APN de Emergência",
    "qci": 5,
    "allocation_retention_priority": 1,
    "apn_ambr_dl_kbps": 10000,
    "apn_ambr_ul_kbps": 10000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": false
  }
}
```

Configuração

Configuração do Serviço Diameter

Aplicação Gx (`config/runtime.exs`):

```
%{
  application_name: :gx,
  application_dictionary: :diameter_gen_3gpp_gx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_238}
  ]
}
```

Aplicação Rx (`config/runtime.exs`):

```
%{
  application_name: :rx,
  application_dictionary: :diameter_gen_3gpp_rx,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_236}
  ]
}
```

Parâmetros de QoS

Os parâmetros de QoS são obtidos de:

- **Portadora Padrão**: Configuração do perfil APN no banco de dados
  - `apn_qos_profile.qci` (Identificador de Classe de QoS)
  - `apn_qos_profile.apn_ambr_ul_kbps` (Taxa Máxima Agregada de Bits do APN Uplink)
  - `apn_qos_profile.apn_ambr_dl_kbps` (Taxa Máxima Agregada de Bits do APN Downlink)
  - `apn_qos_profile.priority_level` (Prioridade de Alocação Retenção)
- **Portadora Dedicada**: Extraído da Descrição do Componente de Mídia do Rx AAR
  - QCI: 1 (Voz Conversacional)
  - Bitrate Garantido: Dos AVPs de Largura de Banda Máxima Solicitada
  - Filtros de Fluxo: Dos AVPs de Descrição de Fluxo

Tratamento de Erros

Código de Resultado	Tipo	Significado	Causa
2001	Sucesso	DIAMETER_SUCCESS	Solicitação processada com sucesso
5001	Experimental	Usuário não encontrado	IMSI não está no banco de dados de assinantes
5002	Experimental	Sessão não encontrada	Sessão PDN não existe para atualização/terminação
5063	Experimental	Serviço não autorizado	Autorização de mídia IMS negada

Detalhes da Implementação

Gerenciamento de Sessão

O PCRF rastreia:

- **Sessões PDN Ativas** - Uma por APN, por assinante
- **Chamadas VoLTE** - Múltiplas chamadas por sessão IMS (suporta chamadas em conferência)
- **Políticas de QoS** - Aplicadas dinamicamente com base na configuração do APN
- **Regras de Cobrança** - Templates de fluxo de tráfego e políticas específicas de serviço

Recursos Avançados de Política

O PCRF suporta controle avançado de políticas, incluindo:

- **Instalação/remoção de regras de cobrança** via interface Gx
- **Correspondência de Traffic Flow Template (TFT)** para diferenciação de serviços
- **Perfis de velocidade dinâmicos** com base em aplicativo ou tipo de tráfego
- **Políticas conscientes do serviço** acionadas por condições de rede ou comportamento do assinante

Entre em contato com o administrador do sistema para informações sobre como configurar regras de cobrança avançadas e políticas baseadas em TFT.

Documentação Relacionada

- Interface Sy - Controle de limite de gastos do assinante contra um OCS externo
- Protocolos Diameter - Especificações detalhadas do protocolo
- Referência da API - Documentação completa da API
- Arquitetura - Arquitetura geral do HSS
- Mapeamento de Dados - Mapeamentos de banco de dados para AVPs Diameter

Tratamento de Erros da API

← Voltar para a Referência da API

Índice

- Respostas de Erro Comuns
- Fluxo de Tratamento de Erros

Respostas de Erro Comuns

400 Solicitação Inválida

```
{
  "error": "Formato JSON inválido"
}
```

Causas:

- JSON malformado
- Campos obrigatórios ausentes
- Tipos de dados inválidos

404 Não Encontrado

```
{
  "error": "Recurso não encontrado"
}
```

Causas:

- Assinante/perfil/entidade não existe
- ID incorreto na URL

422 Entidade Não Processável

```
{
  "errors": {
    "imsi": ["já foi utilizado"],
    "key_set_id": ["não existe"]
  }
}
```

Causas:

- Falhas de validação
- Restrições do banco de dados violadas
- Referências de chave estrangeira não existem

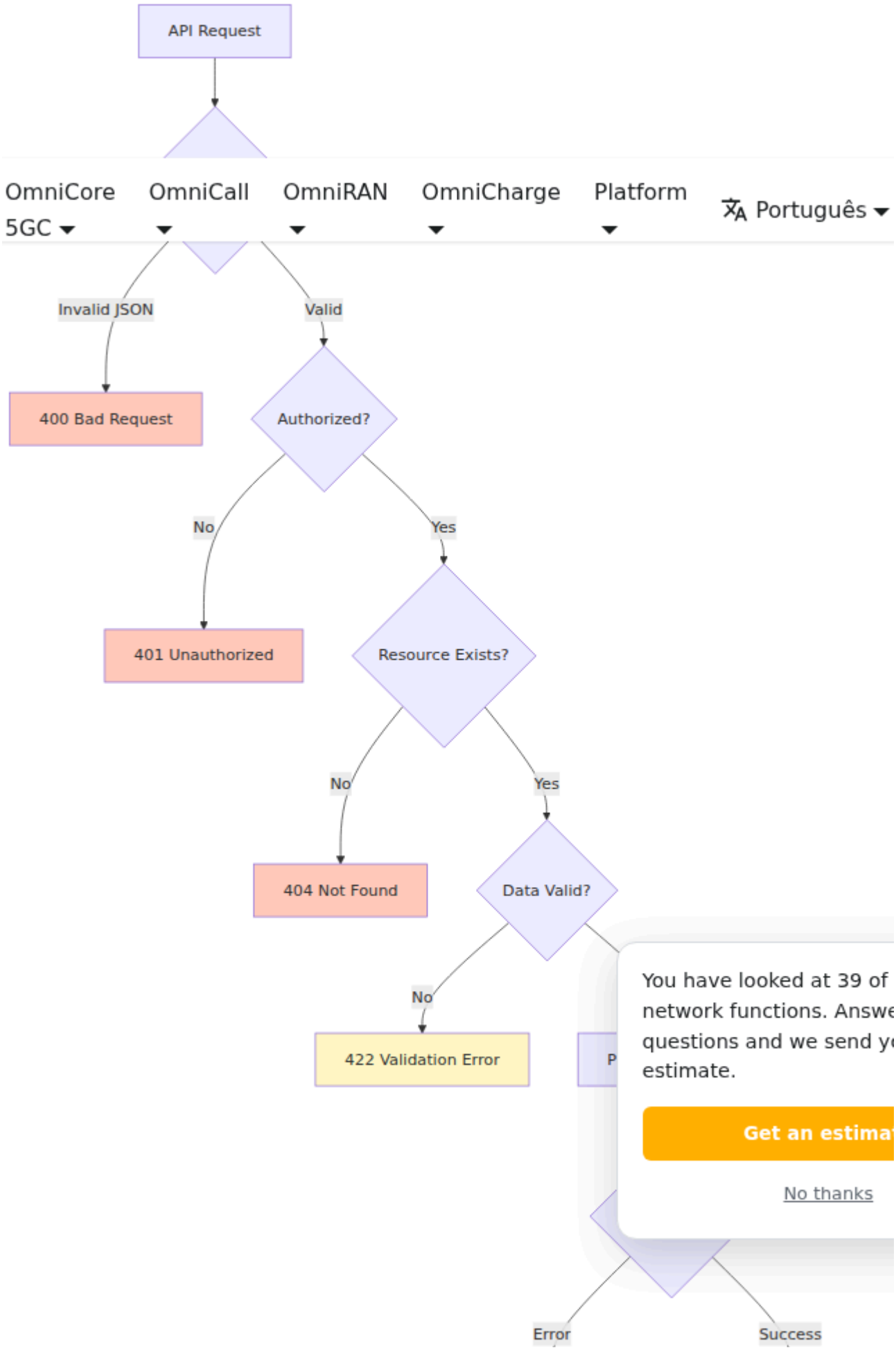
500 Erro Interno do Servidor

```
{
  "error": "Erro interno do servidor"
}
```

Causas:

- Problemas de conectividade com o banco de dados
- Erros inesperados da aplicação

Fluxo de Tratamento de Erros


API Request
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Invalid JSON
Valid
400 Bad Request
Authorized?
No
Yes
401 Unauthorized
Resource Exists?
No
Yes
404 Not Found
Data Valid?
No
422 Validation Error
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks
Error
Success



← Voltar para a Referência da API

Exemplos de Uso da API

← Voltar para a Referência da API

Índice

- Provisionamento Completo de Assinante
- Pré-Provisionamento de Assinante Inativo
- Provisionamento Completo de IP Estático

Provisionamento Completo de Assinante

Este exemplo demonstra o fluxo de trabalho completo para o provisionamento de um novo assinante do zero. O processo envolve a criação de todos os perfis e componentes necessários antes de criar o assinante.

Pré-requisitos: Este exemplo usa `jq` para análise de JSON. Instale com `apt-get install jq` ou `brew install jq`.

Seções Relacionadas:

- Gerenciamento de Conjunto de Chaves
- Perfis APN
- Perfis EPC
- Gerenciamento de Assinantes

```
# 1. Criar Conjunto de Chaves
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Criar Perfil de QoS APN
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Default Internet QoS",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 50000,
    "apn_ambr_ul_kbps": 25000,
    "pre_emption_capability": true,
    "pre_emption_vulnerability": true,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Criar Identificador APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn": "internet",
    "ip_version": "ipv4v6"
  }' | jq -r '.response.id')

# 4. Criar Perfil APN
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$APN_QOS_ID\",
    \"name\": \"Internet APN\"
```

```
  }" | jq -r '.response.id')

# 5. Criar Perfil EPC
EPC_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profiles\": [$APN_PROFILE_ID],
    \"name\": \"Standard Data Plan\",
    \"network_access_mode\": \"packet_only\",
    \"tracking_area_update_interval_seconds\": 600,
    \"ue_ambr_dl_kbps\": 100000,
    \"ue_ambr_ul_kbps\": 50000
  }" | jq -r '.response.id')

# 6. Criar Assinante
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001001123456789\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"epc_profile_id\": \"$EPC_PROFILE_ID\"
  }" | jq -r '.response.id')

echo "Assinante provisionado com sucesso com ID: $SUBSCRIBER_ID"
```

O que isso cria:

Este fluxo de trabalho de provisionamento cria um assinante completo com:

1. **Chaves criptográficas** (Conjunto de Chaves) - Para autenticação
2. **Perfil de serviço de dados** (Perfil EPC) - Configurações de largura de banda e acesso à rede
3. **Configuração APN** (Perfil APN) - Ponto de acesso com QoS
4. **Registro de assinante** (Assinante) - A entidade real do assinante

Próximos Passos:

- Adicionar números de telefone: Veja Gerenciamento de MSISDN
- Habilitar serviços de voz: Criar e atribuir Perfil IMS

- Configurar roaming: Criar e atribuir Perfil de Roaming
- Vincular SIM físico: Criar e atribuir SIM

Veja Também:

- Documentação Multi-MSISDN - Atribuindo múltiplos números de telefone
- Documentação de Perfis - Configuração avançada de perfis

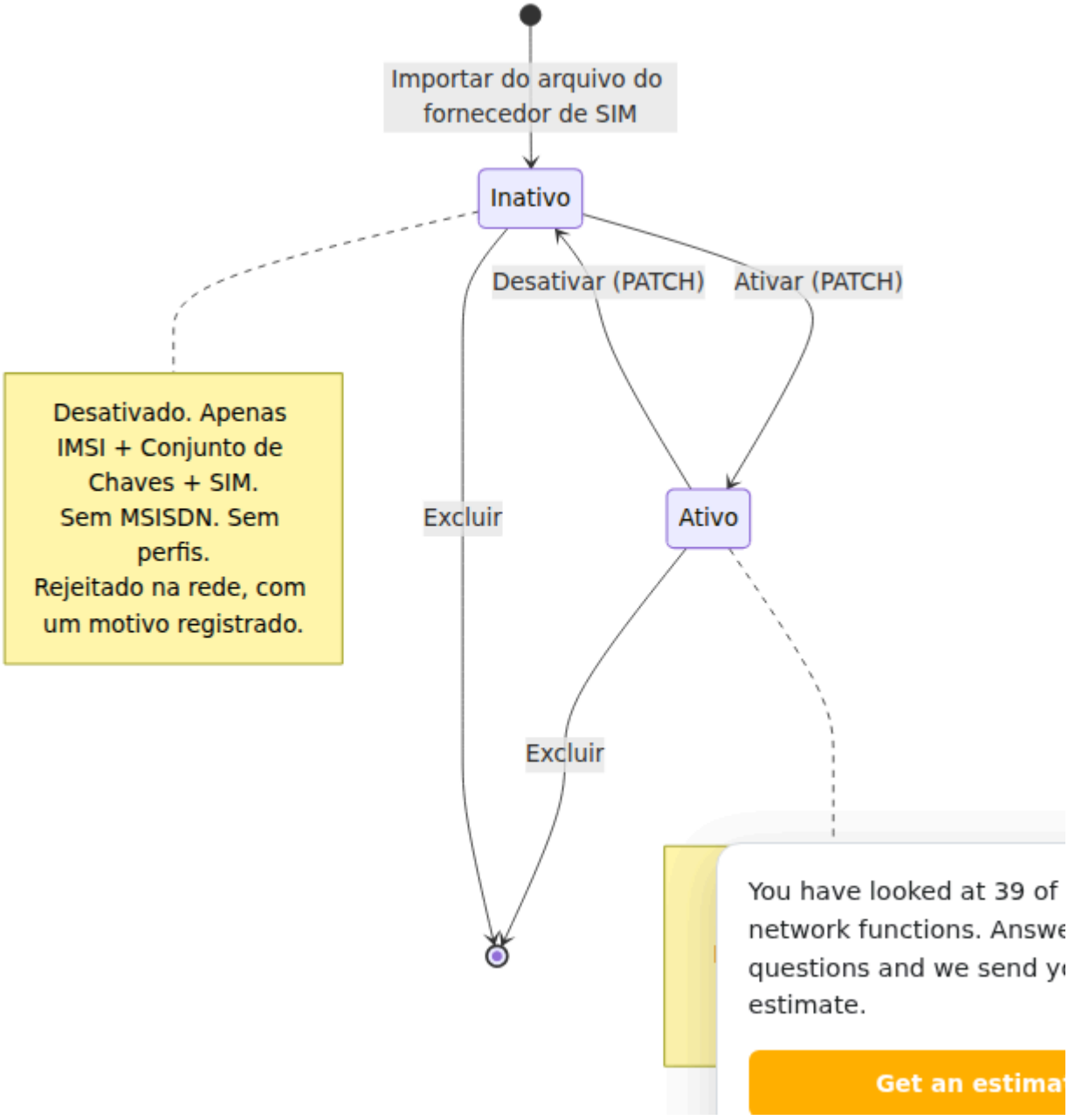
Pré-Provisionamento de Assinante Inativo

Use este fluxo de trabalho para carregar assinantes em massa a partir de um arquivo de saída de fornecedor de SIM e para ativar cada assinante posteriormente. Este é o modelo recomendado para a maioria dos operadores.

Um assinante inativo mantém apenas sua identidade e suas chaves de autenticação. Você adiciona o número de telefone e os perfis de serviço na ativação. Isso corresponde aos dados que um fornecedor de SIM envia para você. O arquivo do fornecedor de SIM contém o ICCID, o PIN, os códigos PUK e ADM, e as chaves de autenticação (Ki, OPc). O arquivo não contém um número de telefone ou um plano de serviço.

O banco de dados precisa apenas do IMSI e do Conjunto de Chaves para inserir um Assinante. Isso é o que torna um registro inativo possível. Isso não significa que os outros objetos são opcionais para o serviço. Cada objeto é necessário para seu próprio serviço: o perfil EPC para dados, o perfil IMS e um MSISDN para voz, e um SIM para autenticação. Um assinante inativo não funciona na rede até que você atribua o objeto correspondente. Você atribui esses objetos quando o cliente ativa o serviço. Para a razão de que o SIM, o Conjunto de Chaves e o Assinante são objetos separados, veja Conjunto de Chaves, SIM e Assinante.

Ciclo de Vida do Provisionamento



Etapa 1: Importar (criar o SIM, o Conjunto de Chaves e um Assinante desativado)

Crie um Conjunto de Chaves, um SIM e um Assinante desativado para cada linha no arquivo do fornecedor de SIM. Este exemplo usa `jq` para ler o identificador retornado de cada resposta.

```
# 1. Criar o Conjunto de Chaves a partir do arquivo do fornecedor
de SIM (Ki, OPc)
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Criar o SIM a partir do arquivo do fornecedor de SIM (ICCID,
PIN/PUK/ADM)
SIM_ID=$(curl -k -X POST https://hss.example.com:8443/api/sim \
  -H "Content-Type: application/json" \
  -d '{
    "iccid": "89440000000000000001",
    "is_esim": false,
    "ind_length": 5
  }' | jq -r '.response.id')

# 3. Criar o Assinante em um estado desativado.
#    imsi e key_set_id são necessários. Vincule o SIM. Adicione
nenhum MSISDN e nenhum perfil.
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001010000000001\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"sim_id\": \"$SIM_ID\",
    \"enabled\": false,
    \"ims_enabled\": false
  }" | jq -r '.response.id')
```

Um Assinante inativo é seguro para manter no banco de dados. O assinante não se conecta, autentica ou registra enquanto estiver desativado. Se a rede consultar um assinante inativo, o HSS rejeita a solicitação. Veja Comportamento do Assinante Inativo na Rede.

Etapa 2: Ativar (atribuir o número e os perfis, em seguida habilitar)

Para ativar um assinante, envie uma solicitação `PATCH`. Adicione o MSISDN, atribua os perfis e defina `enabled` como `true`.

```
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{
    \"enabled\": true,
    \"ims_enabled\": true,
    \"msisdns\": [\"$MSISDN_ID\"],
    \"epc_profile_id\": \"$EPC_PROFILE_ID\",
    \"ims_profile_id\": \"$IMS_PROFILE_ID\",
    \"roaming_profile_id\": \"$ROAMING_PROFILE_ID\"
  }"
```

Uma solicitação `PATCH` altera apenas os campos que você envia. O IMSI, o Conjunto de Chaves e o SIM permanecem como estão. Crie o MSISDN e os perfis primeiro, ou reutilize identificadores existentes. Veja Provisionamento Completo de Assinante para como criar os perfis.

Etapa 3: Desativar (desabilitar o assinante)

Para desativar um assinante, defina `enabled` como `false`. O registro do assinante e seus dados permanecem no banco de dados.

```
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d '{
    "enabled": false,
    "ims_enabled": false
  }'
```

Para remover apenas a voz IMS e manter o serviço de dados, defina `ims_enabled` como `false` e deixe `enabled` como `true`.

Comportamento do Assinante Inativo na Rede

Um assinante falha de forma fechada enquanto está inativo ou desativado. O HSS rejeita a solicitação na interface da rede. O HSS também escreve um log de aviso com o motivo exato. O motivo é um dos seguintes:

- O assinante está desativado (`enabled` é `false`).
- IMS está desativado para o assinante (`ims_enabled` é `false`).
- Um objeto necessário ainda não está atribuído. Por exemplo, não há perfil EPC para um anexo LTE, ou não há perfil IMS para registro IMS.

A rejeição usa o Código de Resultado Experimental `DIAMETER_ERROR_USER_UNKNOWN (5001)` nas interfaces S6a e Cx, conforme 3GPP TS 29.272 e 3GPP TS 29.229. O log de aviso nomeia o IMSI e o motivo exato. O log permite que você diferencie um assinante inativo de um IMSI desconhecido.

Provisionamento Completo de IP Estático

Este exemplo demonstra o provisionamento de um assinante com um endereço IP estático do zero.

Cenário: Provisionar um assinante de dispositivo IoT que precisa de um endereço IPv4 estático no APN "internet".

```
# Pré-requisitos: jq deve estar instalado (apt-get install jq ou
brew install jq)

# 1. Criar Conjunto de Chaves
KEY_SET_ID=$(curl -k -X POST
https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "authentication_algorithm": "milenage",
    "amf": "8000",
    "sqn": 0
  }' | jq -r '.response.id')

# 2. Criar Perfil de QoS APN
APN_QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "IoT Best Effort",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 10000,
    "apn_ambr_ul_kbps": 5000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": false,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Criar Identificador APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{
    "apn": "internet",
    "ip_version": "ipv4"
  }' | jq -r '.response.id')

# 4. Criar Perfil APN
APN_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
```

```
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$APN_QOS_ID\",
    \"name\": \"IoT Internet APN\"
  }" | jq -r '.response.id')

# 5. Criar IP Estático para o APN
STATIC_IP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profile_id\": \"$APN_PROFILE_ID\",
    \"ipv4_static_ip\": \"100.64.1.100\"
  }" | jq -r '.response.id')

# 6. Criar Perfil EPC
EPC_PROFILE_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_profiles\": [$APN_PROFILE_ID],
    \"name\": \"IoT Data Plan\",
    \"network_access_mode\": \"packet_only\",
    \"tracking_area_update_interval_seconds\": 600,
    \"ue_ambr_dl_kbps\": 10000,
    \"ue_ambr_ul_kbps\": 5000
  }" | jq -r '.response.id')

# 7. Criar MSISDN (número de telefone)
MSISDN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{
    "msisdn": "14155551000"
  }' | jq -r '.response.id')

# 8. Criar Assinante com IP Estático
SUBSCRIBER_ID=$(curl -k -X POST
https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"imsi\": \"001001999999999\",
    \"key_set_id\": \"$KEY_SET_ID\",
    \"epc_profile_id\": \"$EPC_PROFILE_ID\",
    \"msisdns\": [$MSISDN_ID],
```

```
    \"static_ips\": [$STATIC_IP_ID]
  }" | jq -r '.response.id')

echo "Assinante IoT provisionado com sucesso!"
echo "  ID do Assinante: $SUBSCRIBER_ID"
echo "  IMSI: 001001999999999"
echo "  MSISDN: 14155551000"
echo "  IPv4 Estático: 100.64.1.100 (no APN 'internet')"
```

O que isso cria:

Este fluxo de trabalho de provisionamento cria um assinante IoT completo com:

1. **Chaves criptográficas** (Conjunto de Chaves) - Para autenticação
2. **Configuração APN** (Perfil APN) - Ponto de acesso "internet"
3. **Atribuição de IP Estático** (IP Estático) - Endereço IPv4 fixo 100.64.1.100
4. **Perfil de serviço de dados** (Perfil EPC) - Limites de largura de banda otimizados para IoT
5. **Número de telefone** (MSISDN) - Para identificação do dispositivo
6. **Registro de assinante** (Assinante) - A entidade completa do assinante

Resultado:

Quando este assinante se conecta à rede e se conecta ao APN "internet", ele receberá o endereço IP estático `100.64.1.100` em vez de um endereço DHCP dinâmico.

Próximos Passos:

- Adicionar APNs adicionais com IPs estáticos: Repita as etapas 2-5 para cada APN
- Habilitar serviços de voz: Criar e atribuir Perfil IMS
- Configurar roaming: Criar e atribuir Perfil de Roaming
- Vincular SIM físico: Criar e atribuir SIM

Veja Também:

- Gerenciamento de IP Estático - Documentação detalhada sobre IP estático

- Provisionamento Completo de Assinante - Provisionamento básico sem IP estático
- Documentação Multi-MSISDN - Atribuindo múltiplos números de telefone

---

← Voltar para a Referência da API

Controle de Acesso à API (RBAC)

Esta página explica como controlar o acesso à API REST do OmniHSS. Ela aborda chaves de API, funções e permissões. Também fornece funções sugeridas para os sistemas que utilizam a API, como OmniCRM, OmniWeb, OmniHLR e ferramentas de monitoramento.

Voltar ao Guia de Operações. Para os endpoints da API, consulte a Referência da API.

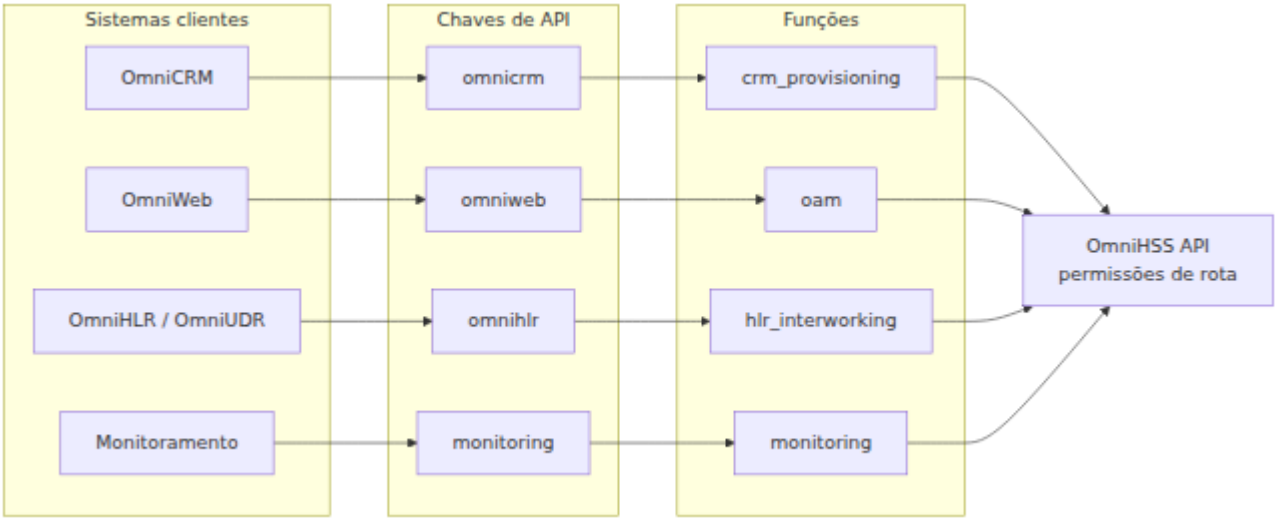
Índice

- Visão Geral
- Modos de Autenticação
- Como o OmniHSS Autoriza uma Solicitação
- Configuração
- Permissões
- Catálogo de Permissões do OmniHSS
- Funções Sugeridas
- Exemplo Completo de Configuração
- Usuário Atuante
- Chamando a API com uma Chave
- Ativar RBAC em um Sistema em Execução
- Rotacionar uma Chave
- Registro e Auditoria
- Solução de Problemas

Visão Geral

A API do OmniHSS pode alterar credenciais de assinantes, números de telefone e perfis de serviço. Ela também pode enviar operações de rede, por exemplo, uma Solicitação de Cancelamento de Localização para um MME. Assim, cada sistema cliente deve obter apenas o acesso que necessita.

Controle de acesso baseado em função (RBAC) fornece a cada sistema cliente sua própria **chave de API**. Cada chave possui uma ou mais **funções**. Cada função é uma lista de **padrões de permissão**. Cada rota da API possui exatamente uma **permissão**. O OmniHSS permite uma solicitação apenas se uma função da chave corresponder à permissão da rota. Todas as outras solicitações são recusadas. Isso é "negação padrão".



As interfaces Diameter (S6a, Cx, Sh, Gx, Rx e outras) não utilizam chaves de API. Os pares Diameter são controlados pela lista de pares na configuração Diameter. Consulte Configuração Diameter.

Modos de Autenticação

A API possui três modos de autenticação. O modo é definido no bloco `config :api_ex, :auth` em `runtime.exs`. O modo depende de quais chaves estão presentes nesse bloco.

`config :api_ex, :auth`	Modo	Comportamento
Não definido	Desativado	A API atende a cada rota para cada chamador sem chave. Um aviso é registrado no log na inicialização.
Contém `key` e não `api_keys`	Legado	Uma chave compartilhada fornece acesso total a todas as rotas.
Contém `api_keys`	RBAC	Cada cliente possui uma chave nomeada com funções. Solicitações que nenhuma função permite são recusadas.

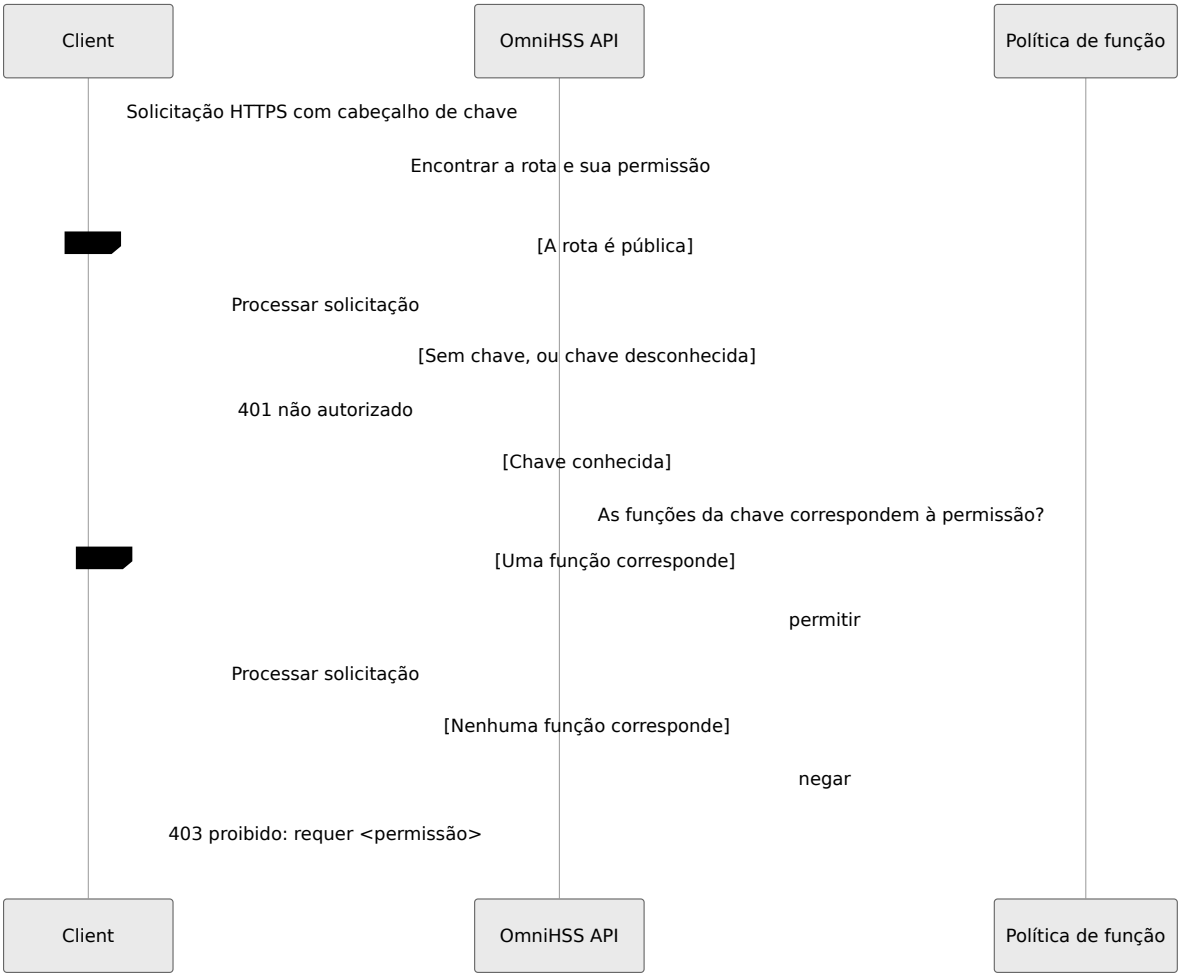
> **Aviso:** Não execute um OmniHSS de produção no modo desativado. Se o bloco `config :api_ex, :auth` não estiver definido, qualquer host que conseguir acessar a porta 8443 pode ler e alterar todos os dados de assinantes.

Use o modo RBAC para todos os sistemas de produção. Use o modo legado apenas como um passo curto durante a migração.

Regras do modo:

- Um valor vazio não desativa a autenticação. Por exemplo, `key: ""` é um erro de configuração. A API então responde a cada solicitação com `503`.
- O OmniHSS define o modo na inicialização. Se a configuração mudar para um modo inferior enquanto o serviço estiver em execução, a API responde a cada solicitação com `503`. Reinicie o serviço para usar um modo inferior.
- No modo RBAC, uma configuração inválida impede a API de iniciar. O log fornece uma lista numerada de cada problema. A API nunca inicia sem proteção.

Como o OmniHSS Autoriza uma Solicitação



O OmniHSS toma a decisão antes de ler o corpo da solicitação. Um chamador que recebe `401`, `403` ou `503` não pode fazer o servidor ler um corpo de solicitação grande.

Configuração

Coloque o bloco `config :api_ex, :auth` no arquivo de configuração em tempo de execução `runtime.exs`. Em um host implantado, este arquivo é `/etc/omnihss/runtime.exs`. Reinicie o OmniHSS após alterar o bloco.

Estrutura da Configuração

```
# runtime.exs
config :api_ex, :auth,
  header: "x-omnihss-key",
  user_header: "x-omnihss-user",
  public: ["status/api:read"],
  roles: %{
    admin: ["*"],
    monitoring: ["status/**:read", "subscriber/**:read"]
  },
  api_keys: [
    %{
      name: "monitoring",
      roles: [:monitoring],
      secrets: ["REPLACE-WITH-A-64-CHARACTER-RANDOM-HEX-SECRET-
FROM-OPENSSL-RAND"]
    },
    %{
      name: "break-glass",
      roles: [:admin],
      secrets: ["REPLACE-WITH-A-DIFFERENT-64-CHARACTER-RANDOM-HEX-
SECRET-VALUE"]
    }
  ],
  log: [deny: :warning, unauthenticated: :info, allow: false,
not_found: false]
```

Parâmetros de Autenticação

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`header`	String	Não	`"x-api-key"`	O cabeçalho HTTP que transporta a chave da API. O nome não é sensível a maiúsculas. Use `"x-omnihss-key"`. Não use o padrão `"x-api-key"`, pois o OmniWeb usa `X-API-Key` para suas próprias chaves. Consulte OmniWeb.
`user_header`	String	Não	Não definido	O cabeçalho HTTP que nomeia o usuário humano para quem um cliente atua. Deve ser diferente de `header`. Consulte Usuário Atuante.
`public`	Lista de strings	Não	`[]`	Padrões de permissão que os chamadores podem usar sem chave. Use isso apenas para verificações de saúde. Mantenha os padrões

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				públicos somente para leitura.
`roles`	Mapa	Sim (RBAC)	-	Nomes de funções e seus padrões de permissão. Cada chave do mapa é um nome de função. Cada valor é uma lista de padrões. Consulte Permissões.
`api_keys`	Lista de mapas	Sim (RBAC)	-	As chaves da API. A presença deste parâmetro seleciona o modo RBAC. Consulte Parâmetros da Chave da API.
`log`	Lista de palavras-chave	Não	Veja a descrição	O nível de log para cada tipo de decisão. Consulte Parâmetros de Log.
`key`	String	Apenas legado	-	A única chave compartilhada do modo legado. Não defina `key` e `api_keys` juntos. Remova `key` ao migrar para RBAC.

Parâmetros da Chave da API

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`name`	String	Sim	-	Um nome único para a chave. Use o nome do sistema cliente, por exemplo, `"omnicrm"`. O nome aparece nos logs. O segredo nunca aparece nos logs.
`roles`	Lista de átomos	Sim	-	As funções da chave. Cada função deve ser definida em `roles`. Uma chave obtém as permissões de todas as suas funções.
`secrets`	Lista de strings	Sim	-	Um ou mais segredos para a chave. O cliente envia um desses segredos no cabeçalho da chave. Dois segredos permitem rotacionar uma

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				chave sem tempo de inatividade. Consulte Rotacionar uma Chave. Cada segredo deve ser único entre todas as chaves.
`may_assert_user`	Booleano	Não	`false`	Se `true`, a chave pode nomear um usuário humano em `user_header`. Defina isso apenas para sistemas que têm seu próprio login de usuário, por exemplo, OmniWeb e OmniCRM.
`metadata`	Mapa	Não	`%{}`	Dados livres sobre a chave, por exemplo, um proprietário ou um número de ticket. O OmniHSS não usa esses dados

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
				para tomar decisões.

Regras de segredo:

- Use um segredo longo e aleatório. Crie um segredo com o comando `openssl rand -hex 32`. Isso gera 64 caracteres hexadecimais.
- O OmniHSS registra um aviso na inicialização para um segredo menor que 16 bytes. Dezesseis bytes é um mínimo, não uma recomendação.
- Dê a cada sistema cliente sua própria chave. Não compartilhe uma chave entre dois sistemas. Se você compartilhar uma chave, não poderá remover o acesso de apenas um sistema, e os logs não poderão diferenciar os dois sistemas.

Parâmetros de Log

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`deny`	Nível de log ou `false`	Não	`:warning`	O nível de log para uma solicitação que as funções de uma chave não permitem (`403`).
`unauthenticated`	Nível de log ou `false`	Não	`:info`	O nível de log para uma solicitação sem chave ou com uma chave errada (`401`).
`allow`	Nível de log ou `false`	Não	`false`	O nível de log para uma solicitação permitida. Defina `:info` para registrar cada chamada de API permitida.
`not_found`	Nível de log ou `false`	Não	`false`	O nível de log para uma solicitação de uma chave conhecida para um caminho que não existe (`404`).

Um nível de log é `:debug`, `:info`, `:warning` ou `:error`. O valor `false` desativa a linha de log para essa decisão.

Permissões

Formato de Permissão

Cada rota possui uma permissão no formato `resource:verb`.

- O **recurso** é o caminho da API sem o prefixo `/api` e sem os parâmetros de caminho. Por exemplo, a rota `GET /api/subscriber/imsi/:imsi` tem o recurso `subscriber/imsi`.
- O **verbo** vem do método HTTP.

Método HTTP	Verbo
`GET` (lista e registro único)	`read`
`POST`	`write`
`PUT`, `PATCH`	`write`
`DELETE`	`delete`

Assim, `POST /api/operation/s6a/clr` tem a permissão `operation/s6a/clr:write`, e `DELETE /api/subscriber/:id` tem a permissão `subscriber:delete`.

Formato de Padrão

Uma função é uma lista de padrões. Um padrão tem o mesmo formato `resource:verb`, e pode conter curingas.

Curinga	Posição	Combina
`*` (padrão inteiro)	Sozinho	Todas as permissões.
`*` (segmento de recurso)	Qualquer segmento	Exatamente um segmento. No final do recurso, combina com um ou mais segmentos.
`**`	Apenas último segmento	Zero ou mais segmentos.
`*` (verbo)	Após `:`	Todos os verbos.

Um segmento é o texto entre dois caracteres `/`. Um `*` não é uma combinação de prefixo. Por exemplo, `sub*` não é um padrão válido.

Exemplos de Padrão

Os curingas `*` e `**` dão resultados diferentes. Use esta tabela ao escrever uma função.

Padrão	`subscriber:read`	`subscriber/imsi:read`	`subscr`
`subscriber:read`	Sim	Não	Não
`subscriber:*`	Sim	Não	Sim
`subscriber/*:read`	Não	Sim	Não
`subscriber/**:read`	Sim	Sim	Não
`subscriber/**:*`	Sim	Sim	Sim
`*:read`	Sim	Sim	Não
`*`	Sim	Sim	Sim

Pontos-chave:

- `subscriber/*:read` dá as rotas de pesquisa (`subscriber/imsi`, `subscriber/msisdn` e outras), mas não a lista `subscriber`.
- `subscriber/**:read` dá as rotas `subscriber` e todas as rotas abaixo de `subscriber`.
- As funções apenas adicionam permissões. Não há padrão de "negação". Para remover uma permissão de uma função, escreva a função com uma lista menor de padrões.

Catálogo de Permissões do OmniHSS

Esta tabela mostra todas as permissões da API do OmniHSS. Para o formato de solicitação e resposta de cada rota, consulte a Referência da API.

Status e Documentação

Permissão	Rotas	Notas
`status/api:read`	`GET /api/status/api`	Saúde da API. Um bom candidato para `public`.
`status/license:read`	`GET /api/status/license`	Status da licença.
`status/database:read`	`GET /api/status/database`	Saúde do banco de dados.
`docs:read`	`GET /api/docs`, `GET /docs`	Swagger UI.
`schema:read`	`GET /api/schema`, `GET /schema`	Documento OpenAPI.
`version:read`	`GET /version`	Versão e build do software.

Dados do Assinante

Permissão	Rotas
`subscriber:read`	`GET /api/subscriber`, `GET /api/subscriber/:id`
`subscriber:write`	`POST /api/subscriber`, `PUT` e `PATCH /api/subscriber/:id`
`subscriber:delete`	`DELETE /api/subscriber/:id`
`subscriber/imsi:read`	`GET /api/subscriber/imsi/:imsi`
`subscriber/iccid:read`	`GET /api/subscriber/iccid/:iccid`
`subscriber/msisdn:read`	`GET /api/subscriber/msisdn/:msisc`
`subscriber/ip:read`	`GET /api/subscriber/ip/:ip_addres`
`subscriber/repository_data:read`	`GET /api/subscriber/repository_data/:`
`subscriber/repository_data:write`	`PUT` e `PATCH /api/subscriber/repository_data/:`

Permissão	Rotas
`subscriber/repository_data:delete`	`DELETE /api/subscriber/repository_data/:`
`msisdn:read`	`GET /api/msisdn`, `GET /api/msisdn/`
`msisdn:write`	`POST /api/msisdn`, `PUT` e `PATCH /api/msisdn/:id`
`msisdn:delete`	`DELETE /api/msisdn/:id`
`msisdn/digits:read`	`GET /api/msisdn/digits/:digits`
`ims_subscription:read`, `:write`, `:delete`	`/api/ims_subscription`
`epc/static_ip:read`, `:write`, `:delete`	`/api/epc/static_ip`

Credenciais

Permissão	Rotas	Notas
`key_set:read`	`GET /api/key_set`, `GET /api/key_set/:id`	A resposta não contém Ki, OPc, OP, PSK, DES ou a senha SIP. Esses campos são removidos de cada leitura.
`key_set:write`	`POST /api/key_set`, `PUT` e `PATCH /api/key_set/:id`	**Sensível.** Um chamador com esta permissão pode definir ou substituir as chaves de autenticação de um assinante.
`key_set:delete`	`DELETE /api/key_set/:id`	**Sensível.**
`sim:read`	`GET /api/sim`, `GET /api/sim/:id`	A resposta não contém KIc, KID ou ADM1.
`sim:write`	`POST /api/sim`, `PUT` e `PATCH /api/sim/:id`	
`sim:delete`	`DELETE /api/sim/:id`	
`sim/iccid:read`	`GET /api/sim/iccid/:iccid`	Pesquisa por ICCID.

Perfis e Regras

Cada recurso nesta tabela possui as três permissões `read`, `write` e `delete`. As rotas são `GET`, `POST`, `PUT`, `PATCH` e `DELETE` no caminho. Uma alteração em um perfil pode alterar o serviço de muitos assinantes ao mesmo tempo.

Recurso	Caminho	Des
`apn/profile`	`/api/apn/profile`	Perfis Consu Geren de Per
`apn/identifier`	`/api/apn/identifier`	Nome
`apn/qos_profile`	`/api/apn/qos_profile`	QoS A
`epc/profile`	`/api/epc/profile`	Perfis (dado
`ims/profile`	`/api/ims/profile`	Perfis (voz).
`roaming/profile`	`/api/roaming/profile`	Perfis roami Consu Contr Roami
`roaming/rule`	`/api/roaming/rule`	Regra roami
`eir/rule`	`/api/eir/rule`	Regra Consu
`pcrf/profile`	`/api/pcrf/profile`	Perfis Consu
`pcrf/charging_rule`	`/api/pcrf/charging_rule`	Regra cobra
`pcrf/flow_information_rule`	`/api/pcrf/flow_information_rule`	Regra fluxo

Interoperabilidade HLR

Permissão	Rotas
`hlr/generate_auth_vectors:write`	`POST /api/hlr/generate_auth_`
`hlr/generate_eutran_auth_vectors:write`	`POST /api/hlr/generate_eutran_aut`
`hlr/circuit_session:write`, `:delete`	`PUT`, `PATCH`, `DELETE /api/hlr/circuit_session/:im`
`hlr/subscriber_state:write`	`PUT` e `PATCH /api/hlr/subscriber_state/:i`
`hlr/pdn_session:write`, `:delete`	`POST`, `PUT`, `PATCH`, `DELETE /api/hlr/pdn_session`

Operações de Rede

Permissão	Rotas	Not
`operation/count:read`	`GET /api/operation/count/:type`	Contage registros
`operation/s6a/clr:write`	`POST /api/operation/s6a/clr`	Envia um Solicitaç Cancelar de Locali para o M assinant desliga. Consulte Fluxos de Protocol
`operation/s6a/idr:write`	`POST /api/operation/s6a/idr`	Envia um Solicitaç Inserção Dados do Assinant o MME.
`operation/pcrf/re_auth:write`	`POST /api/operation/pcrf/re_auth`	Envia um Solicitaç Reautent Gx para PGW. Co PCRF.

Funções Sugeridas

Essas funções fornecem a cada sistema cliente as rotas que ele utiliza, e nada mais. Comece com essas funções. Em seguida, torne cada função menor se um cliente não usar um recurso.

Função	Sistemas clientes	Acesso
`admin`	Chave de quebra de vidro para emergências	Todas as rotas.
`crm_provisioning`	OmniCRM	Provisionamento de assinantes, MSISDN, SIM e conjunto de chaves. Somente leitura para perfis.
`oam`	OmniWeb	Todos os dados de assinantes, perfis e operações de rede. Sem estado de sessão HLR.
`hlr_interworking`	OmniHLR (OmniSS7), OmniUDR	Pesquisas de assinantes, vetores de autenticação, estado comutado.
`monitoring`	Exportadores, painéis, scripts que apenas leem	Pesquisas de assinantes e status somente leitura.
`eir_lookup`	OmniEIR	Ler apenas as regras EIR.

Admin (Quebra de Vidro)

```
admin: ["*"]
```

Como funciona: A chave pode usar todas as rotas, incluindo novas rotas em versões posteriores.

Caso de uso: Acesso de emergência por um operador. Mantenha o segredo em um cofre de senhas. Não configure um sistema cliente com esta chave. Verifique os logs para cada uso desta chave.

OmniCRM (Provisionamento)

```
crm_provisioning: [
  "subscriber:*",
  "subscriber/*:read",
  "msisdn/**:*",
  "sim/**:read",
  "sim:write",
  "key_set:write",
  "status/**:read"
]
```

Como funciona: O OmniCRM cria, altera e exclui assinantes e MSISDNs. Ele cria SIMs e conjuntos de chaves quando provisiona um novo serviço. Ele pesquisa assinantes por IMSI e por endereço IP do UE. Não pode alterar perfis, regras ou operações de rede. Não pode excluir conjuntos de chaves ou SIMs.

Caso de uso: O CRM provisiona serviços para clientes. Defina `may_assert_user: true` na chave do OmniCRM para que o CRM possa nomear o membro da equipe que fez a alteração. Consulte Usuário Atuante.

Se o CRM precisar enviar uma Solicitação de Cancelamento de Localização após uma troca de SIM, adicione `"operation/s6a/clr:write"`.

OmniWeb (OAM)

```
oam: [
  "*:read",
  "subscriber/**:*",
  "msisdn/**:*",
  "sim/**:*",
  "key_set:*",
  "ims_subscription:*",
  "apn/**:*",
  "epc/**:*",
  "ims/**:*",
  "roaming/**:*",
  "eir/**:*",
  "pcrf/**:*",
  "operation/**:write",
  "hlr/subscriber_state:write",
  "hlr/generate_auth_vectors:write",
  "hlr/generate_eutran_auth_vectors:write"
]
```

Como funciona: O OmniWeb gerencia todos os dados de assinantes, perfis e regras. Ele envia operações de rede (CLR, IDR e Reautenticação Gx). Ele pode gerar vetores de autenticação de teste. Não pode alterar o estado da sessão comutada ou PDN que o HLR possui.

Caso de uso: O OmniWeb é o sistema OAM. O OmniWeb tem seus próprios usuários e seu próprio RBAC para cada elemento. O OmniWeb verifica seu próprio RBAC antes de enviar uma solicitação ao OmniHSS. Assim, a chave do OmniWeb no OmniHSS deve cobrir todos os recursos que o OmniWeb exibe. Defina `may_assert_user: true` na chave do OmniWeb para que o OmniHSS possa registrar o usuário do OmniWeb.

> **Cuidado:** Não use `x-api-key` como o cabeçalho da chave OmniHSS. O OmniWeb usa `X-API-Key` para suas próprias chaves de API pessoais, e o proxy do OmniWeb envia esse cabeçalho para o OmniHSS. Defina `header: "x-omnihss-key"`, para que as duas chaves não entrem em conflito.

OmniHLR e OmniUDR (Interoperabilidade HLR)

```
hlr_interworking: [
  "subscriber/imsi:read",
  "subscriber/msisdn:read",
  "key_set:read",
  "hlr/generate_auth_vectors:write",
  "hlr/generate_eutran_auth_vectors:write",
  "hlr/circuit_session:*",
  "status/api:read"
]
```

Como funciona: O HLR e o UDR consultam assinantes, solicitam vetores de autenticação e atualizam a localização de comutação de circuitos. Eles não podem alterar dados ou perfis de assinantes.

Caso de uso: OmniHLR (OmniSS7) para 2G/3G, e OmniUDR para 5G. Esses sistemas recebem vetores que contêm CK e IK. Não forneça essas chaves a nenhum outro sistema.

Monitoramento (Somente Leitura)

```
monitoring: [
  "status/**:read",
  "subscriber/**:read",
  "operation/count:read"
]
```

Como funciona: A chave pode ler status, consultas de assinantes e contagens de registros. Não pode alterar dados.

Caso de uso: Exportadores, painéis e relatórios. As leituras de assinantes contêm IMSIs e MSISDNs, que são dados pessoais. Se uma ferramenta precisa apenas de status, use `"status/**:read"` sozinha.

OmniEIR (Consulta EIR)

```
eir_lookup: ["eir/rule:read"]
```

Como funciona: A chave pode ler apenas as regras do EIR.

Caso de uso: OmniEIR lê as regras que o OmniHSS armazena.

Ferramentas de Laboratório e Teste

As ferramentas de teste, por exemplo, o Simulador RAN, criam e excluem assinantes e perfis. Em um laboratório, forneça a cada ferramenta de teste sua própria chave com o papel `admin`. Não instale chaves de ferramentas de teste em sistemas de produção.

Exemplo Completo de Configuração

Este exemplo usa todos os Papéis Sugeridos. Cada parâmetro é explicado em Configuração.

```
# runtime.exs
config :api_ex, :auth,
  # Não use x-api-key: o OmniWeb envia seu próprio cabeçalho X-
API-Key através de seu proxy
  header: "x-omnihss-key",
  user_header: "x-omnihss-user",

  # Verificação de saúde sem chave
  public: ["status/api:read"],

  roles: %{
    admin: ["*"],
    crm_provisioning: [
      "subscriber:*",
      "subscriber/*:read",
      "msisdn/**:*",
      "sim/**:read",
      "sim:write",
      "key_set:write",
      "status/**:read"
    ],
    oam: [
      "*:read",
      "subscriber/**:*",
      "msisdn/**:*",
      "sim/**:*",
      "key_set:*",
      "ims_subscription:*",
      "apn/**:*",
      "epc/**:*",
      "ims/**:*",
      "roaming/**:*",
      "eir/**:*",
      "pcrf/**:*",
      "operation/**:write",
      "hlr/subscriber_state:write",
      "hlr/generate_auth_vectors:write",
      "hlr/generate_eutran_auth_vectors:write"
    ],
    hlr_interworking: [
      "subscriber/imsi:read",
      "subscriber/msisdn:read",
      "key_set:read",
```

```
      "hlr/generate_auth_vectors:write",
      "hlr/generate_eutran_auth_vectors:write",
      "hlr/circuit_session:*",
      "status/api:read"
    ],
    monitoring: ["status/**:read", "subscriber/**:read",
"operation/count:read"],
    eir_lookup: ["eir/rule:read"]
  },

  api_keys: [
    %{
      name: "omnicrm",
      roles: [:crm_provisioning],
      may_assert_user: true,
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNICRM-KEY-00"]
    },
    %{
      name: "omniweb",
      roles: [:oam],
      may_assert_user: true,
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIWEB-KEY-00"]
    },
    %{
      name: "omnihlr",
      roles: [:hlr_interworking],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIHLR-KEY-00"]
    },
    %{
      name: "omniudr",
      roles: [:hlr_interworking],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
OMNIUDR-KEY-00"]
    },
    %{
      name: "monitoring",
      roles: [:monitoring],
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-THE-
MONITOR-KEY-00"]
    },
    %{
```

```
      name: "break-glass",
      roles: [:admin],
      metadata: %{owner: "NOC", vault: "omnihss/break-glass"},
      secrets: ["REPLACE-WITH-OPENSSL-RAND-HEX-32-OUTPUT-FOR-
BREAK-GLASS-KEY-00"]
    }
  ],

  log: [deny: :warning, unauthenticated: :info, allow: false,
not_found: false]
```

Como funciona: O OmniHSS inicia no modo RBAC, porque `api_keys` está presente. Cada sistema cliente tem sua própria chave. Os balanceadores de carga podem verificar `GET /api/status/api` sem chave. Todas as outras rotas precisam de uma chave, e a chave precisa de um papel que corresponda à rota.

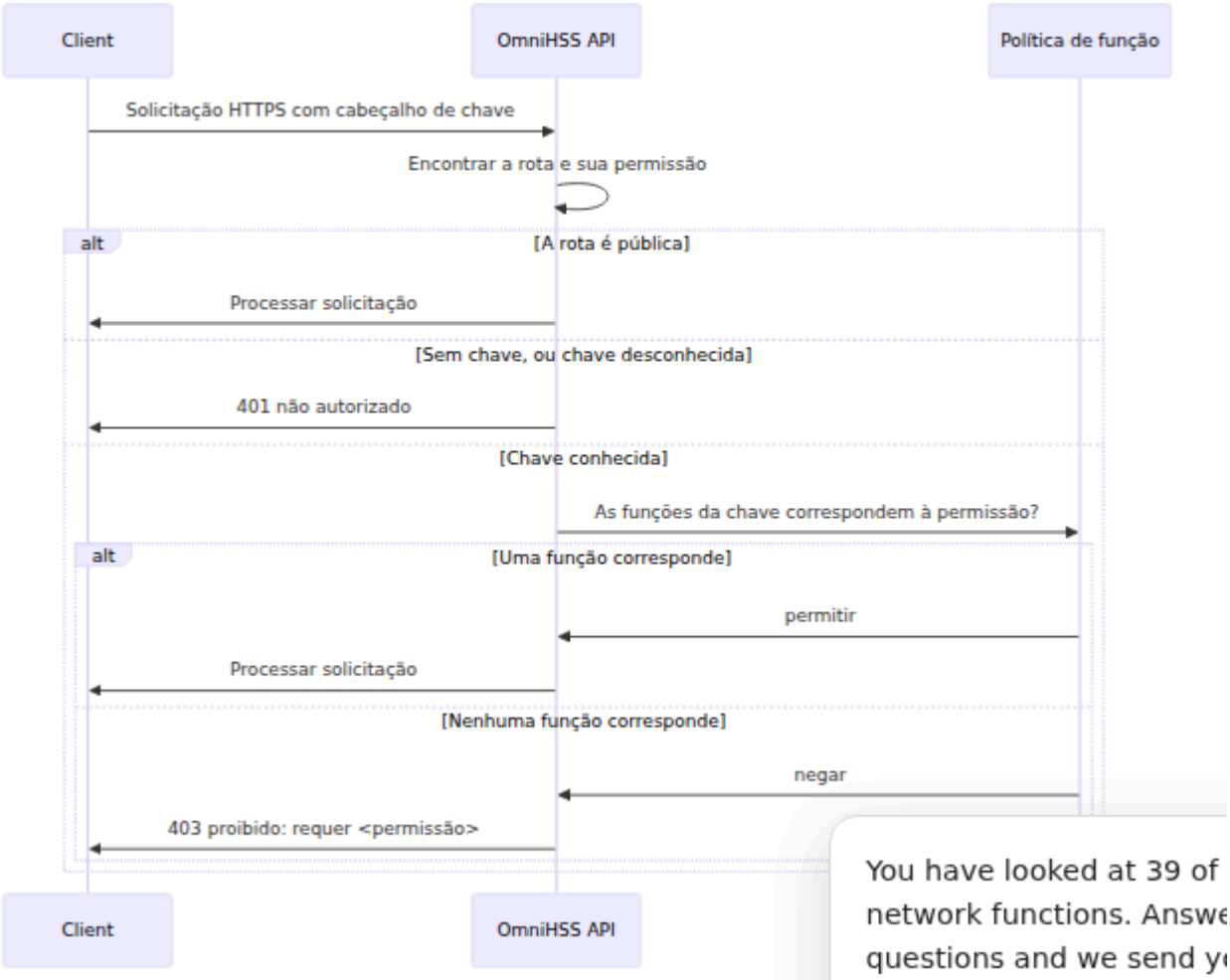
Quando o serviço inicia, o log mostra uma linha de resumo. A linha fornece o modo, os nomes dos cabeçalhos, os nomes das chaves, os papéis e o número de avisos. Nunca mostra um segredo. Por exemplo:

```
[info] ApiEx: autenticação de chave compartilhada habilitada
(modo: rbac, cabeçalho: x-omnihss-key, user_header: x-omnihss-
user, skip_paths: [])
  keys: omnicrm [crm_provisioning, asserts user], omniweb [oam,
asserts user], omnihlr [hlr_interworking], ...
```

Verifique esta linha após cada alteração. Se a linha mostrar `mode: legacy` ou um aviso sobre autenticação desativada, o parâmetro `api_keys` não está carregado.

Usuário Ativo

Um sistema cliente pode agir em nome de um usuário humano. Por exemplo, um usuário do OmniWeb altera um assinante, e o OmniWeb envia a solicitação com sua própria chave. O cliente pode enviar o nome do usuário em `user_header`.



Regras:

- O OmniHSS aceita o nome do usuário apenas se a chave tiver `may_assert_user: true`. Para outras chaves, o OmniHSS ignora o cabeçalho.
- O nome do usuário deve ser UTF-8 válido, no máximo 256 bytes, e não conter caracteres de controle.
- O usuário ativo é um registro de auditoria. Não é um login. O OmniHSS confia no sistema cliente para enviar o nome de usuário correto. Assim, defina `may_assert_user: true` apenas para sistemas que têm seu próprio login de usuário.

Chamando a API com uma Chave

Envie a chave no cabeçalho configurado. Envie o cabeçalho apenas uma vez. Se a solicitação tiver duas cópias do cabeçalho, o OmniHSS a rejeita.

```
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001010000000001 \
  -H "x-omnihss-key: <segredo>"
```

Na interface do Swagger (`/api/docs`), clique em **Autorizar** e cole o segredo. O Swagger UI então envia a chave com cada solicitação "Experimente". A chave precisa de `docs:read` e `schema:read` para abrir a interface do Swagger. O padrão `*:read` contém ambos.

Códigos de Resposta

Todos os erros usam o formato usual da API `{"status":"error","response":"..."}`.

Status	`response`	Causa
`401`	`unauthorized`	Sem chave, uma chave errada ou duas cópias do cabeçalho da chave.
`403`	`forbidden: requires <permission>`	Os papéis da chave não correspondem à permissão da rota. A resposta nomeia a permissão.
`403`	`forbidden: acting user required`	A rota precisa de um usuário ativo, e a solicitação não nomeou um.
`503`	`authentication misconfigured`	A configuração não é válida ou mudou para um modo inferior enquanto o serviço estava em execução.
`404`	`Not Found`	Uma chave conhecida pediu um caminho que não existe.

Habilitar RBAC em um Sistema em Execução

Quando você habilita o RBAC, cada cliente que não enviar uma chave válida recebe `401`. Assim, prepare todos os clientes antes de reiniciar o OmniHSS.

1. Faça uma lista de cada sistema que chama a API OmniHSS neste site.
2. Selecione um papel dos Papéis Sugeridos para cada sistema.
3. Faça um segredo para cada sistema com `openssl rand -hex 32`.
4. Configure cada sistema cliente para enviar seu segredo no cabeçalho `x-omnihss-key`.
5. Adicione o bloco `config :api_ex, :auth` ao `runtime.exs` em cada host do OmniHSS.
6. Reinicie o OmniHSS em um host.
7. Verifique a linha do log de inicialização. Certifique-se de que mostra `mode: rbac` e os nomes corretos das chaves.
8. Procure linhas `403` e `401` no log. Cada linha `403` nomeia a chave e a permissão que ela precisa.
9. Se um cliente precisar de uma permissão que seu papel não possui, adicione o padrão ao papel. Em seguida, reinicie o OmniHSS.
10. Repita os passos 6 a 9 para cada outro host do OmniHSS.

Se um sistema cliente não puder enviar um cabeçalho de chave ainda, não habilite o RBAC nos hosts que o cliente usa. Esse cliente recebe `401` para cada solicitação.

Em uma implantação em múltiplos locais, use o mesmo bloco `config :api_ex, :auth` em todos os hosts do OmniHSS. Um cliente pode então usar a mesma chave em cada site.

Rotacionar uma Chave

Uma chave pode ter mais de um segredo. Use isso para mudar um segredo sem tempo de inatividade.

1. Faça um novo segredo com `openssl rand -hex 32`.
2. Adicione o novo segredo à lista de `secrets` da chave. Mantenha o segredo antigo na lista.
3. Reinicie o OmniHSS. Ambos os segredos agora funcionam.
4. Altere o sistema cliente para enviar o novo segredo.
5. Remova o segredo antigo da lista de `secrets`.
6. Reinicie o OmniHSS. Apenas o novo segredo funciona.

```
%{
  name: "omnicrm",
  roles: [:crm_provisioning],
  may_assert_user: true,
  secrets: [
    "REPLACE-WITH-THE-OLD-OMNICRM-SECRET-UNTIL-THE-CRM-USES-THE-
NEW-1",
    "REPLACE-WITH-THE-NEW-OMNICRM-SECRET-FROM-OPENSSL-RAND-HEX-32-
002"
  ]
}
```

Para remover todo o acesso de um sistema, remova sua chave de `api_keys`. Em seguida, reinicie o OmniHSS.

Registro e Auditoria

- Cada linha de log que o OmniHSS escreve após identificar uma chave contém `api_ex_principal=<nome da chave>`. Use isso para descobrir qual sistema fez uma alteração.
- Uma linha `403` fornece o nome da chave, o padrão da rota e a permissão que a chave precisa. A linha fornece o padrão da rota, por exemplo `/api/subscriber/:imsi`, não o caminho completo.
- Segredos nunca aparecem em linhas de log ou em respostas de erro.
- Para registrar cada solicitação permitida, defina `log: [allow: :info]`. Isso fornece um registro de auditoria de todas as alterações da API, mas adiciona uma linha de log para cada solicitação.

Solução de Problemas

Todos os Clientes Recebem 401 Após um Reinício

Sintomas: Todos os clientes recebem `401 unauthorized`.

Possíveis causas:

- Os clientes não enviam o cabeçalho da chave.
- Os clientes enviam a chave em um cabeçalho diferente, por exemplo, `x-api-key`.
- Os segredos nos clientes não correspondem aos segredos em `runtime.exs`.

Resolução:

1. Verifique a linha do log de inicialização para o nome do cabeçalho configurado.
2. Certifique-se de que cada cliente envia sua chave nesse cabeçalho.
3. Compare o segredo na configuração do cliente com o segredo em `api_keys`.

Um Cliente Recebe 403

Sintomas: Um cliente recebe `403 forbidden: requires <permission>`.

Possíveis causas:

- O papel da chave não contém um padrão para essa permissão.
- O padrão usa `*` onde `**` é necessário. Veja Exemplos de Padrões.

Resolução:

1. Encontre a permissão na resposta ou na linha do log.
2. Encontre a rota no Catálogo de Permissões do OmniHSS.

3. Se o cliente deve usar a rota, adicione um padrão para a permissão ao papel da chave.
4. Reinicie o OmniHSS.

A API Responde 503 a Cada Solicitação

Sintomas: Cada solicitação recebe `503 authentication misconfigured`.

Possíveis causas:

- O parâmetro legado `key` tem um valor vazio.
- A configuração mudou para um modo inferior enquanto o serviço estava em execução.

Resolução:

1. Procure no log a linha que lista os problemas de configuração.
2. Corrija o bloco `config :api_ex, :auth`.
3. Reinicie o OmniHSS.

OmniHSS Não Inicia Após a Alteração

Sintomas: O OmniHSS para na inicialização. O log mostra `ApiEx: invalid RBAC configuration`.

Possíveis causas:

- Uma chave usa um papel que não está definido em `roles`.
- Duas chaves têm o mesmo `name`.
- Duas chaves compartilham um segredo.
- `user_header` é o mesmo que `header`.
- Uma chave não tem segredo.

Resolução:

1. Leia a lista numerada de problemas sob a linha do log.
2. Corrija cada problema em `runtime.exs`.

3. Reinicie o OmniHSS.

O Log de Inicialização Mostra um Aviso

Sintomas: O log de inicialização contém linhas `ApiEx auth warning:`.

Possíveis causas:

- Um segredo é mais curto que 16 bytes.
- Um padrão de papel não corresponde a nenhuma rota. Isso geralmente é um erro de digitação no padrão.
- Um padrão `public` abre uma rota que pode alterar dados.

Resolução:

1. Leia cada linha de aviso.
2. Faça segredos longos com `openssl rand -hex 32`.
3. Corrija a ortografia do padrão. Compare-o com o Catálogo de Permissões do OmniHSS.
4. Mantenha os padrões `public` somente para leitura.

Referência da API OmniHSS

← Voltar ao Guia de Operações

Índice

- Visão Geral da API
- Autenticação e Controle de Acesso
- Gerenciamento de Conjuntos de Chaves
- Gerenciamento de Assinantes
- Dados do Repositório Sh (Dados Transparentes)
- Gerenciamento de MSISDN
- Gerenciamento de Assinaturas IMS
- Gerenciamento de SIM
- Gerenciamento de Conjuntos de Chaves
- Gerenciamento de Perfis
- Gerenciamento de IPs Estáticos
- Gerenciamento de Roaming
- Gerenciamento de EIR
- Status e Saúde
- Tratamento de Erros
- Exemplos de Uso da API

Visão Geral da API

URL Base

```
https://[hostname]:8443/api
```

Autenticação

Cada solicitação deve conter uma chave da API no cabeçalho de chave configurado, por exemplo `x-omnihss-key`. Cada chave tem funções que controlam quais rotas ela pode usar. Para a configuração, a permissão de cada rota e funções sugeridas para cada sistema cliente, veja Controle de Acesso da API (RBAC).

Formato da Solicitação

- **Content-Type:** `application/json`
- **Protocolo:** Apenas HTTPS
- **Porta:** 8443 (configurável)

Importante: Todos os endpoints da API esperam cargas úteis JSON "planas" sem objetos de encapsulamento.

Formato Correto:

```
{
  "name": "value",
  "field": "value"
}
```

Formato Incorreto (Não Use):

```
{
  "subscriber": {
    "name": "value",
    "field": "value"
  }
}
```

Exemplo:

```
# ✓ Correto
curl -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{"name": "default", "ifc_template": "..."}'

# ✗ Incorreto
curl -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{"ims_profile": {"name": "default", "ifc_template": "..."}}'
```

Formato da Resposta

Todas as respostas são JSON com a seguinte estrutura:

Resposta de Sucesso:

```
{
  "status": "success",
  "response": { ... }
}
```

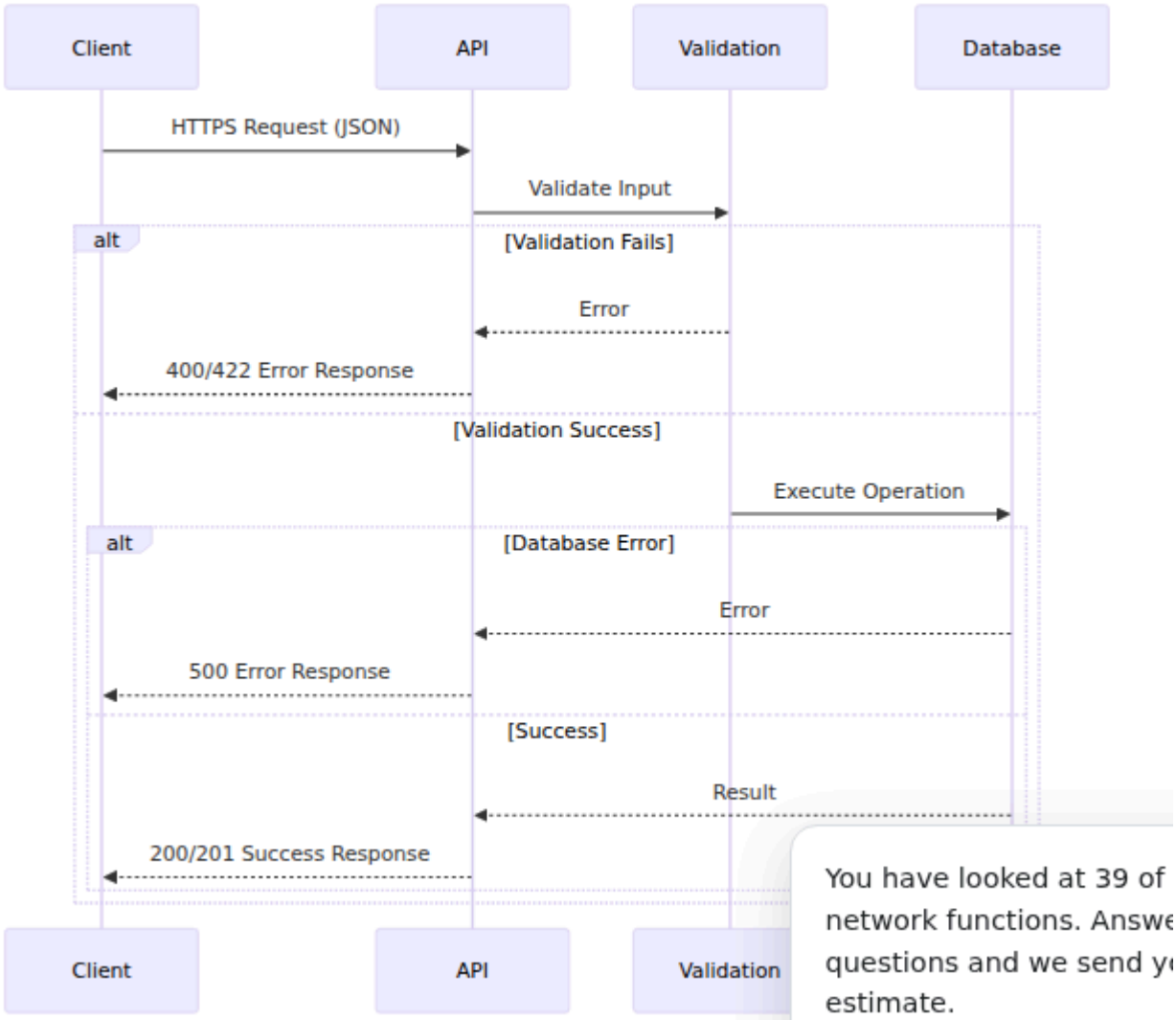
Resposta de Erro:

```
{
  "status": "error",
  "response": {
    "invalid_fields": {
      "field_name": "error message"
    }
  }
}
```

Códigos de Status HTTP

Código	Significado	Caso de Uso
200	OK	GET, PUT, DELETE bem-sucedidos
201	Criado	POST bem-sucedido
400	Solicitação Inválida	Dados de entrada inválidos
401	Não Autorizado	Sem chave da API, ou a chave não é conhecida. Veja Controle de Acesso da API
403	Proibido	As funções da chave não permitem a rota. Veja Controle de Acesso da API
404	Não Encontrado	Recurso não existe
422	Entidade Não Processável	Erro de validação
500	Erro Interno do Servidor	Erro do lado do servidor
503	Serviço Indisponível	A configuração de controle de acesso da API não é válida. Veja Controle de Acesso da API

Fluxo de Solicitação da API



Gerenciamento de Assinantes

Listar Assinantes

Recuperar todos os assinantes ou filtrar por critérios.

Endpoint: `GET /api/subscriber`

Parâmetros de Consulta:

Parâmetro	Tipo	Descrição
`enabled`	boolean	Filtrar por status habilitado
`ims_enabled`	boolean	Filtrar por status habilitado IMS

Exemplo de Solicitação:

```
curl -k https://hss.example.com:8443/api/subscriber
```

Exemplo de Resposta:

```
{
  "data": [
    {
      "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
      "imsi": "001001123456789",
      "enabled": true,
      "ims_enabled": true,
      "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "roaming_profile_id": "9c858901-8a57-4791-81fe-
4c455b099bc9",
      "custom_attributes": {},
      "inserted_at": "2025-10-15T10:30:00Z",
      "updated_at": "2025-10-15T10:30:00Z"
    }
  ]
}
```

Obter Assinante por ID

Recuperar um assinante específico pelo ID do banco de dados.

Endpoint: `GET /api/subscriber/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do assinante no banco de dados

Exemplo de Solicitação:

```
curl -k https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-
4b01-90e6-d701748f0851
```

Obter Assinante por IMSI

Recuperar um assinante pelo seu IMSI.

Endpoint: `GET /api/subscriber/imsi/:imsi`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição	Formato
`imsi`	string	Identidade Internacional de Assinante Móvel	14-15 dígitos

Exemplo de Solicitação:

```
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001001123456789
```

Caso de Uso: Solucionar problemas de um assinante específico pelo seu IMSI.

Obter Assinante por MSISDN

Recuperar um assinante pelo seu número de telefone.

Endpoint: `GET /api/subscriber/msisdn/:msisdn`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição	Formato
`msisdn`	string	Número ISDN da Estação Móvel	1-15 dígitos (E.164)

Exemplo de Solicitação:

```
curl -k
https://hss.example.com:8443/api/subscriber/msisdn/14155551234
```

Caso de Uso: Procurar informações do assinante quando você só tem o número de telefone.

Criar Assinante

Provisionar um novo assinante.

Endpoint: `POST /api/subscriber`

Corpo da Solicitação:

```
{
  "subscriber": {
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
    "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
    "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
    "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
    "roaming_profile_id": "9c858901-8a57-4791-81fe-4c455b099bc9",
    "custom_attributes": {
      "note": "Assinante de teste"
    }
  }
}
```

Campos Obrigatórios:

- `imsi` - Deve ter 14-15 dígitos, único
- `key_set_id` - Deve referenciar um Conjunto de Chaves existente
- `epc_profile_id` - Deve referenciar um Perfil EPC existente

Campos Opcionais:

- `enabled` - Padrão: true
- `ims_enabled` - Padrão: true
- `sim_id` - Referência a SIM card
- `ims_profile_id` - Referência a Perfil IMS (obrigatório para serviços IMS)
- `roaming_profile_id` - Referência a Perfil de Roaming (obrigatório para controle de roaming)
- `msisdns` - Array de IDs de MSISDN (números de telefone)
- `static_ips` - Array de IDs de IP Estático para atribuições de APN
- `ims_subscription_id` - Referência a uma Assinatura IMS para agrupar vários assinantes sob uma identidade pública compartilhada e S-CSCF
- `custom_attributes` - Pares chave-valor personalizados

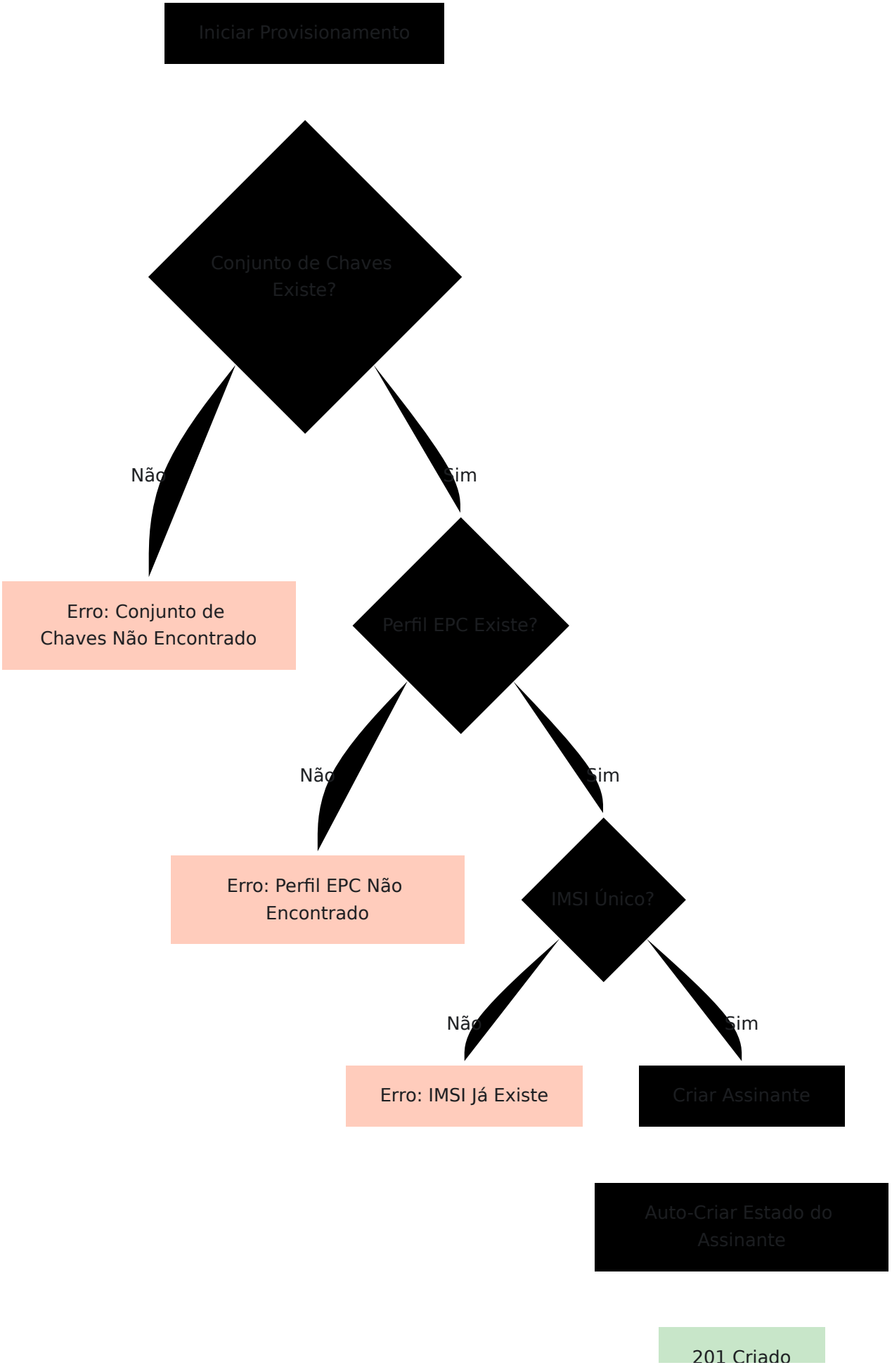
Veja Também:

- Exemplo Completo de Provisionamento de Assinante - Fluxo de trabalho de ponta a ponta
- Documentação Multi-MSISDN - Atribuindo números de telefone a assinantes
- Gerenciamento de IP Estático - Atribuindo IPs estáticos a APNs

Exemplo de Solicitação:

```
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3"
    }
  }'
```

Fluxo de Provisionamento:


Iniciar Provisionamento
Conjunto de Chaves Existe?
Não
Sim
Erro: Conjunto de Chaves Não Encontrado
Perfil EPC Existe?
Não
Sim
Erro: Perfil EPC Não Encontrado
IMSI Único?
Não
Sim
Erro: IMSI Já Existe
Criar Assinante
Auto-Criar Estado do Assinante
201 Criado

Atualizar Assinante

Modificar um assinante existente.

Endpoint: `PUT /api/subscriber/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do assinante no banco de dados

Corpo da Solicitação:

```
{
  "subscriber": {
    "enabled": false,
    "ims_enabled": false,
    "epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "custom_attributes": {
      "note": "Temporariamente desativado"
    }
  }
}
```

Campos Atualizáveis:

- `enabled` - Habilitar/desabilitar todos os serviços
- `ims_enabled` - Habilitar/desabilitar serviços IMS
- `sim_id` - Alterar atribuição de SIM card
- `key_set_id` - Alterar chaves criptográficas (cuidado!)
- `epc_profile_id` - Alterar perfil de serviço de dados
- `ims_profile_id` - Alterar perfil de serviço de voz
- `roaming_profile_id` - Alterar política de roaming
- `msisdns` - Atualizar números de telefone atribuídos ao assinante (`[]` remove todos)

- `static_ips` - Atualizar atribuições de IP estático para APNs (`[]` remove todos)
- `custom_attributes` - Atualizar dados personalizados

Não Atualizável:

- `imsi` - Não é possível alterar o IMSI (excluir e recriar em vez disso)
- `subscriber_state` - Estado em tempo de execução escrito pelo HSS; não pode ser definido através deste endpoint. Defina ou limpe `assigned_scscf` com `PATCH /api/hlr/subscriber_state/:imsi`

Veja Também:

- Gerenciamento de Perfis - Gerenciando perfis de serviço

Exemplo de Solicitação:

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "enabled": false
    }
  }'
```

Casos de Uso:

- Desativar temporariamente o assinante: `{"enabled": false}`
- Desativar apenas os serviços de voz: `{"ims_enabled": false}`
- Alterar perfil de serviço: `{"epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479"}` (veja Perfis EPC)
- Atualizar política de roaming: `{"roaming_profile_id": "1b671a64-40d5-491e-99b0-da01ff1f3341"}` (veja Gerenciamento de Roaming)

Deletar Assinante

Remover um assinante do sistema.

Endpoint: `DELETE /api/subscriber/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do assinante no banco de dados

Exemplo de Solicitação:

```
curl -k -X DELETE
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851
```

Aviso: Isso exclui permanentemente o assinante e todos os dados de estado associados (sessões PDN, chamadas, etc.). O IMSI pode ser reutilizado após a exclusão.

Nota: Excluir um assinante NÃO exclui o associado:

- Conjunto de Chaves - Pode ser reutilizado para outros assinantes
- SIM - Pode ser reatribuído a um novo assinante
- Perfis - Recursos compartilhados usados por vários assinantes
- MSISDNs - Devem ser excluídos separadamente, se desejado

Cancelar Solicitação de Localização (Desconexão Forçada)

Enviar uma Solicitação de Cancelamento de Localização (CLR) para forçar a desconexão de um assinante de seu MME registrado atualmente.

Endpoint: `POST /api/subscriber/cancel_location`

Corpo da Solicitação:

```
{
  "imsi": "001001123456789"
}
```

Parâmetros:

Parâmetro	Tipo	Obrigatório	Descrição
`imsi`	string	Sim	IMSI do assinante a ser desconectado (14-15 dígitos)

Exemplo de Solicitação:

```
curl -k -X POST
https://hss.example.com:8443/api/subscriber/cancel_location \
  -H "Content-Type: application/json" \
  -d '{"imsi": "001001123456789"}'
```

Resposta de Sucesso (200 OK):

```
{
  "data": {
    "message": "Solicitação de Cancelamento de Localização enviada
com sucesso",
    "imsi": "001001123456789",
    "destination_host": "mme01.operator.com",
    "destination_realm": "epc.operator.com"
  }
}
```

Resposta de Erro (404 Não Encontrado):

```
{
  "error": "Assinante não encontrado ou não registrado atualmente
em nenhum MME"
}
```

Comportamento:

- Envia S6a CLR para o MME onde o assinante está atualmente registrado (`subscriber_state.last_seen_mme`)
- Usa `Cancellation-Type: subscription_withdrawal` (força desconexão total)
- Define `CLR-Flags: {s6a_indicator: 1, reattach_required: 1}` (UE deve reautenticar)
- Retorna 404 se o assinante nunca registrou ou `last_seen_mme` é nulo
- **Afeta todos os MSISDNs** associados ao IMSI (mesmo dispositivo físico/SIM)

Casos de Uso:

- **Prevenção de Fraude:** Desconectar imediatamente assinante suspeito
- **Término de Assinatura:** Forçar logout quando a conta é desativada
- **Solução de Problemas:** Limpar registro MME obsoleto para depuração
- **Migração:** Forçar reautenticação para aplicar novas configurações de perfil
- **Segurança:** Desconectar imediatamente assinante comprometido

Considerações sobre Multi-IMSI:

Ao usar CLR com cenários multi-MSISDN:

1. **Múltiplos MSISDNs, Um Único IMSI:**

```
// Assinante tem IMSI 001001123456789 com MSISDNs
["+1234567890", "+9876543210"]
POST /api/subscriber/cancel_location
{"imsi": "001001123456789"}

// Resultado: Um CLR enviado, ambos os MSISDNs afetados (mesmo
dispositivo)
```

2. **IMSI Diferentes (Dispositivos Diferentes):**

```
// Dois assinantes com o mesmo MSISDN, mas IMSIs diferentes
(cenário de portabilidade de número)
// Assinante A: IMSI 001001111111111, MSISDN "+1234567890"
// Assinante B: IMSI 001001222222222, MSISDN "+1234567890"

POST /api/subscriber/cancel_location
{"imsi": "001001111111111"}

// Resultado: Apenas Assinante A desconectado, Assinante B não
afetado
```

Notas Importantes:

- **Baseado em IMSI:** CLR é sempre enviado por IMSI, não por MSISDN
- **Assíncrono:** CLR é enviado de forma assíncrona; resposta de sucesso significa que o CLR foi enviado, não que o MME o processou
- **Sem validação do status do MME:** CLR é enviado mesmo se o MME estiver inacessível (comportamento padrão do HSS)
- **Idempotente:** Seguro chamar várias vezes para o mesmo IMSI

Documentação Relacionada:

- Fluxo do Protocolo de Solicitação de Cancelamento de Localização
- Cenários Multi-IMSI
- Arquitetura da Interface S6a

Dados do Repositório Sh (Dados Transparentes)

Provisionamento de acesso aos documentos transparentes Sh de um assinante (MMTEL-Services, IMS-ODB-Information e outros). Estes são os mesmos documentos opacos que um Servidor de Aplicação lê e escreve pela interface Diameter Sh. O acesso é via `GET`/`PUT`/`DELETE` `/api/subscriber/repository_data/:imsi`.

Veja Dados do Repositório Sh (Dados Transparentes) para o modelo de dados, fluxos de mensagens, regras de concorrência SequenceNumber e a referência completa da API REST.

Gerenciamento de MSISDN

MSISDNs (números de telefone) podem ser atribuídos a assinantes para habilitar serviços de voz. Veja Documentação Multi-MSISDN para detalhes sobre a atribuição de vários números a um único assinante.

A bandeira `shared`

As respostas que carregam um MSISDN incluem um `shared` booleano computado, somente leitura. Ele é `true` quando o número é mantido por **mais de um assinante** (um número compartilhado multi-IMSI, por exemplo, um telefone e um smartwatch em SIMs separados), e `false` caso contrário. Ele aparece na entidade MSISDN (`GET /api/msisdn` e `GET /api/msisdn/:id`), em cada número incorporado em um assinante (`GET /api/subscriber/:id`), e nos números de assinantes retornados por `GET /api/subscriber/msisdn/:msisdn`. Ele nunca é aceito em gravações; o HSS o computa. Veja Resolução de MSISDN entre Interfaces.

Listar MSISDNs

Recuperar todos os números de telefone.

Endpoint: `GET /api/msisdn`

Exemplo de Solicitação:

```
curl -k https://hss.example.com:8443/api/msisdn
```

Obter MSISDN

Recuperar um número de telefone específico.

Endpoint: `GET /api/msisdn/:id`

Exemplo de Solicitação:

```
curl -k https://hss.example.com:8443/api/msisdn/16fd2706-8baf-
433b-82eb-8c7fada847da
```

Exemplo de Resposta:

```
{
  "status": "success",
  "response": {
    "id": "16fd2706-8baf-433b-82eb-8c7fada847da",
    "msisdn": "14155551234",
    "shared": false
  }
}
```

Criar MSISDN

Criar um novo número de telefone.

Endpoint: `POST /api/msisdn`

Corpo da Solicitação:

```
{
  "msisdn": {
    "msisdn": "14155551234"
  }
}
```

Validação:

- Deve ter 1-15 dígitos
- Deve ser único
- Deve seguir o formato E.164 (formato internacional sem o sinal de +)

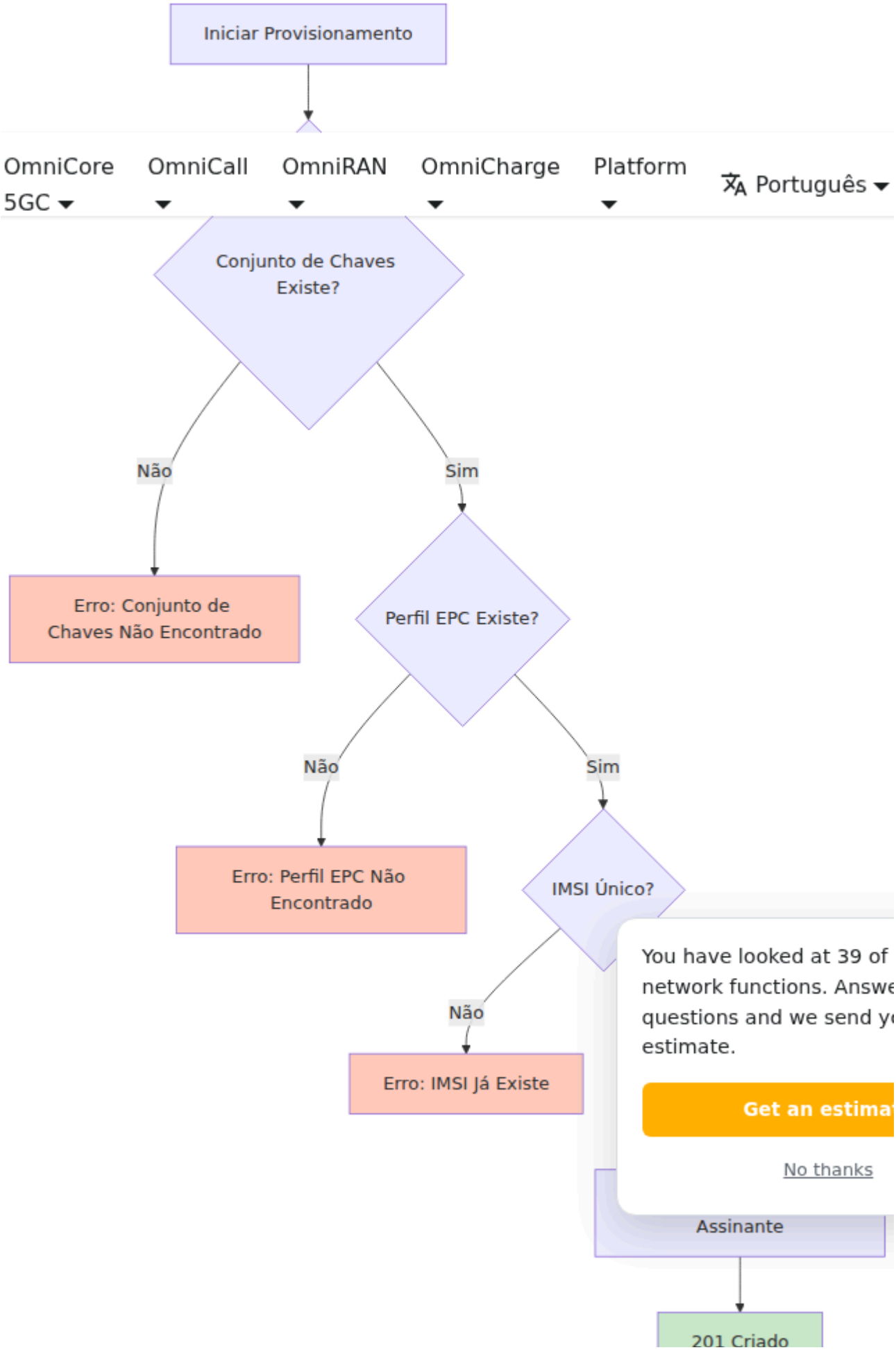
Exemplo de Solicitação:

```
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{
    "msisdn": {
      "msisdn": "14155551234"
    }
  }'
```

Atribuir MSISDN a Assinante

Para atribuir um número de telefone a um assinante, você precisa criar um registro de junção. Isso é tipicamente feito através do endpoint de atualização do assinante ou por meio de manipulação direta do banco de dados.

Padrão Multi-MSISDN:


Iniciar Provisionamento
OmniCore 5GC
OmniCall
OmniRAN
OmniCharge
Platform
Português
Conjunto de Chaves Existe?
Não
Sim
Erro: Conjunto de Chaves Não Encontrado
Perfil EPC Existe?
Não
Sim
Erro: Perfil EPC Não Encontrado
IMSI Único?
Não
Erro: IMSI Já Existe
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks
Assinante
201 Criado

Veja Recursos Multi-MSISDN e Multi-IMSI para uso detalhado.

Deletar MSISDN

Remover um número de telefone.

Endpoint: `DELETE /api/msisdn/:id`

Exemplo de Solicitação:

```
curl -k -X DELETE
https://hss.example.com:8443/api/msisdn/16fd2706-8baf-433b-82eb-
8c7fada847da
```

Gerenciamento de Assinaturas IMS

Uma Assinatura IMS agrupa vários assinantes (IMSIs / IMPIs) em uma única assinatura IMS que compartilha uma identidade pública e um único S-CSCF de atendimento. Veja Identidade Pública de Usuário Compartilhada.

Listar Assinaturas IMS

Endpoint: `GET /api/ims_subscription`

```
curl -k https://hss.example.com:8443/api/ims_subscription
```

Obter Assinatura IMS

Endpoint: `GET /api/ims_subscription/:id`

```
curl -k
https://hss.example.com:8443/api/ims_subscription/16fd2706-8baf-
433b-82eb-8c7fada847da
```

Criar Assinatura IMS

Endpoint: `POST /api/ims_subscription`

```
curl -k -X POST https://hss.example.com:8443/api/ims_subscription
\
  -H "Content-Type: application/json" \
  -d '{"name": "Família Smith - número compartilhado"}'
```

Campos:

- `name` - Nome legível por humanos (obrigatório, único)
- `ims_profile_id` - Perfil IMS compartilhado opcional para as identidades compartilhadas do grupo

`assigned_scscf` e `last_seen_scscf_timestamp` são estados em tempo de execução escritos pelo HSS e não são definidos pelo cliente.

Atualizar Assinatura IMS

Endpoint: `PATCH /api/ims_subscription/:id`

Deletar Assinatura IMS

Endpoint: `DELETE /api/ims_subscription/:id`

Excluir uma Assinatura IMS não exclui seus assinantes; seu `ims_subscription_id` é limpo. Para adicionar um assinante a um grupo, defina `ims_subscription_id` no assinante (veja Atualizar Assinante).

Gerenciamento de SIM

Os registros de cartão SIM armazenam informações físicas do cartão SIM, incluindo ICCID, detalhes do fornecedor, códigos PIN/PUK e chaves OTA. Os registros de SIM podem ser opcionalmente vinculados a assinantes.

Veja Também:

- Documentação Multi-IMSI - Múltiplos assinantes em um único SIM físico

Listar SIMs

Recuperar todos os cartões SIM.

Endpoint: `GET /api/sim`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/sim
```

Obter SIM

Recuperar um cartão SIM específico. `adm1`-`adm10`, `kic` e `kid` são somente para escrita: eles são retornados apenas na resposta à requisição de criação ou atualização que os definiu, e nunca em qualquer outra resposta da API.

Endpoint: `GET /api/sim/:id`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/sim/3f2504e0-4f89-41d3-
9a0c-0305e82c3301
```

Criar SIM

Criar um novo registro de cartão SIM.

Endpoint: `POST /api/sim`

Corpo da Requisição:

```
{
  "sim": {
    "iccid": "89911012000003204510",
    "sim_vendor": "Gemalto",
    "batch_name": "2025-Q1-Batch-01",
    "is_esim": false,
    "pin1": "1234",
    "pin2": "5678",
    "puk1": "12345678",
    "puk2": "87654321",
    "adm1": "admin-code-1",
    "kic": "0123456789ABCDEF0123456789ABCDEF",
    "kid": "FEDCBA9876543210FEDCBA9876543210"
  }
}
```

Campos Obrigatórios:

- `iccid` - 19-20 dígitos, único

Campos Opcionais, mas Importantes:

- `sim_vendor` - Nome do fabricante
- `batch_name` - Para rastreamento
- `is_esim` - Flag booleana para eSIM
- `pin1`, `pin2` - Códigos PIN do usuário final
- `puk1`, `puk2` - Códigos de desbloqueio PIN
- `adm1-adm10` - Códigos administrativos
- `kic`, `kid` - Chaves de segurança OTA (string hexadecimal)

Exemplo de Requisição:

```
curl -k -X POST https://hss.example.com:8443/api/sim \
  -H "Content-Type: application/json" \
  -d '{
    "sim": {
      "iccid": "89911012000003204510",
      "sim_vendor": "Gemalto"
    }
  }'
```

Atualizar SIM

Modificar os dados do cartão SIM.

Endpoint: `PUT /api/sim/:id`

Exemplo de Requisição:

```
curl -k -X PUT https://hss.example.com:8443/api/sim/3f2504e0-4f89-
41d3-9a0c-0305e82c3301 \
  -H "Content-Type: application/json" \
  -d '{
    "sim": {
      "batch_name": "Nome-Batch-Atualizado"
    }
  }'
```

Deletar SIM

Remover um registro de cartão SIM.

Endpoint: `DELETE /api/sim/:id`

Aviso: Certifique-se de que nenhum assinante faça referência a este SIM antes de deletá-lo.

Gerenciamento de Conjuntos de Chaves

Os conjuntos de chaves contêm o material criptográfico (Ki, OPC/OP, AMF, SQN) usado para autenticação de assinantes via o algoritmo Milenage. Cada assinante deve referenciar um conjunto de chaves.

Veja Também:

- Fluxos de Protocolo - Procedimentos de autenticação usando conjuntos de chaves

Listar Conjuntos de Chaves

Recuperar todos os conjuntos de chaves criptográficas.

Endpoint: `GET /api/key_set`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/key_set
```

Obter Conjunto de Chaves

Recuperar um conjunto de chaves específico. `ki`, `opc`, `op`, `psk`, `des` e `sip_password` são somente para escrita: eles são retornados apenas na resposta à requisição de criação ou atualização que os definiu, e nunca em qualquer outra resposta da API.

Endpoint: `GET /api/key_set/:id`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/key_set/6ba7b810-9dad-
41d1-80b4-00c04fd430c8
```

Exemplo de Resposta:

```
{
  "data": {
    "id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage",
    "ota_counter": 0
  }
}
```

Criar Conjunto de Chaves

Criar um novo conjunto de chaves criptográficas.

Endpoint: `POST /api/key_set`

Corpo da Requisição:

```
{
  "key_set": {
    "ki": "0123456789ABCDEF0123456789ABCDEF",
    "opc": "FEDCBA9876543210FEDCBA9876543210",
    "amf": "8000",
    "sqn": 0,
    "authentication_algorithm": "milenage"
  }
}
```

Campos Obrigatórios:

- `ki` - Chave de 128 bits (32 caracteres hexadecimais)
- Ou `opc` OU `op` (OPC pode ser derivado de OP)
- `authentication_algorithm` - Atualmente apenas "milenage"

Campos Opcionais:

- `amf` - Padrão: "8000"

- `sqn` - Padrão: 0
- `ota_counter` - Padrão: 0

Formato da Chave:

- Todas as chaves são strings hexadecimais
- Ki, OPC, OP: 32 caracteres hexadecimais (128 bits)
- AMF: 4 caracteres hexadecimais (16 bits)

Exemplo de Requisição:

```
curl -k -X POST https://hss.example.com:8443/api/key_set \
  -H "Content-Type: application/json" \
  -d '{
    "key_set": {
      "ki": "0123456789ABCDEF0123456789ABCDEF",
      "opc": "FEDCBA9876543210FEDCBA9876543210",
      "authentication_algorithm": "milenage"
    }
  }'
```

Aviso de Segurança: Conjuntos de chaves contêm material criptográfico altamente sensível. Proteja o acesso à API de acordo.

Atualizar Conjunto de Chaves

Modificar um conjunto de chaves existente.

Endpoint: `PUT /api/key_set/:id`

Aviso: Alterar chaves para um assinante ativo causará falhas de autenticação. Atualize as chaves apenas durante janelas de manutenção ou para novos assinantes.

Impacto: Atualizações afetam todos os assinantes que usam este conjunto de chaves imediatamente. Assinantes ativos falharão na autenticação na próxima tentativa de conexão.

Deletar Conjunto de Chaves

Remover um conjunto de chaves.

Endpoint: `DELETE /api/key_set/:id`

Aviso: Certifique-se de que nenhum assinante faça referência a este conjunto de chaves antes de deletar. Consulte os assinantes primeiro para verificar referências.

Gerenciamento de Perfis

Perfis EPC

Os perfis EPC (Evolved Packet Core) definem parâmetros de serviço de dados para assinantes. Esses perfis são referenciados ao criar assinantes.

Listar Perfis EPC

Endpoint: `GET /api/epc/profile`

Obter Perfil EPC

Endpoint: `GET /api/epc/profile/:id`

Criar Perfil EPC

Endpoint: `POST /api/epc/profile`

Corpo da Requisição:

```
{
  "apn_profiles": [],
  "name": "Plano de Dados Padrão",
  "network_access_mode": "packet_only",
  "tracking_area_update_interval_seconds": 600,
  "ue_ambr_dl_kbps": 100000,
  "ue_ambr_ul_kbps": 50000
}
```


Campos:

Campo	Descrição	Unidades	V
`name`	Nome do perfil	Texto	Ide
`ue_ambr_dl_kbps`	Limite de largura de banda de download	Kbps	10
`ue_ambr_ul_kbps`	Limite de largura de banda de upload	Kbps	50
`network_access_mode`	Tipo de acesso	String	"p "p
`tracking_area_update_interval_seconds`	Temporizador TAU	Segundos	60
`apn_profiles`	Lista de IDs de perfil APN	Array	[]

Exemplo de Requisição:

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profiles": [],
    "name": "Premium 100Mbps",
    "network_access_mode": "packet_only",
    "tracking_area_update_interval_seconds": 600,
    "ue_ambr_dl_kbps": 100000,
    "ue_ambr_ul_kbps": 50000
  }'
```

Veja Também:

- Documentação de Perfis - Guia detalhado de configuração de perfis
- Provisionamento Completo de Assinantes - Usando perfis EPC no provisionamento

Atualizar Perfil EPC

Endpoint: `PUT /api/epc/profile/:id`

Nota: Alterações nos perfis EPC afetam todos os assinantes que usam este perfil. Sessões ativas podem precisar ser restabelecidas.

Deletar Perfil EPC

Endpoint: `DELETE /api/epc/profile/:id`

Aviso: Certifique-se de que nenhum assinante faça referência a este perfil antes de deletar.

Perfis IMS

Os perfis IMS (IP Multimedia Subsystem) definem parâmetros de serviço de voz e Critérios de Filtro Iniciais (IFC) para assinantes. Esses perfis são referenciados ao criar assinantes com serviços IMS habilitados.

Listar Perfis IMS

Endpoint: `GET /api/ims/profile`

Criar Perfil IMS

Endpoint: `POST /api/ims/profile`

Corpo da Requisição:

```
{
  "name": "VoLTE Padrão",
  "ifc_template": "<IMS-XML-Template-Here>"
}
```

Campos Obrigatórios:

- `name` - Nome do perfil (deve ser único)
- `ifc_template` - Template XML IFC (Critérios de Filtro Iniciais) com variáveis de template Liquid

Variáveis do Template IFC:

O template IFC suporta as seguintes variáveis de template Liquid que são substituídas dinamicamente:

Variável	Descrição	Valor Exemplo
`{{ imsi }}`	IMSI do assinante	`001001123456789`
`{{ msisdns }}`	Array de MSISDNs (para loops)	`["14155551234", "14155555678"]`
`{{ mcc }}`	Código do País Móvel	`001`
`{{ mnc }}`	Código da Rede Móvel	`001`

Como Funciona a Renderização do Template:

O template IFC é armazenado como um **template Liquid** (semelhante ao Jinja2) e é renderizado **dinamicamente** durante operações IMS:

1. **Armazenamento**: Quando você cria um perfil IMS, o template é armazenado como está com variáveis como `{{ imsi }}` e `{% for msisdn in msisdns %}`
2. **Validação**: A API valida o template renderizando-o com dados de teste para garantir uma sintaxe XML válida
3. **Renderização em Tempo de Execução**: Quando um assinante realiza o registro IMS (MAA/SAA), o HSS:
   - Recupera o perfil IMS do assinante
   - Renderiza o template com os dados reais do assinante:
     - `{{ imsi }}` → IMSI do assinante
     - `{{ msisdns }}` → Números de telefone do assinante
     - `{{ mcc }}` → Código do País Móvel configurado
     - `{{ mnc }}` → Código da Rede Móvel configurado
   - Retorna o XML renderizado para o S-CSCF via Cx/Diameter

Sintaxe do Template:

```
<!-- Substituição simples de variável -->
{{ imsi }}

<!-- Para loops sobre arrays -->
{% for msisdn in msisdns %}
  <MSISDN>{{ msisdn }}</MSISDN>
{% endfor %}

<!-- Combinando variáveis -->
{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc }}.3gppnetwork.org
```

Exemplo de Template IFC:

```
<?xml version="1.0" encoding="UTF-8"?>
<IMSSubscription>
<PrivateID>{{ imsi }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</PrivateID>
<ServiceProfile>
{% for msisdn in msisdns %}
<PublicIdentity>
<Identity>sip:{{ msisdn }}@ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
<PublicIdentity>
<Identity>tel:{{ msisdn }}</Identity>
<Extension>
<IdentityType>0</IdentityType>
</Extension>
</PublicIdentity>
{% endfor %}
<InitialFilterCriteria>
<Priority>10</Priority>
<TriggerPoint>
<ConditionTypeCNF>0</ConditionTypeCNF>
<SPT>
<ConditionNegated>0</ConditionNegated>
<Group>0</Group>
<Method>REGISTER</Method>
</SPT>
</TriggerPoint>
<ApplicationServer>
<ServerName>sip:as.ims.mnc{{ mnc }}.mcc{{ mcc
}}.3gppnetwork.org</ServerName>
<DefaultHandling>0</DefaultHandling>
</ApplicationServer>
</InitialFilterCriteria>
</ServiceProfile>
</IMSSubscription>
```

Exemplo de Requisição (curl):

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "default",
    "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
<IMSSubscription><ServiceProfile>...</ServiceProfile>
</IMSSubscription>"
  }'
```

Exemplo de Requisição (Python):

```
import requests

response = requests.post(
    "https://hss.example.com:8443/api/ims/profile",
    json={
        "name": "default",
        "ifc_template": ifc_template_string
    },
    verify=False  # Para certificados autoassinados
)
```

Resposta de Sucesso (201 Criado):

```
{
  "status": "success",
  "response": {
    "id": "0f8fad5b-d9cb-469f-a165-70867728950e",
    "name": "default",
    "ifc_template": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>..."
  }
}
```

Validação:

- A API valida que o template IFC é um XML válido
- Variáveis do template são renderizadas com dados de teste para verificar a sintaxe

- O campo `name` deve ser único e não vazio

Veja Também:

- Documentação de Perfis - Detalhes e exemplos do template IFC
- Fluxos de Protocolo - Registro IMS e fluxos de chamadas
- Template IFC Padrão - Implementação de referência

Perfis APN

Os perfis APN (Access Point Name) consistem em três componentes que trabalham juntos:

1. **Identificador APN** - Define o nome do APN e a versão IP
2. **Perfil QoS APN** - Define parâmetros de Qualidade de Serviço
3. **Perfil APN** - Combina identificador e QoS, vinculado aos Perfis EPC

Veja Documentação PCRF para configuração detalhada de políticas, gerenciamento de QoS e re-autenticação automática. Veja também Documentação de Perfis para exemplos de configuração de APN.

Listar Identificadores APN

Endpoint: `GET /api/apn/identifier`

Criar Identificador APN

Endpoint: `POST /api/apn/identifier`

Corpo da Requisição:

```
{
  "apn": "internet",
  "ip_version": "ipv4v6"
}
```

Valores da Versão IP:

- `"ipv4"` - Apenas IPv4

- `"ipv6"` - Apenas IPv6
- `"ipv4v6"` - IPv4v6 (dual stack)
- `"ipv4_or_ipv6"` - IPv4 ou IPv6 (escolha da rede)

Listar Perfis QoS APN

Endpoint: `GET /api/apn/qos_profile`

Criar Perfil QoS APN

Endpoint: `POST /api/apn/qos_profile`

Corpo da Requisição:

```
{
  "name": "Internet de Melhor Esforço",
  "allocation_retention_priority": 8,
  "apn_ambr_dl_kbps": 50000,
  "apn_ambr_ul_kbps": 25000,
  "pre_emption_capability": false,
  "pre_emption_vulnerability": true,
  "qci": 9
}
```

Listar Perfis APN

Endpoint: `GET /api/apn/profile`

Criar Perfil APN

Endpoint: `POST /api/apn/profile`

Corpo da Requisição:

```
{
  "apn_identifier_id": "4fa85f64-5717-4562-b3fc-2c963f66afa6",
  "apn_qos_profile_id": "8d8ac610-566d-4ef0-9c22-186b2a5ed793",
  "name": "APN de Internet"
}
```

Campos Obrigatórios:

- `apn_identifier_id` - Deve referenciar um Identificador APN existente
- `apn_qos_profile_id` - Deve referenciar um Perfil QoS APN existente

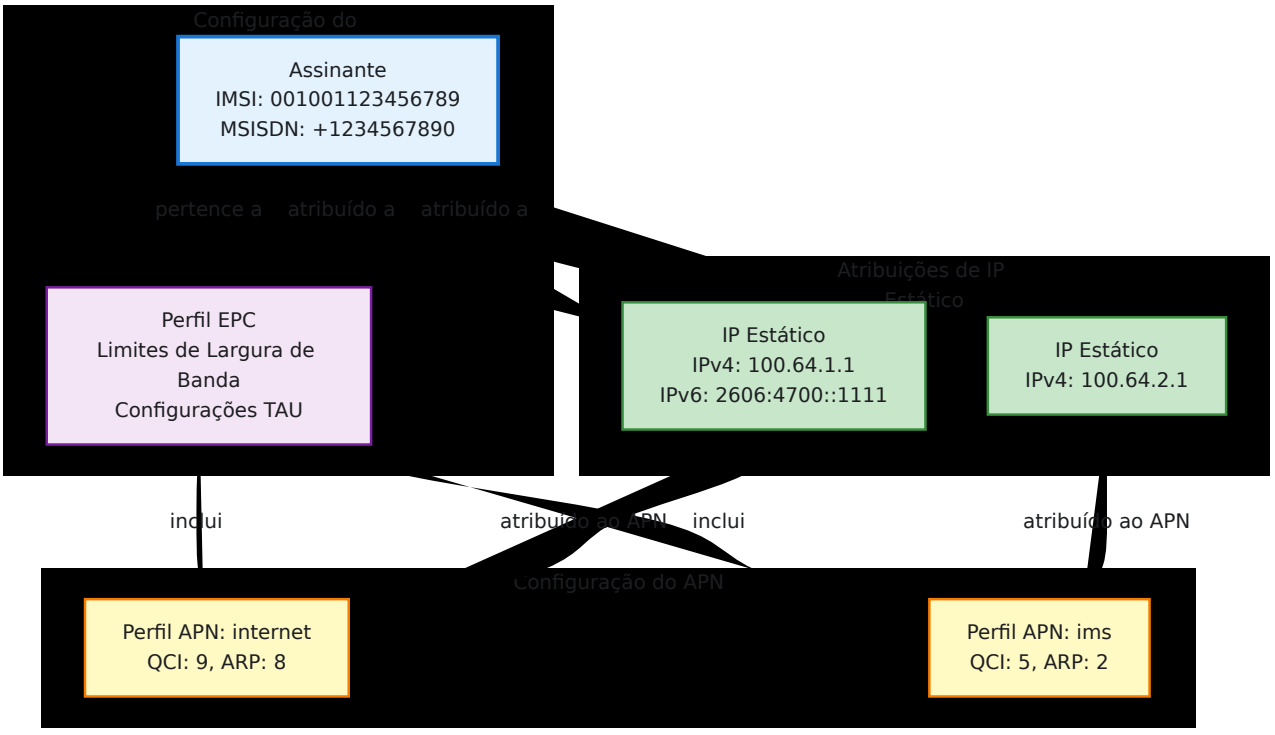
Veja Também:

- Provisionamento Completo de Assinantes - Exemplo completo incluindo configuração de APN
- Perfis EPC - Perfis APN estão vinculados a perfis EPC

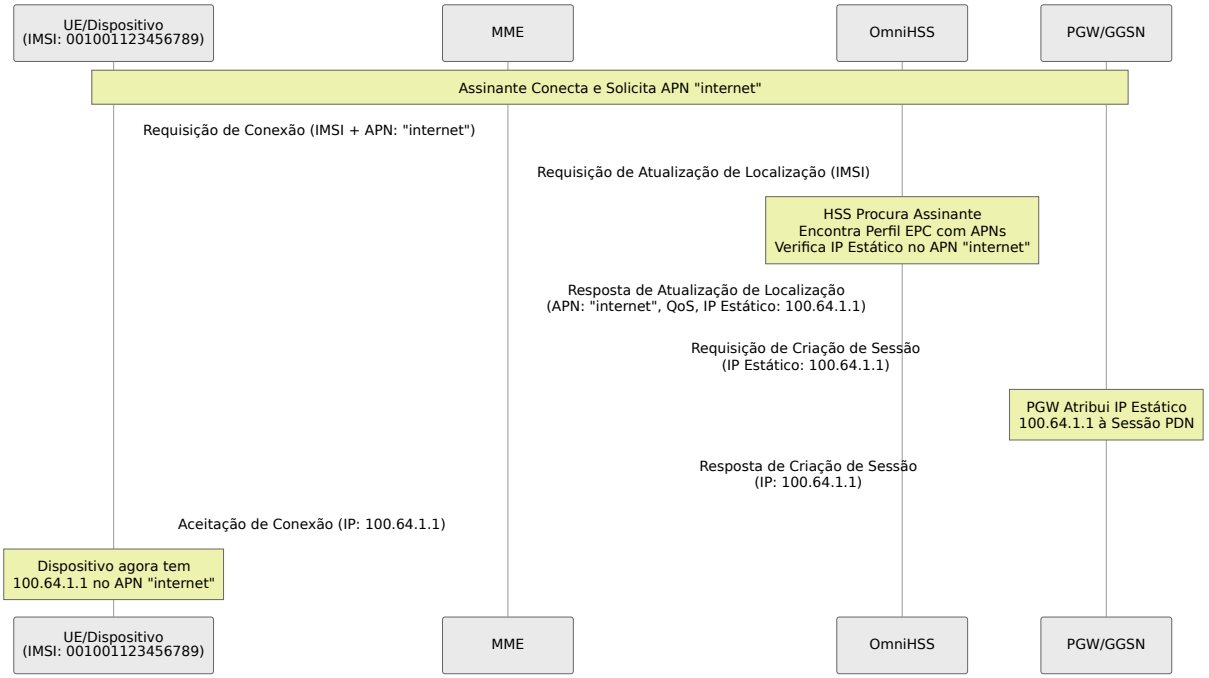
Gerenciamento de IP Estático

Endereços IP estáticos podem ser atribuídos a APNs específicos para assinantes individuais. Isso permite que os assinantes recebam um endereço IPv4 e/ou IPv6 predeterminado ao se conectar a um APN específico, em vez de receber um endereço dinâmico de um pool DHCP.

Arquitetura:

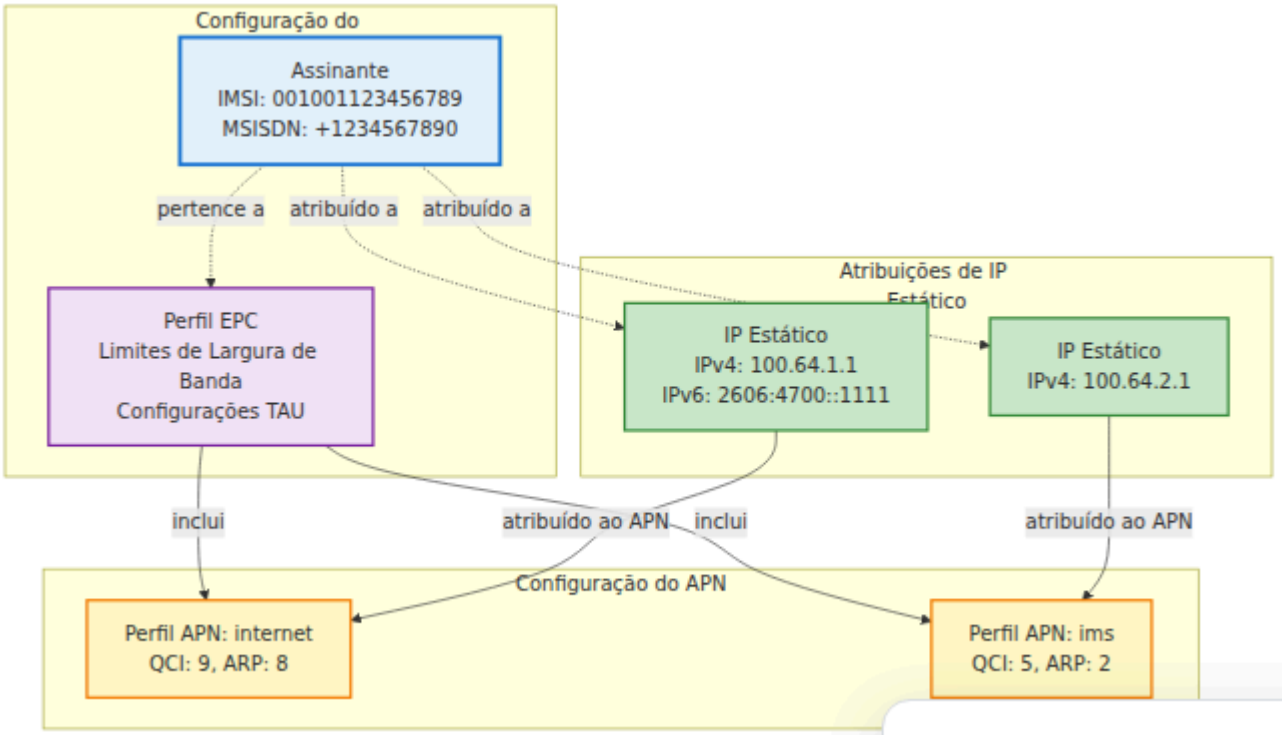


Fluxo de Dados Quando o Assinante se Conecta:



Resposta de Atualização de Localização - Mapeamento de Dados de Configuração do APN:

Este diagrama mostra exatamente de onde cada campo no AVP de Configuração do APN da Resposta de Atualização de Localização S6a vem no banco de dados:



Conceitos Chave:

- **Atribuição por APN**: Cada IP Estático está vinculado a um Perfil APN específico
- **Um IP por APN por Assinante**: Um assinante pode ter apenas uma atribuição de IP estático por APN
- **Suporte a IPv4 e IPv6**: IPs estáticos podem ser apenas IPv4, apenas IPv6 ou dual-stack
- **Unicidade Global de IP**: Cada endereço IP deve ser globalmente único em **todos** os registros de IP estático no sistema
  - O mesmo endereço IPv4 ou IPv6 não pode ser atribuído a múltiplos assinantes (mesmo em diferentes APNs)
  - Isso previne conflitos de roteamento e ambiguidade de endereço IP
  - Aplicado por índices únicos no banco de dados nos campos `ipv4_static_ip` e `ipv6_static_ip`
- **Relação Muitos-para-Muitos**: Assinantes e IPs Estáticos estão vinculados através de uma tabela de junção

Casos de Uso:

- Endereços IP fixos para dispositivos IoT
- Hospedagem de servidores em dispositivos móveis (requer IP estático para conexões de entrada)
- Aplicações legadas que requerem endereços IP específicos
- Roteamento de políticas de rede baseado em IP de origem
- Conformidade regulatória exigindo rastreamento de endereços IP

Listar IPs Estáticos

Recuperar todas as atribuições de IP estático.

Endpoint: `GET /api/epc/static_ip`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/epc/static_ip
```

Exemplo de Resposta:

```
{
  "data": [
    {
      "id": "c56a4180-65aa-42ec-a945-5fd21dec0538",
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111",
      "apn_profile": {
        "id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
        "name": "Perfil APN Internet",
        "apn_identifier": {
          "apn": "internet",
          "ip_version": "ipv4v6"
        }
      },
      "inserted_at": "2025-11-15T10:30:00Z",
      "updated_at": "2025-11-15T10:30:00Z"
    }
  ]
}
```


Obter IP Estático

Recuperar uma atribuição de IP estático específica.

Endpoint: `GET /api/epc/static_ip/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do banco de dados do IP estático

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-a945-5fd21dec0538
```

Criar IP Estático

Criar uma nova atribuição de IP estático para um APN.

Endpoint: `POST /api/epc/static_ip`

Corpo da Requisição:

```
{
  "static_ip": {
    "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111"
  }
}
```

Campos Obrigatórios:

- `apn_profile_id` - Deve referenciar um Perfil APN existente
- Pelo menos um de `ipv4_static_ip` OU `ipv6_static_ip` deve ser especificado

Campos Opcionais:

- `ipv4_static_ip` - Endereço IPv4 (notação decimal pontuada)
- `ipv6_static_ip` - Endereço IPv6 (notação padrão)

Validação de Formato de IP:

- IPv4: Formato padrão decimal pontuado (ex: `100.64.1.1`)
- IPv6: Formato padrão hexadecimal separado por dois pontos (ex: `2606:4700:4700::1111`)
- Ambos os endereços IPv4 e IPv6 devem ser **globalmente únicos em todos os registros de IP estático**
  - Isso previne conflitos de endereços IP na rede
  - O mesmo IP não pode ser atribuído a múltiplos assinantes, mesmo em diferentes APNs

- Esta é uma restrição de nível de banco de dados aplicada por índices únicos

Opções de Configuração:

Configuração	IPv4	IPv6	Exemplo
Apenas IPv4	✓	-	`{"ipv4_static_ip": "100.64.1.1"}`
Apenas IPv6	-	✓	`{"ipv6_static_ip": "2606:4700:4700::1111"}`
Dual Stack	✓	✓	Ambos os campos especificados

Exemplos de Requisições:

IP Estático apenas IPv4:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1"
    }
  }'
```

IP Estático apenas IPv6:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "b2c3d4e5-f6a7-4b1c-9d2e-3f4a5b6c7d8e",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }'
```

IP Estático dual-stack:

```
curl -k -X POST https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }'
```

Resposta de Sucesso (201 Criado):

```
{
  "data": {
    "id": "c56a4180-65aa-42ec-a945-5fd21dec0538",
    "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
    "ipv4_static_ip": "100.64.1.1",
    "ipv6_static_ip": "2606:4700:4700::1111",
    "inserted_at": "2025-11-15T10:30:00Z",
    "updated_at": "2025-11-15T10:30:00Z"
  }
}
```

Veja Também:

- Atribuir IP Estático a Assinante - Como vincular isso a um assinante
- Perfis APN - Gerenciando configurações de APN

Atualizar IP Estático

Modificar uma atribuição de IP estático existente.

Endpoint: `PUT /api/epc/static_ip/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do banco de dados do IP estático

Corpo da Requisição:

```
{
  "static_ip": {
    "ipv4_static_ip": "100.64.1.2",
    "ipv6_static_ip": "2606:4700:4700::1112"
  }
}
```

Campos Atualizáveis:

- `ipv4_static_ip` - Alterar endereço IPv4
- `ipv6_static_ip` - Alterar endereço IPv6
- `apn_profile_id` - Alterar atribuição de APN

Não Atualizável:

- `id` - Chave primária (somente leitura)

Aviso: Alterar o endereço IP de um assinante ativo afetará sua próxima conexão PDN. Sessões PDN ativas continuarão a usar o IP antigo até desconectarem e reconectarem.

Exemplo de Requisição:

```
curl -k -X PUT
https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-
a945-5fd21dec0538 \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "ipv4_static_ip": "100.64.1.2"
    }
  }'
```

Deletar IP Estático

Remover uma atribuição de IP estático.

Endpoint: `DELETE /api/epc/static_ip/:id`

Parâmetros de Caminho:

Parâmetro	Tipo	Descrição
`id`	string (UUID)	ID do banco de dados do IP estático

Exemplo de Requisição:

```
curl -k -X DELETE
https://hss.example.com:8443/api/epc/static_ip/c56a4180-65aa-42ec-
a945-5fd21dec0538
```

Comportamento:

- Remove a atribuição de IP estático
- Não afeta o Perfil APN (APN permanece disponível para outros assinantes)
- Assinantes usando este IP estático receberão IPs dinâmicos na próxima conexão
- O endereço IP se torna disponível para reutilização após a exclusão

Aviso: Se um assinante estiver usando ativamente este IP estático, excluí-lo fará com que ele receba um IP dinâmico em sua próxima conexão PDN. Certifique-se de que os assinantes estejam offline ou envie uma Solicitação de Cancelamento de Localização antes de excluir.

Atribuir IP Estático a Assinante

Para atribuir um IP estático a um assinante, você precisa associar o registro de IP Estático com o Assinante durante a criação ou atualização.

Padrão de Atribuição:

1. **Criar o IP Estático** (veja Criar IP Estático)
2. **Atribuir ao Assinante** usando o campo `static_ips`

Criar Assinante com IP Estático:

```
# Passo 1: Criar IP estático para APN "internet"
STATIC_IP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1",
      "ipv6_static_ip": "2606:4700:4700::1111"
    }
  }' | jq -r '.data.id')

# Passo 2: Criar assinante com IP estático atribuído
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"subscriber\": {
      \"imsi\": \"001001123456789\",
      \"key_set_id\": \"6ba7b810-9dad-41d1-80b4-00c04fd430c8\",
      \"epc_profile_id\": \"e58ed763-928c-4155-bee9-
fdbaaadc15f3\",
      \"static_ips\": [$STATIC_IP_ID]
    }
  }"
```

Atualizar Assinante Existente com IP Estático:

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "static_ips": ["c56a4180-65aa-42ec-a945-5fd21dec0538",
"5a8f9c3e-2d4b-4e6f-8a1b-3c5d7e9f0a2b"]
    }
  }'
```

Múltiplos IPs Estáticos (APNs Diferentes):

Um assinante pode ter múltiplos IPs estáticos, desde que cada um seja para um APN diferente:

```
# Criar IP estático para APN "internet"
INTERNET_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "a1b2c3d4-e5f6-4a3b-8c2d-1e0f9a8b7c6d",
      "ipv4_static_ip": "100.64.1.1"
    }
  }' | jq -r '.data.id')

# Criar IP estático para APN "ims"
IMS_IP=$(curl -k -X POST
https://hss.example.com:8443/api/epc/static_ip \
  -H "Content-Type: application/json" \
  -d '{
    "static_ip": {
      "apn_profile_id": "b2c3d4e5-f6a7-4b1c-9d2e-3f4a5b6c7d8e",
      "ipv4_static_ip": "100.64.2.1"
    }
  }' | jq -r '.data.id')

# Atribuir ambos ao assinante
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d "{
    \"subscriber\": {
      \"imsi\": \"001001123456789\",
      \"key_set_id\": \"6ba7b810-9dad-41d1-80b4-00c04fd430c8\",
      \"epc_profile_id\": \"e58ed763-928c-4155-bee9-
fdbaaadc15f3\",
      \"static_ips\": [$INTERNET_IP, $IMS_IP]
    }
  }"
```

Regras de Validação:

- ✓ **Permitido**: Múltiplos IPs estáticos para APNs diferentes
- ✗ **Rejeitado**: Múltiplos IPs estáticos para o mesmo APN

Exemplo de Erro - APN Duplicada:

```
# Isso irá FALHAR se ambos os IPs estáticos referenciam o mesmo
APN
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "static_ips": ["c56a4180-65aa-42ec-a945-5fd21dec0538",
"5a8f9c3e-2d4b-4e6f-8a1b-3c5d7e9f0a2b"]
    }
  }'

# Resposta de Erro:
{
  "errors": {
    "static_ips": [
      "static ips per apn per subscriber must be unique. eg a
subscriber may not be assigned static ip 100.64.1.1 for internet
and also 100.64.1.2 for internet"
    ]
  }
}
```

Veja Também:

- Criar Assinante - Provisionamento de assinante
- Atualizar Assinante - Modificando a configuração do assinante
- Exemplo Completo de Provisionamento de IP Estático - Fluxo de trabalho de ponta a ponta

---

Gerenciamento de Roaming

Perfis de roaming controlam se os assinantes podem acessar dados e serviços IMS em redes visitadas. Perfis são atribuídos a assinantes e consistem em regras correspondidas por MCC/MNC.

Listar Perfis de Roaming

Endpoint: `GET /api/roaming/profile`

Criar Perfil de Roaming

Endpoint: `POST /api/roaming/profile`

Corpo da Requisição:

```
{
  "roaming_profile": {
    "name": "Apenas Operadoras dos EUA",
    "data_action_if_no_rules_match": "deny",
    "ims_action_if_no_rules_match": "deny",
    "roaming_rules": []
  }
}
```

Valores de Ação:

- `"allow"` - Permitir
- `"deny"` - Negar

Ações Padrão:

- `data_action_if_no_rules_match` - Ação quando nenhuma regra de roaming corresponde
- `ims_action_if_no_rules_match` - Ação padrão específica de IMS

Listar Regras de Roaming

Endpoint: `GET /api/roaming/rule`

Criar Regra de Roaming

Endpoint: `POST /api/roaming/rule`

Corpo da Requisição:

```
{
  "roaming_rule": {
    "name": "Permitir AT&T",
    "mcc": "310",
    "mnc": "410",
    "data_action": "allow",
    "ims_action": "allow"
  }
}
```

Campos:

- `mcc` - Código do País Móvel (3 dígitos)
- `mnc` - Código da Rede Móvel (2-3 dígitos)
- `data_action` - `"allow"` ou `"deny"` serviços de dados
- `ims_action` - `"allow"` ou `"deny"` serviços IMS/vocais

Veja Também:

- Documentação de Roaming - Configuração detalhada e exemplos
- Fluxos de Protocolo - Como o controle de roaming funciona em fluxos Diameter

---

Gerenciamento de EIR

OmniHSS funciona como um Registro de Identidade de Equipamento (EIR) através da interface Diameter S13. As regras de EIR são expressões regulares correspondidas contra o IMEISV de um dispositivo, depois seu IMEI, e então o IMSI do assinante. Dentro de cada etapa, as regras são avaliadas em ordem crescente de `precedence` e a primeira correspondência vence.

Veja Documentação de EIR para verificação detalhada de identidade de equipamento, fluxos da interface S13, validação de IMEI e precedência de regras.

Listar Regras de EIR

Recuperar todas as regras de EIR.

Endpoint: `GET /api/eir/rule`

A lista é retornada na ordem de `id`. Classifique por `precedence` para ver a ordem de avaliação.

Obter Regra de EIR

Recuperar uma única regra de EIR.

Endpoint: `GET /api/eir/rule/:id`

Criar Regra de EIR

Adicionar uma regra e colocá-la na ordem de avaliação.

Endpoint: `POST /api/eir/rule`

Corpo da Requisição:

```
{
  "action": "blacklist",
  "regex": "35979139461640",
  "precedence": 1
}
```

Campos Obrigatórios:

- `action` - `"whitelist"` (permitir), `"blacklist"` (negar) ou `"greylist"` (permitir, mas rastrear)
- `regex` - Expressão regular testada contra o IMEISV, depois o IMEI, depois o IMSI. Deve compilar

Campos Opcionais:

- `precedence` - Inteiro de 1 a 65535. Valores mais baixos são avaliados primeiro. Omiti-lo adiciona a regra após a atual de maior precedência. Fornecer um valor insere a regra naquela posição: cada regra existente com uma precedência igual ou superior se move para baixo em uma unidade, independentemente de o valor já estar em uso

Resposta de Sucesso (201 Criado):

```
{
  "status": "success",
  "response": {
    "id": "0362b7ab-9bd6-4d8a-8b1a-2f4c6e8d0a1c",
    "action": "blacklist",
    "regex": "35979139461640",
    "precedence": 1
  }
}
```

A `precedence` na resposta é a posição que a regra recebeu. Falhas de validação (ação desconhecida, regex vazio ou não compilável, `precedence` fora de 1 a 65535) retornam HTTP 422 no formato de erro padrão.

Atualizar Regra de EIR

Alterar a ação, padrão ou posição de uma regra.

Endpoint: `PATCH /api/eir/rule/:id` ou `PUT /api/eir/rule/:id`

Ambos os métodos aplicam apenas os campos presentes no corpo; campos omitidos mantêm seus valores atuais. Um `PUT` para um id que não existe cria a regra com aquele id (HTTP 200), colocada pelas mesmas regras de precedência que um `POST`.

Alterar `precedence` move a regra para aquela posição e desloca apenas as regras entre suas antigas e novas posições. Omitir `precedence`, ou enviar seu valor atual, deixa a ordenação inalterada. Não envie `null`: isso limpa a precedência da regra e a regra é então avaliada após todas as regras numeradas.

Deletar Regra de EIR

Remover uma regra.

Endpoint: `DELETE /api/eir/rule/:id`

Resposta: HTTP 200 com `"response": ""`.

Excluir uma regra deixa um espaço na numeração de precedência. As regras restantes mantêm sua ordem relativa.

Casos de Uso:

- Bloquear dispositivos roubados (lista negra de IMEIs específicos)
- Restringir tipos de dispositivos (lista negra por padrão TAC)
- Permitir apenas dispositivos aprovados (padrões de lista branca, com `unknown_equipment_behaviour` alterado de seu padrão de `whitelist` para `blacklist` ou `reject_unknown_equipment`; veja Comportamento de Equipamento Desconhecido)
- Substituir uma regra ampla para um dispositivo dando à regra específica uma precedência mais baixa

Veja Também:

- Documentação de EIR - Lógica de correspondência, precedência de regras e fluxos S13
- Fluxos de Protocolo - Fluxo da interface S13 e verificação de EIR
- Visão Geral da Arquitetura - Função EIR do OmniHSS

Documentação Adicional

Para mais informações, veja a seguinte documentação:

- **Status e Saúde** - Endpoints de verificação de saúde da API
- **Tratamento de Erros** - Erros comuns e solução de problemas

- **Exemplos de Uso da API** - Fluxos de trabalho completos de provisionamento

---

← Voltar ao Guia de Operações | Próximo: Painel de Controle →

Status e Saúde da API

← Voltar para a Referência da API

Status do Sistema

Verifique se a API está respondendo.

Endpoint: `GET /api/status`

Exemplo de Requisição:

```
curl -k https://hss.example.com:8443/api/status
```

Exemplo de Resposta:

```
{
  "status": "ok"
}
```

Caso de Uso: Verificação de saúde para balanceadores de carga e sistemas de monitoramento.

← Voltar para a Referência da API

Visão Geral da Arquitetura do OmniHSS

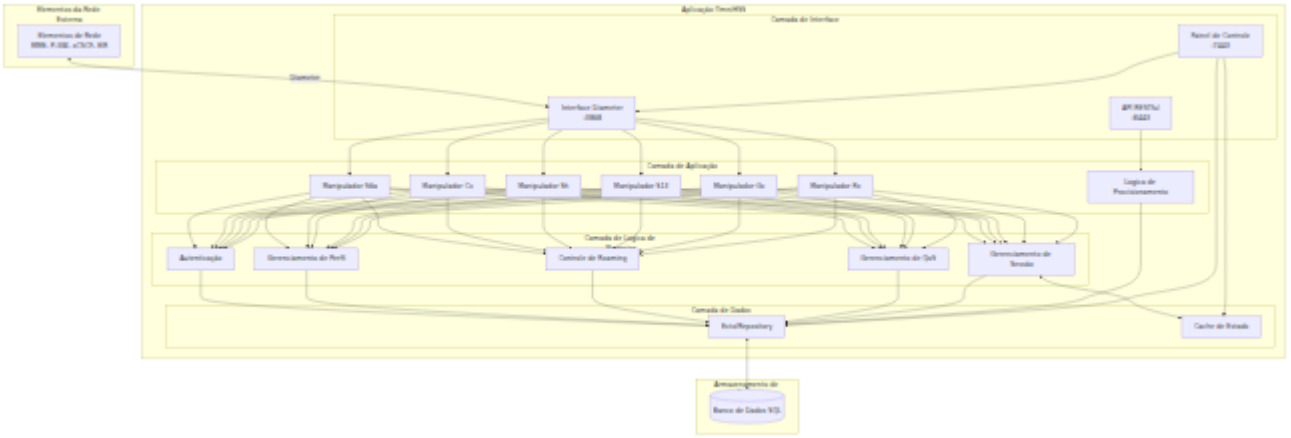
← Voltar ao Guia de Operações

Índice

- Visão Geral do Sistema
- Arquitetura dos Componentes
- Pilha Diameter
- Camada de Aplicação
- Camada de Dados
- Interfaces Externas
- Arquitetura de Implantação

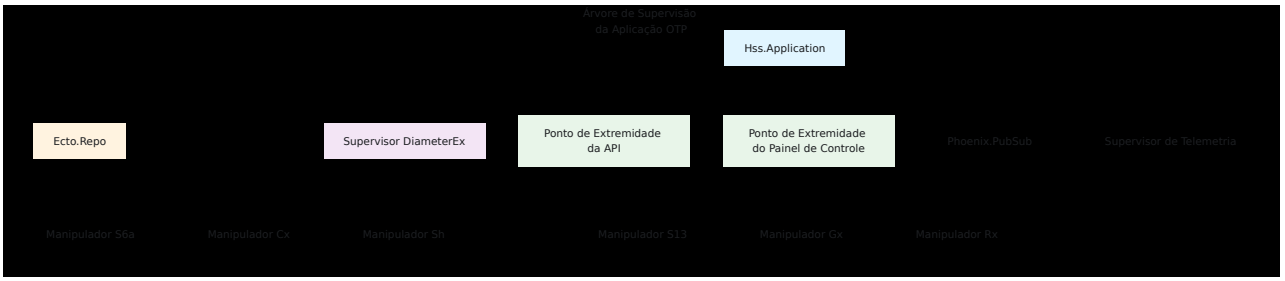
Visão Geral do Sistema

OmniHSS é construído sobre Elixir e a plataforma Erlang/OTP, fornecendo um sistema altamente concorrente e tolerante a falhas, projetado para cargas de trabalho de telecomunicações. A arquitetura segue uma abordagem em camadas com clara separação de preocupações.



Arquitetura dos Componentes

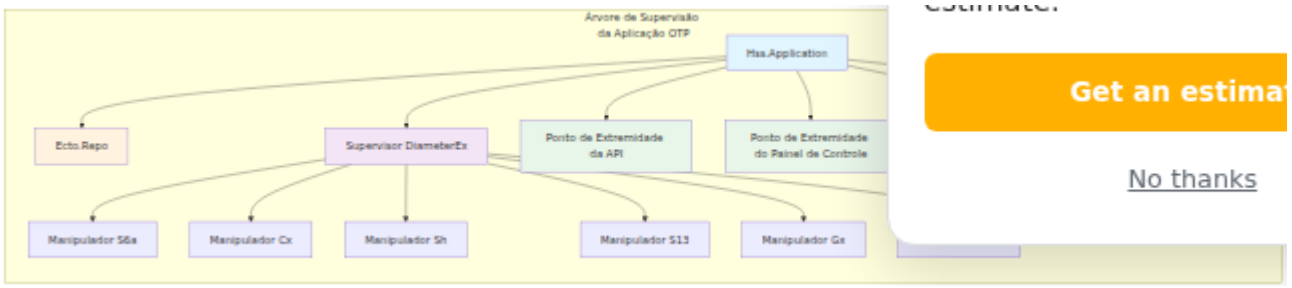
Componentes Principais



Manipuladores de Aplicação Diameter

Cada aplicação Diameter (S6a, Cx, Sh, S13, Gx, Rx) é implementada como um módulo manipulador DiameterEx que:

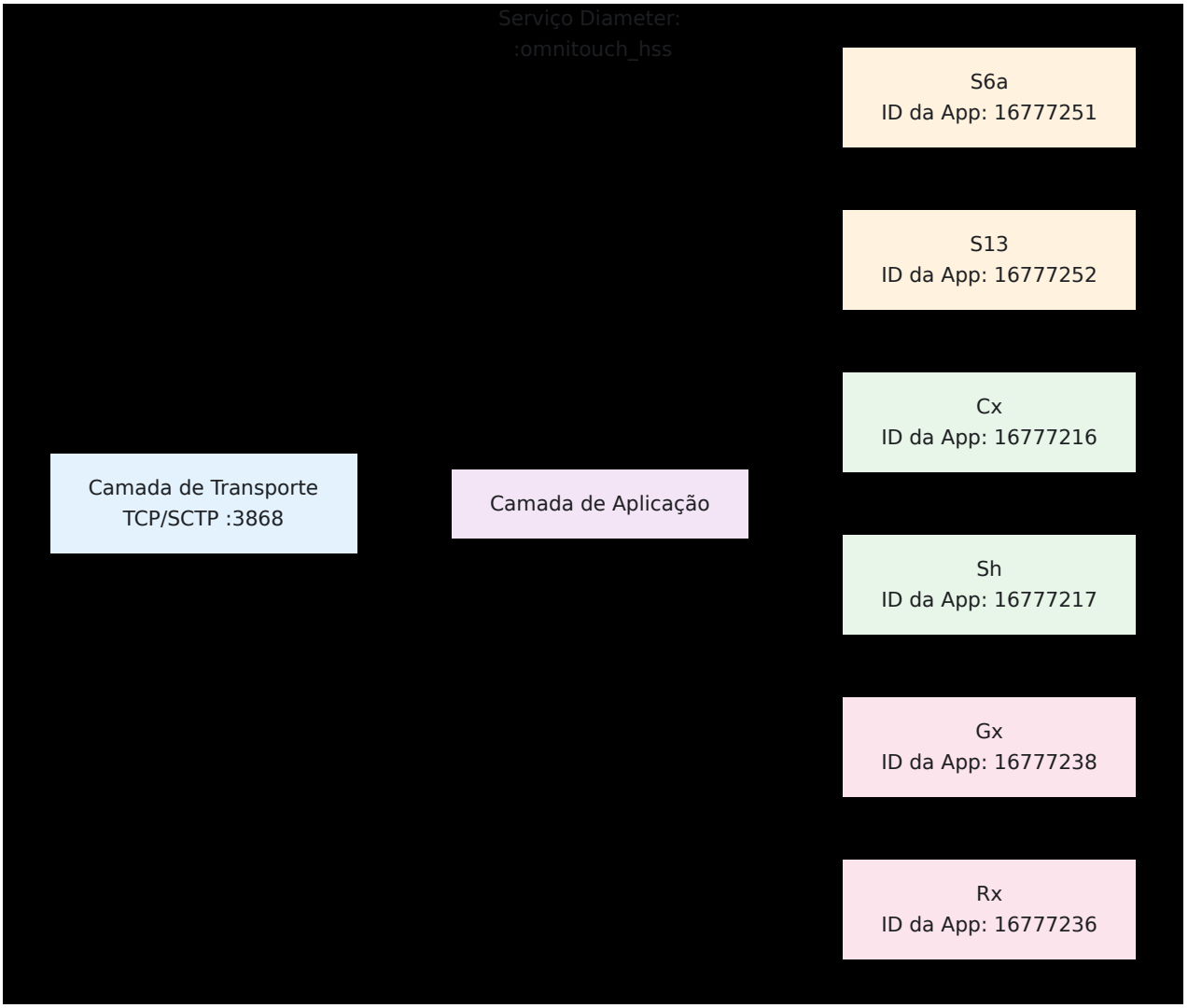
1. **Registra-se no DiameterEx** - Inscreve-se em IDs de Aplicação Diameter específicos
2. **Valida Solicitações** - Extrai AVPs, valida o estado do assinante
3. **Processa Lógica de Negócios** - Chama módulos de lógica de negócios apropriados
4. **Constrói Respostas** - Monta mensagens de resposta Diameter com AVPs
5. **Lida com Erros** - Retorna códigos de resultado Diameter apropriados



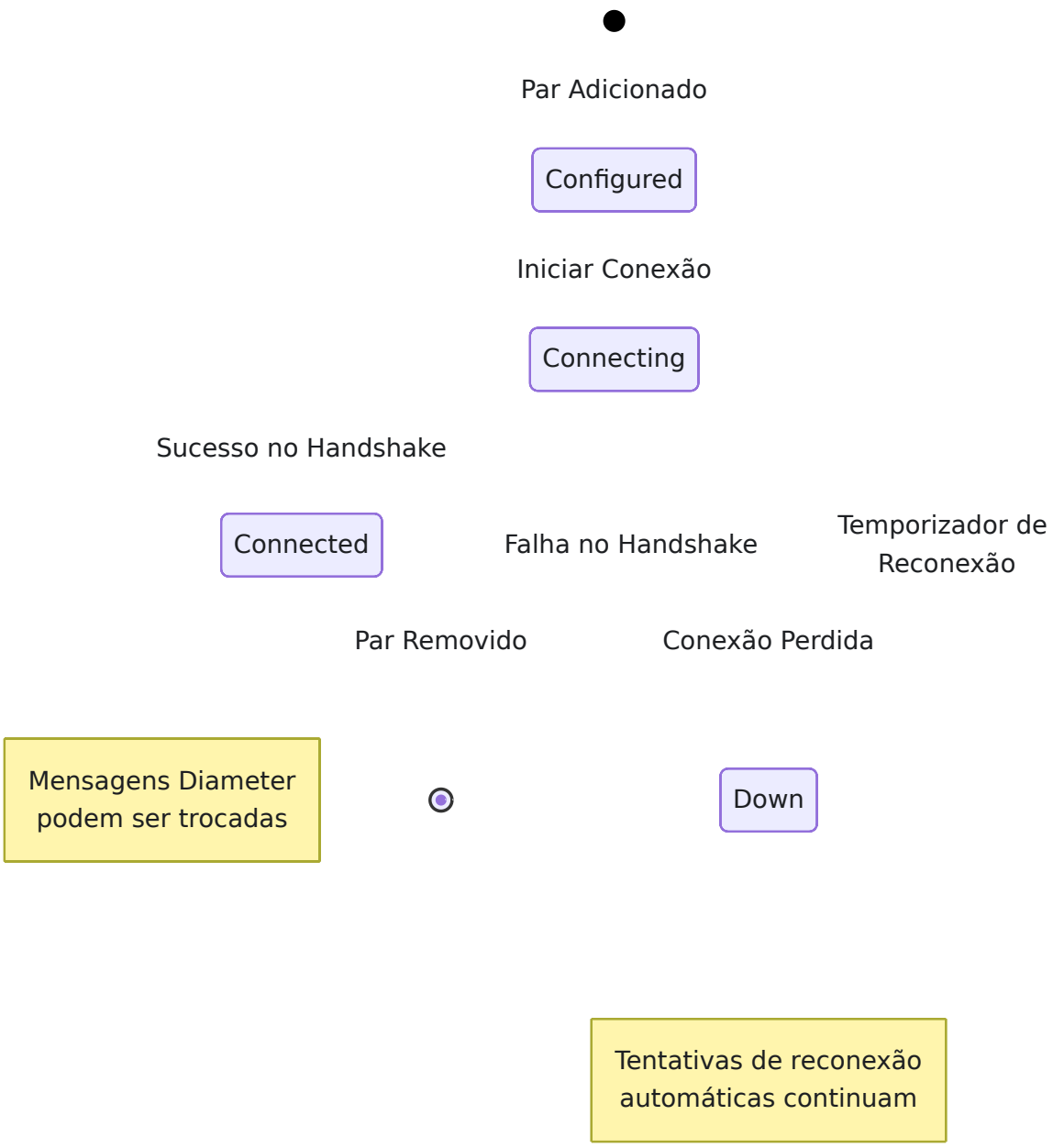
Pilha Diameter

Configuração do Serviço Diameter

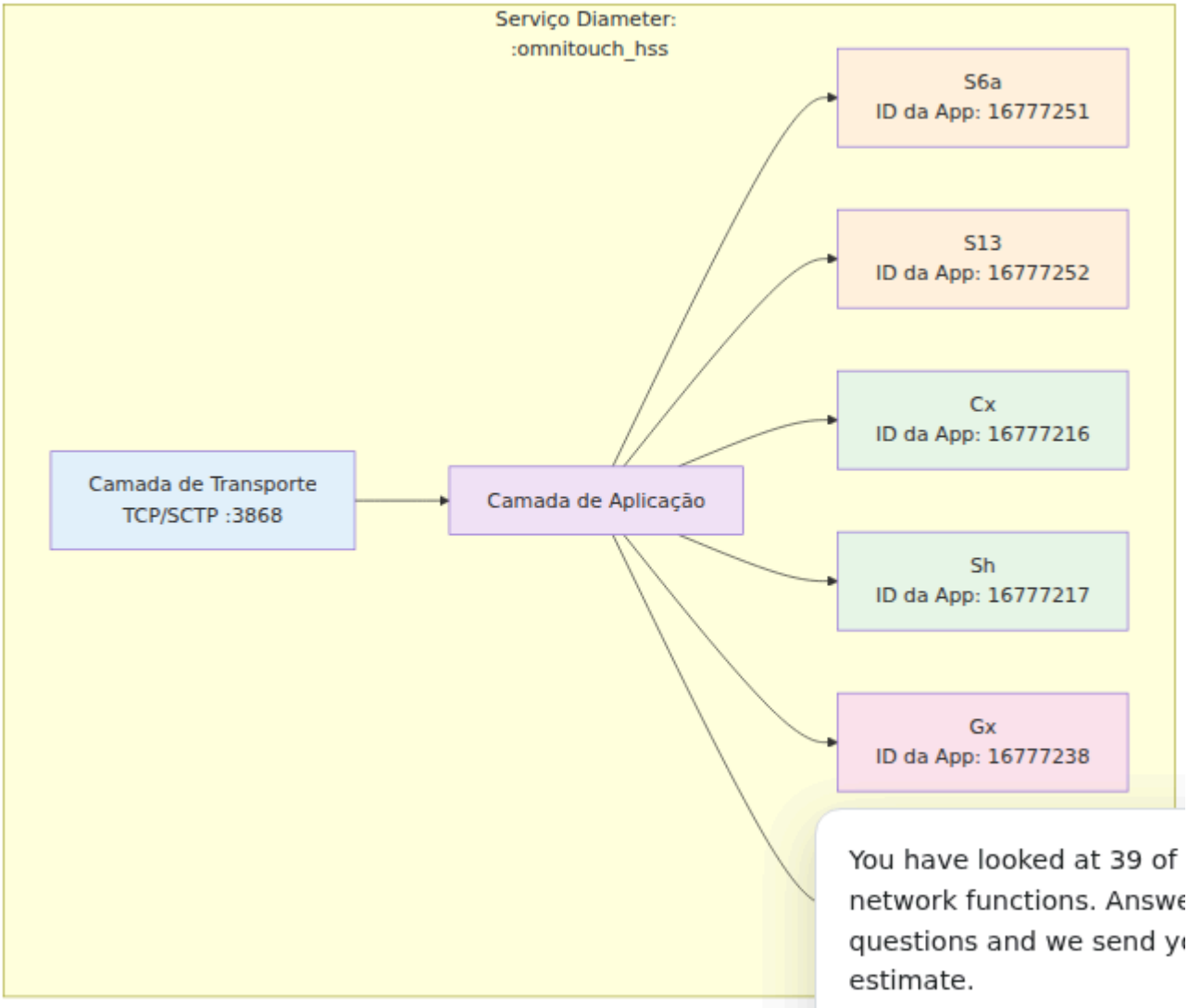
OmniHSS configura um único serviço Diameter com várias aplicações suportadas:



Gerenciamento de Conexão de Par



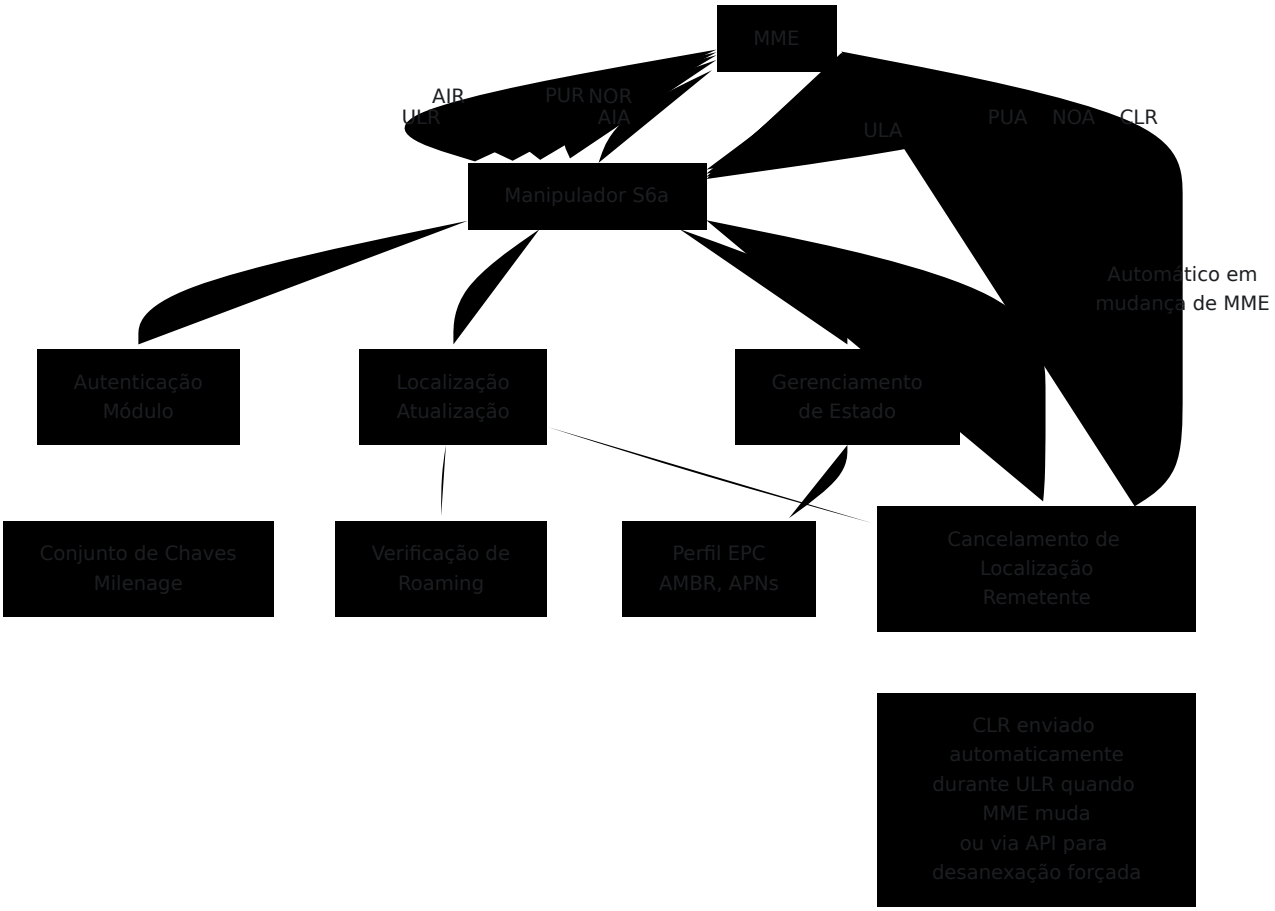
Fluxo de Mensagens Diameter



Camada de Aplicação

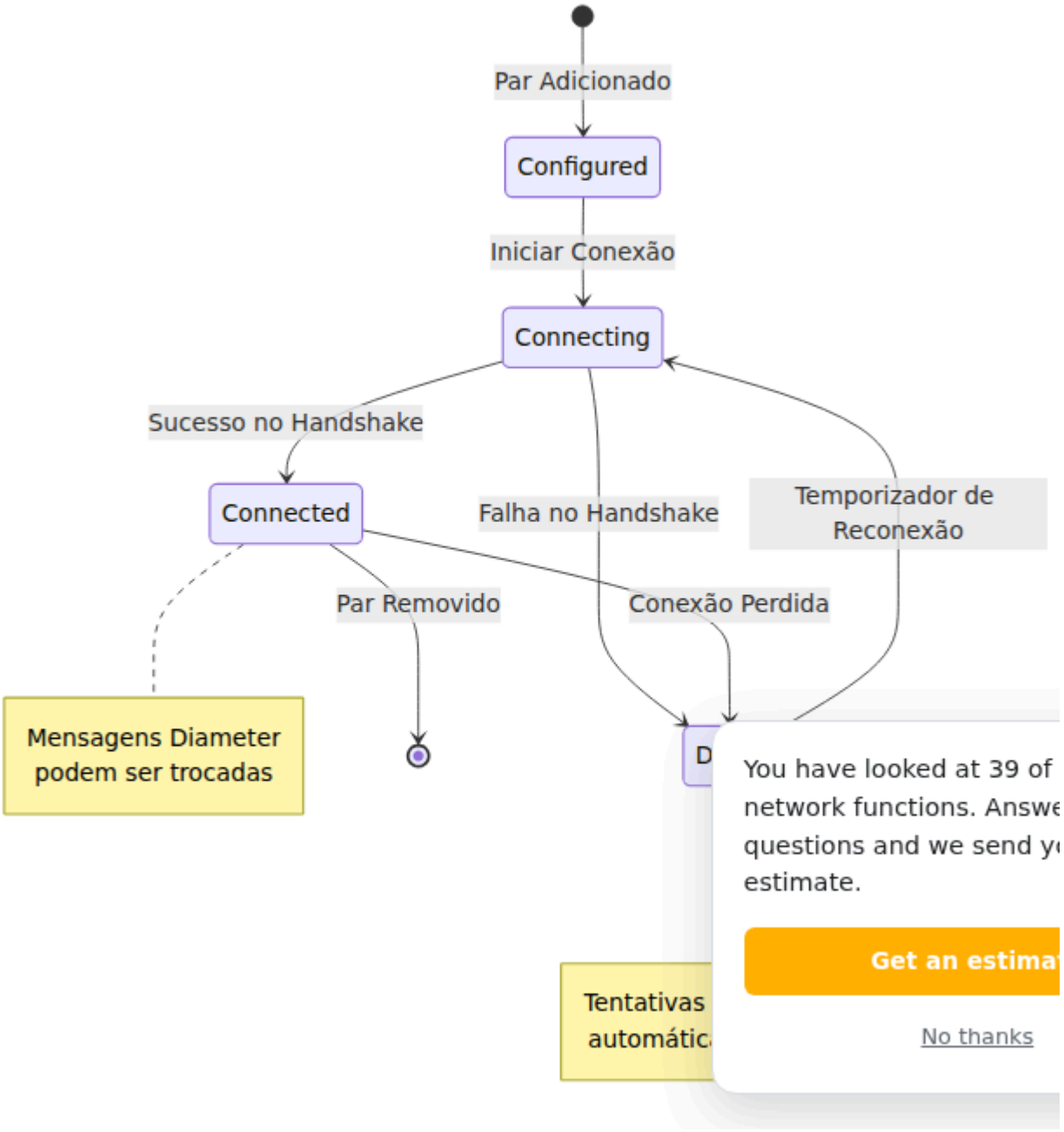
Interface S6a (LTE/EPC)

Lida com autenticação e gerenciamento de mobilidade para redes LTE.



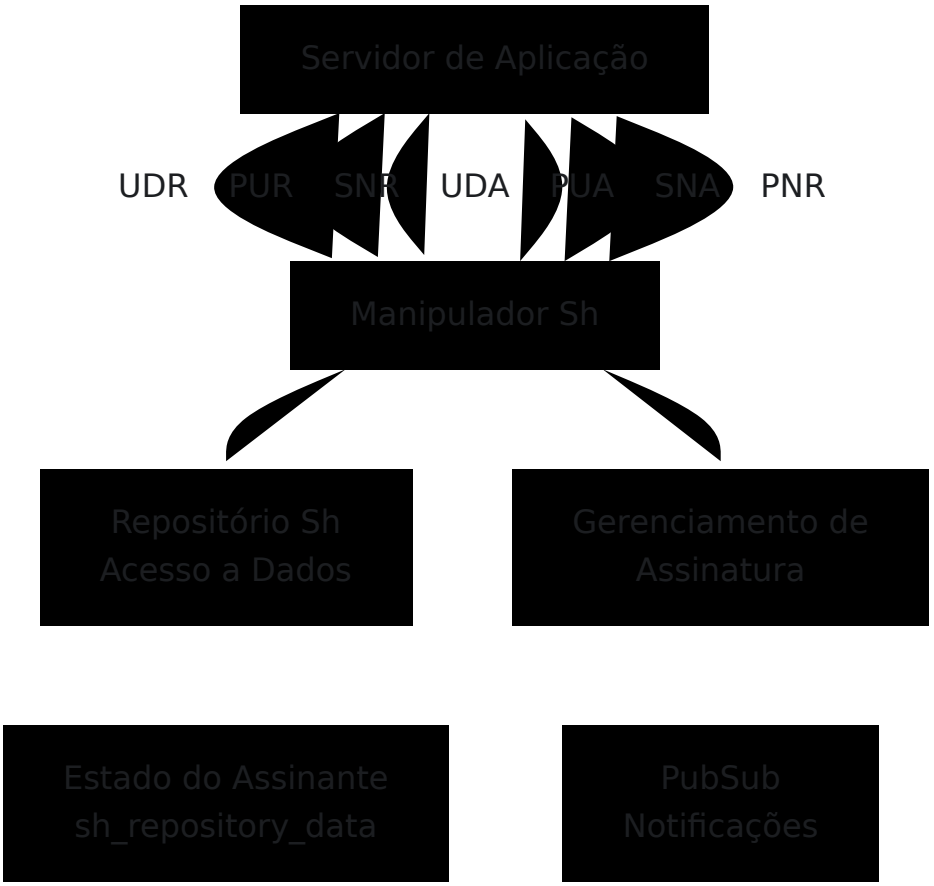
Interface Cx (IMS)

Lida com registro e autenticação IMS.



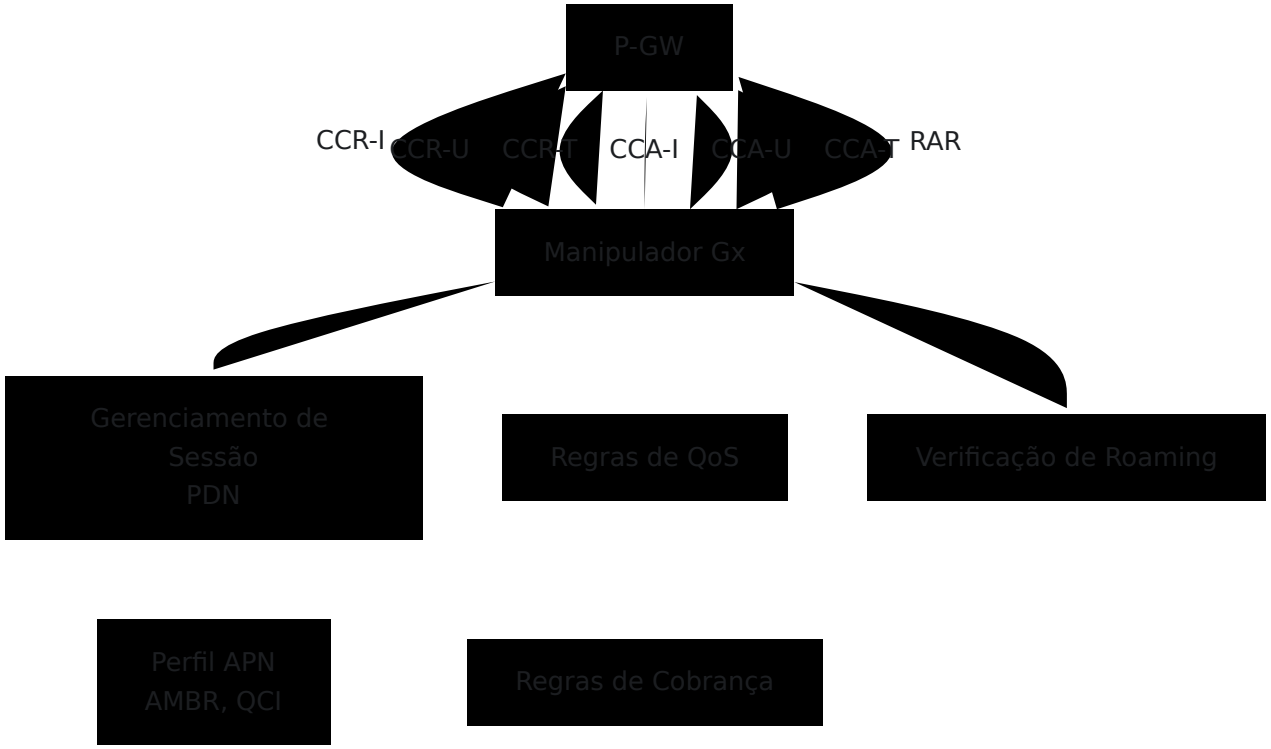
Interface Sh (Dados de Perfil IMS)

Fornece servidores de aplicação IMS acesso aos dados de perfil do assinante.



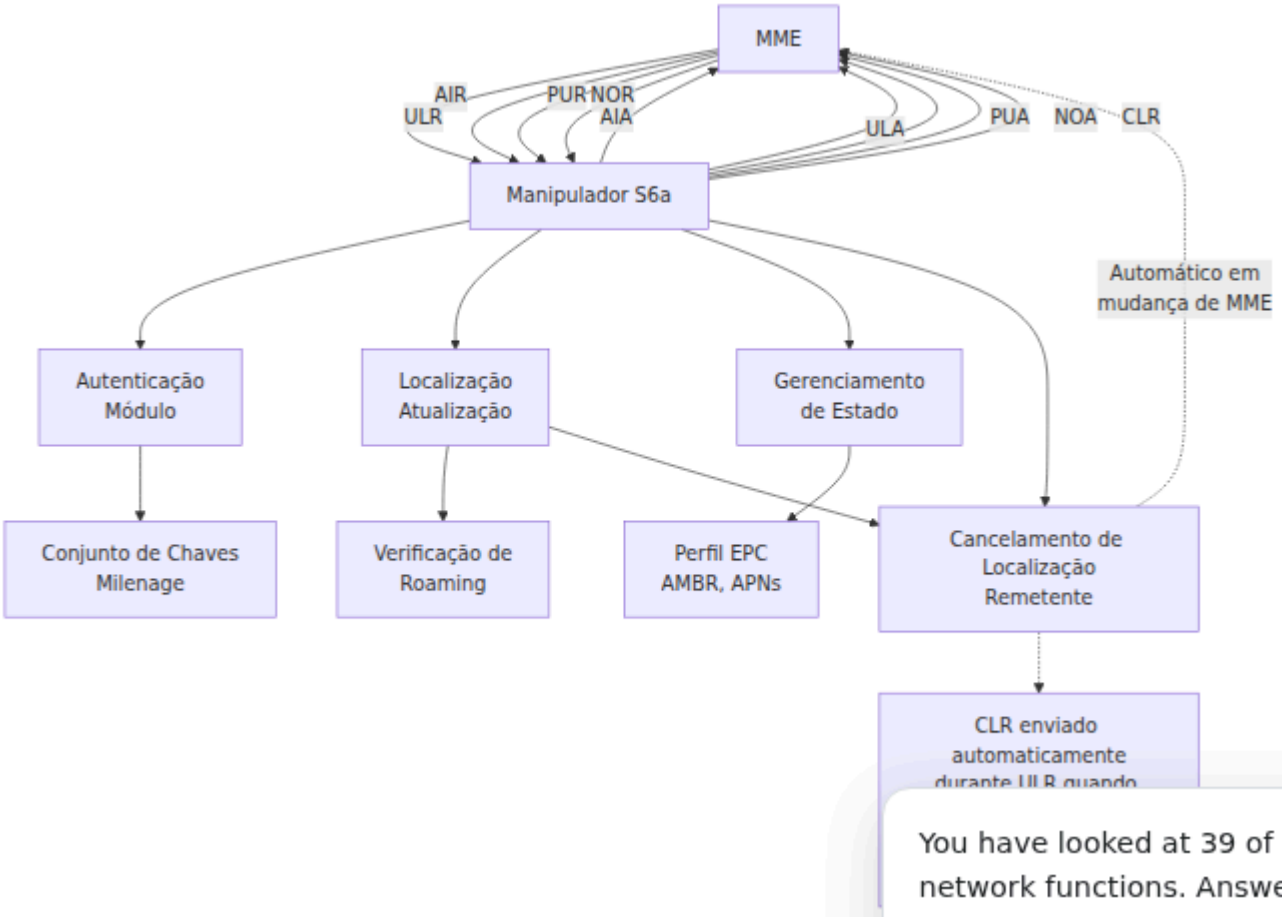
Interface Gx (Controle de Políticas)

Gerencia controle de políticas e cobrança para sessões de dados. **Veja Documentação PCRF para detalhes.**



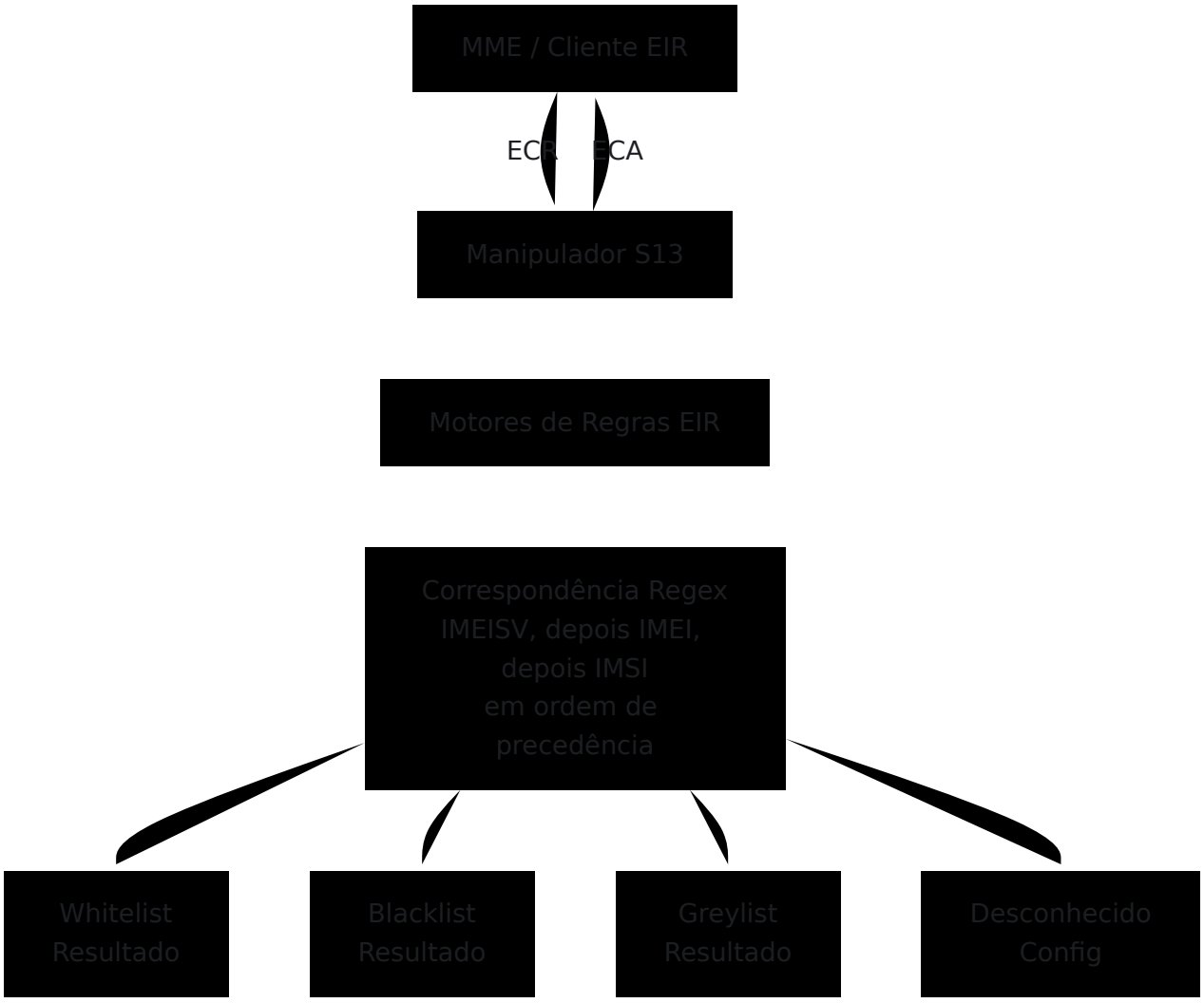
Interface Rx (Mídia IMS)

Controla a política de mídia IMS e portadoras dedicadas para VoLTE. **Veja Documentação PCRF para detalhes.**



Interface S13 (EIR)

OmniHSS atua como o EIR. Ele valida o IMEISV e IMEI do dispositivo, e opcionalmente o IMSI do assinante quando o ECR carrega um AVP de Nome de Usuário, contra regras de identidade de equipamento avaliadas em ordem de precedência ascendente. **Veja Documentação EIR para detalhes.**



Camada de Dados

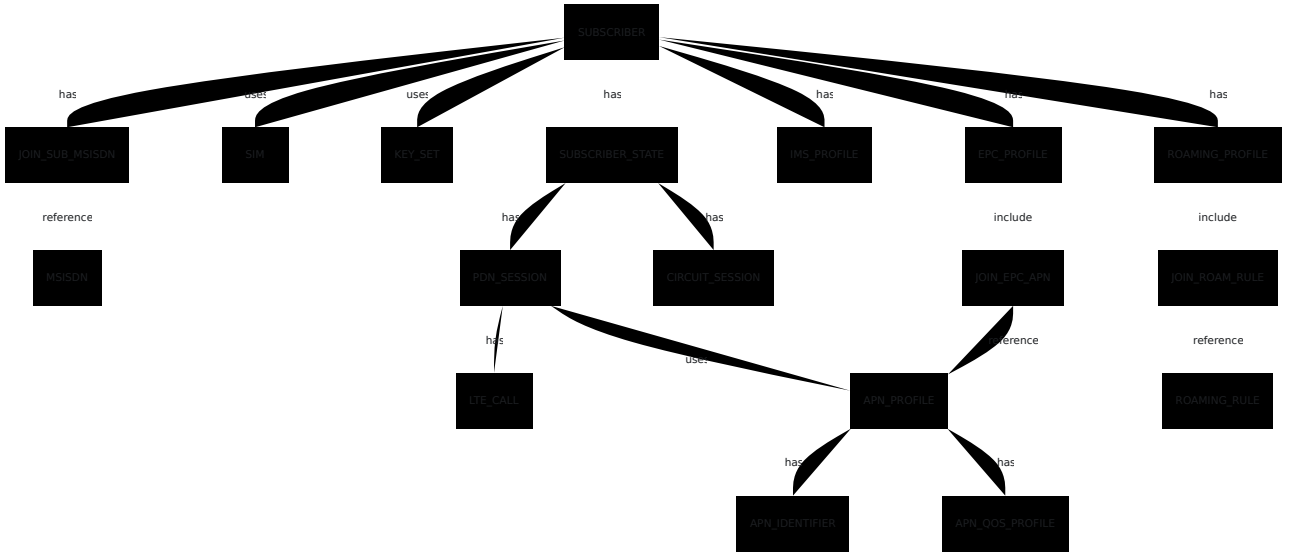
Backend do Banco de Dados

OmniHSS usa **PostgreSQL** como seu backend de banco de dados e **Ecto** como sua camada de abstração de banco de dados. Todas as tabelas usam chaves primárias UUID para identificação globalmente única entre sites.

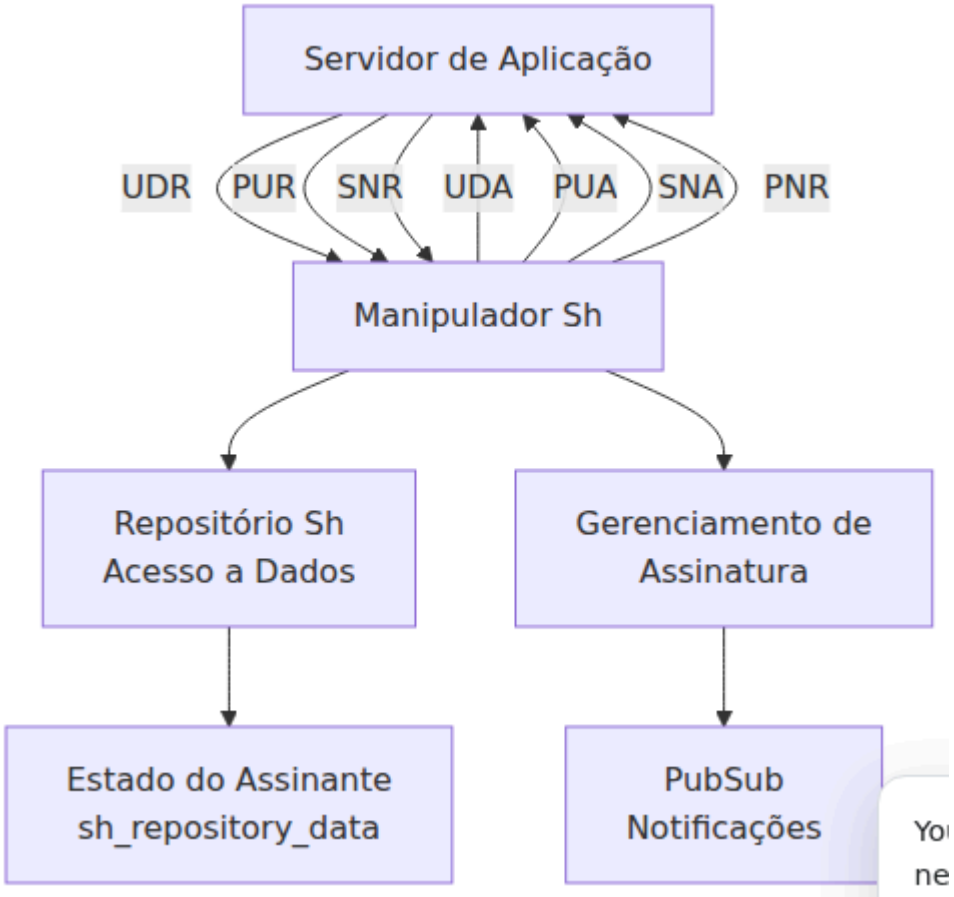
Para implantações em múltiplos sites, OmniHSS utiliza a replicação lógica bidirecional do PostgreSQL. Cada site opera como um primário de leitura/escrita independente com replicação assíncrona baseada em WAL entre sites.

Veja Replicação de Banco de Dados para configuração e operações de replicação em múltiplos sites.

Visão Geral do Esquema do Banco de Dados

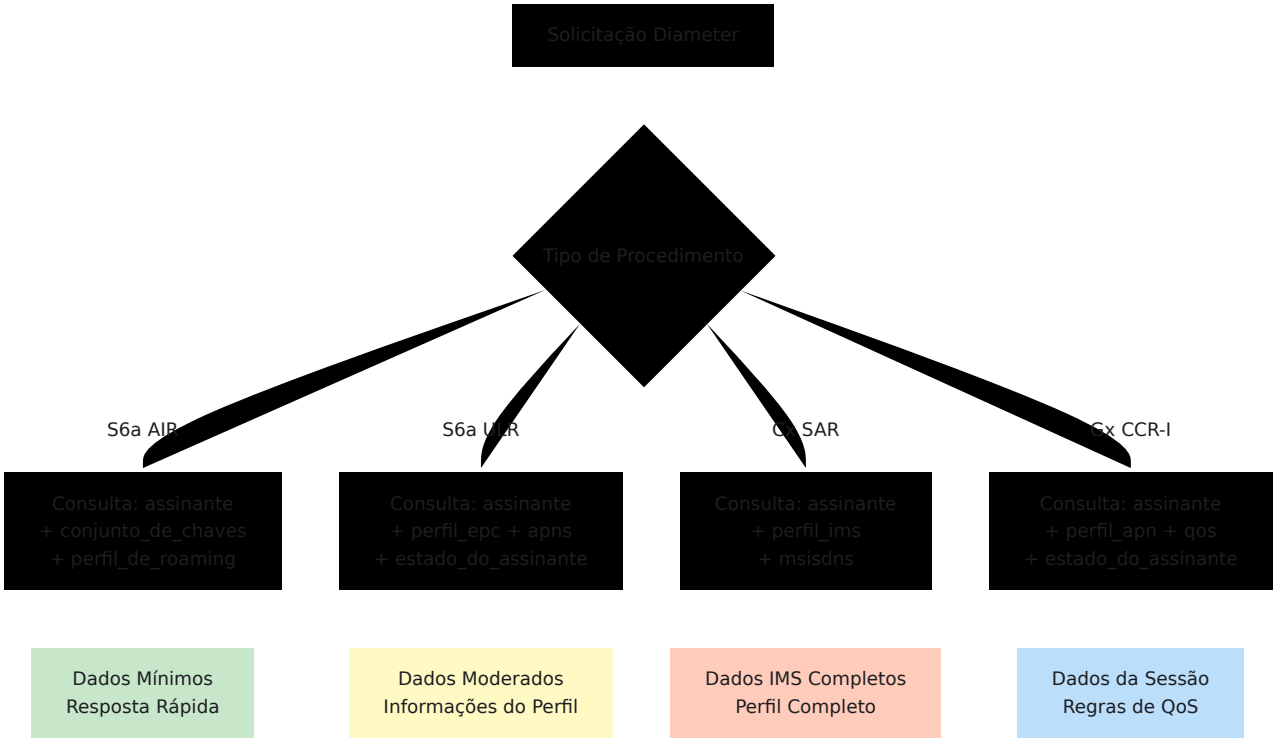


Padrão de Repositório Ecto



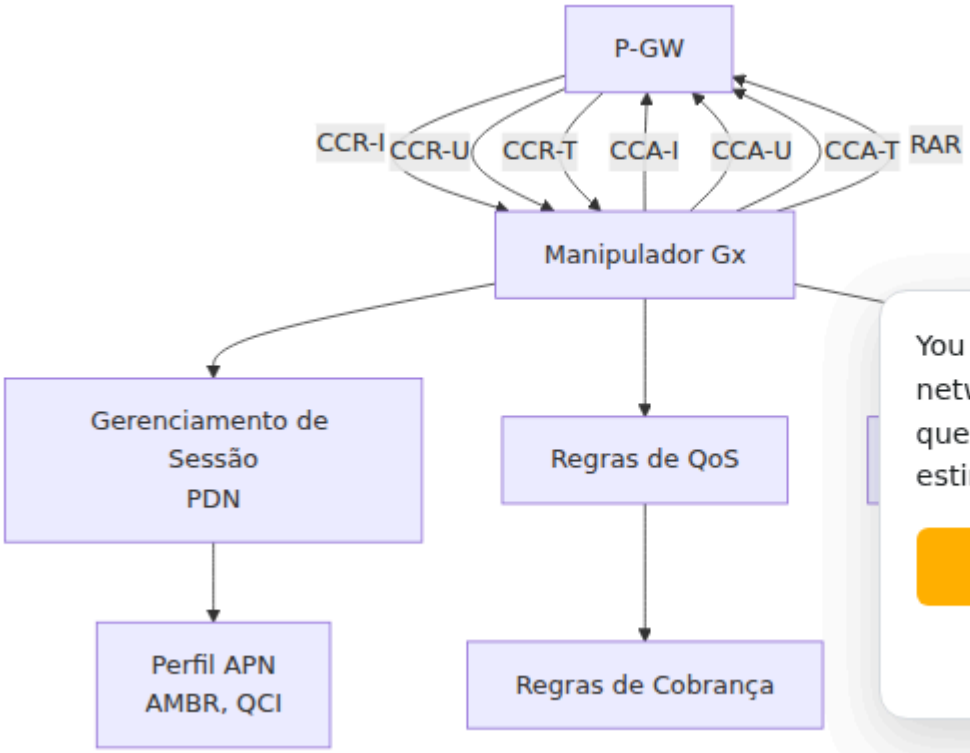
Estratégia de Consulta Otimizada

Cada procedimento Diameter utiliza consultas otimizadas que pré-carregam apenas associações necessárias:



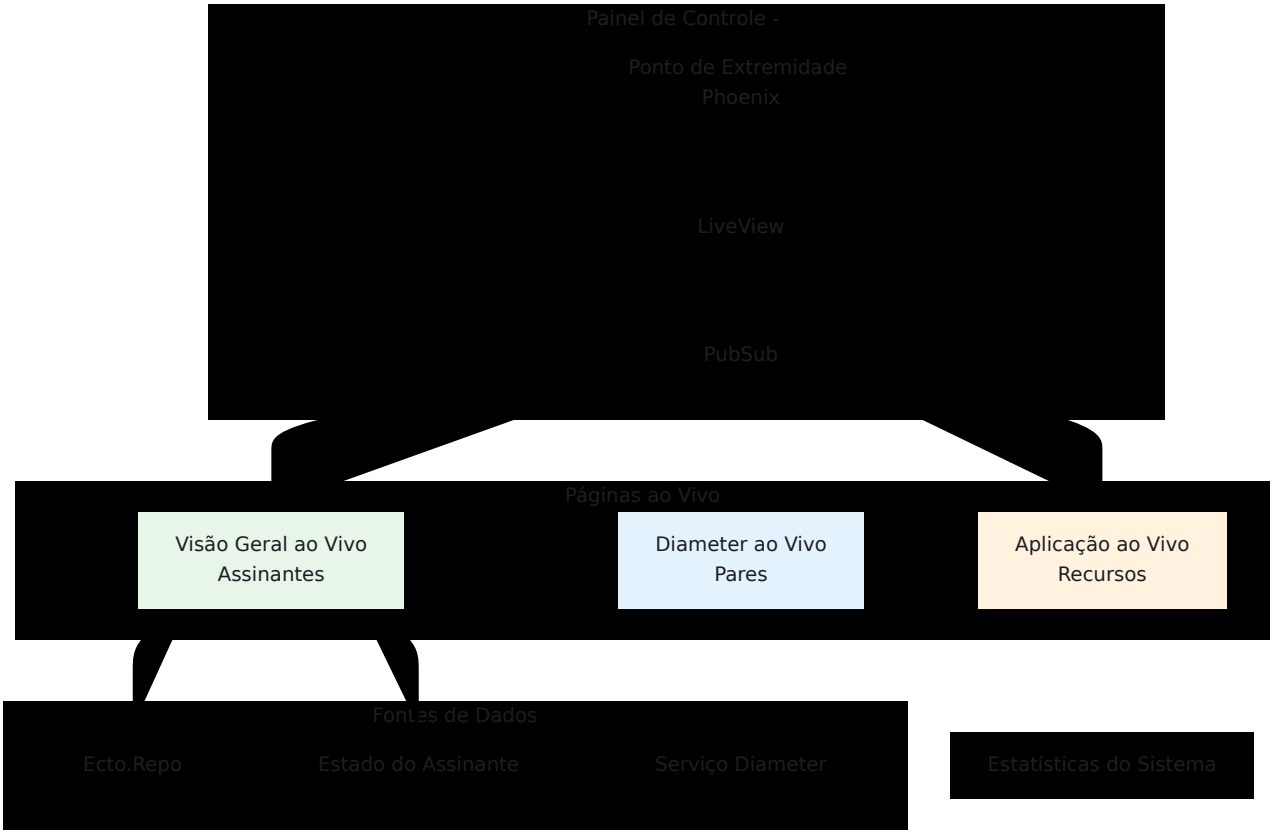
Interfaces Externas

Arquitetura da API



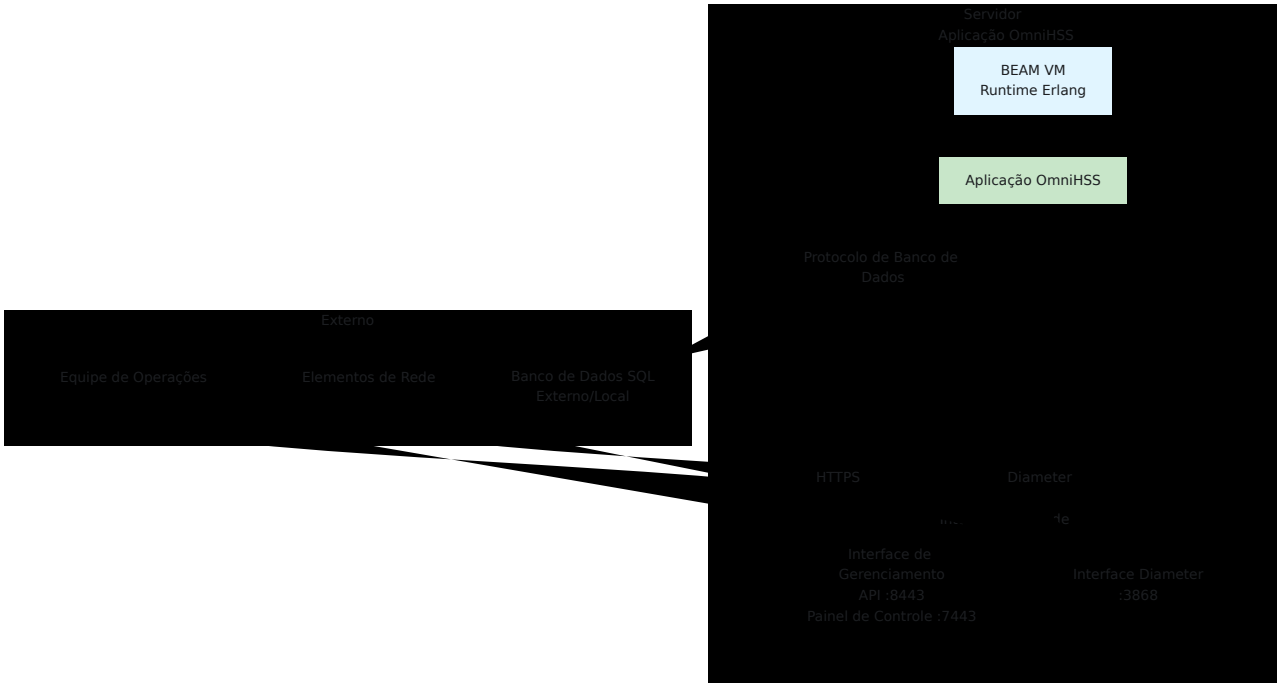
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima
No thanks

Arquitetura do Painel de Controle



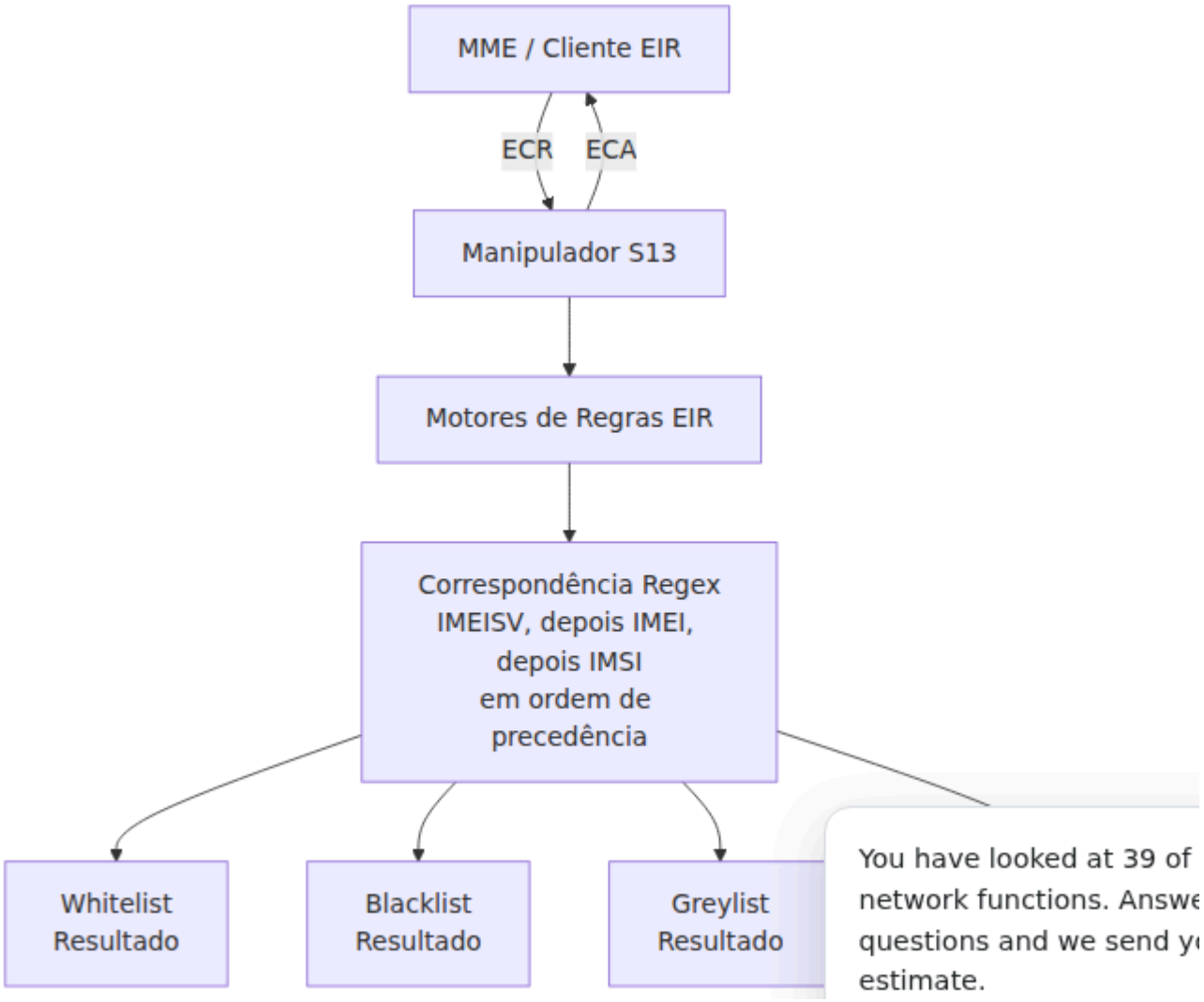
Arquitetura de Implantação

Implantação em Um Único Nó



Implantação HA em Múltiplos Sites (Replicação Lógica)

Para implantações de alta disponibilidade em sites geográficos, o OmniHSS utiliza replicação lógica bidirecional do PostgreSQL. Cada site tem um banco de dados de leitura/escrita independente.



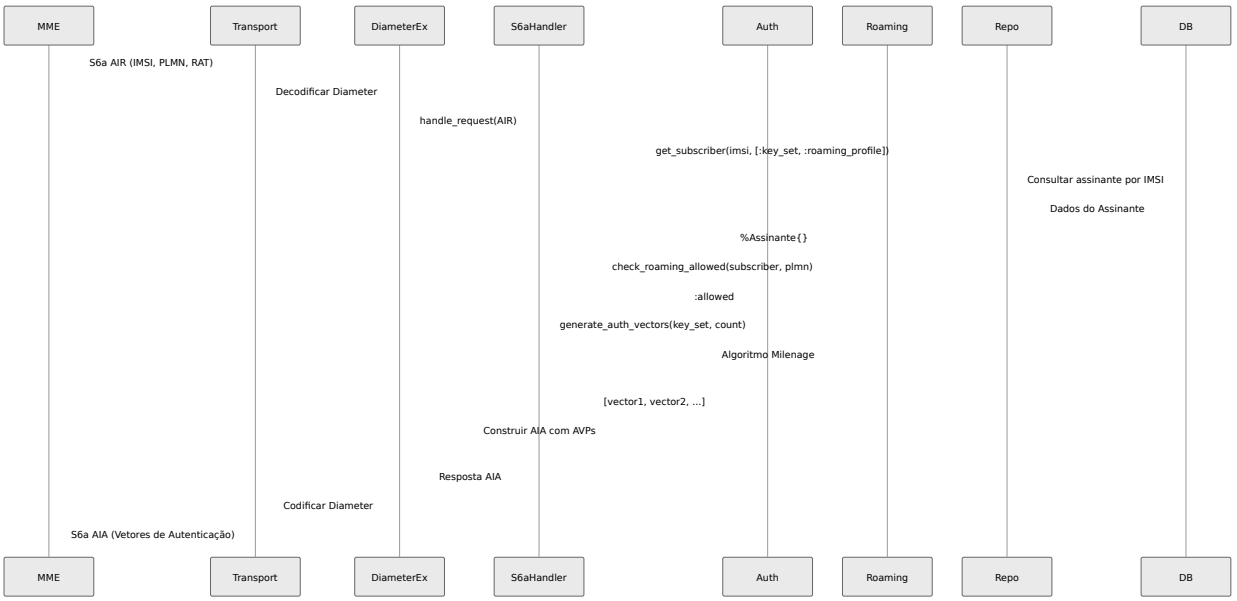
Características Principais:

- **Ambos os Sites Leem/Escrevem**: Escritas locais são instantâneas, nunca bloqueadas por site remoto
- **Replicação Assíncrona**: Mudanças são transmitidas via WAL. A replicação não adiciona penalidade de latência de escrita
- **Tolerante a Partições**: Ambos os sites continuam operando se o link intersite cair. WAL enfileira localmente e se recupera na restauração
- **Sem Quorum**: Nenhum número mínimo de nós é necessário. Funciona de forma idêntica com um site ou muitos
- **Chaves Primárias UUID**: IDs globalmente únicos eliminam conflitos entre sites

Veja Replicação de Banco de Dados para configuração e operações detalhadas.

Exemplo de Fluxo de Processo: Autenticação

Este exemplo mostra o fluxo completo para uma solicitação de autenticação:



Princípios Arquitetônicos Chave

1. Tolerância a Falhas

- Árvores de supervisão Erlang/OTP reiniciam automaticamente processos com falha
- Manipuladores Diameter isolados evitam falhas em cascata
- Pooling de conexão de banco de dados com reconexão automática

2. Concorrência

- Cada solicitação Diameter é tratada em seu próprio processo
- Sem estado compartilhado entre manipuladores de solicitação
- Pooling de conexão de banco de dados para consultas paralelas

3. Modularidade

- Cada aplicação Diameter em módulo separado
- Clara separação entre interface, lógica de negócios e camadas de dados
- Algoritmos de autenticação plugáveis

4. Desempenho

- Consultas de banco de dados otimizadas com pré-carregamento seletivo
- Transferência mínima de dados para cada tipo de procedimento
- Pooling de conexão e keepalive

5. Observabilidade

- Monitoramento em tempo real via Painel de Controle
- Registro estruturado em toda a aplicação
- Rastreamento de status de pares Diameter
- Rastreamento do estado do assinante com timestamps

← Voltar ao Guia de Operações | Próximo: Configuração →

Guia de Configuração do OmniHSS

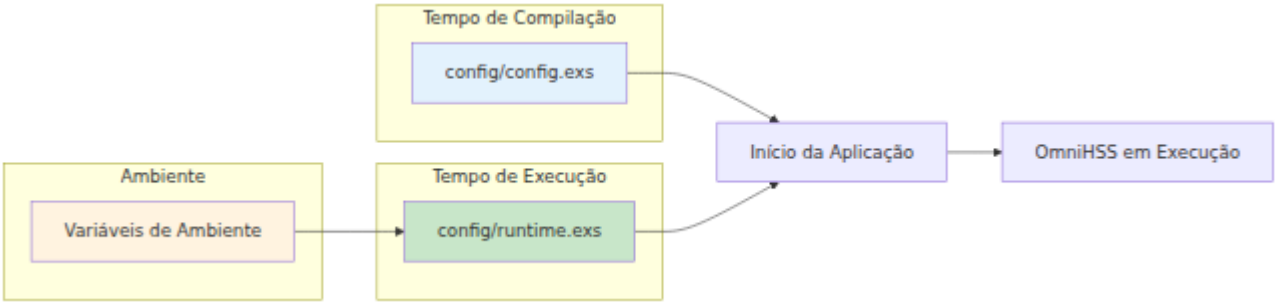
← Voltar ao Guia de Operações

Índice

- Visão Geral do Arquivo de Configuração
- Configuração do Cliente de Licença
- Configuração em Tempo de Execução
- Configuração do Banco de Dados
- Configuração do Diameter
- Configuração de Rede
  - Configuração do PLMN de Origem
  - Configuração do Núcleo HSS
- Configuração do IMS
- Configuração do EIR
- Configuração da API e do Painel de Controle
- Fluxo de Trabalho de Configuração

Visão Geral do Arquivo de Configuração

OmniHSS utiliza dois arquivos de configuração principais:



config/config.exs (Tempo de Compilação)

Contém configuração estática que não muda entre ambientes:

- Configuração da página do Painel de Controle
- Configuração do endpoint da API
- Configurações de telemetria

config/runtime.exs (Tempo de Execução)

Contém configuração específica do ambiente que muda por implantação:

- Parâmetros de conexão do banco de dados
- Configuração de pares Diameter
- Configurações do PLMN de origem
- Seleção de S-CSCF do IMS
- Vínculos de interface de rede

Configuração do Cliente de Licença

O Cliente de Licença valida a licença do HSS com um servidor de licença remoto:

```
# config/runtime.exs

config :license_client,
  # Endpoints da API do servidor de licença (lista para failover)
  license_server_api_urls:
["https://license.example.com:8443/api"],

  # Nome da organização licenciada
  licensee: "Seu Nome de Organização",

  # Identificador do produto
  product_name: "omnihss"
```

Parâmetros de Configuração da Licença:

Parâmetro	Descrição	Necessário	Exem
`license_server_api_urls`	Lista de URLs do servidor de licença	Sim	`["https://10.0.`
`licensee`	Nome da organização na licença	Sim	`"ACME Telecom"`
`product_name`	Identificador do produto para licença	Sim	`"omnihss"`

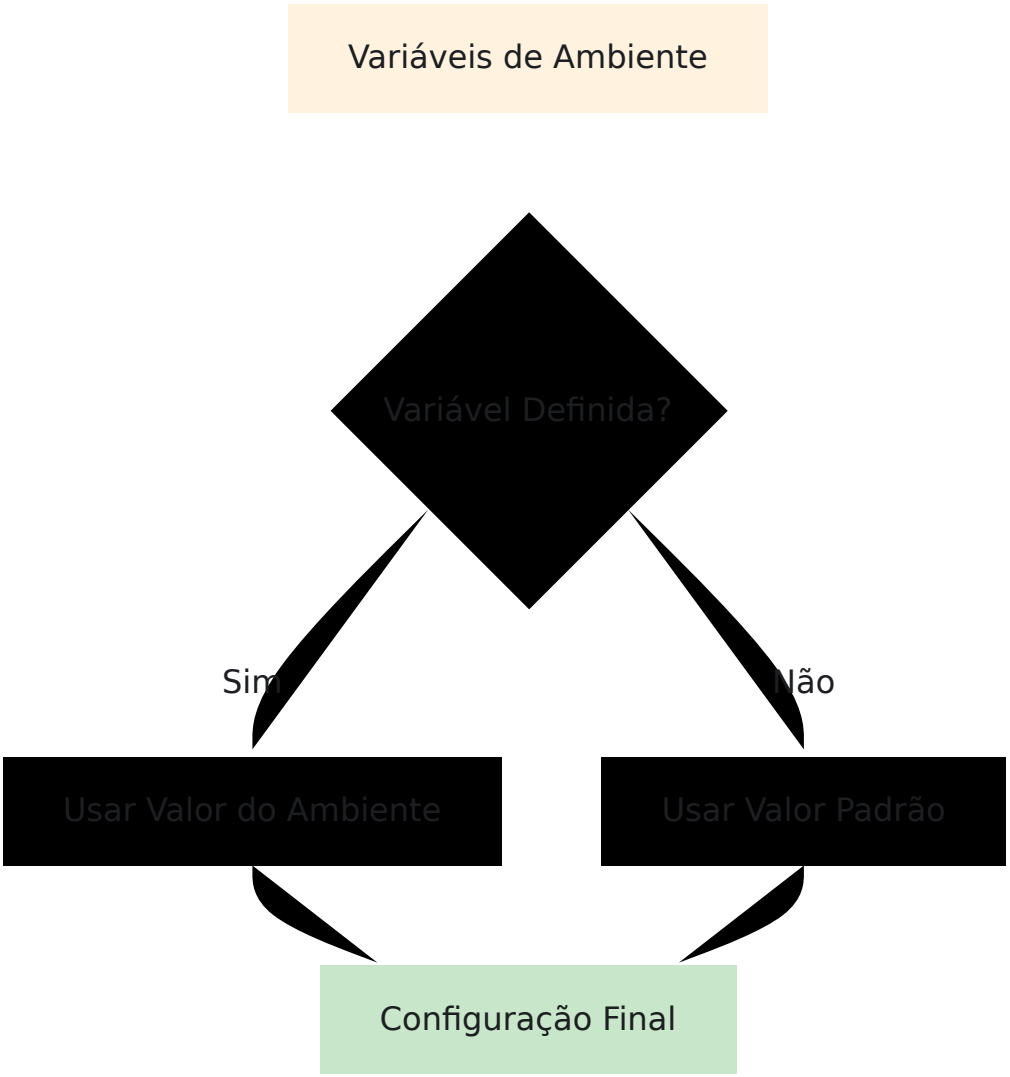
Notas Importantes:

- O servidor de licença deve ser acessível a partir do HSS
- Use HTTPS para validação segura da licença
- Múltiplas URLs fornecem capacidade de failover
- A validação da licença ocorre na inicialização e periodicamente

---

Configuração em Tempo de Execução

Prioridade de Configuração



Padrão de Variável de Ambiente

OmniHSS segue este padrão para configuração:

- Nomes de variáveis de ambiente são MAIÚSCULAS com sublinhados
- Valores padrão são fornecidos em runtime.exs
- Credenciais do banco de dados devem usar variáveis de ambiente em produção

Configuração do Banco de Dados

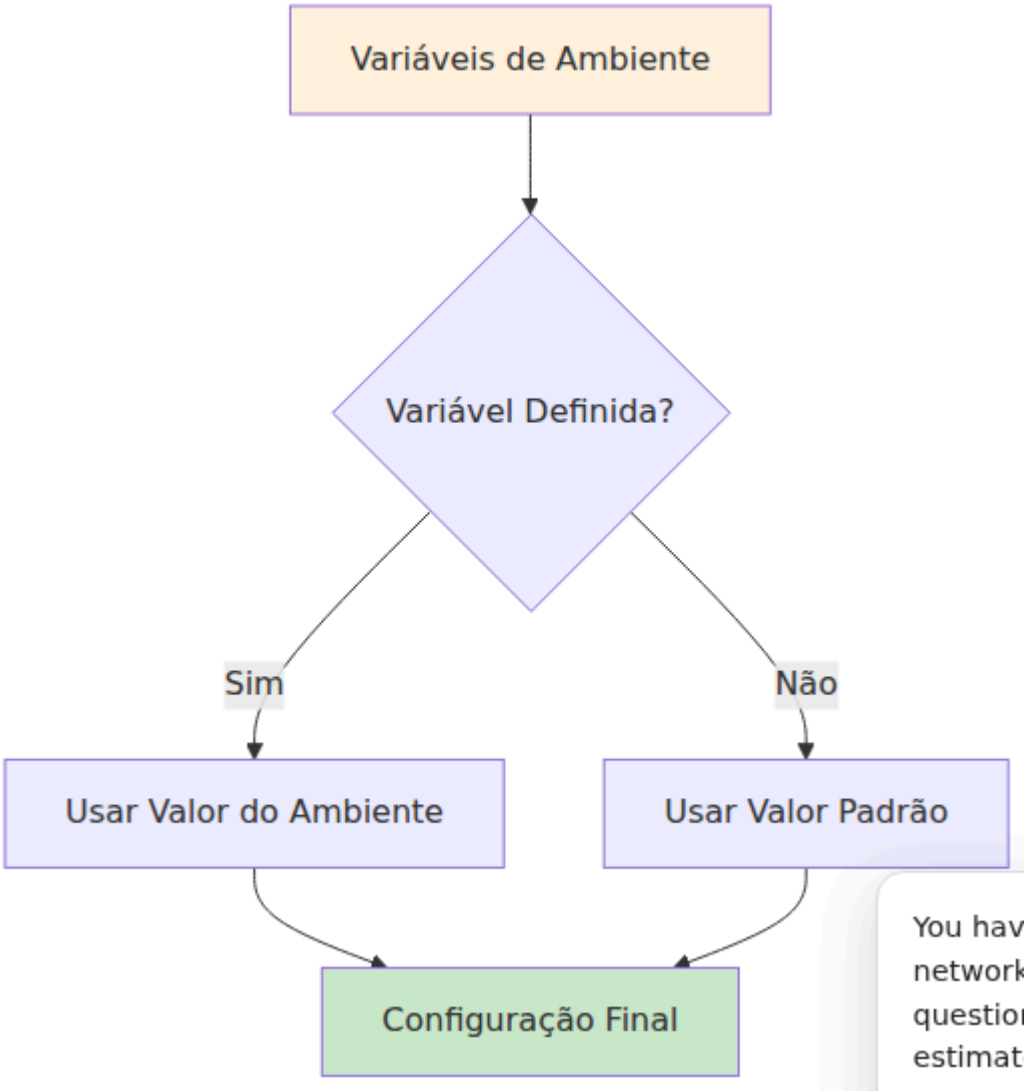
Configuração Básica do Banco de Dados

```
# config/runtime.exs

config :hss, Hss.Repo,
  # Parâmetros de conexão do banco de dados
  username: System.get_env("DATABASE_USERNAME", "root"),
  password: System.get_env("DATABASE_PASSWORD", "password"),
  hostname: System.get_env("DATABASE_HOSTNAME", "localhost"),
  database: System.get_env("DATABASE_NAME", "omnihss"),

  # Configurações do pool de conexão
  pool_size:
String.to_integer(System.get_env("DATABASE_POOL_SIZE", "20")),

  # Timeouts (em milissegundos)
  timeout: 15_000,
  connect_timeout: 15_000,

  # Opções adicionais
  show_sensitive_data_on_connection_error: false
```

Parâmetros de Configuração do Banco de Dados

Parâmetro	Descrição	Padrão	Recomendação
`username`	Nome de usuário do banco de dados SQL	`"root"`	Use um usuário dedicado em produção
`password`	Senha do banco de dados SQL	`"password"`	Use uma senha forte, armazene em var de ambiente
`hostname`	Nome do servidor do banco de dados SQL	`"localhost"`	Use FQDN ou IP em produção
`database`	Nome do banco de dados	`"omnihss"`	Mantenha o padrão, a menos que múltiplas instâncias
`pool_size`	Tamanho do pool de conexão	`20`	Ajuste com base na carga (10-50 típico)

Ajuste do Tamanho do Pool



Diretrizes:

- Comece com 20 conexões
- Monitore por erros de "timeout do pool de conexão"
- Aumente em 10 se timeouts ocorrerem sob carga normal
- Cada conexão usa ~4MB de memória
- Muitas conexões podem degradar o desempenho do Banco de Dados SQL

Exemplo: Configuração do Banco de Dados em Produção

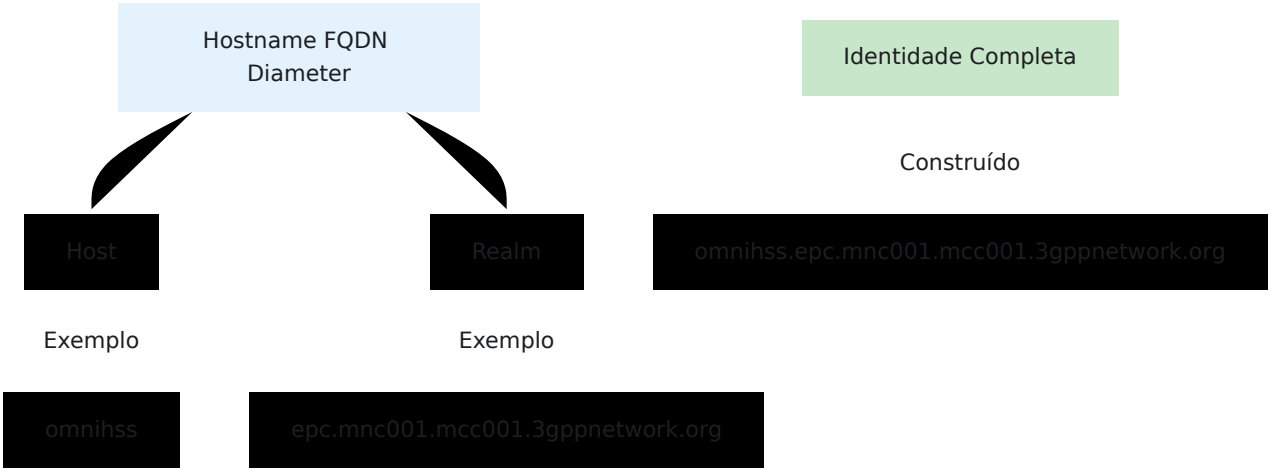
```
# config/runtime.exs - Exemplo de Produção

config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),      #
Necessário em produção
  password: System.fetch_env!("DATABASE_PASSWORD"),      #
Necessário em produção
  hostname: System.get_env("DATABASE_HOSTNAME",
"db.internal.example.com"),
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT",
"5432")),
  pool_size:
String.to_integer(System.get_env("DATABASE_POOL_SIZE", "30")),
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/postgres-ca.pem",
    verify: :verify_peer
  ]
```

Configuração do Diameter

Configuração do Serviço Diameter

```
# config/runtime.exs

diameter_config = %{
  service_name: :omnitouch_hss,

  # Vínculo de rede
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port:
String.to_integer(System.get_env("DIAMETER_LISTEN_PORT", "3868")),

  # Identidade Diameter
  host: System.get_env("DIAMETER_HOST", "omnihss"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),

  # Identificação do produto
  product_name: "OmniHSS",
  vendor_id: 10415,  # 3GPP
  supported_vendor_ids: [5535, 10415],

  # Configurações de protocolo
  request_timeout: 5000,

  # Configuração de pares
  peers: [
    # Adicione configurações de pares aqui
  ]
}

config :hss, :diameter, diameter_config
```

Configuração de Identidade Diameter



Diretrizes:

- **Host:** Nome curto do HSS (ex: "omnihss", "hss01")
- **Realm:** Realm Diameter que corresponde ao seu PLMN (ex: "epc.mnc001.mcc001.3gppnetwork.org")
- **Identidade Completa:** Construída como `{host}.{realm}`

Adicionando Pares Diameter

Configuração de Par Estático (Modo Conectar)

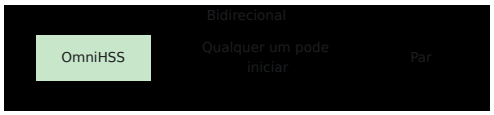
```
# config/runtime.exs

peers: [
  # Exemplo de Par MME
  %{
    host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.100",
    port: 3868,
    transport: :sctp,  # ou :tcp
    applications: [:s6a]
  },

  # Exemplo de Par P-GW
  %{
    host: "pgw01.epc.mnc001.mcc001.3gppnetwork.org",
    realm: "epc.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.101",
    port: 3868,
    transport: :sctp,
    applications: [:gx]
  },

  # Exemplo de Par I-CSCF
  %{
    host: "icscf01.ims.mnc001.mcc001.3gppnetwork.org",
    realm: "ims.mnc001.mcc001.3gppnetwork.org",
    ip: "10.7.25.102",
    port: 3868,
    transport: :tcp,
    applications: [:cx]
  }
]
```

Modo Apenas Escuta

Para ambientes onde os pares iniciam conexões com o HSS:

```
# config/runtime.exs

diameter_config = %{
  # ... outras configurações ...
  peers: []  # Vazio - aceitar apenas conexões de entrada
}
```

Modos de Conexão de Par Diameter







Seleção de Protocolo de Transporte

Transporte	Vantagens	Desvantagens	Recomendação
SCTP	Multi-streaming, melhor detecção de falhas	Requer suporte do kernel, configuração de firewall	Preferido para Diameter
TCP	Suporte universal, regras de firewall mais simples	Fluxo único, detecção de falhas mais lenta	Use se SCTP não estiver disponível

---

Configuração de Rede

Configuração do PLMN de Origem

O PLMN de origem identifica seu operador de rede:

```
# config/runtime.exs

config :hss, :home_plmn, %{
  mcc: System.get_env("HOME_PLMN_MCC", "001"),  # Código do País
Móvel
  mnc: System.get_env("HOME_PLMN_MNC", "001")   # Código da Rede
Móvel
}
```

Configuração do Núcleo HSS

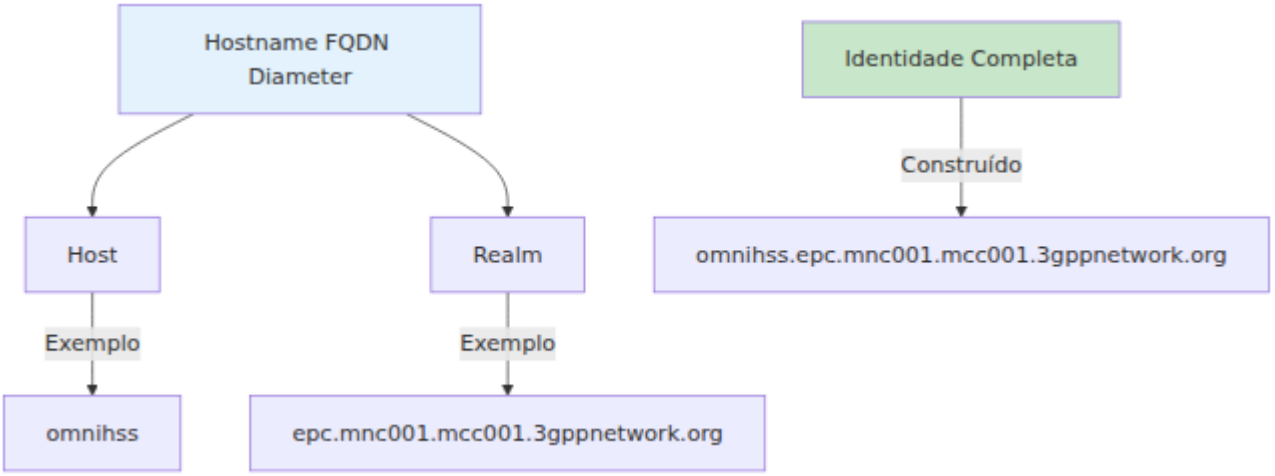
Essas configurações controlam o comportamento e os recursos do HSS:

```
# config/runtime.exs

config :hss,
  # Repositórios Ecto para operações de banco de dados
  ecto_repos: [Hss.Repo],

  # CLR (Cancel Location Request) na mudança de MME
  send_clr_on_mme_change: true,

  # Parar o serviço Diameter durante falhas no banco de dados
  stop_diameter_on_database_failure: true,

  # Configuração de aplicação de licença
  license_enforced: true,
  license_module: LicenseClient
```

Parâmetros do Núcleo HSS:

Parâmetro	Descrição	Padrão	Rec
`ecto_repos`	Lista de repositórios Ecto usados pela aplicação	`[Hss.Repo]`	Nec ope ban
`send_clr_on_mme_change`	Enviar Cancel Location Request quando o assinante muda de MME	`true`	Mar hab mol ade
`stop_diameter_on_database_failure`	Desabilitar o serviço Diameter se o banco de dados se tornar indisponível	`true`	Hab con dad
`license_enforced`	Habilitar aplicação de licença	`true`	Nec pro
`license_module`	Módulo que trata verificações de licença	`LicenseClient`	Não

Formato do Código PLMN



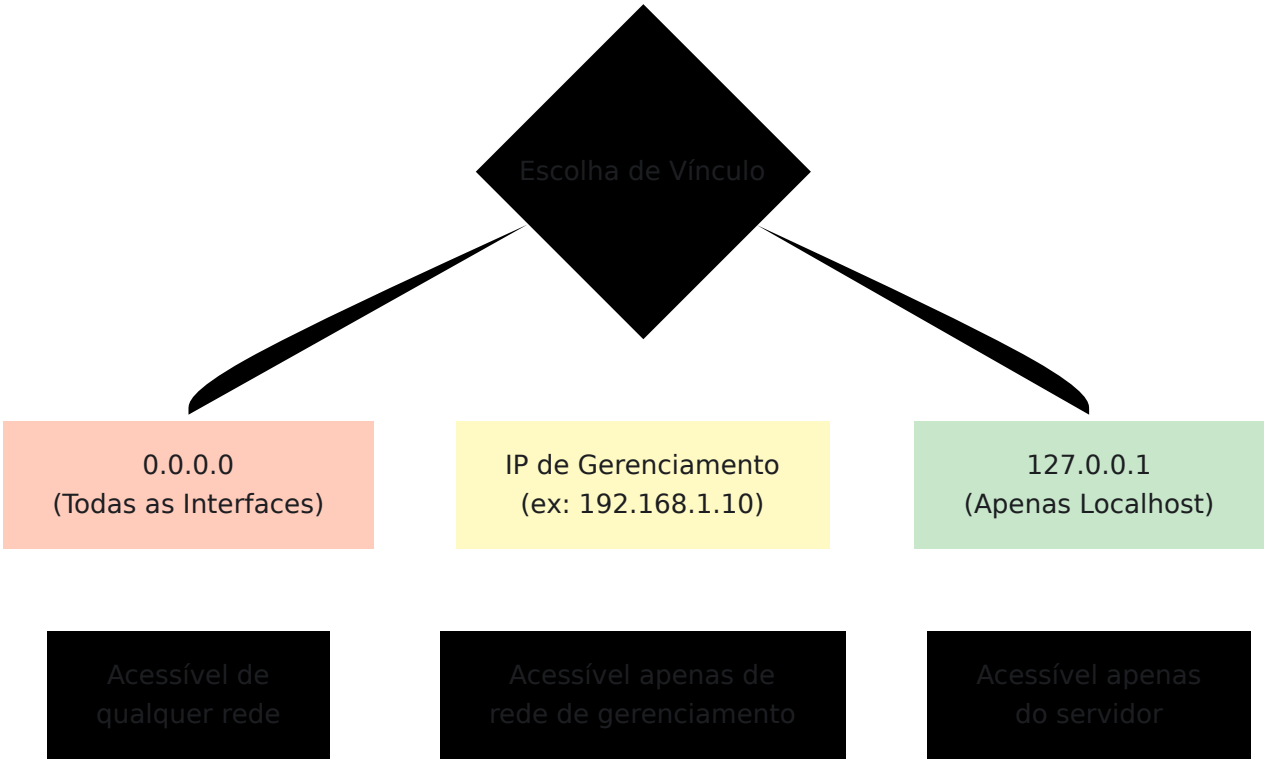
Exemplos:

- AT&T (EUA): MCC=310, MNC=410
- Verizon (EUA): MCC=311, MNC=480
- Vodafone (Reino Unido): MCC=234, MNC=15
- Rede de Teste: MCC=001, MNC=01

Vínculo da Interface de Rede

```
# config/runtime.exs

# Interface Diameter
listen_ip: System.get_env("DIAMETER_LISTEN_IP", "0.0.0.0"),  #
Todas as interfaces
# Ou interface específica:
# listen_ip: "10.7.25.186",

# Interface da API
config :hss, HssWeb.Api.Endpoint,
  http: [
    ip: {0, 0, 0, 0},  # Todas as interfaces
    port: 8443
  ]

# Interface do Painel de Controle
config :hss, HssWeb.ControlPanel.Endpoint,
  http: [
    ip: {0, 0, 0, 0},  # Todas as interfaces
    port: 7443
  ]
```

Opções de Vínculo de Interface:



Configuração do IMS

Configuração de Seleção do S-CSCF

O HSS seleciona um S-CSCF apenas no primeiro REGISTER, quando nenhum S-CSCF está registrado para o assinante. Após um nome ser armazenado (veja Estado de Atribuição do S-CSCF), o HSS retorna o nome armazenado e não re-seleciona. Isso mantém cada REGISTER em um ciclo de desafio IMS-AKA no mesmo S-CSCF, de modo que o ciclo converge.

O nome armazenado é sempre um nó S-CSCF real. O HSS aprende isso do SAR, porque o SAR é a primeira mensagem que informa ao HSS qual nó assumiu o registro. Cada UAR de re-registro e cada LIR de término então retorna esse nó. Uma assinatura IMS compartilhada armazena um desses nós para todos os seus membros. Veja Um S-CSCF Servindo por Assinatura.

```
# config/runtime.exs

config :hss, :ims, %{
  scscf: %{
    # Método de seleção do primeiro REGISTER: :pool_uri ou
:random_peer
    selection_method: :pool_uri,

    # Nome do pool lógico S-CSCF para :pool_uri. Sem porta, para
que o I-CSCF faça um
    # lookup SRV (_sip._tcp/_udp.<nome>) através do pool. O HSS
retorna esse
    # nome apenas em um primeiro REGISTER. Ele nunca o armazena
como um S-CSCF servindo.
    pool_uri: "sip:scscf.ims.mnc001.mcc001.3gppnetwork.org",

    # Quanto tempo um desafio IMS-AKA pode permanecer pendente
antes que o HSS pare
    # de retornar o S-CSCF desafiador em um UAR.
    auth_pending_ttl_seconds: 30,

    # Nós S-CSCF para :random_peer (também o inventário S-CSCF da
implantação)
    peers: [
      %{
        host: "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060"
      },
      %{
        host: "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060"
      }
    ]
  }
}
```

Parâmetros:

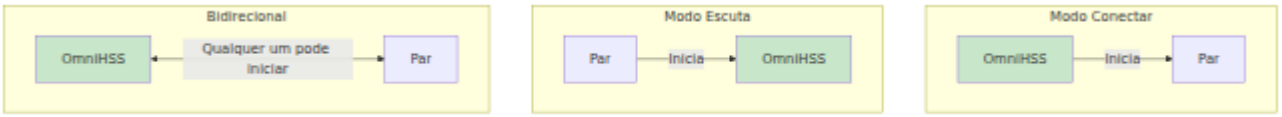
Parâmetro	Tipo	Necessário	Padrão	Des
`selection_method`	Átomo	Sim	`:pool_uri`	Métod seleçã CSCF r primei REGIS `:pool_` `:rand` Veja M de Sel S-CSC
`pool_uri`	String	Sim para `:pool_uri`	-	Nome lógico retorn primei REGIS Não de porta, que o faça u lookup DNS d a carg pool. C nunca armaz esse n como CSCF s de um assina `pool_` estiver definic HSS re

Parâmetro	Tipo	Necessário	Padrão	Des
				um av recua nó de
`auth_pending_ttl_seconds`	Inteiro	Não	30	Quant o HSS um S-(pende autent em se Após e tempo ignora pende usa o registr vez dis Estado Atribui S-CSCI
`peers`	Lista	Sim para `:random_peer`	-	Lista d CSCF. tabela Parâm Par ab

Parâmetros de Par:

Parâmetro	Tipo	Necessário	Padrão	Descrição
`host`	String	Sim	-	URI SIP S-CSCF, com FQDN e porta. O I-CSCF deve ser capaz de resolvê-lo.

Métodos de Seleção do S-CSCF



Métodos de Seleção:

Método	Descrição	Caso de Uso
`:pool_uri`	Retorna o único `pool_uri`. O I-CSCF o resolve por DNS SRV e distribui pelo pool. O DNS distribui a carga e ignora um nó inativo no primeiro REGISTER.	Distribuição de carga mais failover no primeiro REGISTER. Padrão recomendado.
`:random_peer`	Retorna um nó de `peers`, escolhido de forma determinística por assinante (um hash do IMSI sobre a lista de pares ordenados). O mesmo assinante sempre mapeia para o mesmo nó.	S-CSCF fixo por assinante, sem dependência de registros SRV.

O nome `:random_peer` é histórico. A seleção é determinística por assinante, não aleatória. Um mapeamento estável é necessário, para que cada REGISTER em um ciclo de desafio retorne o mesmo S-CSCF.

Se `selection_method` contiver um valor desconhecido, o HSS registra um aviso e recua para o comportamento `:random_peer`. O HSS sempre responde ao UAR.

A seleção se aplica apenas ao primeiro REGISTER. A tabela abaixo mostra o que o UAA retorna em cada estado.

Estado do Assinante	Nome do Servidor UAA	Código de resultado UAA
Nenhum S-CSCF registrado	Uma nova seleção: o `pool_uri`, ou um nó de `peers`	2001 (primeiro registro)
Um desafio IMS-AKA está ativo	O nó S-CSCF desafiador	2002 (registro subsequente)
Registrado, ou não registrado com o nome mantido	O nó S-CSCF servindo	2002 (registro subsequente)
Um membro de uma assinatura IMS compartilhada que outro membro já serve	O nó S-CSCF ancorado da assinatura	2001 ou 2002

Configuração do Realm IMS

Normalmente, o IMS usa um realm separado do EPC:

```
# Realm EPC
"epc.mnc001.mcc001.3gppnetwork.org"

# Realm IMS
"ims.mnc001.mcc001.3gppnetwork.org"
```

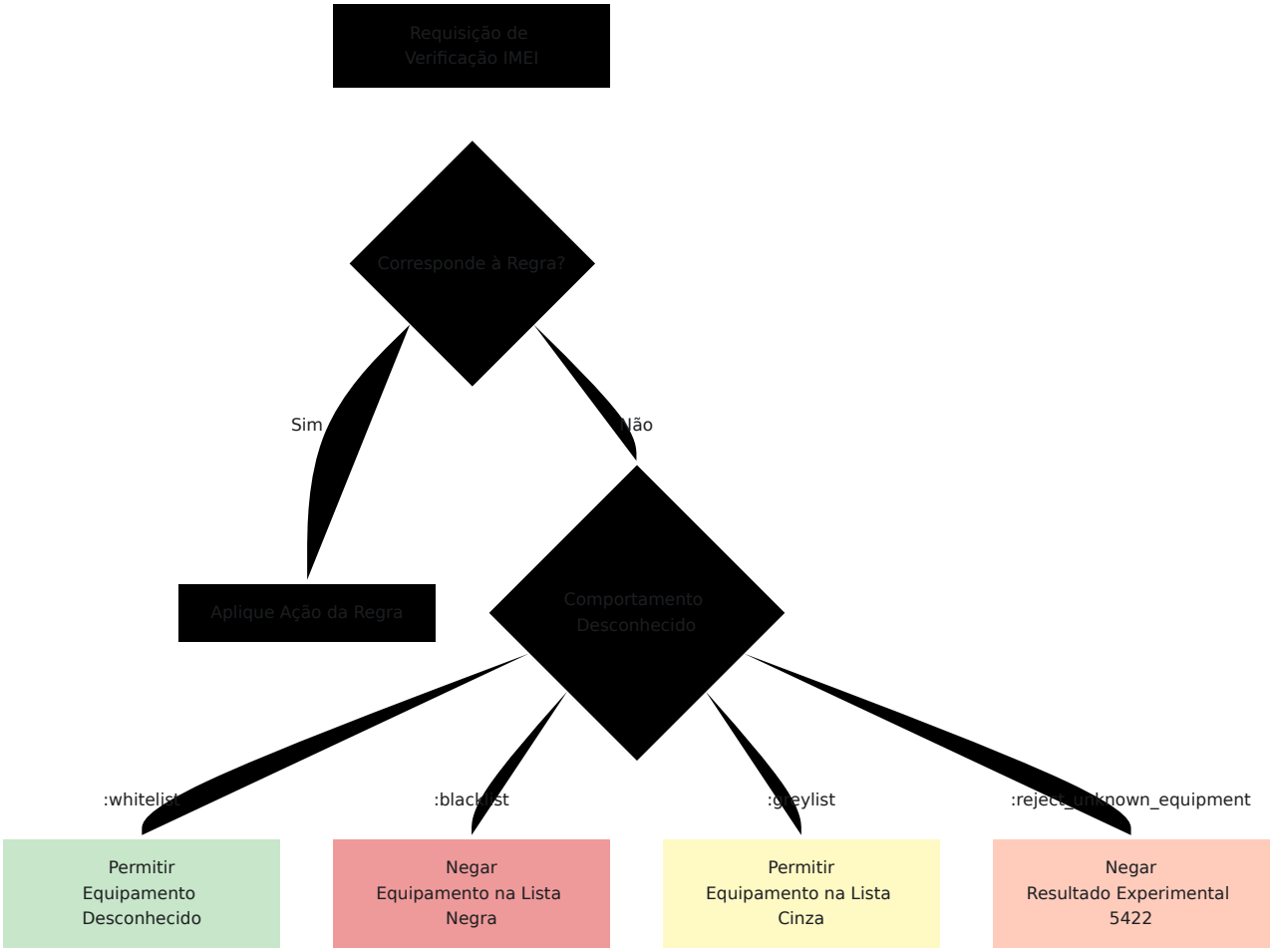
Configuração do EIR

Veja Documentação do EIR para detalhes completos sobre verificação de identidade de equipamentos.

Configurações do Registro de Identidade de Equipamento

```
# config/runtime.exs

config :hss, :eir, %{
  # Comportamento para equipamentos desconhecidos (sem regra
correspondente)
  unknown_equipment_behaviour: :whitelist
  # Opções:
  #   :whitelist - Permitir equipamentos desconhecidos
  #   :blacklist - Bloquear equipamentos desconhecidos
  #   :greylist - Rastrear, mas permitir equipamentos
desconhecidos
  #   :reject_unknown_equipment - Rejeitar com código de resultado
específico
}
```

Comportamento de Equipamento Desconhecido



Opções de Comportamento:

Opção	Resultado	Caso de Uso
`:whitelist`	Permitir todos os IMEI desconhecidos	Rede aberta, testes
`:blacklist`	Bloquear todos os IMEI desconhecidos	Segurança moderada
`:greylist`	Permitir, mas rastrear IMEI desconhecidos	Modo de monitoramento
`:reject_unknown_equipment`	Rejeitar com código específico	Alta segurança

Recomendação: Comece com `:whitelist` durante os testes, passe para `:greylist` para monitoramento em produção e, em seguida, `:blacklist` para segurança rigorosa.

Configuração da API e do Painel de Controle

Configuração do Endpoint da API

```
# config/config.exs

config :hss, HssWeb.Api.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(json)],
  pubsub_server: Hss.PubSub,

  # Configuração HTTPS
  https: [
    port: 8443,
    cipher_suite: :strong,
    certfile: "priv/cert/omnitouch.crt",
    keyfile: "priv/cert/omnitouch.pem"
  ]
```

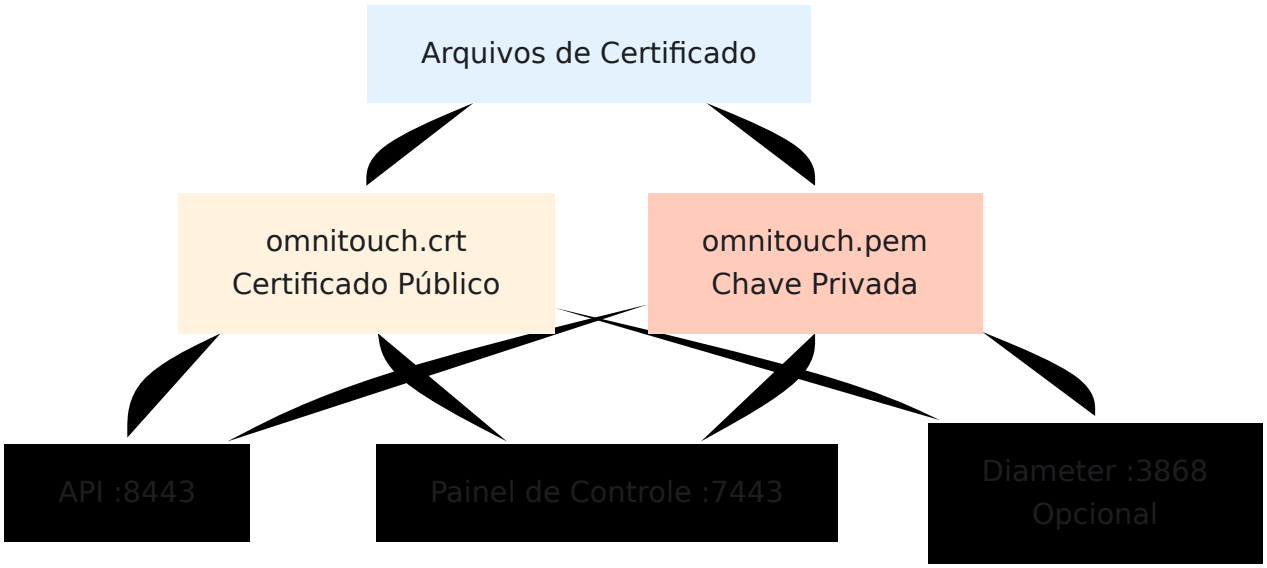
Controle de Acesso da API

O bloco `config :api_ex, :auth` em `runtime.exs` define as chaves e funções da API. Para todos os parâmetros, veja Controle de Acesso da API (RBAC).

Configuração do Painel de Controle

```
# config/config.exs

config :hss, HssWeb.ControlPanel.Endpoint,
  url: [host: "localhost"],
  render_errors: [view: HssWeb.ErrorView, accepts: ~w(html json)],
  pubsub_server: Hss.PubSub,
  live_view: [signing_salt: "some-secret"],

  # Configuração HTTPS
  https: [
    port: 7443,
    cipher_suite: :strong,
    certfile: "priv/cert/omnitouch.crt",
    keyfile: "priv/cert/omnitouch.pem"
  ]
```

Configuração do Certificado TLS



Requisitos do Certificado:

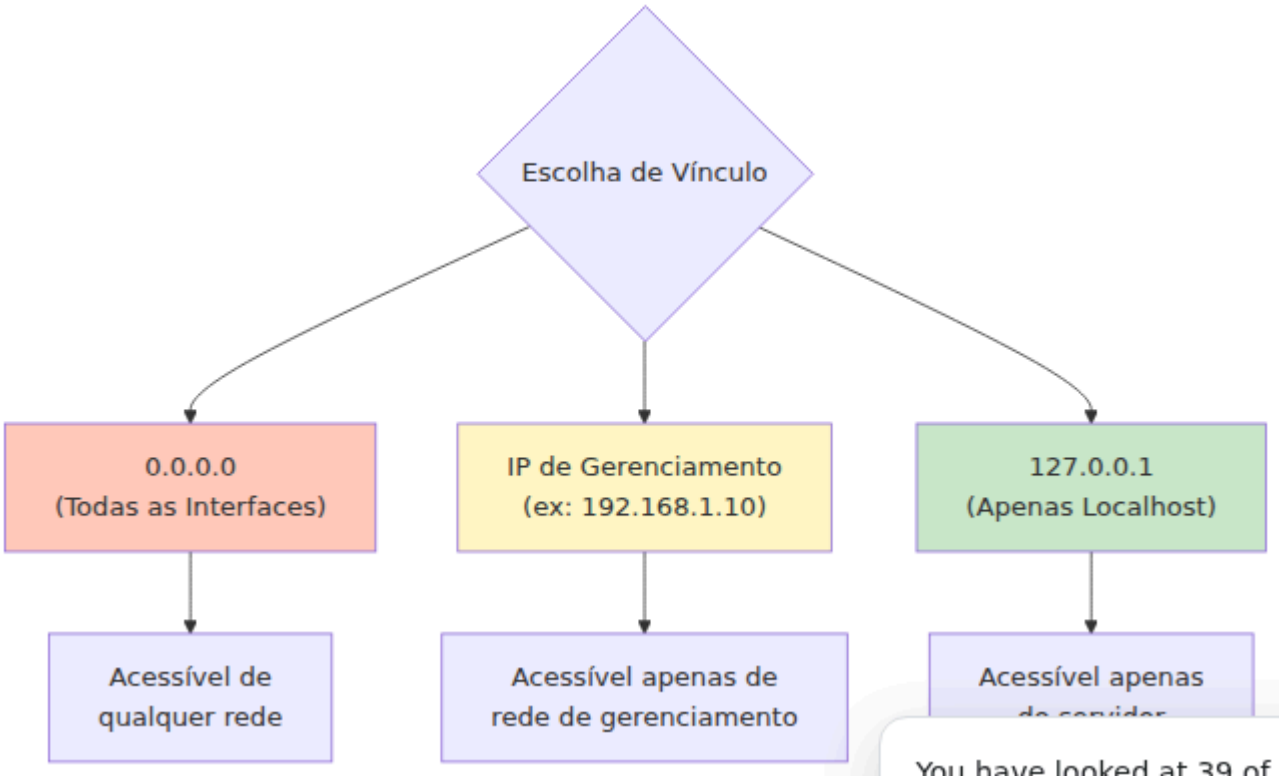
- Certificado X.509 válido
- Chave privada correspondente
- Incluir certificados intermediários, se necessário
- CN ou SAN devem corresponder ao nome do host

Para Produção:

```
https: [
  port: 8443,
  cipher_suite: :strong,
  certfile: System.get_env("TLS_CERT_FILE",
"/etc/ssl/certs/omnihss.crt"),
  keyfile: System.get_env("TLS_KEY_FILE",
"/etc/ssl/private/omnihss.key"),
  cacertfile: System.get_env("TLS_CA_FILE", "/etc/ssl/certs/ca-
bundle.crt")
]
```

Fluxo de Trabalho de Configuração

Configuração Inicial de Implantação



Lista de Verificação de Configuração

Configuração Essencial

- [ ] Conexão com o banco de dados (nome do host, credenciais)
- [ ] PLMN Doméstico (MCC, MNC)
- [ ] Host e domínio Diameter
- [ ] IP e porta de escuta Diameter
- [ ] Certificados TLS para API e Painel de Controle
- [ ] Configuração do cliente de licença (URLs do servidor, licenciado, nome_do_produto)
- [ ] Configurações principais do HSS (send_clr_on_mme_change, stop_diameter_on_database_failure)

Integração de Elementos de Rede

- [ ] Pares Diameter configurados (se usando modo de conexão)
- [ ] Regras de firewall permitem tráfego Diameter (porta 3868)
- [ ] Regras de firewall permitem tráfego HTTPS (portas 7443, 8443)
- [ ] Resolução DNS para identidades Diameter

Configuração IMS (se usando recursos IMS)

- [ ] Lista de pares S-CSCF configurada
- [ ] Método de seleção S-CSCF escolhido
- [ ] Domínio IMS configurado

Configuração Opcional

- [ ] Comportamento do EIR configurado
- [ ] Tamanho do pool de banco de dados ajustado
- [ ] Vínculo da interface de rede restrito

Verificando a Configuração

Após modificar a configuração:

1. **Verificação de Sintaxe:**

```
Verifique os logs para erros de carregamento da configuração
```

2. **Acesso ao Painel de Controle:**

```
Acesse https://[hostname]:7443
Verifique se a página de Visão Geral carrega
```

3. **Acesso à API:**

```
curl -k https://[hostname]:8443/api/status
```

4. **Status do Diameter:**

```
Verifique a página Diameter do Painel de Controle
Verifique as conexões de pares
```

5. **Conectividade com o Banco de Dados:**

```
Verifique o Painel de Controle para dados do assinante
Ou conecte-se diretamente ao Banco de Dados SQL
```

Exemplo Completo de Configuração

em Tempo de Execução

```
# config/runtime.exs - Exemplo completo de produção

import Config

#
==========================================================================
# CONFIGURAÇÃO DO BANCO DE DADOS
#
==========================================================================
config :hss, Hss.Repo,
  username: System.fetch_env!("DATABASE_USERNAME"),
  password: System.fetch_env!("DATABASE_PASSWORD"),
  hostname: System.get_env("DATABASE_HOSTNAME", "db.omnihss.internal'
  database: System.get_env("DATABASE_NAME", "omnihss"),
  port: String.to_integer(System.get_env("DATABASE_PORT", "5432")),
  pool_size: String.to_integer(System.get_env("DATABASE_POOL_SIZE", '
  timeout: 15_000,
  connect_timeout: 15_000,
  ssl: true,
  ssl_opts: [
    cacertfile: "/etc/ssl/certs/postgres-ca.pem",
    verify: :verify_peer
  ]

#
==========================================================================
# CONFIGURAÇÃO DO CLIENTE DE LICENÇA
#
==========================================================================
config :license_client,
  license_server_api_urls: [System.get_env("LICENSE_SERVER_URL",
"https://license.example.com:8443/api")],
  licensee: System.get_env("LICENSE_ORGANIZATION", "Sua Organização")
  product_name: "omnihss"

#
==========================================================================
# CONFIGURAÇÃO DO PLMN DOMÉSTICO E DO NÚCLEO HSS
#
==========================================================================
```

```
config :hss,
  ecto_repos: [Hss.Repo],
  home_plmn: %{
    mcc: System.get_env("HOME_PLMN_MCC", "001"),
    mnc: System.get_env("HOME_PLMN_MNC", "001")
  },
  send_clr_on_mme_change: true,
  stop_diameter_on_database_failure: true,
  license_enforced: true,
  license_module: LicenseClient

#
==========================================================================
# CONFIGURAÇÃO DO DIAMETER
#
==========================================================================
diameter_config = %{
  service_name: :omnitouch_hss,
  listen_ip: System.get_env("DIAMETER_LISTEN_IP", "10.7.25.186"),
  listen_port: String.to_integer(System.get_env("DIAMETER_LISTEN_PORT
"3868")),
  host: System.get_env("DIAMETER_HOST", "omnihss01"),
  realm: System.get_env("DIAMETER_REALM",
"epc.mnc001.mcc001.3gppnetwork.org"),
  product_name: "OmniHSS",
  vendor_id: 10415,
  supported_vendor_ids: [5535, 10415],
  request_timeout: 5000,
  peers: [
    %{
      host: "mme01.epc.mnc001.mcc001.3gppnetwork.org",
      realm: "epc.mnc001.mcc001.3gppnetwork.org",
      ip: "10.7.25.100",
      port: 3868,
      transport: :sctp,
      applications: [:s6a]
    }
  ]
}

config :hss, :diameter, diameter_config

#
==========================================================================
```

```
# CONFIGURAÇÃO IMS
#
============================================================================
config :hss, :ims, %{
  scscf: %{
    selection_method: :random_peer,
    peers: [
      %{host: "sip:scscf01.ims.mnc001.mcc001.3gppnetwork.org:5060"},
      %{host: "sip:scscf02.ims.mnc001.mcc001.3gppnetwork.org:5060"}
    ]
  }
}

#
============================================================================
# CONFIGURAÇÃO DO EIR
#
============================================================================
config :hss, :eir, %{
  unknown_equipment_behaviour: :whitelist
}

#
============================================================================
# CONFIGURAÇÃO DO PONTO DE EXTENSÃO DA API
#
============================================================================
config :hss, HssWeb.Api.Endpoint,
  http: [ip: {0, 0, 0, 0}, port: 8443],
  https: [
    port: 8443,
    cipher_suite: :strong,
    certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss
    keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss
  ],
  url: [host: System.get_env("API_HOST", "api.omnihss.internal"), por
8443]

#
============================================================================
# CONFIGURAÇÃO DO PONTO DE EXTENSÃO DO PAINEL DE CONTROLE
#
============================================================================
config :hss, HssWeb.ControlPanel.Endpoint,
```

```
  http: [ip: {0, 0, 0, 0}, port: 7443],
  https: [
    port: 7443,
    cipher_suite: :strong,
    certfile: System.get_env("TLS_CERT_FILE", "/etc/ssl/certs/omnihss
    keyfile: System.get_env("TLS_KEY_FILE", "/etc/ssl/private/omnihss
  ],
  url: [host: System.get_env("CP_HOST", "hss.omnihss.internal"), port
```

← Voltar ao Guia de Operações | Próximo: Relacionamentos de Entidade →

Guia do Painel de Controle do OmniHSS

← Voltar ao Guia de Operações

Índice

- Visão Geral do Painel de Controle
- Acessando o Painel de Controle
- Página de Visão Geral
- Página de Diâmetro
- Página de Aplicação
- Página de Configuração
- Navegação e Interface

Visão Geral do Painel de Controle

O Painel de Controle do OmniHSS é uma interface de monitoramento baseada na web que fornece visibilidade em tempo real sobre o status do sistema, atividade dos assinantes e conectividade de Diâmetro. Construído com Phoenix LiveView, ele se atualiza automaticamente sem exigir atualizações de página.

Principais Recursos

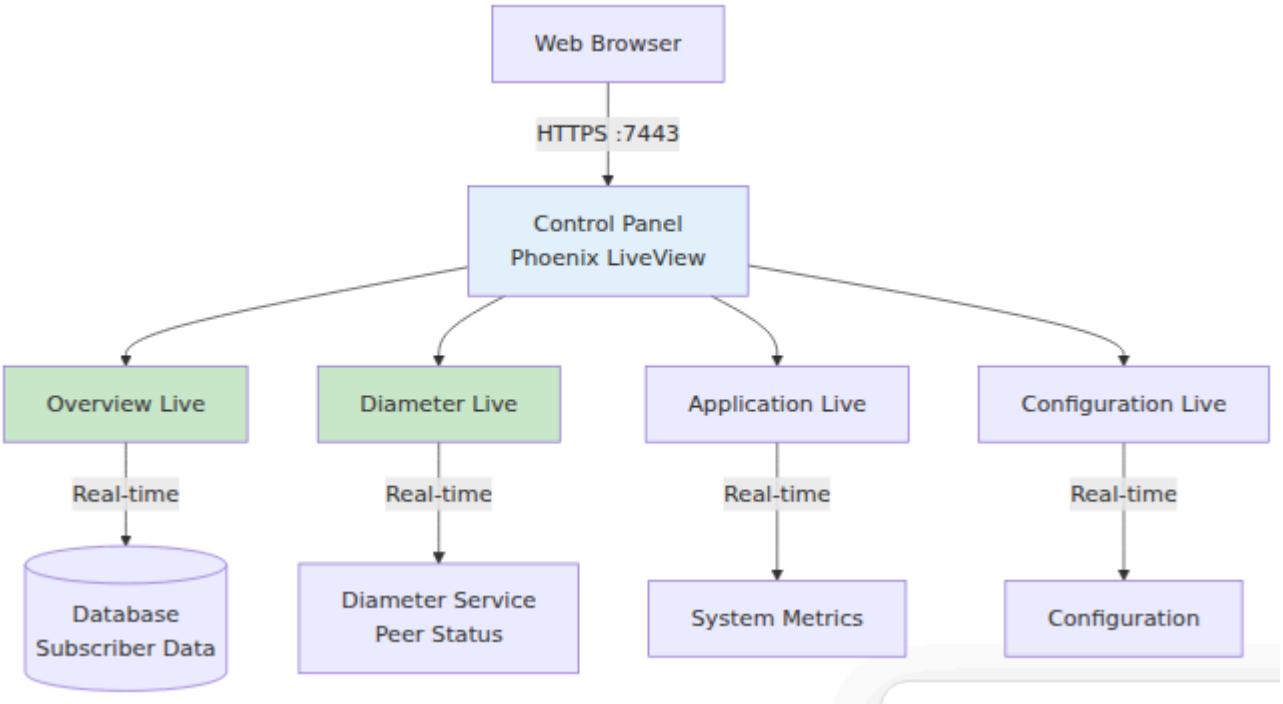
- **Atualizações em Tempo Real** - Atualiza automaticamente a cada segundo
- **Monitoramento de Assinantes** - Visualize assinantes ativos e seu estado atual
- **Status de Diâmetro** - Monitore conexões de pares em tempo real
- **Recursos do Sistema** - Acompanhe o desempenho da aplicação

- **Visualizador de Configuração** - Inspecione a configuração em tempo de execução

Informações de Acesso

```
URL: https://[hostname]:7443
Protocolo: Apenas HTTPS
Porta: 7443 (configurável)
Certificado: Configurado em config/config.exs
```

Arquitetura do Painel de Controle



Acessando o Painel de Controle

Acesso Inicial

1. Abra um navegador web
2. Navegue até `https://[hostname]:7443`
3. Aceite o certificado TLS (se autoassinado)

4. Você será apresentado à página de Visão Geral por padrão

Avisos de Certificado TLS

Se estiver usando certificados autoassinados, os navegadores mostrarão avisos de segurança. Isso é esperado para implantações internas.

Para Produção: Use certificados assinados por uma Autoridade Certificadora confiável.

Requisitos de Rede

- **A porta 7443** deve ser acessível a partir da sua rede de gerenciamento
- **HTTPS** é obrigatório - HTTP não é suportado
- **Regras de firewall** devem permitir tráfego para a porta 7443

Compatibilidade do Navegador

O Painel de Controle utiliza tecnologias web modernas (LiveView, WebSockets):

- Chrome/Chromium (recomendado)
- Firefox
- Safari
- Edge

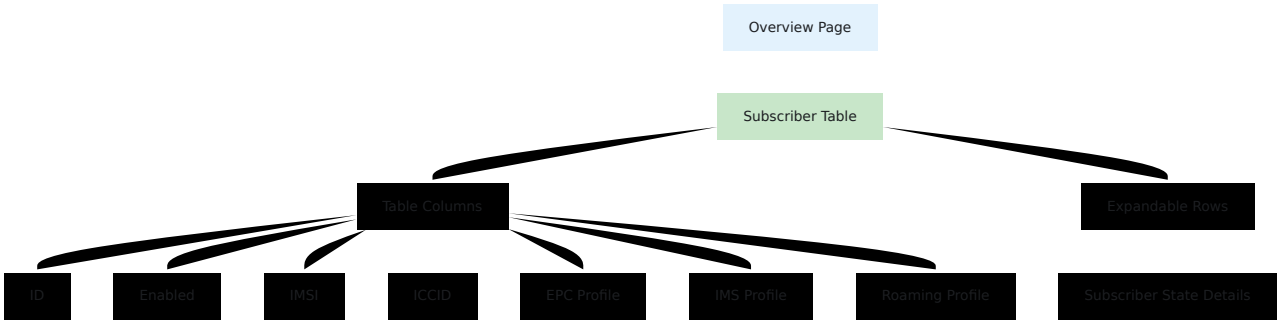
Nota: Internet Explorer não é suportado.

Página de Visão Geral

URL: `https://[hostname]:7443/overview`

A página de Visão Geral exibe todos os assinantes e suas informações de estado em tempo real.

Layout da Página



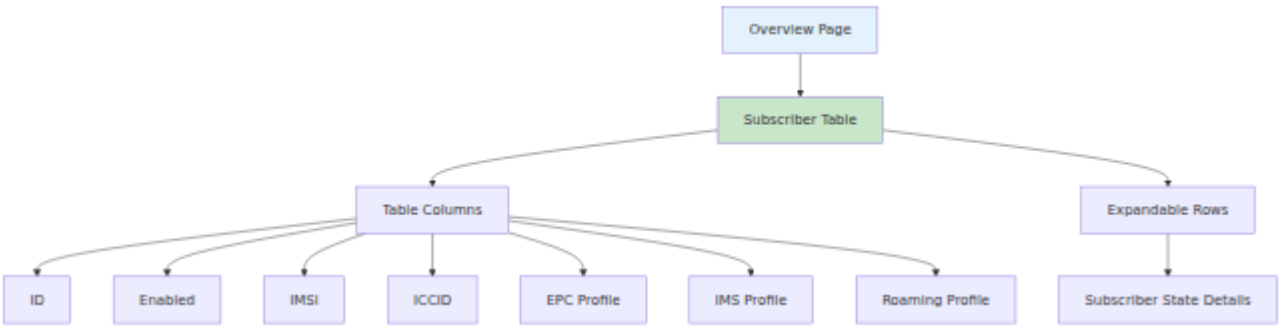
Colunas da Tabela

Coluna	Descriç��o	Valores
ID	ID do banco de dados do assinante	String (UUID)
Enabled	Status do serviço	✓ (habilitado) / ✗ (desabilitado)
IMSI	Identidade Internacional do Assinante Móvel	14-15 dígitos
ICCID	ID do cartão SIM	19-20 dígitos ou "N/A"
EPC Profile	Nome do perfil de serviço de dados	Nome ou ID do perfil
IMS Profile	Nome do perfil de serviço de voz	Nome do perfil, ID ou "N/A"
Roaming Profile	Nome da política de roaming	Nome do perfil, ID ou "N/A"

Detalhes da Linha Expansível

Clique em qualquer linha para expandir e visualizar o estado detalhado do assinante:

Informações de Localização



Campos:

- **MCC** - Código do País Móvel (3 dígitos)
- **MNC** - Código da Rede Móvel (2-3 dígitos)
- **TAC** - Código da Área de Rastreamento
- **Cell ID** - Identificador da célula de serviço
- **eNodeB ID** - Identificador da estação base
- **ECI** - Identificador da Célula E-UTRAN

Informações da Rede

Campos:

- **Último MME Visto** - Nome do host do MME em serviço atual
- **Último Realm Visto** - Realm de Diâmetro do MME
- **Tipo de RAT** - Tecnologia de Acesso Rádio (por exemplo, "E-UTRAN" para LTE)
- **Último Visto Em** - Timestamp da última mensagem de Diâmetro

Informações de IMS

Campos:

- **S-CSCF Atribuído** - URI SIP do S-CSCF atualmente atribuído

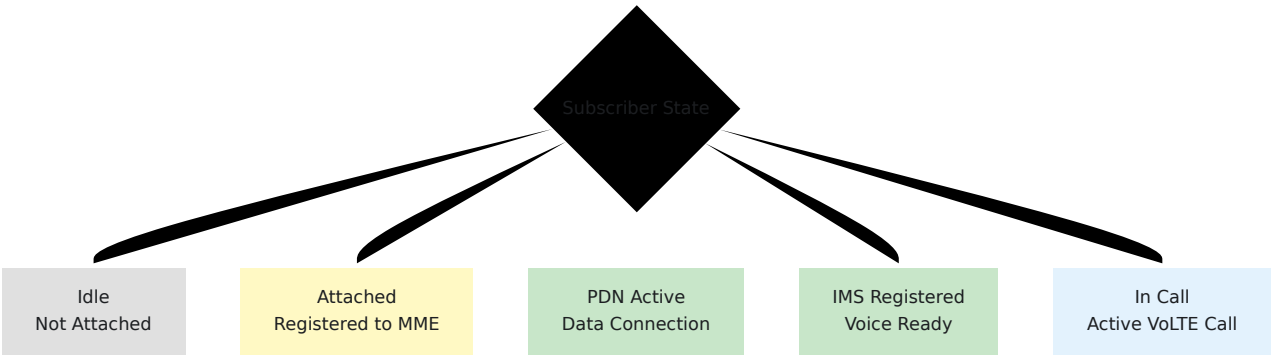
- **Identidade Pública IMS** - URI SIP (por exemplo, sip:+14155551234@ims.example.com)
- **Último P-CSCF Visto** - Último P-CSCF que contatou o HSS
- **Último I-CSCF Visto** - Último I-CSCF que contatou o HSS

Informações da Sessão

Campos:

- **Sessões PDN** - Número de conexões de dados ativas
- **Chamadas Ativas** - Número de chamadas VoLTE ativas

Indicadores de Estado



Como identificar o estado:

- **Idle:** Sem informações de localização, sem MME
- **Attached:** Último MME Visto presente, informações de localização disponíveis
- **PDN Active:** Contagem de sessões PDN > 0
- **IMS Registered:** S-CSCF Atribuído presente
- **In Call:** Contagem de chamadas ativas > 0

Auto-Atualização

A página de Visão Geral atualiza automaticamente **a cada 1 segundo** para mostrar atualizações em tempo real.

Indicadores visuais:

- Novos dados aparecem sem recarregar a página
- Timestamps atualizam em tempo real
- Nenhuma atualização manual necessária

Casos de Uso

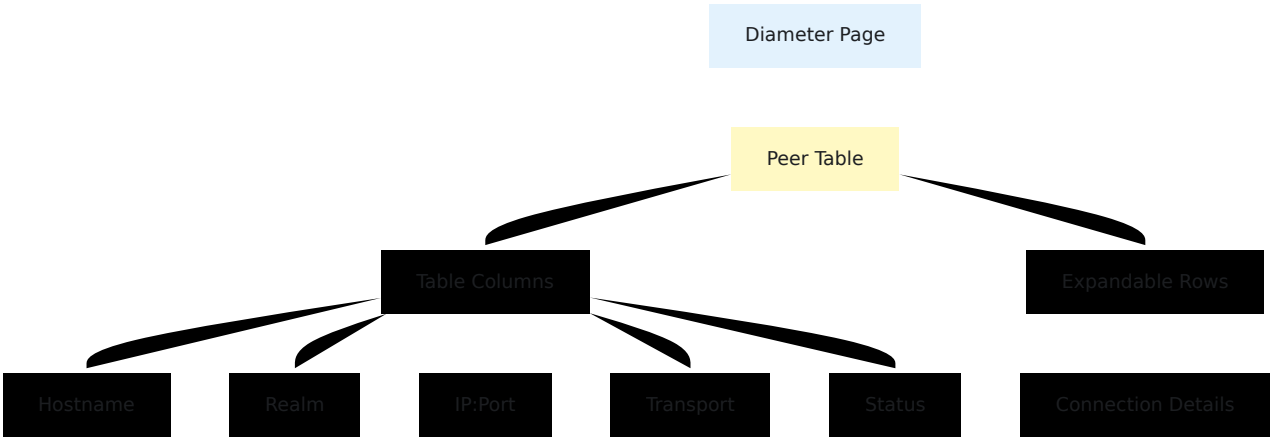
1. **Monitorar Assinantes Ativos**
    - Veja quais assinantes estão atualmente conectados
    - Verifique a rede de serviço atual (para roaming)
    - Verifique o status de registro IMS
2. **Solução de Problemas**
    - Verifique se o assinante está habilitado
    - Verifique o timestamp do último visto (o assinante está responsivo?)
    - Confirme as atribuições de perfil
    - Visualize as informações de localização atuais
3. **Monitoramento de Capacidade**
    - Conte o total de assinantes conectados
    - Monitore as contagens de sessões PDN
    - Acompanhe chamadas VoLTE ativas

Página de Diâmetro

URL: `https://[hostname]:7443/diameter`

A página de Diâmetro mostra o status em tempo real de todas as conexões de pares de Diâmetro.

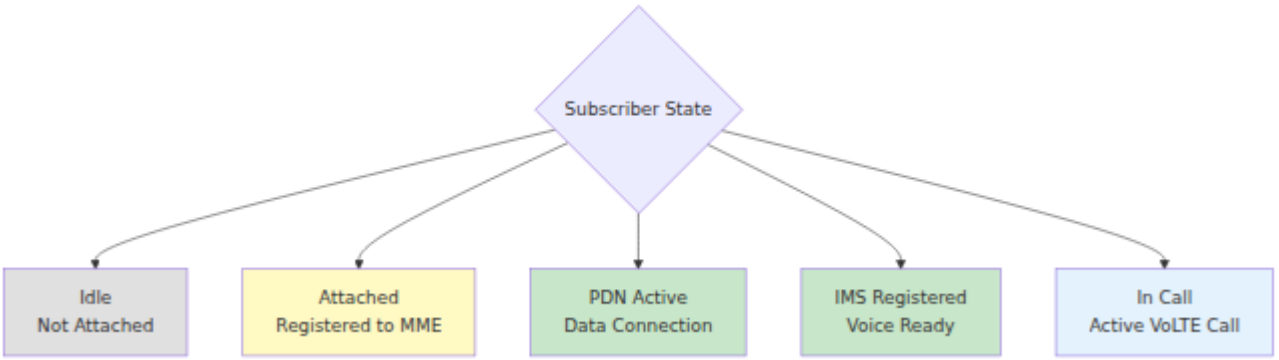
Layout da Página



Colunas da Tabela

Coluna	Descrição	Valores
Hostname	Nome do host do par de Diâmetro	FQDN
Realm	Realm de Diâmetro	Nome de domínio
IP:Port	Endereço de rede	Endereço IP e porta
Transport	Protocolo de transporte	TCP ou SCTP
Status	Status da conexão	Conectado / Desconectado

Status da Conexão



Detalhes da Linha Expansível

Clique em qualquer par para visualizar informações adicionais:

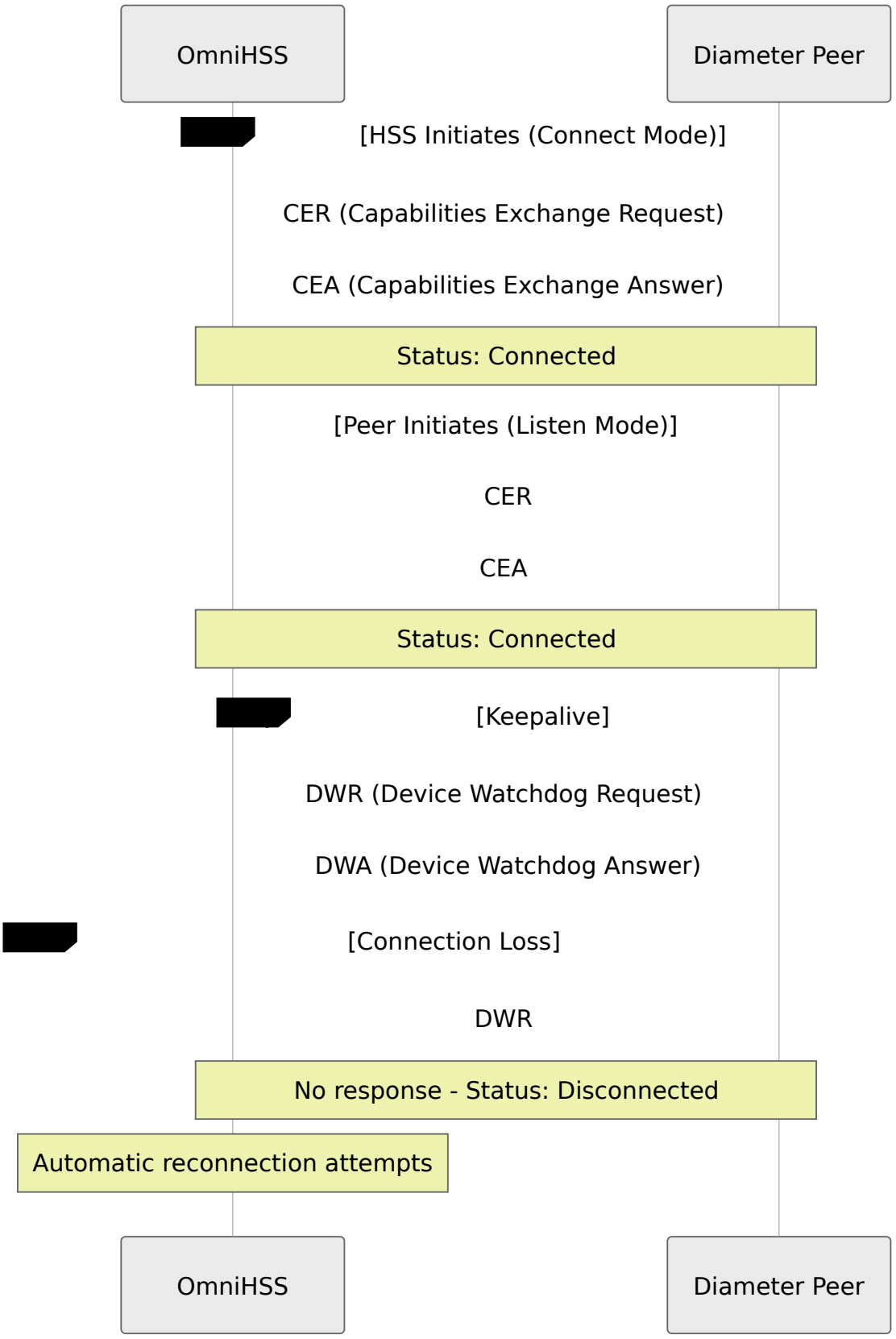
Informações de Conexão:

- **Tipo de Conexão** - Iniciada pelo HSS ou par
- **Nome do Produto** - Identificação do produto do par
- **IDs de Aplicação** - Aplicações de Diâmetro suportadas

Exemplos de ID de Aplicação:

- 16777251 - S6a (MME)
- 16777238 - Gx (P-GW)
- 16777216 - Cx (I-CSCF, S-CSCF)
- 16777217 - Sh (Servidor de Aplicação)
- 16777236 - Rx (P-CSCF)
- 16777252 - S13 (cliente EIR, se externo)

Fluxo de Conexão de Par



Auto-Atualização

A página de Diâmetro atualiza automaticamente **a cada 1 segundo**.

Casos de Uso

1. **Verificar Conectividade**
    - Garantir que todos os pares esperados estão conectados
    - Identificar pares desconectados imediatamente
    - Monitorar conexões instáveis
2. **Solução de Problemas**
    - Verifique se o par é acessível
    - Verifique o protocolo de transporte (TCP vs SCTP)
    - Confirme se os IDs de aplicação correspondem às expectativas
    - Identifique qual lado iniciou a conexão
3. **Planejamento de Capacidade**
    - Conte o total de pares conectados
    - Monitore a estabilidade da conexão
    - Planeje capacidade adicional para pares

Problemas Comuns

Par Mostra Desconectado

Possíveis Causas:

1. Problema de conectividade de rede
2. Par está fora do ar ou reiniciando
3. Firewall bloqueando tráfego
4. Desajuste de configuração de Diâmetro
5. Problema de certificado (se usando TLS)

Passos de Solução de Problemas:

1. Verifique a conectividade de rede: `ping [peer-ip]`
2. Verifique se a porta está acessível: `telnet [peer-ip] 3868`
3. Verifique as regras de firewall
4. Revise os logs do HSS para mensagens de erro
5. Verifique se a configuração de Diâmetro do par corresponde ao HSS

Par Conecta e Desconecta Repetidamente

Possíveis Causas:

1. Instabilidade da rede
2. Desajuste de tempo limite de keepalive
3. Problemas de recursos no par
4. Desajuste de aplicação de Diâmetro

Passos de Solução de Problemas:

1. Verifique a estabilidade da rede
2. Revise os temporizadores de keepalive em ambos os lados
3. Verifique os recursos do sistema do par
4. Verifique se os IDs de aplicação correspondem em ambos os lados

---

Página de Aplicação

URL: `https://[hostname]:7443/application`

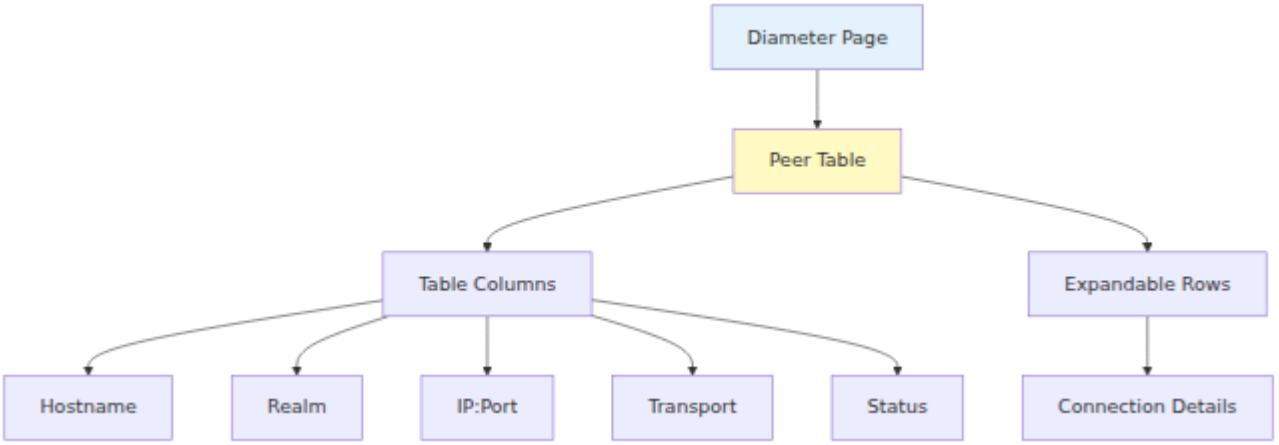
A página de Aplicação fornece informações de monitoramento em nível de sistema e uso de recursos.

Recursos

- **Informações do Processo** - Contagem de processos da VM Erlang e memória
- **Memória do Sistema** - Memória total e usada

- **Tempo de Atividade da Aplicação** - Quanto tempo o OmniHSS está em execução
- **Versão da VM Erlang** - Informações sobre a versão em execução

Métricas Principais



Casos de Uso

1. **Monitoramento de Saúde**
    - Verifique se a aplicação está em execução
    - Verifique se há vazamentos de memória (memória aumentando ao longo do tempo)
    - Monitore o crescimento da contagem de processos
2. **Planejamento de Capacidade**
    - Acompanhe tendências de uso de memória
    - Planeje expansão com base na contagem de processos
    - Verifique se há recursos de sistema adequados
3. **Solução de Problemas**
    - Identifique exaustão de recursos
    - Verifique se é necessário reiniciar
    - Verifique a versão da VM Erlang

Página de Configuração

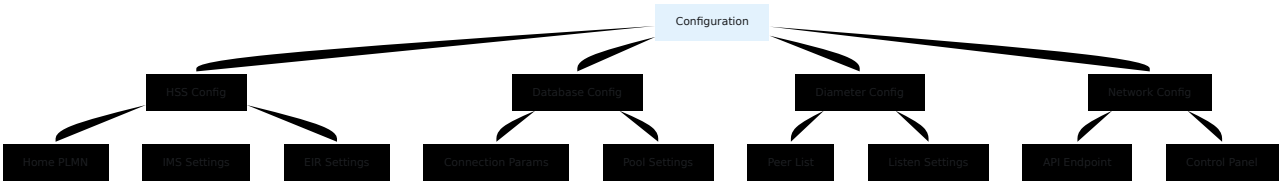
URL: `https://[hostname]:7443/configuration`

A página de Configuração exibe a configuração atual em tempo de execução do OmniHSS.

Recursos

- **Visualizar Configuração** - Inspecione todos os parâmetros de configuração
- **Pesquisar Configuração** - Encontre configurações específicas
- **Variáveis de Ambiente** - Veja os valores resolvidos

Categorias de Configuração



Casos de Uso

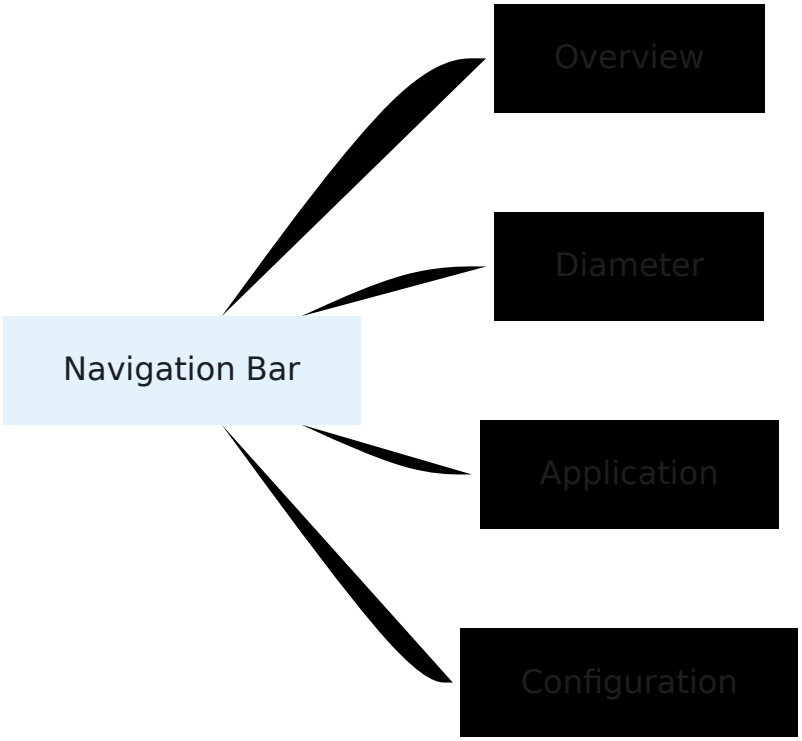
1. **Verificação de Configuração**
    - Verifique se as configurações de runtime.exs estão aplicadas
    - Confirme os parâmetros de conexão do banco de dados
    - Verifique a configuração do par de Diâmetro
2. **Solução de Problemas**
    - Identifique desajustes de configuração
    - Verifique se as variáveis de ambiente estão definidas corretamente
    - Compare a configuração esperada com a real
3. **Documentação**
    - Exporte a configuração atual para documentação

- Compartilhe a configuração com a equipe de suporte

Nota de Segurança: A página de configuração pode exibir informações sensíveis (senhas de banco de dados, chaves). Restrinja o acesso adequadamente.

Navegação e Interface

Barra de Navegação Superior



A navegação está sempre visível na parte superior da página para acesso rápido.

Atalhos de Teclado

O Painel de Controle não implementa atalhos de teclado personalizados. Atalhos padrão do navegador funcionam:

- **Ctrl+R / F5** - Atualização manual da página (embora a autoatualização torne isso desnecessário)

- **Ctrl+F** - Pesquisa na página
- **Ctrl+T** - Abrir nova aba (para várias páginas)

Monitoramento em Múltiplas Abas

Você pode abrir várias páginas do Painel de Controle em abas separadas do navegador para monitoramento simultâneo:

Configuração de Exemplo:

- Aba 1: Página de Visão Geral (monitorar assinantes)
- Aba 2: Página de Diâmetro (monitorar conectividade)
- Aba 3: Página de Aplicação (monitorar recursos)

Todas as abas serão atualizadas automaticamente de forma independente.

Design Responsivo

O Painel de Controle é otimizado para navegadores de desktop. Navegadores móveis são suportados, mas podem exigir rolagem horizontal para tabelas.

Resolução Recomendada: 1920x1080 ou superior para visualização confortável.

Melhores Práticas de Monitoramento

Operações Diárias

1. **Início do Turno**
    - Abra a página de Visão Geral do Painel de Controle
    - Verifique se o número esperado de assinantes está conectado
    - Verifique a página de Diâmetro - todos os pares conectados

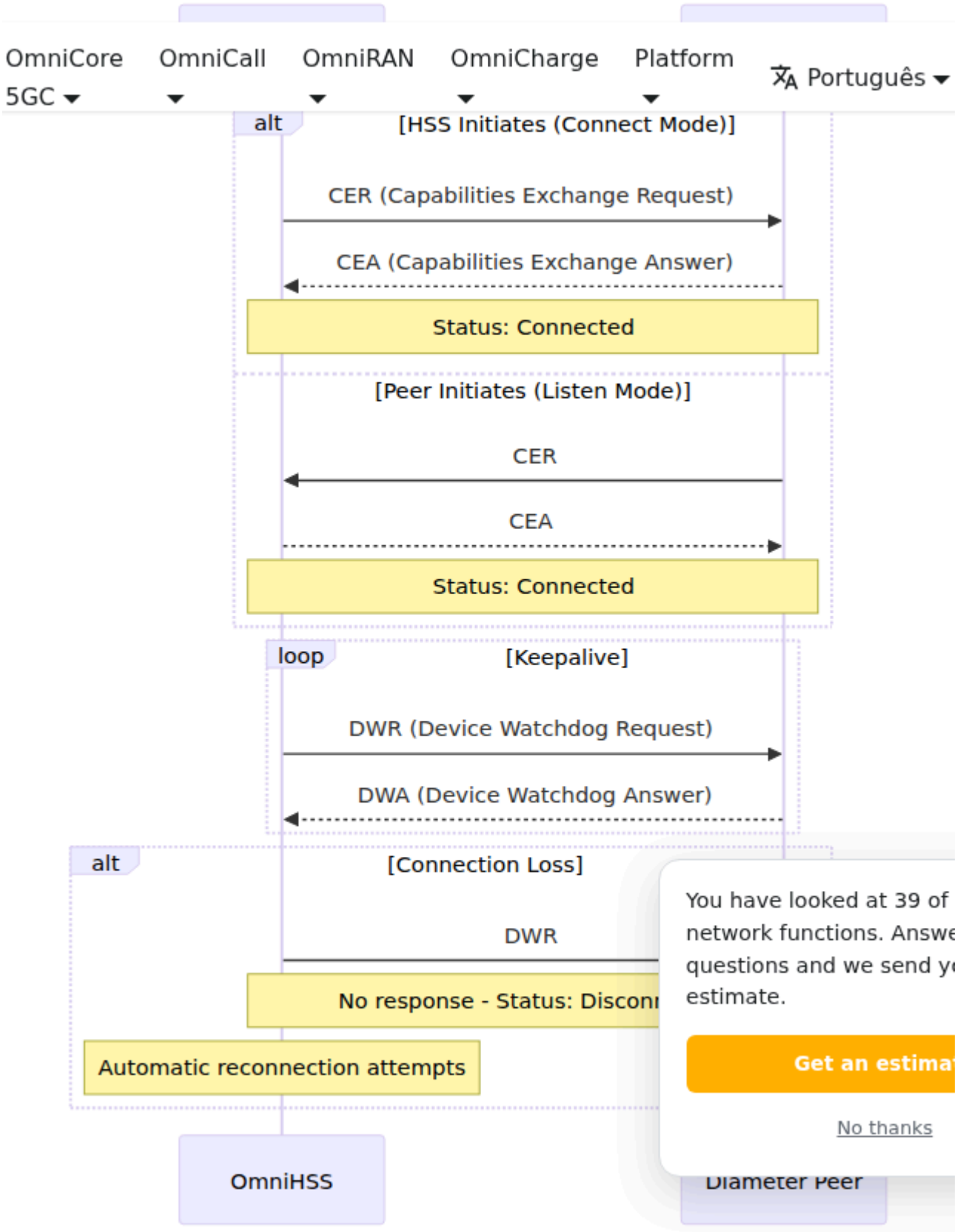
2. **Durante o Turno**

   - Mantenha a página de Visão Geral aberta para monitoramento em tempo real
   - Fique atento a mudanças de estado incomuns
   - Monitore pares desconectados na página de Diâmetro

3. **Fim do Turno**

   - Verifique se o sistema está estável
   - Verifique a página de Aplicação para tendências de uso de recursos
   - Documente quaisquer anomalias

Fluxo de Trabalho de Solução de Problemas



Limites de Alerta

Estabeleça limites de monitoramento para alertas proativos:

Métrica	Aviso	Crítico
Pares de Diâmetro Desconectados	1 par	2+ pares ou par crítico
Uso de Memória	> 80%	> 90%
Falhas de Autenticação de Assinantes	> 5%	> 10%
Contagem de Processos	> 80% do limite	> 95% do limite

← Voltar ao Guia de Operações | Próximo: Métricas & Monitoramento →

Replicação de Banco de Dados

← Voltar ao Guia de Operações

Índice

- Visão Geral
- Como Funciona
- Arquiteturas de Implantação
- Referência de Configuração
- Gerenciamento de Pares
- Monitoramento
- Solução de Problemas

Visão Geral

OmniHSS utiliza **PostgreSQL** com **replicação lógica bidirecional** para implantações de alta disponibilidade em múltiplos locais. Cada local opera como um primário independente de leitura/escrita. Não há um único mestre ou réplicas somente de leitura.

Características Principais

- **Ambos os Locais Leem/Escrevem**: Cada local possui um banco de dados primário completo. Escritas locais são instantâneas, independentemente da qualidade do link entre locais
- **Replicação Assíncrona**: Alterações se replicam através do Write-Ahead Log (WAL) do PostgreSQL. Nenhuma escrita é bloqueada aguardando a confirmação de um local remoto

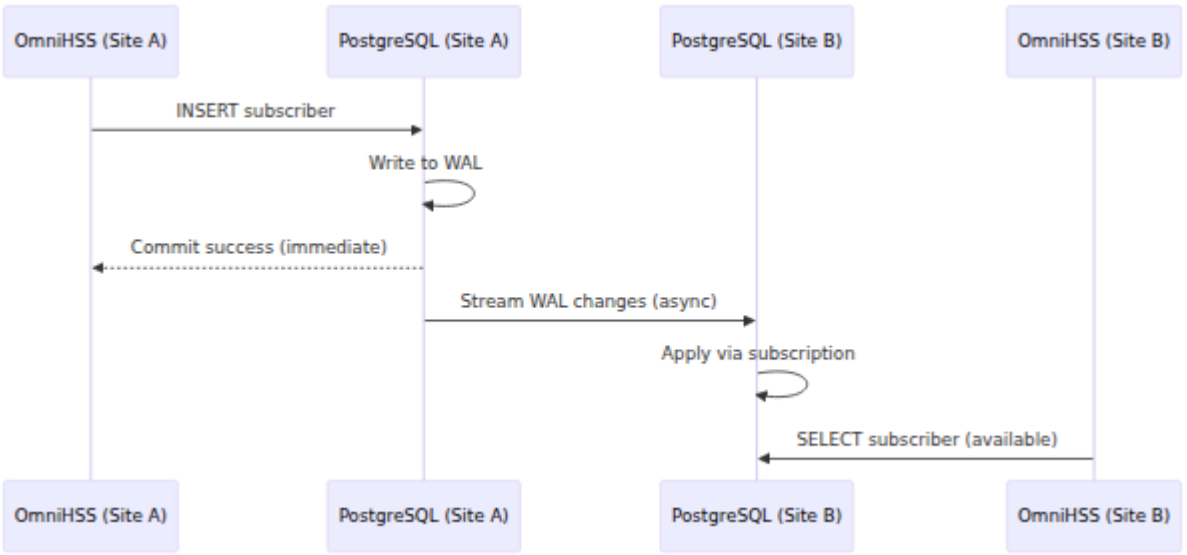
- **Tolerante a Partições**: Se o link entre locais falhar, ambos os locais continuam operando de forma independente. Segmentos de WAL se acumulam localmente e são reproduzidos automaticamente quando o link se recupera
- **Sem Quorum Necessário**: Ao contrário das abordagens de cluster síncronas, não há conceito de quorum. Uma implantação em um único local opera de forma idêntica a uma implantação em múltiplos locais
- **Chaves Primárias UUID**: Todos os registros usam chaves primárias UUIDv4, eliminando colisões de ID entre locais sem coordenação

Como Isso Difere da Replicação Síncrona

	Síncrona	**Replicação Lógica Assíncrona**
Latência de escrita	Bloqueada pelo nó mais lento	Somente local: sub-milissegundos
Falha de link	Local minoritário fica somente leitura	Ambos os locais continuam leitura/escrita
Quorum necessário	Sim (maioria dos nós)	Não
Consistência de dados	Forte (todos os nós idênticos)	Eventual (breve atraso de replicação)
Adequado para links de satélite	Não (latência prejudica o desempenho de escrita)	Sim (WAL enfileira e se recupera)

Como Funciona

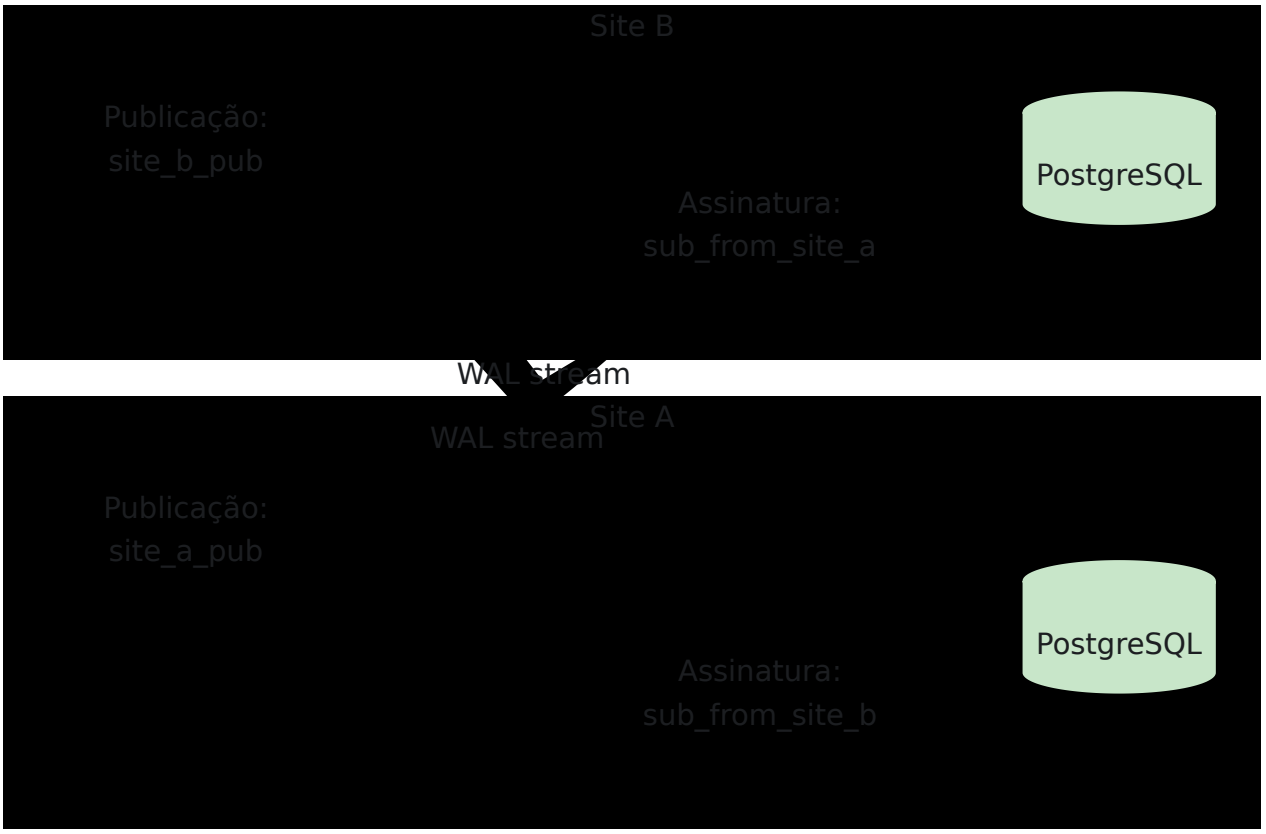
Fluxo de Replicação Lógica



Modelo de Publicação / Assinatura

Cada local possui:

- Uma **publicação** que transmite todas as alterações de tabela (exceto `schema_migrations`)
- Uma **assinatura** para a publicação de cada local par



Enfileiramento de WAL Durante Falha de Link

Quando o link entre locais não está disponível:

1. Escritas locais continuam sem interrupção
2. Segmentos de WAL se acumulam no lado da publicação
3. O slot de replicação rastreia onde cada assinante parou
4. Quando o link se recupera, o WAL enfileirado é reproduzido a partir da última posição confirmada
5. Ambos os locais convergem para um estado idêntico

A retenção de WAL é limitada por `max_slot_wal_keep_size` para evitar crescimento ilimitado do disco durante interrupções prolongadas.

Prevenção de Conflitos

Chaves primárias UUID eliminam a principal fonte de conflitos na replicação bidirecional. Como cada local gera identificadores globalmente únicos de forma independente, operações INSERT nunca colidem.

A opção de assinatura `origin = none` previne loops de replicação. Alterações que chegaram via replicação não são republicadas para outros assinantes.

A tabela `schema_migrations` é excluída das publicações porque ambos os locais executam migrações idênticas de forma independente.

Última Escrita Vence para Estado Dinâmico

Tabelas de estado dinâmico (`subscriber_state`, `pdn_session`, `lte_call`) são atualizadas frequentemente por sinalização Diameter. Cada ULR atualiza o MME em serviço, cada CCR atualiza a sessão do PGW, e cada AAR atualiza a chamada ativa.

Em uma implantação de múltiplos locais, um assinante é atendido por um local de cada vez. Quando o assinante se move entre locais, ambos os locais podem ter atualizações para a mesma linha de `subscriber_state` em momentos diferentes. Sem resolução de conflitos, a ordem de replicação determinaria qual valor sobrevive. Isso poderia sobrescrever o MME em serviço atual com um valor obsoleto.

OmniHSS utiliza **triggers de Última Escrita Vence (LWW)** em tabelas de estado dinâmico:

```
CREATE TRIGGER lww_subscriber_state_trigger
  BEFORE UPDATE ON subscriber_state
  FOR EACH ROW
  EXECUTE FUNCTION lww_subscriber_state();
```

O trigger compara o timestamp `updated_at` da atualização replicada recebida com a linha existente. Se a linha existente for mais nova, a atualização é suprimida silenciosamente. Isso garante:

- Quando um assinante se move do Site A para o Site B, a atualização mais nova do Site B sempre vence
- Atualizações obsoletas do Site A (de antes do movimento do assinante) são descartadas
- Ambos os locais convergem para o mesmo valor: a escrita mais recente

Tabelas com triggers LWW:

Tabela	Atualizado Por	Frequência
`subscriber_state`	ULR, SAR, AIR, PUR, ECR, CCR	Cada atualização de anexo/localização
`pdn_session`	CCR-I, CCR-T	Cada criação/exclusão de sessão PDN
`lte_call`	AAR, STR	Cada configuração/desmontagem de chamada VoLTE
`ims_subscription`	UAR, SAR (S-CSCF compartilhado)	Cada registro de um membro agrupado

Dados de provisionamento estático (assinantes, perfis, chaves, APNs) não precisam de LWW. A API de provisionamento apenas atualiza esses registros, tipicamente de um único plano de gerenciamento.

Convenção de Nomenclatura (Obrigatória para Novas Tabelas)

Os nomes da função e do trigger LWW são um **contrato**, não uma escolha livre. Para uma tabela `foo`, a função deve ser nomeada `lww_foo()` e o trigger deve ser nomeado `lww_foo_trigger`.

O papel Ansible de replicação `omnihss` do OmniCore **não** mantém sua própria lista dessas tabelas. Ele descobre cada trigger cujo nome corresponde a `lww_<table>_trigger` e garante que cada um esteja no modo `ENABLE REPLICA` em toda a malha. Um trigger nomeado de outra forma é invisível a essa reconciliação, portanto a resolução de conflitos entre locais não seria aplicada silenciosamente nessa tabela.

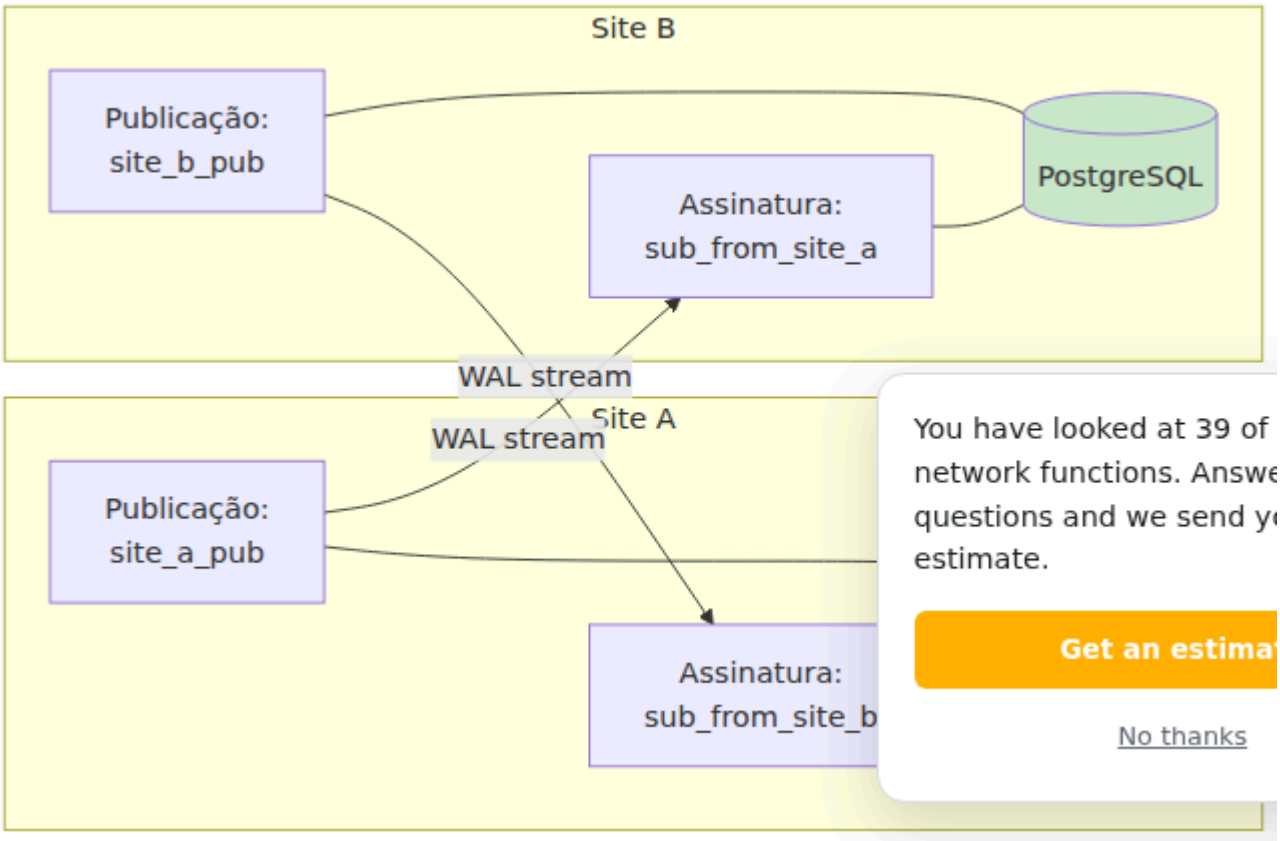
Quando você adicionar uma nova tabela de estado dinâmico (uma tabela escrita a partir do tráfego Diameter ao vivo em mais de um local ao mesmo tempo), siga estes passos:

1. Dê à tabela uma coluna `updated_at` (o macro `timestamps()` do Ecto a adiciona).
2. Na mesma migração, crie `lww_<table>()` e `lww_<table>_trigger` com exatamente esses nomes.
3. Na mesma migração, execute `ALTER TABLE <table> ENABLE REPLICA TRIGGER lww_<table>_trigger`, para que o trigger dispare durante a aplicação da replicação, não em escritas locais.

A tabela `ims_subscription` é o exemplo de referência: sua migração cria `lww_ims_subscription()` e `lww_ims_subscription_trigger`, e então habilita o trigger em modo de réplica. Siga essa mesma estrutura e não renomeie a função ou o trigger.

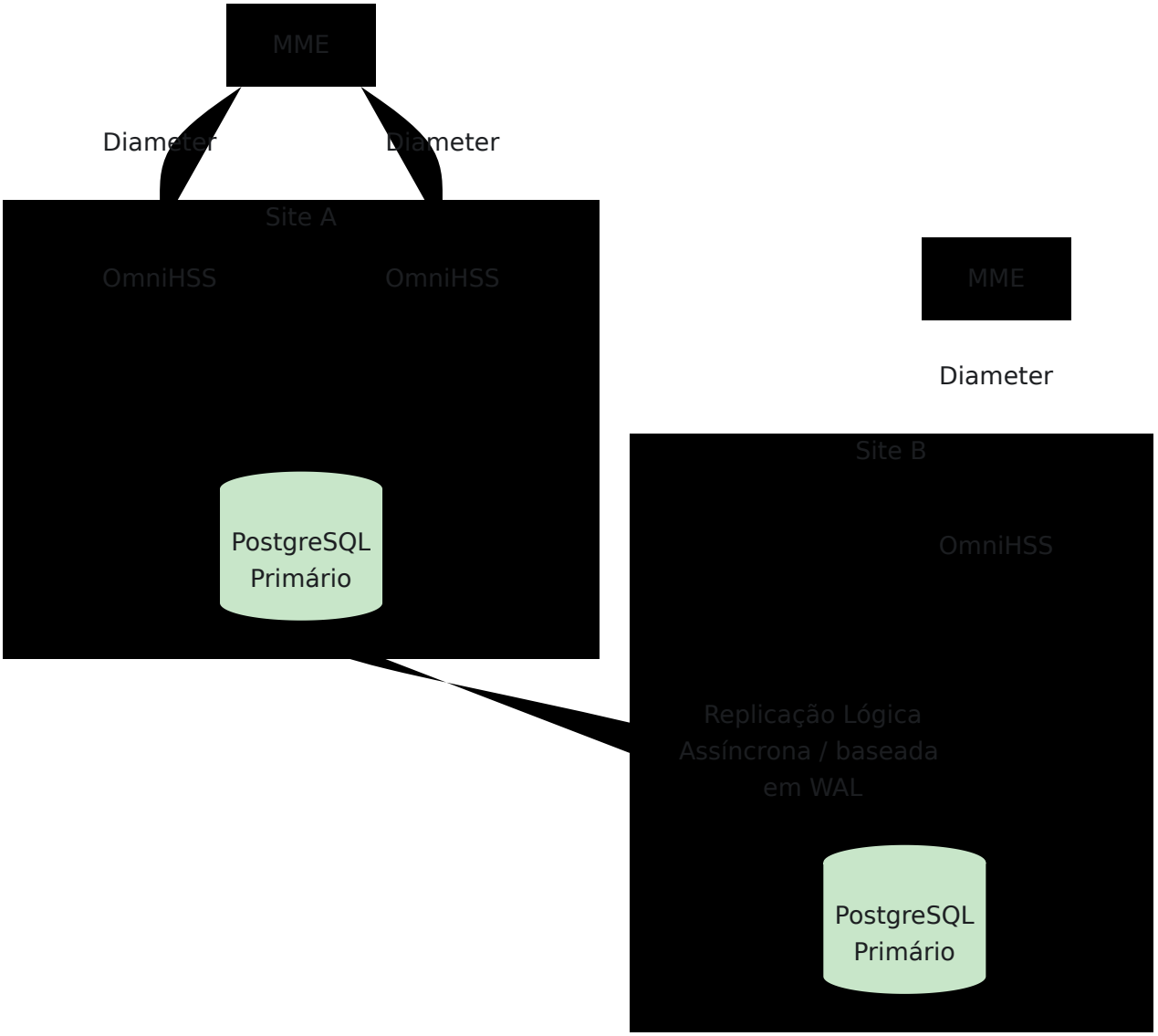
Arquiteturas de Implantação

Site Único (Sem Replicação)



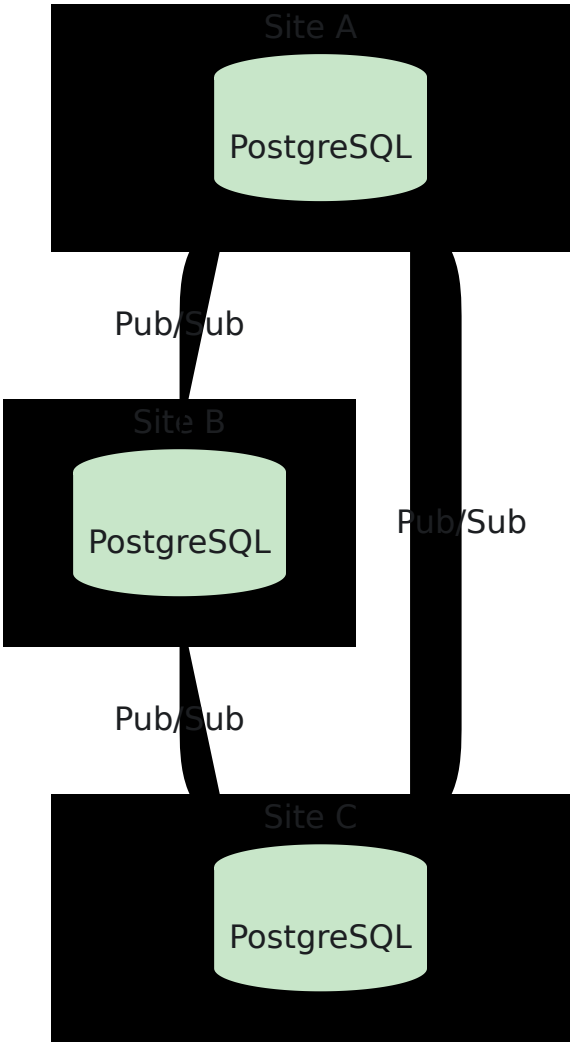
Múltiplas instâncias de aplicativo HSS compartilham um único banco de dados PostgreSQL. A replicação padrão de streaming do PostgreSQL pode ser usada para failover local do banco de dados, se necessário.

Implantação em Dois Locais



Ambos os locais são primários completos. O provisionamento de assinantes em qualquer um dos locais se replica para o outro. Cada MME se conecta ao seu HSS local. Não há dependência de Diameter entre locais.

Implantação em Múltiplos Locais



Cada local se inscreve na publicação de todos os outros locais. Os pares são descobertos automaticamente a partir do inventário do Ansible. Para adicionar um local, adicione-o ao grupo de inventário `hss` e execute o playbook.

Requisitos de Rede

Porta	Protocolo	Propósito
5432	TCP	Conexões de cliente e replicação do PostgreSQL

Referência de Configuração

Variáveis Ansible

A replicação é configurada por grupo HSS em seu inventário:

```yaml
hss:
  hosts:
    site-a-hss01:
      ansible_host: 10.80.12.140
    site-a-hss02:
      ansible_host: 10.80.12.141
  vars:
    omnihss:
      database_type: postgres
      database_host: localhost
      database_port: 5432
      database_name: omnihss
      database_username: omnihss
      database_password: "secure_password"
      replication:
        enabled: true
```

Parâmetros Ansible

Parâmetro	Tipo	Obrigatório	Padrão	Descri
`database_type`	String	Não	`postgres`	Backend d de dados. ser `postgr` para replic
`database_host`	String	Não	`localhost`	Host do PostgreSQ `localhost` quando o Postgres es rodando no mesmo ho o OmniHSS
`database_port`	Inteiro	Não	`5432`	Porta do PostgreSQ
`database_name`	String	Não	`omnihss`	Nome do b de dados.
`database_username`	String	Sim	-	Usuário do PostgreSQ ter privilég `CREATEDB` `REPLICATI`
`database_password`	String	Sim	-	Senha do PostgreSQ
`replication.enabled`	Booleano	Não	`false`	Habilitar replicação bidireciona Quando `tr`

Parâmetro	Tipo	Obrigatório	Padrão	Descri
				papel Ansi configura publicaçõe assinatura entradas `pg_hba.co` automatic

Descoberta de Pares

Os pares são descobertos automaticamente a partir de duas fontes:

1. **Grupo de inventário local**: Todos os outros hosts no grupo Ansible `hss`
2. **Locais remotos**: Entradas em `connected_sites.hss` do `prod_master_peers.yml`

Nenhuma configuração de par por host é necessária. Adicionar um host ao grupo `hss` ou a `connected_sites.hss` e executar o playbook é suficiente.

Pares de Locais Remotos (prod_master_peers.yml)

Para replicação entre locais em diferentes locais geográficos:

```
# group_vars/prod_master_peers.yml
connected_sites:
  hss:
    - remote-site-hss01: 10.80.20.140
    - remote-site-hss02: 10.80.20.141
```

Configuração do PostgreSQL

O papel Ansible configura automaticamente os seguintes parâmetros do PostgreSQL quando a replicação é habilitada:

Parâmetro	Valor	Propósito
`wal_level`	`logical`	Necessário para replicação lógica. Inclui dados de linha completos no WAL.
`max_wal_senders`	`10`	Máximo de processos WAL sender concorrentes (um por assinante).
`max_replication_slots`	`10`	Máximo de slots de replicação. Cada par usa um slot.
`listen_addresses`	`*`	Aceitar conexões de locais pares.

O papel também adiciona entradas `pg_hba.conf` para permitir conexões de replicação de cada IP de par descoberto.

Gerenciamento de Pares

Adicionando um Novo Local

1. Adicione o(s) novo(s) host(s) ao grupo de inventário `hss`, ou adicione a `connected_sites.hss` em `prod_master_peers.yml`
2. Execute o playbook do OmniHSS

O papel irá:

- Instalar PostgreSQL e OmniHSS no novo host
- Executar migrações de banco de dados
- Criar uma publicação para o novo local
- Criar assinaturas para todos os pares existentes
- Atualizar `pg_hba.conf` em todos os pares existentes para permitir o novo local
- Criar assinaturas de pares existentes para o novo local

Removendo um Local

```
ansible-playbook -i hosts site.yml --tags hss-remove-peer \
  -e "peer_name=site_b confirm_remove=yes"
```

O parâmetro `confirm_remove=yes` é necessário como uma medida de segurança. Sem ele, o playbook se recusa a prosseguir e exibe o que aconteceria.

Remover um par:

- Remove a assinatura (para a replicação de entrada daquele par)
- Remove o IP do par de `pg_hba.conf`
- Os dados já replicados do par permanecem no banco de dados local
- A assinatura do local par para este local deve ser removida separadamente no par

Adicionando um Par Ad-Hoc

Para casos em que você precisa adicionar um par fora do fluxo padrão do playbook:

```
ansible-playbook -i hosts site.yml --tags hss-add-peer \
  -e "peer_name=site_b peer_host=10.80.20.140"
```

Monitoramento

Métricas do Prometheus

O serviço **postgres_exporter** roda em cada host HSS na porta `9187`, expondo métricas do PostgreSQL para o Prometheus.

Métricas de Replicação

Métrica	Tipo	Descrição
`pg_replication_slots_active`	Gauge	`1` se o slot de replicação está ativo (par conectado), `0` se inativo
`pg_replication_slots_pg_wal_lsn_diff`	Gauge	Atraso de replicação em bytes. Distância entre a posição atual do WAL e a última posição confirmada do par
`pg_stat_replication_pg_current_wal_lsn_bytes`	Gauge	Posição atual do WAL neste nó

Exemplos de Consultas Prometheus

```
# Slot de replicação ativo (1 = conectado, 0 = desconectado)
pg_replication_slots_active{slot_name=~"sub_from_.*"}

# Atraso de replicação em megabytes
pg_replication_slots_pg_wal_lsn_diff / 1048576

# Alerta: slot de replicação inativo por mais de 60 segundos
pg_replication_slots_active == 0
```

Alertas Recomendados

Alerta	Condição	Severidade	De
Slot de Replicação Inativo	`pg_replication_slots_active == 0` por 60s	Crítico	Par e desc WAL acur
Atraso de WAL Excessivo	`pg_replication_slots_pg_wal_lsn_diff > 104857600` por 5m	Aviso	Atra repli exce 100M pode lento degr
Slot de WAL Perdido	O status do WAL é `lost`	Crítico	Slot inva par r novo com

Validação Ansible

O playbook do OmniHSS valida a saúde da replicação ao final de cada execução. Ele afirma:

1. Todos os trabalhadores de assinatura estão vivos
2. Todos os slots de replicação estão ativos
3. O atraso de WAL está dentro do limite (100MB padrão)
4. O status do slot de WAL é `reserved` ou `extended` (não `lost`)

Se qualquer afirmação falhar, o playbook falha com uma mensagem de erro clara identificando o problema.

Validação Manual

```
# Verifique se os trabalhadores de assinatura estão vivos
sudo -u postgres psql -d omnihss -c \
  "SELECT subname, pid IS NOT NULL as worker_alive FROM
pg_stat_subscription;"

# Verifique os slots de replicação e o atraso
sudo -u postgres psql -d omnihss -c \
  "SELECT slot_name, active,
    pg_wal_lsn_diff(pg_current_wal_lsn(), confirmed_flush_lsn) as
lag_bytes,
    wal_status
  FROM pg_replication_slots WHERE slot_type = 'logical';"

# Compare a contagem de assinantes entre os locais
sudo -u postgres psql -d omnihss -c "SELECT count(*) FROM
subscriber;"
```

Solução de Problemas

Trabalhador de Assinatura Morto

Sintomas: `pg_stat_subscription` mostra `pid IS NULL` para uma assinatura. A validação do Ansible falha com "Trabalhadores mortos."

Causas Comuns:

- A tabela existe no publicador, mas não no assinante (incompatibilidade DDL)
- Conflito de chave duplicada de um registro inserido manualmente
- Par inacessível por um período prolongado causando invalidação do slot de WAL

Resolução:

1. Verifique os logs do PostgreSQL para o erro específico:

```
journalctl -u postgresql | grep "logical replication" | tail
-20
```

2. Se incompatibilidade DDL: aplique a migração ausente neste local, então o trabalhador será reiniciado automaticamente
3. Se chave duplicada: resolva o conflito manualmente, então reabilite a assinatura:

```
ALTER SUBSCRIPTION sub_from_<peer> ENABLE;
```

Slot de Replicação Inativo

Sintomas: `pg_replication_slots` mostra `active = false`. Alerta do Prometheus dispara.

Causas Comuns:

- Problema de conectividade de rede entre locais
- PostgreSQL no local par está parado
- Firewall bloqueando a porta 5432 do IP do par

Resolução:

1. Verifique a conectividade de rede com o par:

```
pg_isready -h <peer_ip> -U <db_user> -d omnihss
```

2. Verifique se o PostgreSQL do local par está em execução
3. Verifique se as regras do firewall permitem a porta 5432 do IP deste local

O trabalhador de assinatura se reconectará automaticamente quando o par se tornar acessível.

Slot de WAL Perdido

Sintomas: `pg_replication_slots` mostra `wal_status = 'lost'`. A validação do Ansible falha com "WAL SLOT CRÍTICO."

Causa: O par foi desconectado por tanto tempo que o WAL retido excedeu `max_slot_wal_keep_size` e foi limpo.

Resolução: O par requer um novo seed completo:

1. Remova a assinatura quebrada neste local:

```
DROP SUBSCRIPTION sub_from_<peer>;
```

2. Remova o slot quebrado no local par:

```
SELECT pg_drop_replication_slot('sub_from_<this_site>');
```

3. Reexecute o playbook do OmniHSS. Ele recria a assinatura com `copy_data = true` para realizar uma sincronização inicial.

Atraso Excessivo de WAL

Sintomas: `pg_replication_slots_pg_wal_lsn_diff` é grande e crescente. Alerta do Prometheus dispara.

Causas Comuns:

- Link entre locais de alta latência ou congestionado
- Local par sob carga pesada (o trabalhador de aplicação não consegue acompanhar)
- Grande operação em massa (por exemplo, provisionamento em massa) gerando volume significativo de WAL

Resolução:

1. Verifique se o atraso está crescendo ou estável:

```
SELECT slot_name,
  pg_wal_lsn_diff(pg_current_wal_lsn(), confirmed_flush_lsn) as
lag_bytes
FROM pg_replication_slots WHERE slot_type = 'logical';
```

2. Se estável: o par está aplicando mudanças, mas está atrasado. Ele irá alcançar.
3. Se crescente: investigue o link de rede ou a carga do local par.
4. Para operações em massa: o atraso durante a operação é esperado e será resolvido após a conclusão da operação.

Coordenação de Migração de Esquema

Importante: Alterações no esquema do banco de dados (migrações) não se replicam via replicação lógica. Ao implantar uma nova versão do OmniHSS que inclui alterações de esquema:

1. Aplique a migração em **todos os locais** antes de implantar o código do aplicativo que depende de novas colunas/tabelas
2. O playbook Ansible lida com isso automaticamente. Ele executa migrações em cada host
3. Se implantando manualmente, execute migrações em todos os locais primeiro:

```
/opt/omnihss/bin/hss rpc "Hss.Command.Database.migrate()"
```

Relações de Entidades do OmniHSS

← Voltar para o Guia de Operações

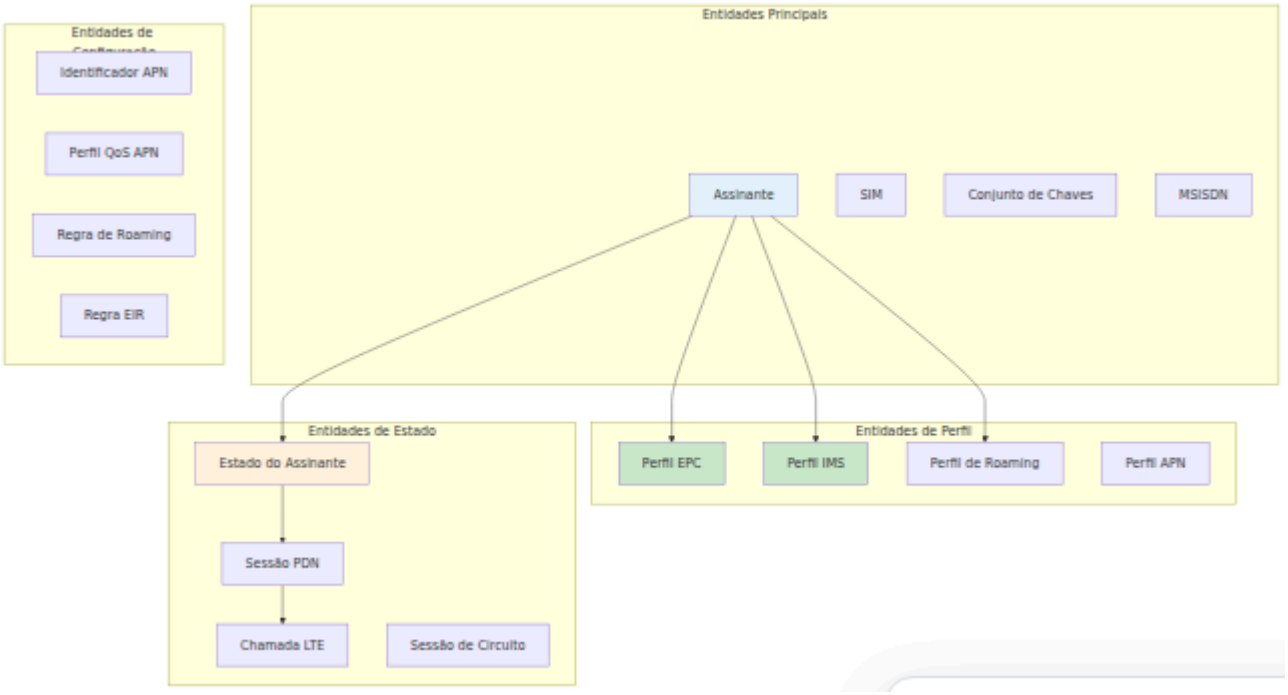
Índice

- Visão Geral da Entidade
- Entidades Principais
- Entidades de Perfil
- Entidades de Estado
- Diagramas de Relação de Entidades
- Ciclo de Vida da Entidade
- Padrões de Fluxo de Dados

Visão Geral da Entidade

O OmniHSS organiza os dados dos assinantes em entidades lógicas com relações claras. Compreender essas entidades é crucial para tarefas operacionais como provisionamento, solução de problemas e planejamento de capacidade.

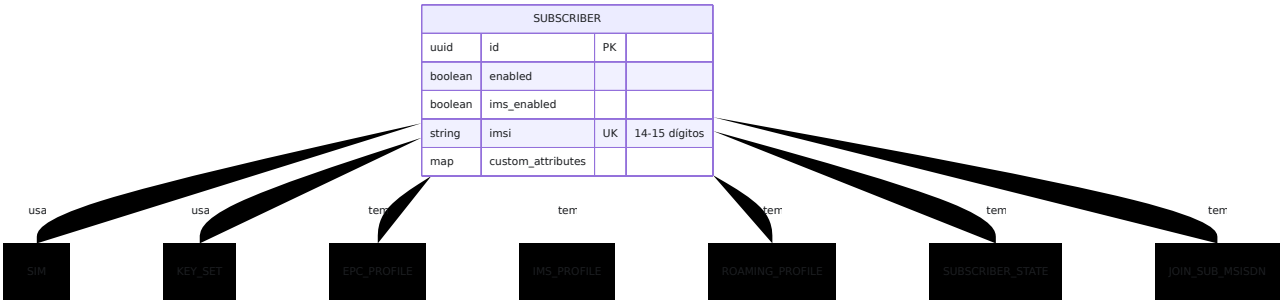
Categorias de Entidade



Entidades Principais

Assinante

O **Assinante** é a entidade central que representa um usuário móvel.



Campos:

Campo	Tipo	Descrição	Restrições
`id`	uuid	Chave primária	UUID, gerado (`gen_random_uuid()`)
`enabled`	boolean	Sinalizador de serviço habilitado	Padrão: true
`ims_enabled`	boolean	Serviços IMS habilitados	Padrão: true
`imsi`	string	Identidade Internacional do Assinante Móvel	14-15 dígitos, único
`custom_attributes`	map	Dados personalizados em chave-valor	Opcional
`sim_id`	uuid	Chave estrangeira para SIM	Necessário para autenticação (pode ser nulo enquanto inativo)
`key_set_id`	uuid	Chave estrangeira para Conjunto de Chaves	Necessário
`epc_profile_id`	uuid	Chave estrangeira para Perfil EPC	Necessário para serviço de dados (pode ser nulo enquanto inativo)

Campo	Tipo	Descrição	Restrições
`ims_profile_id`	uuid	Chave estrangeira para Perfil IMS	Necessário para IMS/vocais (pode ser nulo enquanto inativo)
`roaming_profile_id`	uuid	Chave estrangeira para Perfil de Roaming	Opcional; se nulo, o assinante não pode fazer roaming (apenas rede doméstica)
`subscriber_state_id`	uuid	Chave estrangeira para Estado do Assinante	Criado automaticamente
`ims_subscription_id`	uuid	Chave estrangeira para Assinatura IMS	Opcional

A coluna "Restrições" acima indica para o que cada objeto é necessário **para um serviço**, não o que o banco de dados precisa para inserir uma linha. O banco de dados aceita um assinante que tem apenas um IMSI e um Conjunto de Chaves. Tal assinante está *inativo*: ele existe, mas nenhum serviço funciona até que você atribua o objeto correspondente. Use isso para pré-provisionar assinantes e, em seguida, ativá-los mais tarde. Veja Pré-Provisionamento de Assinante Inativo.

O que cada objeto habilita:

Para fazer isso funcionar	O assinante precisa
Autenticação (anexar, autenticação IMS)	`key_set` e `sim`
Serviço de dados LTE (anexar S6a)	`epc_profile`
IMS / voz (registro Cx)	`ims_profile`, e pelo menos um MSISDN
Roaming em uma rede visitada	um `roaming_profile` com uma regra correspondente

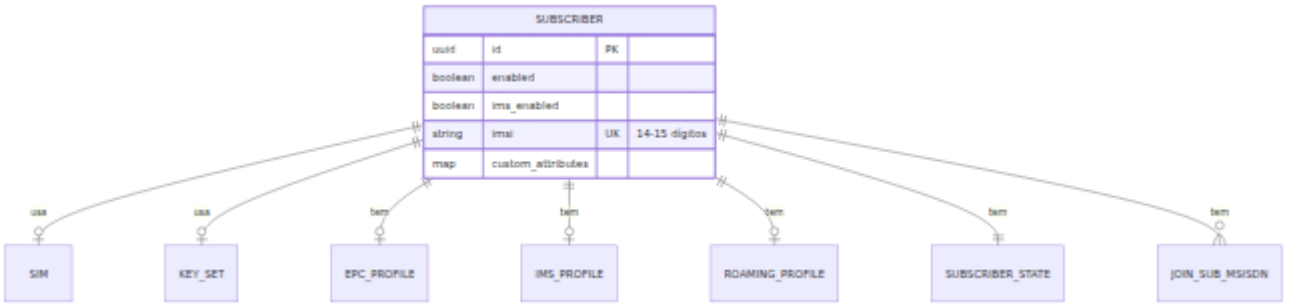
Se um objeto necessário estiver faltando quando o assinante aparecer na rede, o HSS rejeita a solicitação e registra o motivo exato. Veja Comportamento do Assinante Inativo na Rede.

Pontos-Chave:

- Cada assinante deve ter exatamente um IMSI.
- O IMSI deve ter 14-15 dígitos (sem letras ou caracteres especiais).
- Um assinante pode ter vários MSISDNs (números de telefone).
- O estado do assinante é criado automaticamente quando o assinante é criado.
- O sinalizador `enabled` controla todos os serviços (dados e IMS).
- O sinalizador `ims_enabled` controla apenas os serviços IMS.
- O banco de dados precisa apenas do IMSI e do Conjunto de Chaves para inserir um assinante. Todos os outros objetos são necessários para seu próprio serviço, conforme listado acima. Um assinante sem perfil IMS não pode usar IMS, mesmo que o registro seja salvo sem um.
- Um assinante sem `ims_subscription_id` é autônomo. Defina `ims_subscription_id` para agrupar vários assinantes (IMSIs) em uma única assinatura IMS que compartilha uma identidade pública e um único S-CSCF. Veja Assinatura IMS e Identidade Pública de Usuário Compartilhada.

SIM

A entidade **SIM** representa um cartão SIM físico ou embutido.



Campos:

Campo	Tipo	Descrição	Nível de Segurança
`iccid`	string	Identificador de Cartão de Circuito Integrado	Público
`sim_vendor`	string	Fabricante do SIM	Público
`batch_name`	string	Lote de fabricação	Público
`is_esim`	boolean	Sinalizador de SIM embutido	Público
`pin1`, `pin2`	string	Códigos PIN	Sensível
`puk1`, `puk2`	string	Códigos PUK	Sensível
`adm1` - `adm10`	string	Códigos administrativos	Altamente Sensíveis
`kic`, `kid`	binary	Chaves de segurança OTA	Altamente Sensíveis

Pontos-Chave:

- O ICCID identifica exclusivamente o cartão SIM

- Um SIM pode ser atribuído a um assinante por vez
- Códigos PIN/PUK são para bloqueio de SIM pelo usuário final
- Códigos ADM são para operações administrativas de SIM
- KIC/KID são para atualizações OTA (Over-The-Air) do SIM

Conjunto de Chaves

O **Conjunto de Chaves** contém chaves criptográficas para autenticação.

KEY_SET			
uuid	id	PK	
binary	ki		128-bit
binary	opc		128-bit
binary	op		128-bit
binary	amf		16-bit
bigint	sqn		48-bit sequence
string	authentication_algorithm		

usado pc



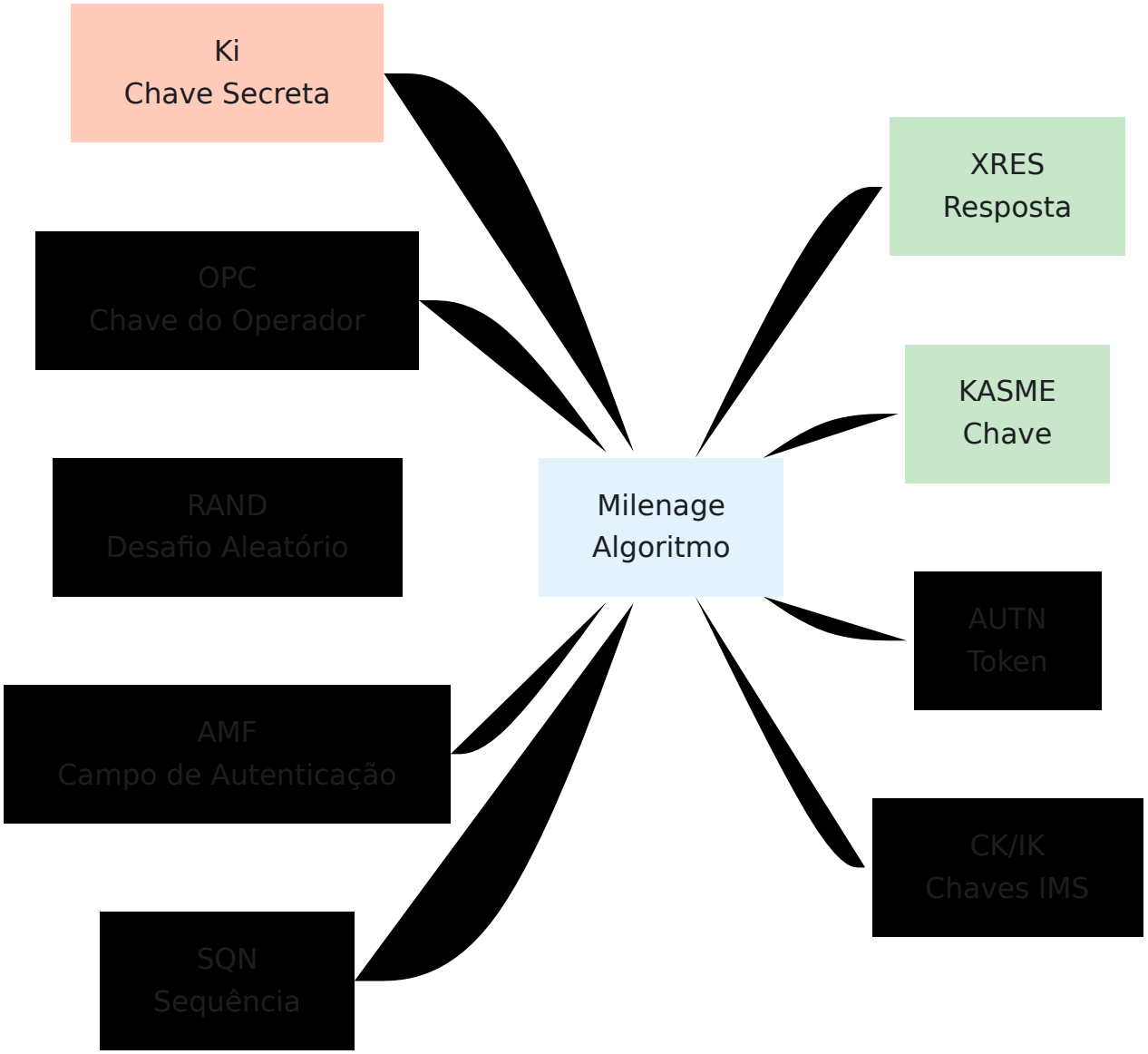
Campos:

Campo	Tipo	Descrição	Tamanho
`ki`	binary	Chave secreta	128 bits (16 bytes)
`opc`	binary	Chave variante do operador (derivada)	128 bits
`op`	binary	Chave do operador (para derivar OPC)	128 bits
`amf`	binary	Campo de Gerenciamento de Autenticação	16 bits (2 bytes)
`sqn`	bigint	Número de sequência (anti-replay)	48 bits
`authentication_algorithm`	string	Nome do algoritmo	Atualmente "milenage"
`ota_counter`	bigint	Contador de operação OTA	Inteiro

Pontos-Chave:

- Vários assinantes podem compartilhar o mesmo conjunto de chaves
- Ki é o segredo mestre compartilhado com o SIM
- OPC ou OP devem ser fornecidos (OPC pode ser derivado de OP)
- SQN é incrementado a cada autenticação
- Milenage é atualmente o único algoritmo suportado

Algoritmo de Autenticação:

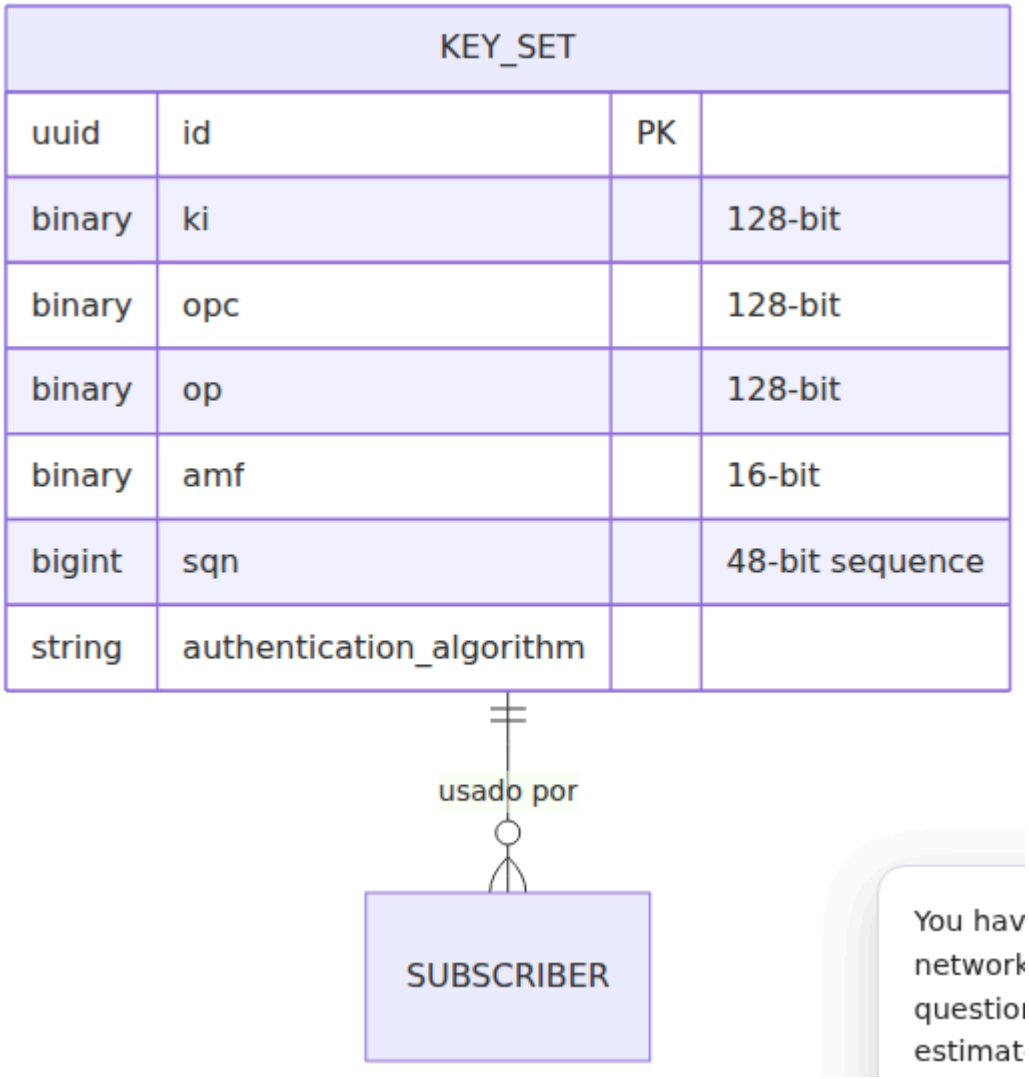


Conjunto de Chaves, SIM e Assinante

O SIM, o Conjunto de Chaves e o Assinante são objetos separados. Cada objeto tem seu próprio ciclo de vida. O Assinante é o registro que os vincula.

- **SIM** - o cartão físico ou o eSIM. Ele contém o ICCID e os códigos PIN, PUK e ADM. É inventário.
- **Conjunto de Chaves** - as chaves de autenticação (Ki, OPc, AMF, SQN). A rede autentica contra essas chaves.
- **Assinante** - a identidade da rede e o serviço (IMSI, MSISDN, perfis, estado habilitado).

Um Assinante referencia um SIM (`sim_id`) e um Conjunto de Chaves (`key_set_id`). Um Conjunto de Chaves não referencia um SIM. Para descobrir a qual SIM um Conjunto de Chaves pertence, leia o Assinante que referencia ambos.



Por que os objetos são separados

Os objetos são separados porque não se mapeiam um a um em todos os casos:

- **Troca de SIM.** Um assinante deixa cair o telefone e precisa de um novo SIM. Você mantém o Assinante, com seu MSISDN, perfis e dados IMS. Você o aponta para um novo SIM e um novo Conjunto de Chaves. O serviço continua.
- **SIM Multi-IMSI.** Um ICCID carrega vários IMSIs. Cada IMSI é um Assinante separado. Alguns IMSIs compartilham um Conjunto de Chaves. Alguns IMSIs têm seu próprio Conjunto de Chaves. As chaves pertencem a um IMSI, não a um cartão. Veja Suporte a SIM Multi-IMSI.

- **Roaming hospedado ou patrocinado.** O applet SIM autentica com um IMSI visitado. A rede hospedeira traduz o IMSI visitado para o seu IMSI doméstico antes que a solicitação chegue ao HSS. Os dois IMSIs compartilham um Conjunto de Chaves.

Um Conjunto de Chaves pode, portanto, atender mais de um Assinante. É por isso que o Conjunto de Chaves é um objeto separado, e não um campo no SIM.

Correlação na importação

O arquivo do fornecedor de SIM fornece o ICCID e as chaves em uma linha. Na importação, você cria o Conjunto de Chaves, cria o SIM e cria o Assinante que referencia ambos. O Assinante é o registro conectivo. Para o fluxo de trabalho de importação, veja Pré-Provisionamento de Assinante Inativo.

MSISDN

O **MSISDN** representa um número de telefone.

MSISDN			
uuid	id	PK	
string	msisdn	UK	1-15 dígitos

atribuído

JOIN_SUB_MSISDN		
uuid	subscriber_id	FK
uuid	msisdn_id	FK

pertence



Campos:

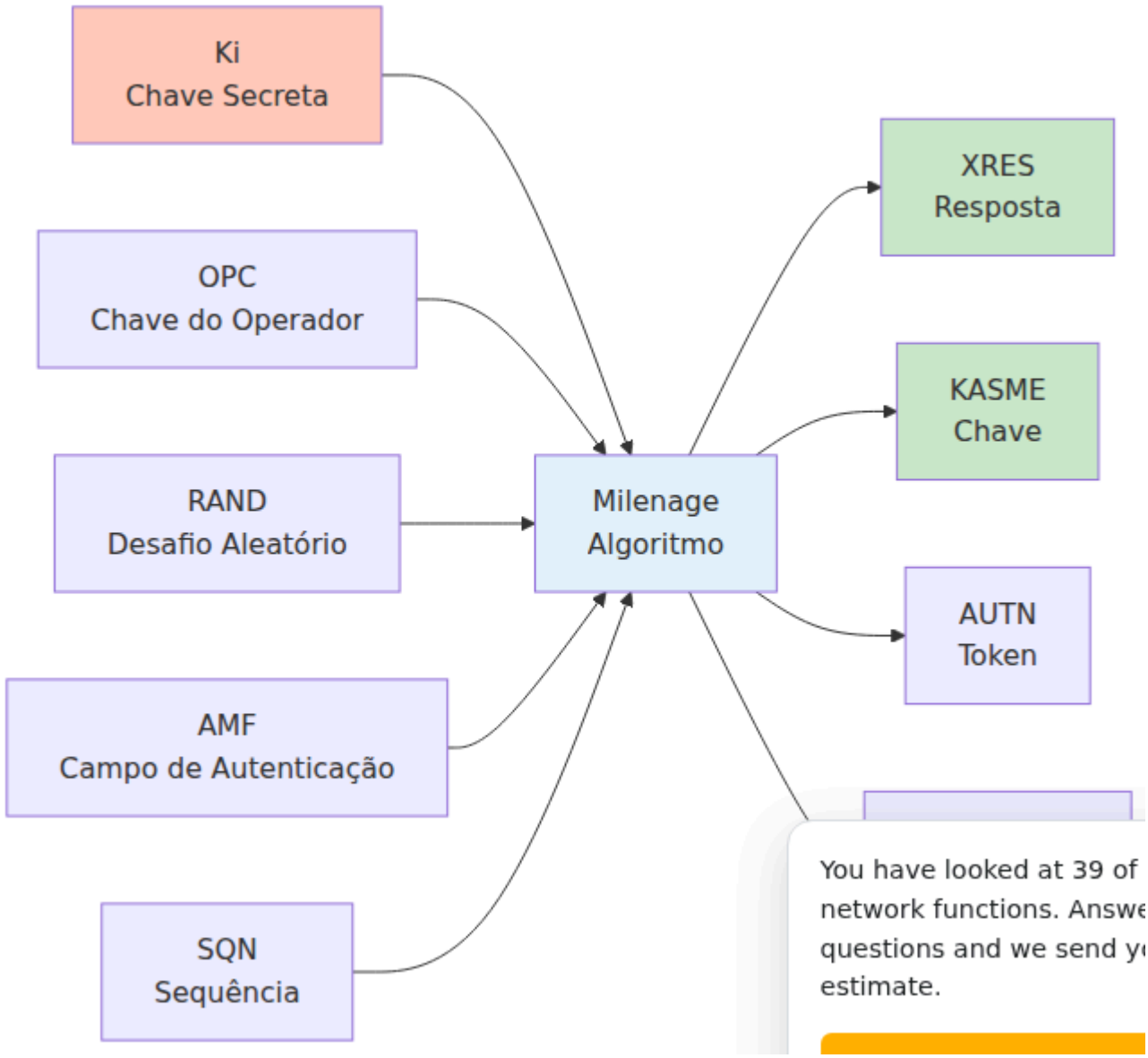
Campo	Tipo	Descrição	Formato
`msisdn`	string	Número ISDN da Estação Móvel	1-15 dígitos, formato E.164

Pontos-Chave:

- MSISDN é o número de telefone no formato internacional.
- Um assinante pode ter muitos MSISDNs (um IMSI, muitos números).
- Um MSISDN pode ser atribuído a muitos assinantes (muitos IMSIs, um número). Use isso para um número compartilhado, por exemplo, um telefone e um smartwatch. O número é armazenado como um registro MSISDN, e cada assinante mantém uma linha de junção para ele.

- Uma pesquisa reversa por MSISDN (por exemplo, a API `GET /subscriber/msisdn/:msisdn`, ou uma pesquisa Diameter indexada no número) retorna um assinante. Quando um número é compartilhado, use o IMSI ou o ICCID para selecionar um assinante específico.
- Formato: Código do país + Número nacional, apenas dígitos - sem prefixo `+` (por exemplo, `14155551234` não `+14155551234`)
- O prefixo `+` é removido automaticamente durante pesquisas Diameter Cx/Rx quando o S-CSCF ou P-CSCF envia um URI SIP com ou sem o `+`

Padrão Multi-MSISDN:

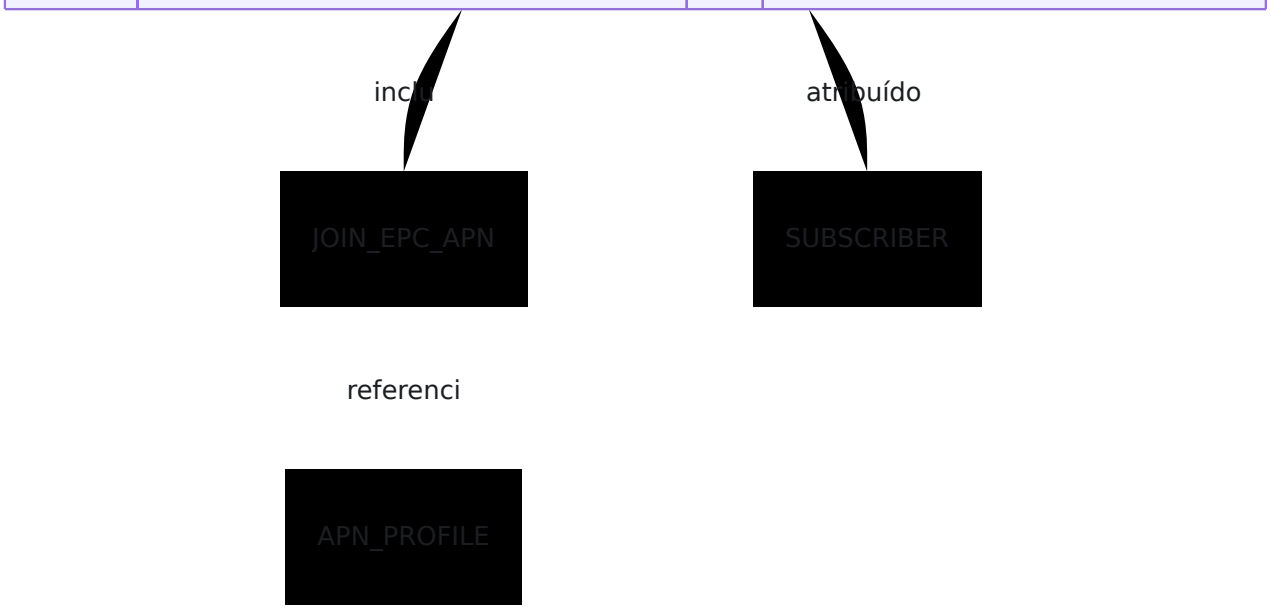


Entidades de Perfil

Perfil EPC

O **Perfil EPC** define as características do serviço de dados para LTE.

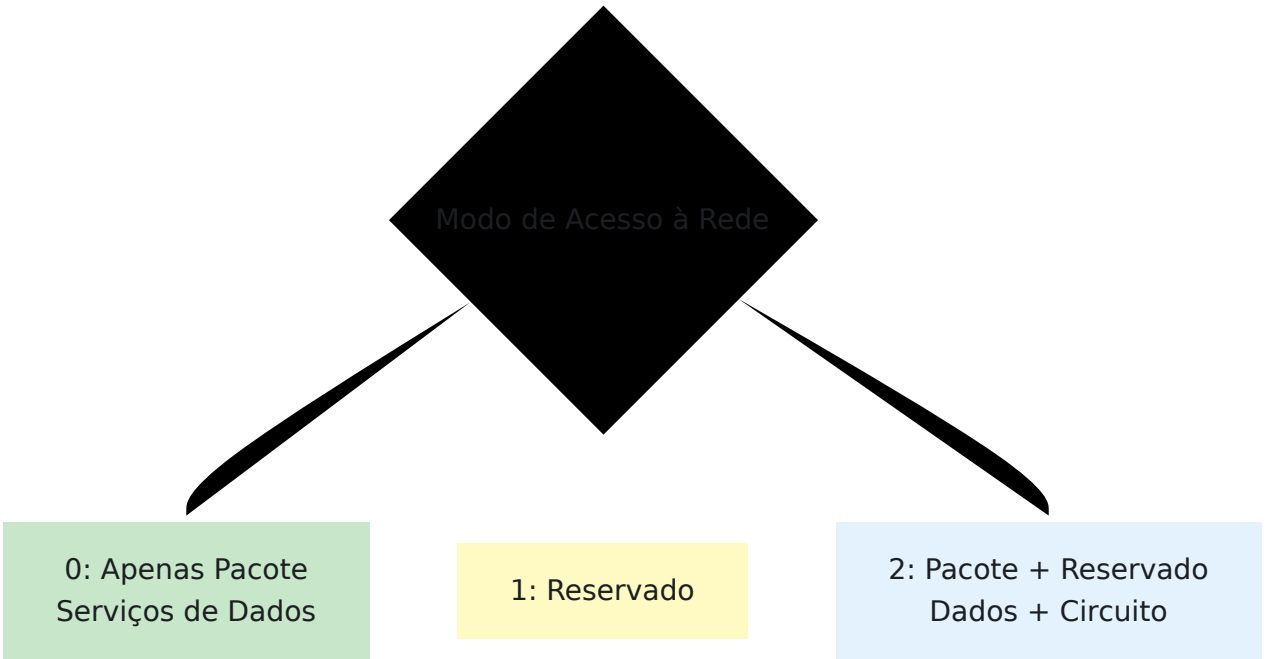
EPC_PROFILE			
uuid	id	PK	
string	name	UK	
integer	ue_ambr_dl_kbps		
integer	ue_ambr_ul_kbps		
string	network_access_mode		apenas pacote ou pacote_e_circuito
integer	tracking_area_update_interval_seconds		



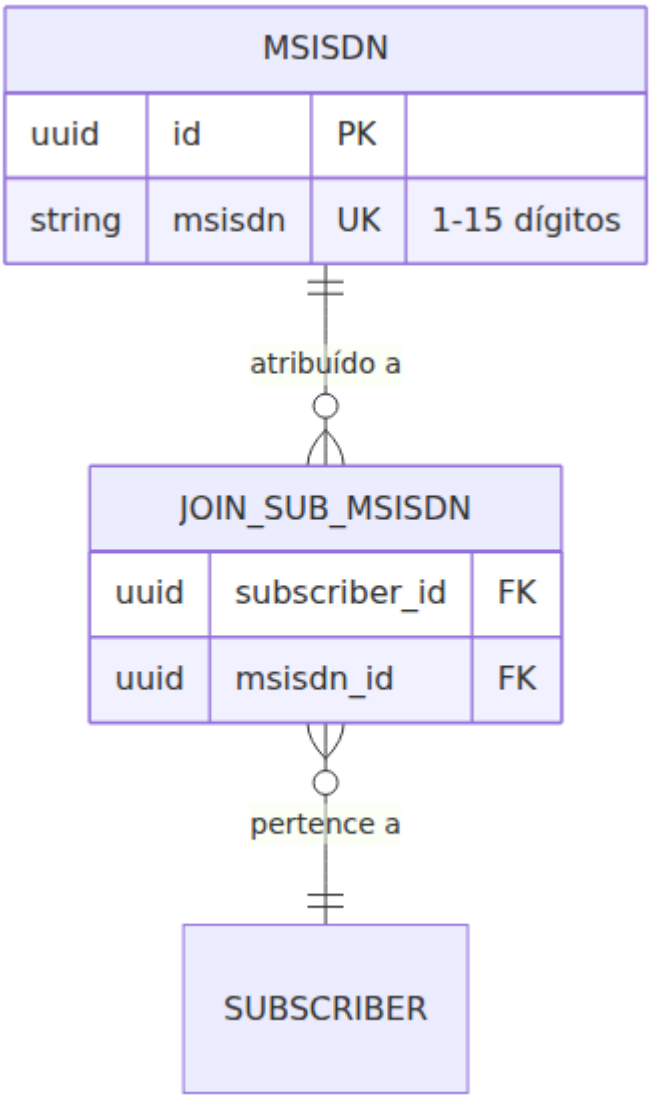
Campos:

Campo	Tipo	Descrição	Uni
`name`	string	Nome do perfil	Texto
`ue_ambr_dl_kbps`	integer	Limite de largura de banda de download	Kbps
`ue_ambr_ul_kbps`	integer	Limite de largura de banda de upload	Kbps
`network_access_mode`	string	Restrições de acesso	"apenas ou "pacote_
`tracking_area_update_interval_seconds`	integer	Intervalo de TAU	Segundo

Modos de Acesso à Rede:



AMBR (Taxa Máxima Agregada):



Perfil IMS

O **Perfil IMS** define as características do serviço de voz/vídeo.

IMS_PROFILE		
uuid	id	PK
string	name	UK
text	ifc_template	

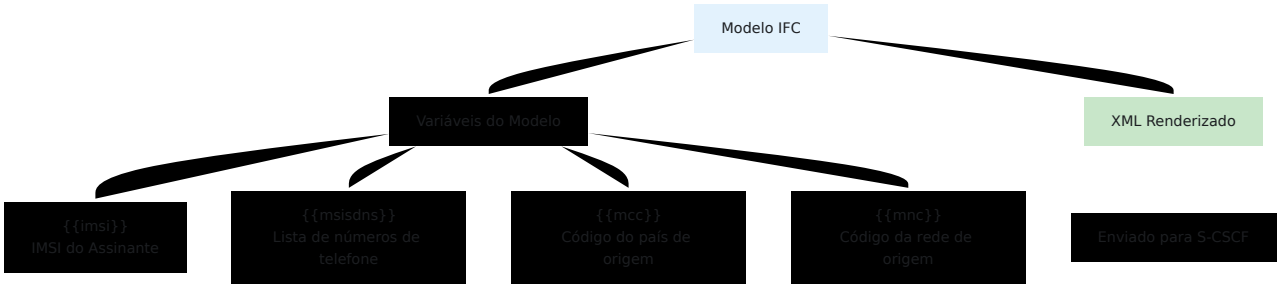
atribuído



Campos:

Campo	Tipo	Descrição	Formato
`name`	string	Nome do perfil	Texto
`ifc_template`	text	Modelo XML de Critérios de Filtro Inicial	XML com variáveis

Variáveis do Modelo IFC:



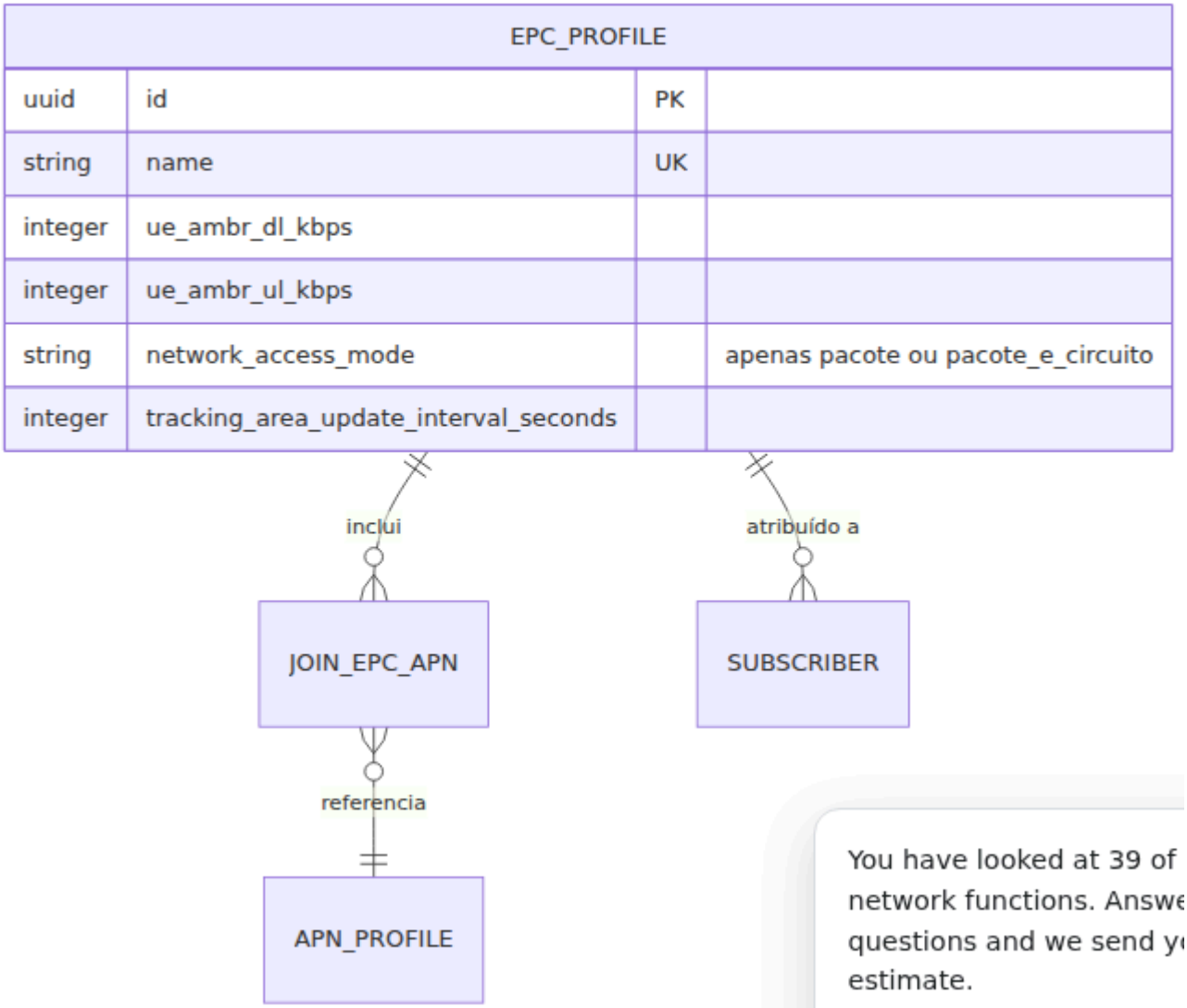
Pontos-Chave:

- IFC (Critérios de Filtro Inicial) controla o roteamento de chamadas no IMS
- O modelo é renderizado quando o assinante se registra
- As variáveis são substituídas pelos dados reais do assinante

- Enviado para S-CSCF durante o registro IMS

Perfil APN

O **Perfil APN** define características para um ponto de acesso de dados específico.



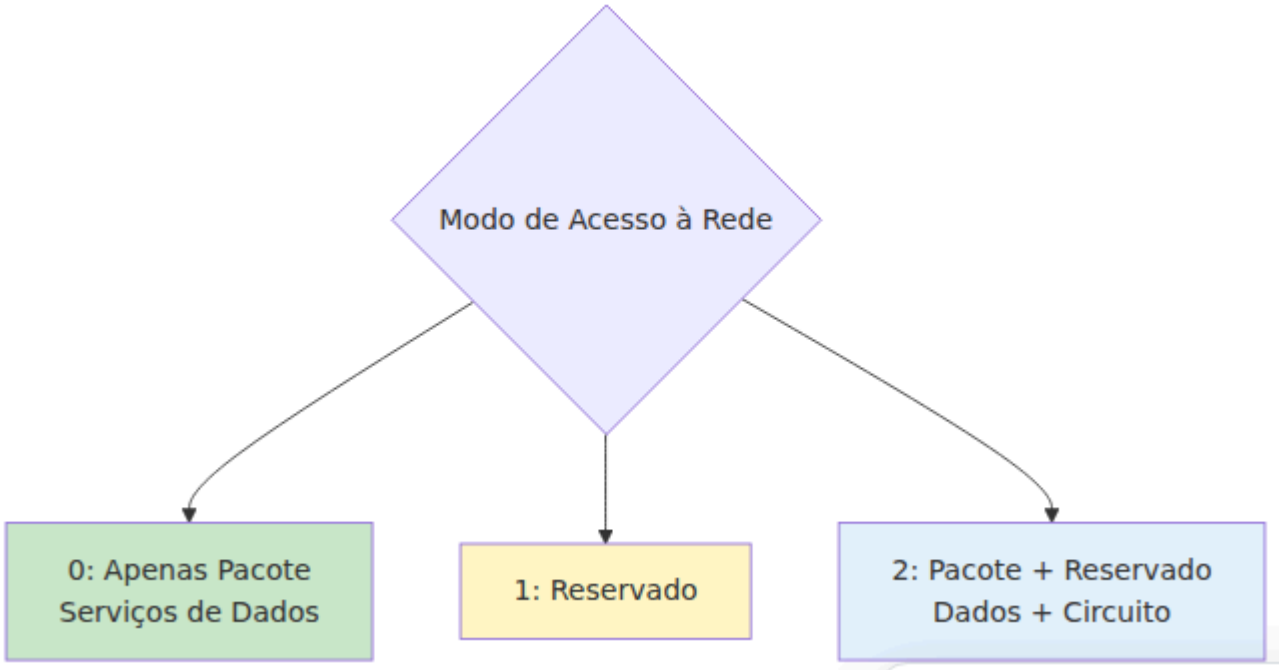
Entidades Relacionadas:

Identificador APN

APN_IDENTIFIER			
uuid	id	PK	
string	apn	UK	
string	ip_version		ipv4, ipv6, ipv4v6, ou ipv4_ou_ipv6

Campo	Tipo	Descrição	Exemplo
`apn`	string	Nome da APN	"internet", "ims", "mms"
`ip_version`	string	Suporte ao protocolo IP	Veja abaixo

Opções de Versão IP:



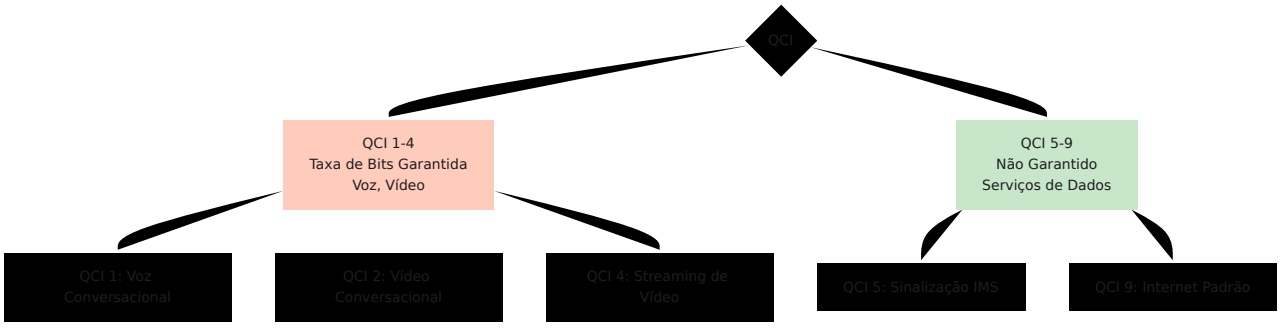
Perfil QoS APN

APN_QOS_PROFILE		
uuid	id	PK
string	name	UK
integer	qci	
integer	allocation_retention_priority	
integer	apn_ambr_dl_kbps	
integer	apn_ambr_ul_kbps	
boolean	pre_emption_capability	
boolean	pre_emption_vulnerability	

Parâmetros QoS:

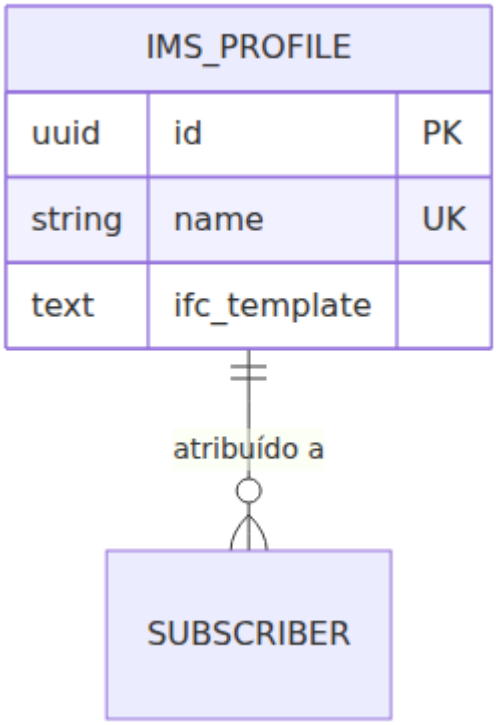
Parâmetro	Descrição	Faixa	Bearer Padrão
`qci`	Identificador de Classe QoS	1-9	QCI 9 (Internet)
`allocation_retention_priority`	Prioridade ARP	1-15	8 (prioridade mais baixa)
`apn_ambr_dl_kbps`	Limite de download da APN	0+	Varia
`apn_ambr_ul_kbps`	Limite de upload da APN	0+	Varia
`pre_emption_capability`	Pode preemptar outros	true/false	false
`pre_emption_vulnerability`	Pode ser preemptado	true/false	true

Valores de QCI:



Perfil de Roaming

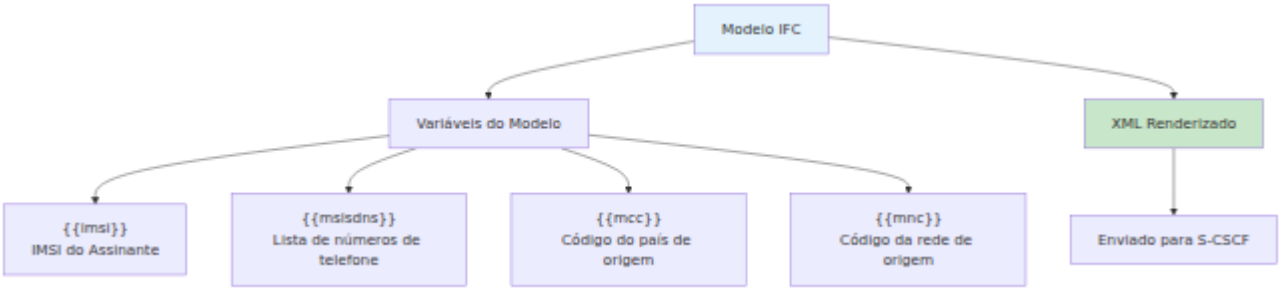
O **Perfil de Roaming** controla o acesso quando o assinante visita outras redes.



Regra de Roaming:

ROAMING_RULE			
uuid	id	PK	
string	name	UK	
string	mcc		3 dígitos
string	mnc		2-3 dígitos
string	data_action		permitir ou negar
string	ims_action		permitir ou negar

Avaliação de Regras:



Regra EIR

Regras de identidade de equipamento avaliadas pela função EIR S13. As Regras EIR são independentes e não estão anexadas a um Assinante ou perfil. Veja Documentação EIR.

EIR_RULE			
uuid	id	PK	
string	action		lista branca, lista negra ou lista cinza
string	regex		testado contra IMEISV, depois IMEI, depois IMSI
integer	precedence		1-65535, menor é avaliado primeiro

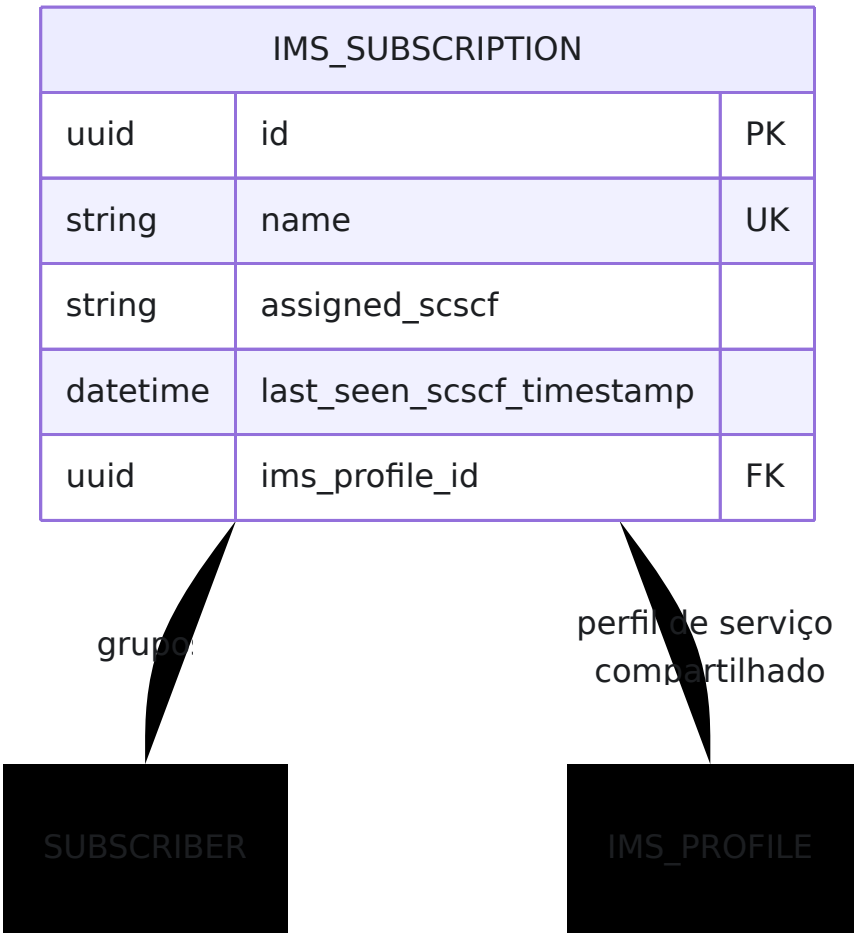
Campo	Tipo	Descrição	Exemplo
`action`	string	Resultado retornado quando a regra corresponde	"lista negra"
`regex`	string	Expressão regular testada contra o IMEISV, depois o IMEI, depois o IMSI	"35979139461640"
`precedence`	integer	Ordem de avaliação dentro de cada etapa de correspondência, mais baixa primeiro. Omitir na criação para anexar; forneça um valor para inserir nessa posição e deslocar as regras a partir dela para baixo por uma	1

Veja Precedência de Regras para o comportamento completo de colocação.

Entidades de Estado

Assinatura IMS

A **Assinatura IMS** agrupa vários assinantes (IMPIs / IMSIs) em uma única assinatura IMS que compartilha uma Identidade Pública de Usuário e um único S-CSCF de atendimento (3GPP TS 23.228 IMPU compartilhado). Use-a para atender um número em vários dispositivos, por exemplo, um telefone e um smartwatch em SIMs separados.


IMS_SUBSCRIPTION
uuid
id
PK
string
name
UK
string
assigned_scscf
datetime
last_seen_scscf_timestamp
uuid
ims_profile_id
FK
grupo
perfil de serviço
compartilhado
SUBSCRIBER
IMS_PROFILE

Campos:

Campo	Tipo	Descrição	Restrições
`name`	string	Nome legível por humanos	Obrigatório, único
`assigned_scscf`	string	S-CSCF de serviço compartilhado para o grupo	Estado em tempo de execução, definido pelo HSS
`last_seen_scscf_timestamp`	datetime	Quando o S-CSCF compartilhado foi escrito pela última vez	Estado em tempo de execução
`ims_profile_id`	uuid	Perfil de Serviço compartilhado opcional para as identidades compartilhadas do grupo	Opcional

Pontos Chave:

- Membros referenciam o grupo via `subscriber.ims_subscription_id`.
- O HSS atribui um S-CSCF quando o primeiro membro se registra e o libera quando o último membro se desregistra. Não defina `assigned_scscf` você mesmo.
- Quando `ims_profile_id` está definido, o perfil IMS do grupo renderiza as identidades compartilhadas; caso contrário, cada membro renderiza seu próprio perfil IMS.
- Excluir uma Assinatura IMS não exclui seus assinantes; seu `ims_subscription_id` é limpo.

Veja Identidade Pública de Usuário Compartilhada para o modelo completo e provisionamento.

Estado do Assinante

O **Estado do Assinante** rastreia o status do assinante em tempo real.

APN_IDENTIFIER			
uuid	id	PK	
string	apn	UK	
string	ip_version		ipv4, ipv6, ipv4v6, ou ipv4_ou_ipv6

Campos Chave:

Informações de Localização:

- `last_seen_mcc`, `last_seen_mnc` - Rede visitada
- `last_seen_tac` - Código da Área de Rastreamento
- `last_seen_cell_id` - ID da Célula
- `last_seen_enodeb_id` - ID do eNodeB
- `last_seen_eci` - Identificador da Célula E-UTRAN

Elementos de Rede:

- `last_seen_mme` - MME atual atendendo o assinante
- `last_seen_realm` - domínio Diameter do MME
- `last_seen_rat_type` - Tecnologia de Acesso Rádio (LTE, 5G, etc.)

Informações IMS:

- `assigned_scscf` - S-CSCF atual atendendo o assinante
- `ims_public_identity` - URI SIP (ex.: sip:+14155551234@ims.example.com)
- `sh_repository_data` - Dados personalizados do perfil IMS

Equipamento:

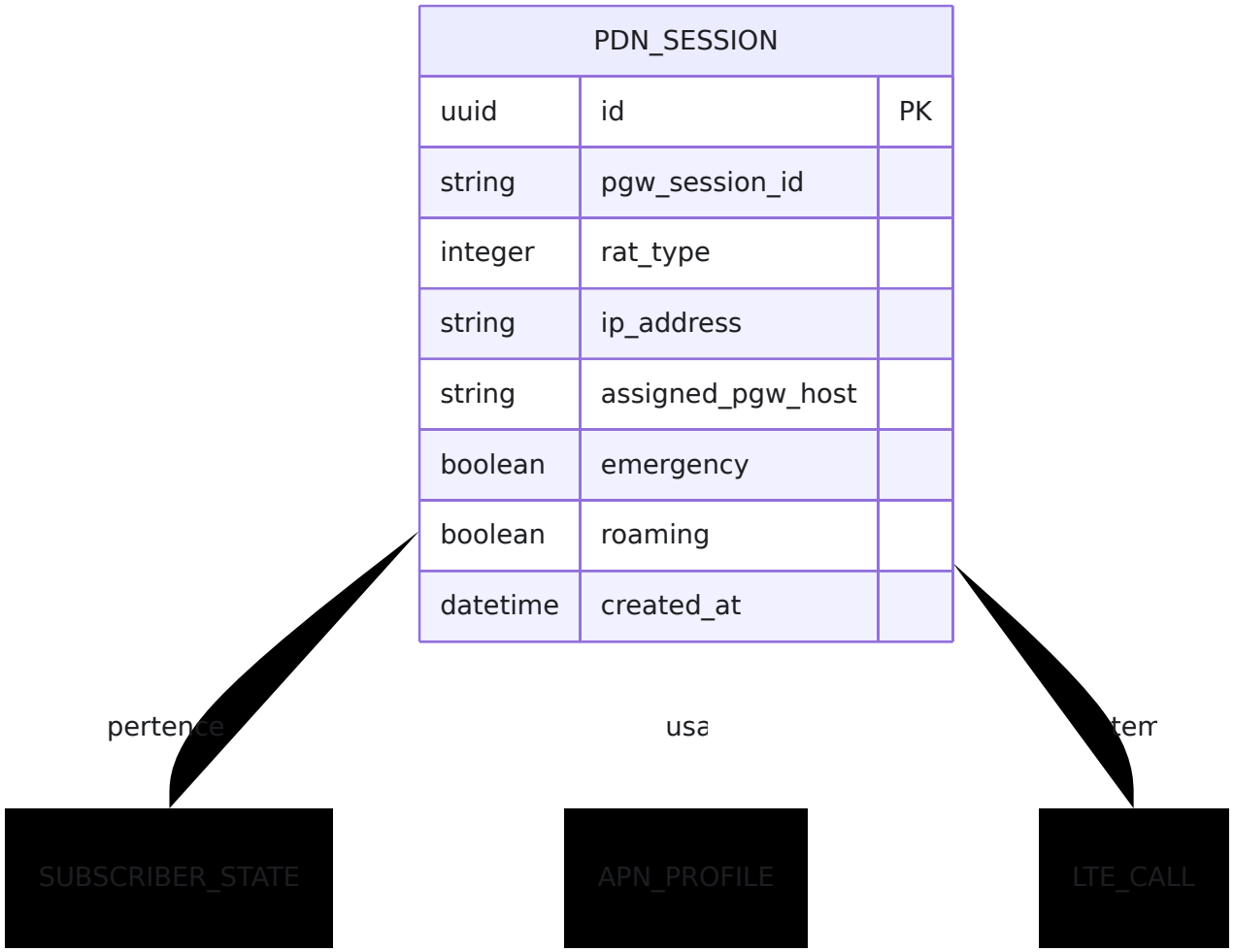
- `last_seen_imei` - IMEI reportado pela última vez no AVP de Informação do Terminal de um ECR S13 ou ULR S6a
- `last_seen_imei_software_version` - Versão do Software reportada com aquele IMEI (IMEISV = IMEI + Versão do Software); mantido do relatório anterior quando o mesmo IMEI chega sem um
- `last_seen_imei_timestamp` - Quando o IMEI foi reportado pela última vez

Timestamps:

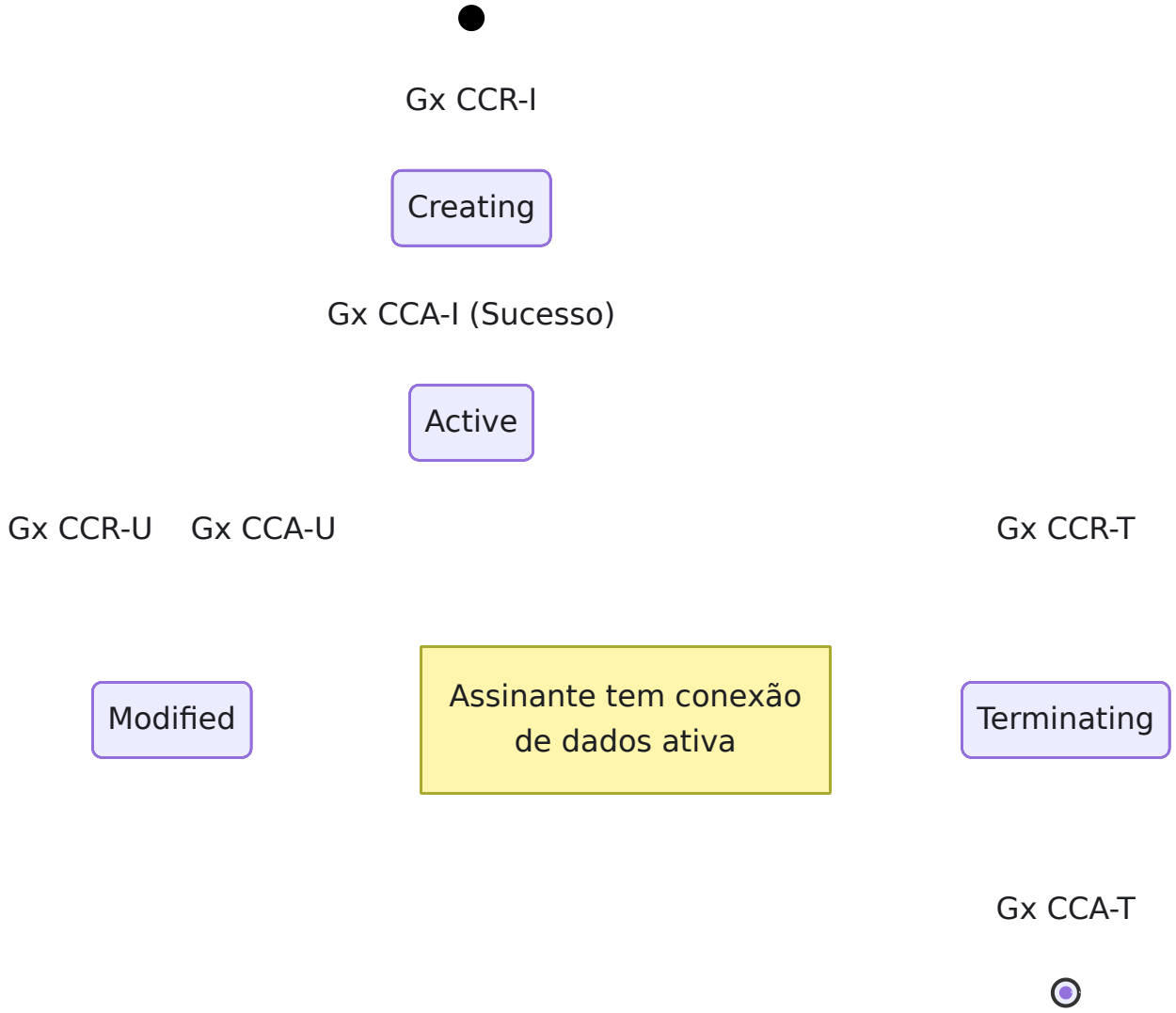
- `last_seen_at` - Última mensagem Diameter recebida
- Vários timestamps `last_*_at` para diferentes procedimentos

Sessão PDN

A **Sessão PDN** representa uma conexão de dados ativa.



Ciclo de Vida da Sessão PDN:

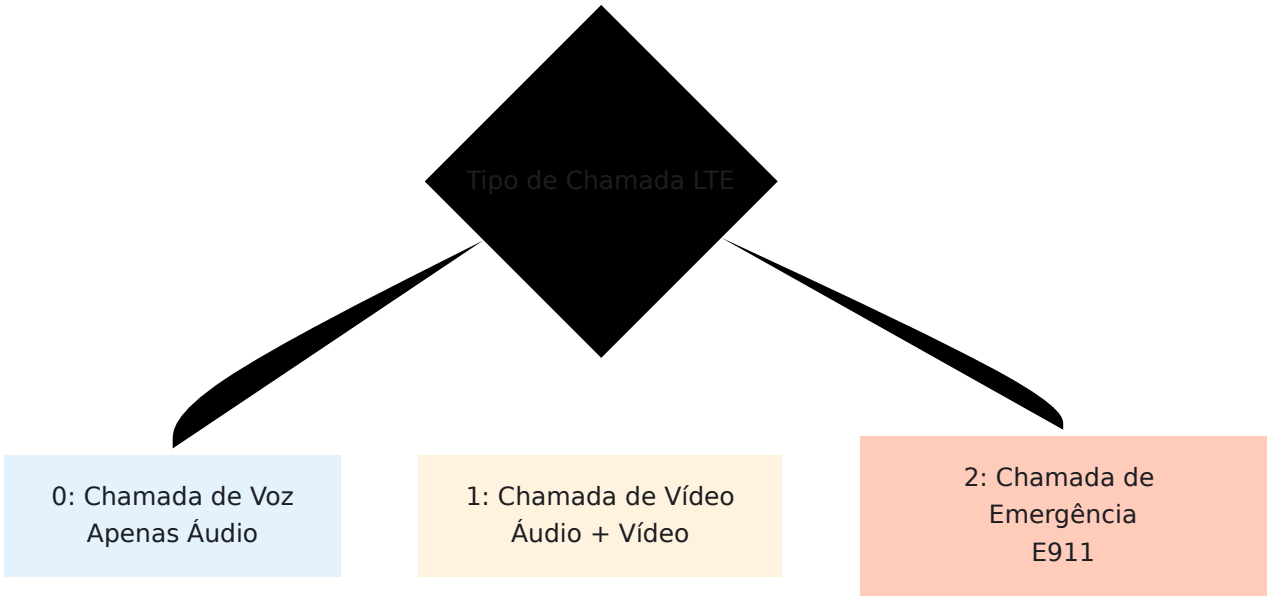


Chamada LTE

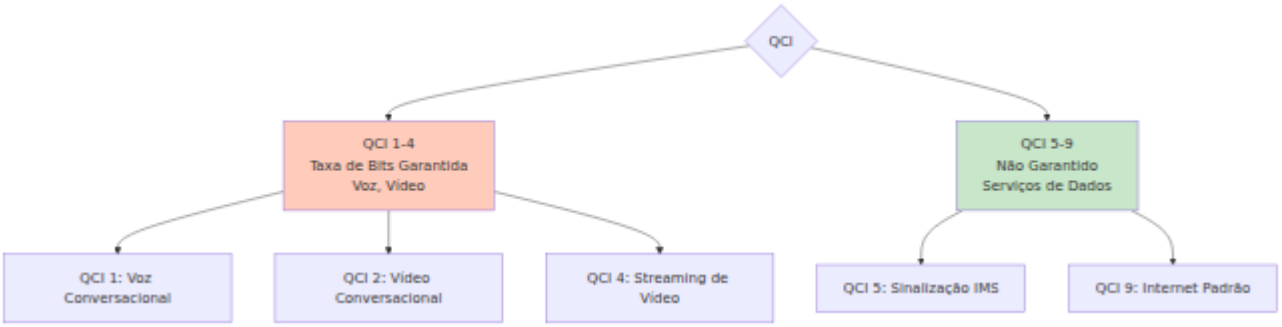
A **Chamada LTE** representa uma chamada de voz/vídeo VoLTE ativa.

APN_QOS_PROFILE		
uuid	id	PK
string	name	UK
integer	qci	
integer	allocation_retention_priority	
integer	apn_ambr_dl_kbps	
integer	apn_ambr_ul_kbps	
boolean	pre_emption_capability	
boolean	pre_emption_vulnerability	

Tipos de Chamada:

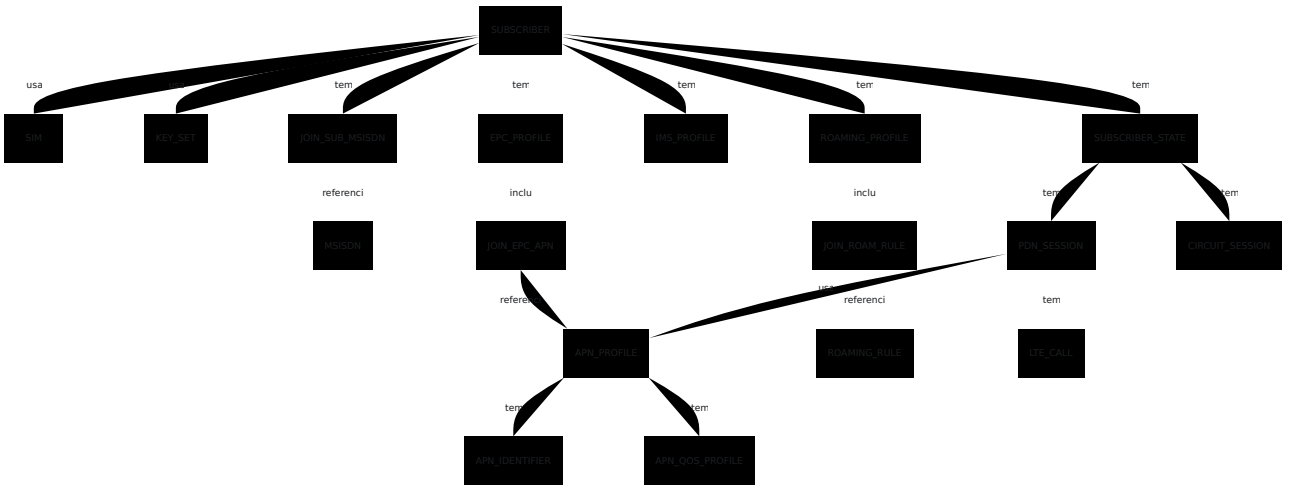


Fluxo da Chamada VoLTE:



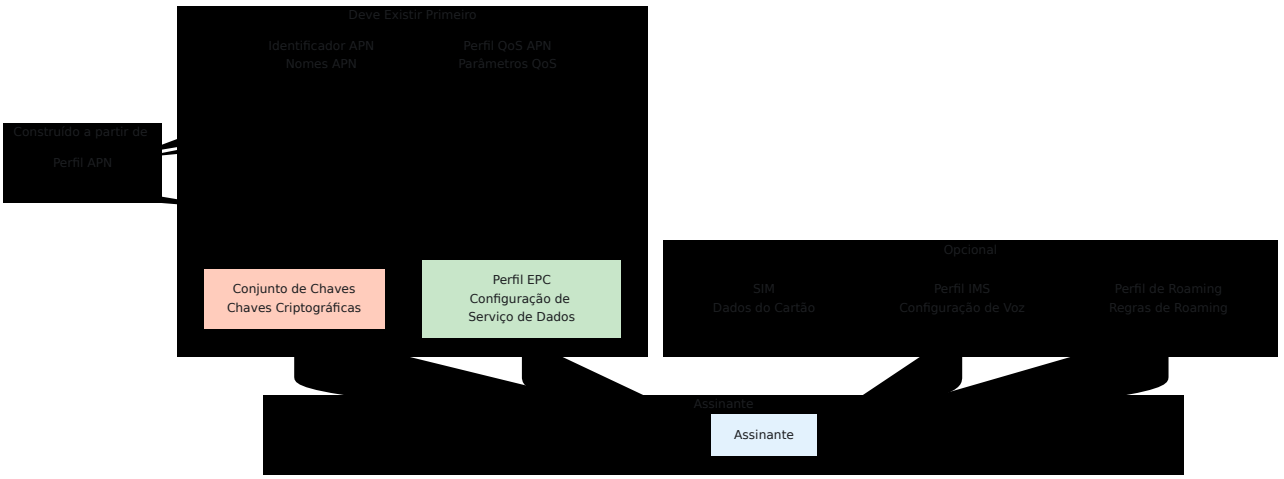
Diagramas de Relacionamento de Entidades

Relacionamentos de Entidades Completos



Relacionamentos de Provisionamento

Este diagrama mostra o que deve existir antes de criar um assinante:

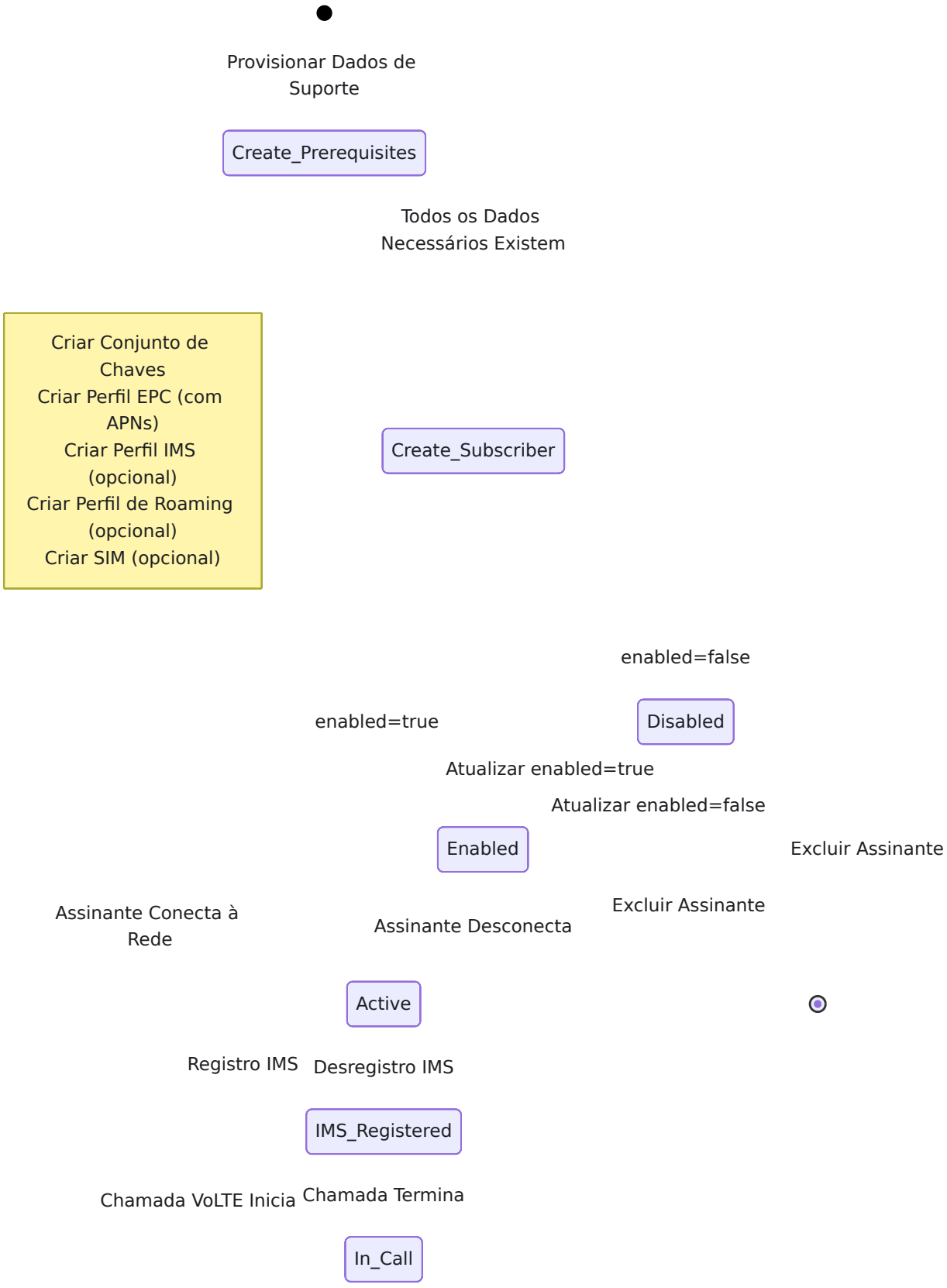


Relacionamentos de Estado da Sessão

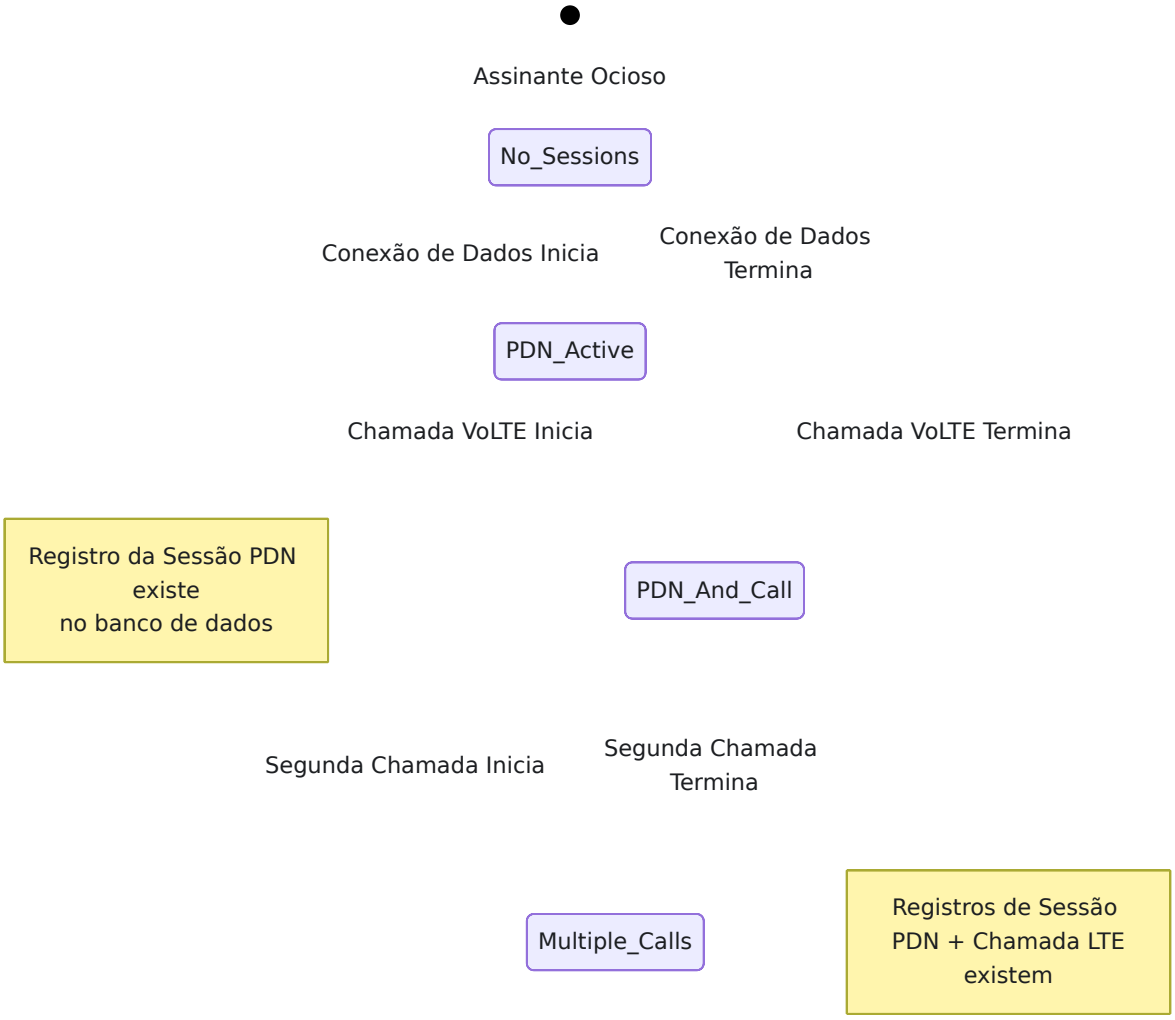
ROAMING_RULE			
uuid	id	PK	
string	name	UK	
string	mcc		3 dígitos
string	mnc		2-3 dígitos
string	data_action		permitir ou negar
string	ims_action		permitir ou negar

Ciclo de Vida da Entidade

Ciclo de Vida do Provisionamento do Assinante



Ciclo de Vida da Sessão

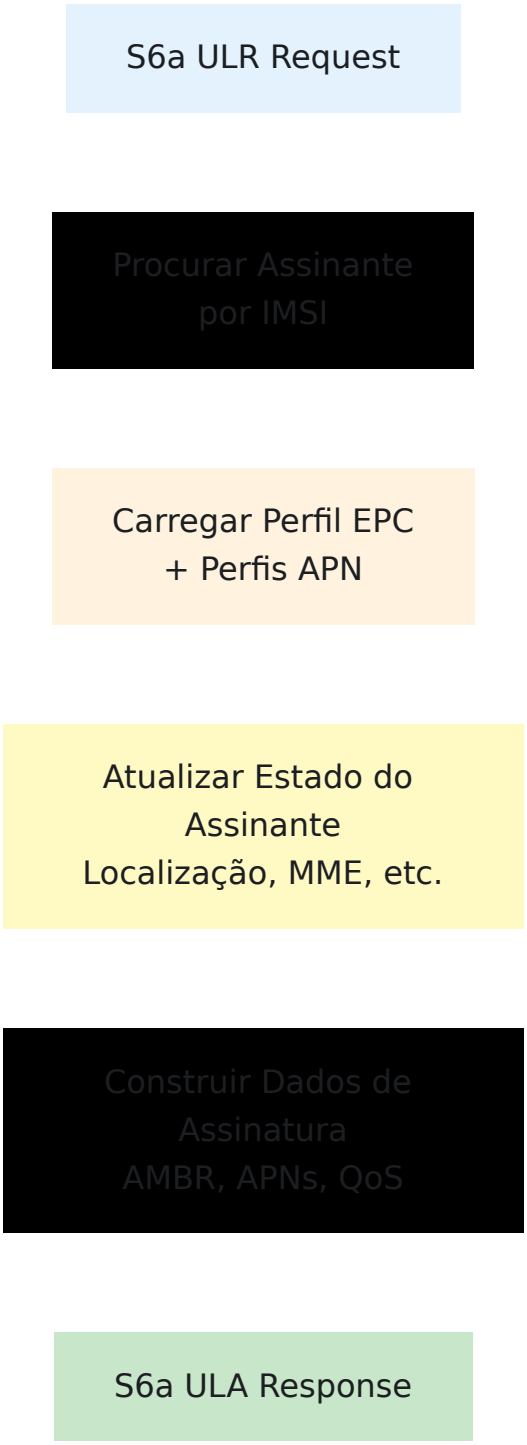


Padrões de Fluxo de Dados

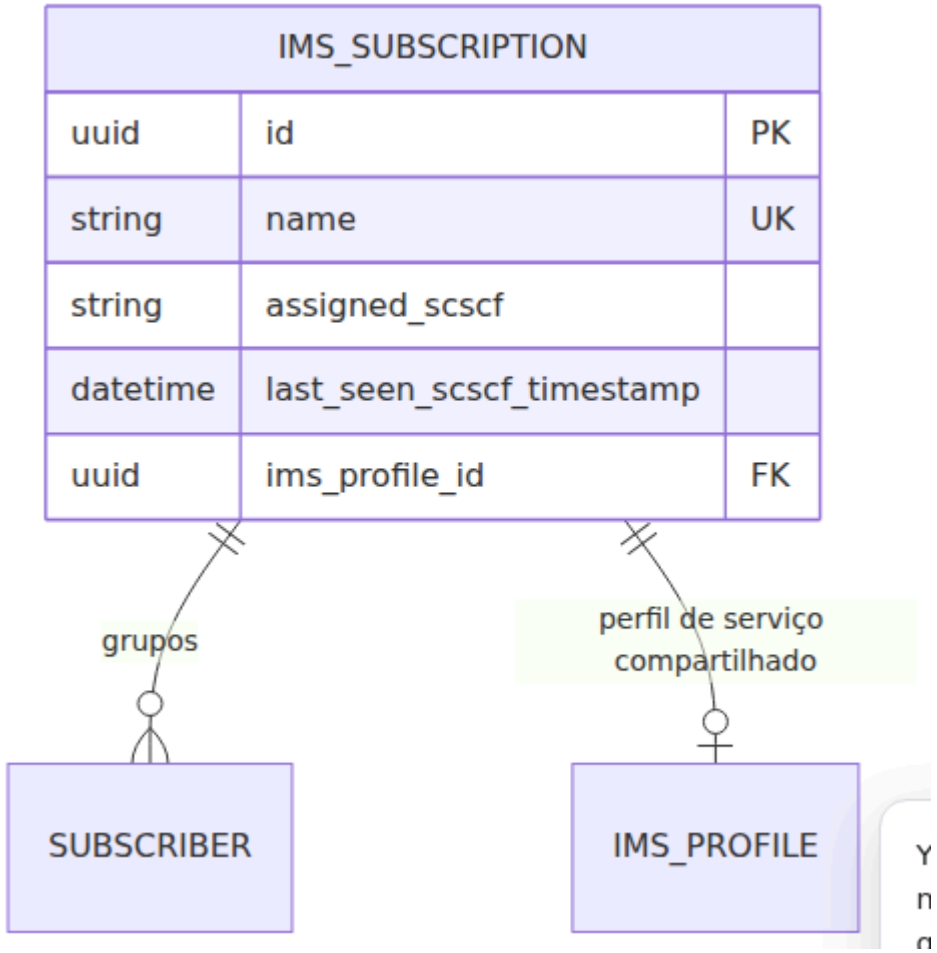
Fluxo de Autenticação

EIR_RULE			
uuid	id	PK	
string	action		lista branca, lista negra ou lista cinza
string	regex		testado contra IMEISV, depois IMEI, depois IMSI
integer	precedence		1-65535, menor é avaliado primeiro

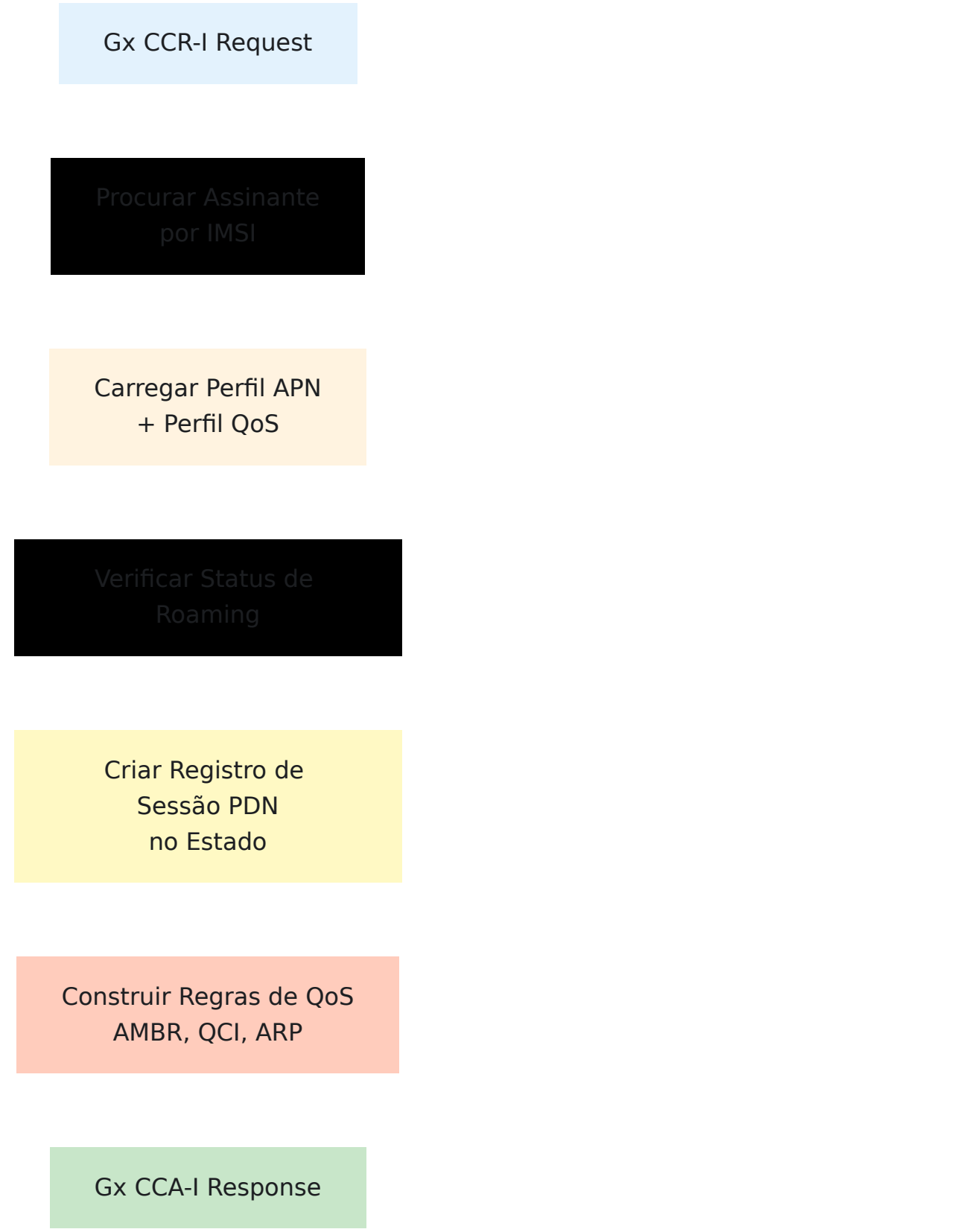
Fluxo de Atualização de Localização



Fluxo de Registro IMS



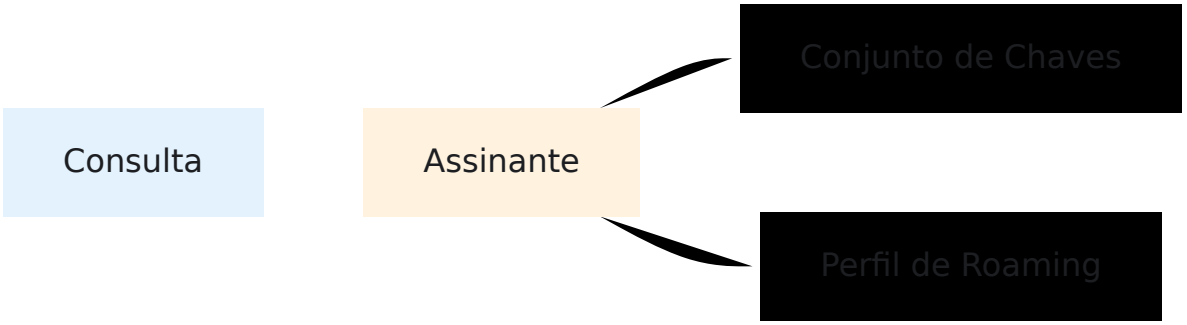
Fluxo de Estabelecimento de Sessão



Padrões de Otimização de Consultas

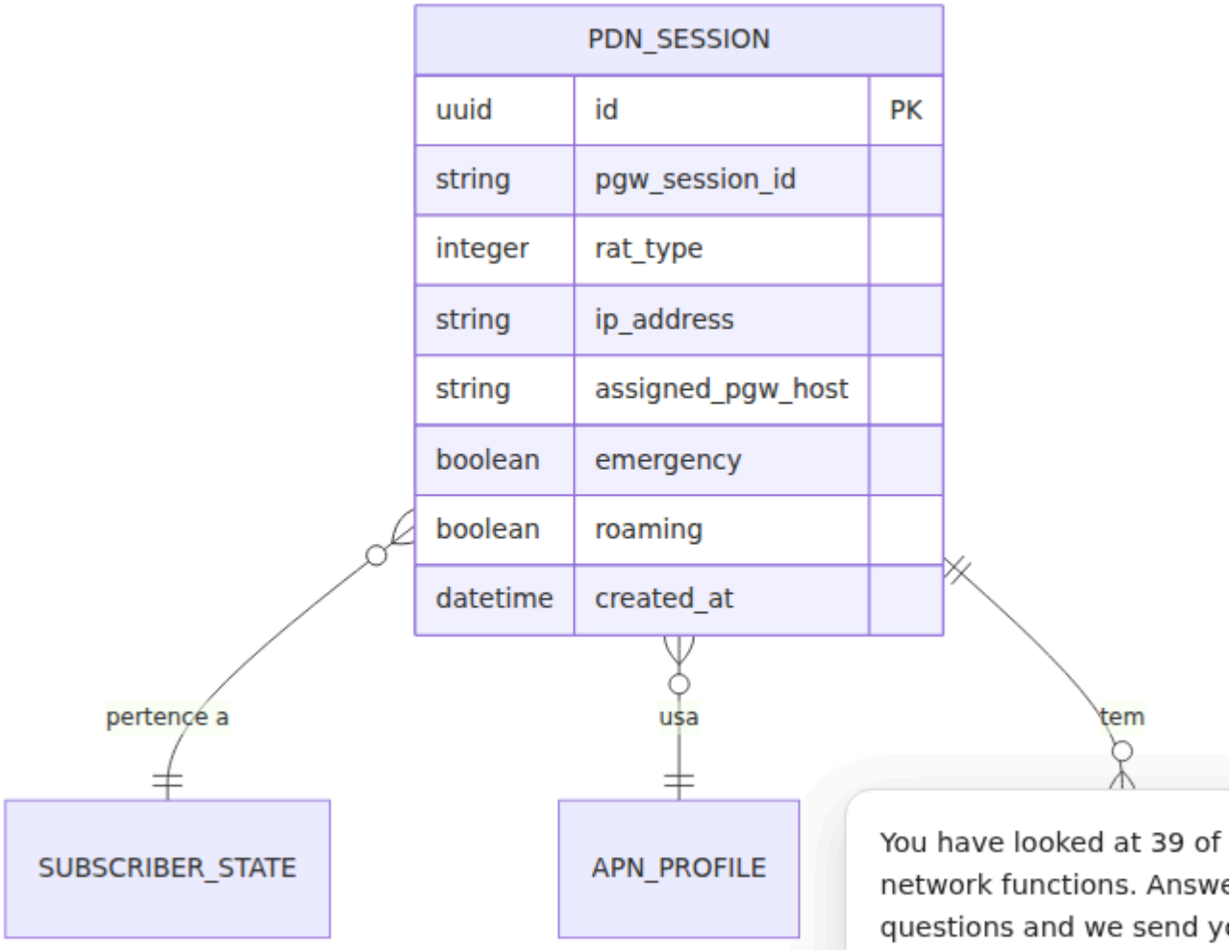
OmniHSS otimiza consultas ao banco de dados pré-carregando seletivamente apenas as associações necessárias para cada operação:

Consulta Mínima (Autenticação)



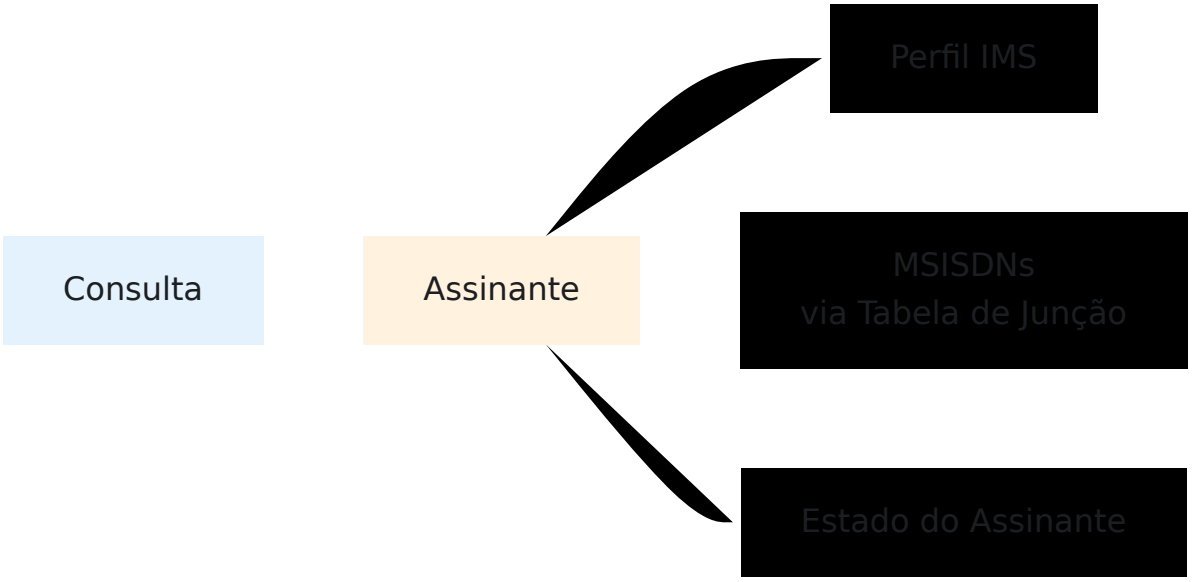
Caso de Uso: S6a AIR - Apenas precisa de chaves criptográficas e regras de roaming

Consulta Moderada (Atualização de Localização)



Caso de Uso: S6a ULR - Precisa de dados completos do perfil EPC

Consulta Completa (Registro IMS)



Caso de Uso: Cx SAR - Precisa do perfil IMS e todos os números de telefone

---

← Voltar para o Guia de Operações | Próximo: Referência da API →

Mapeamento de Dados da Resposta Diameter

← Voltar ao Guia de Operações

Este documento fornece diagramas mermaid detalhados mostrando de onde cada campo nas respostas do protocolo Diameter é originado no sistema OmniHSS.

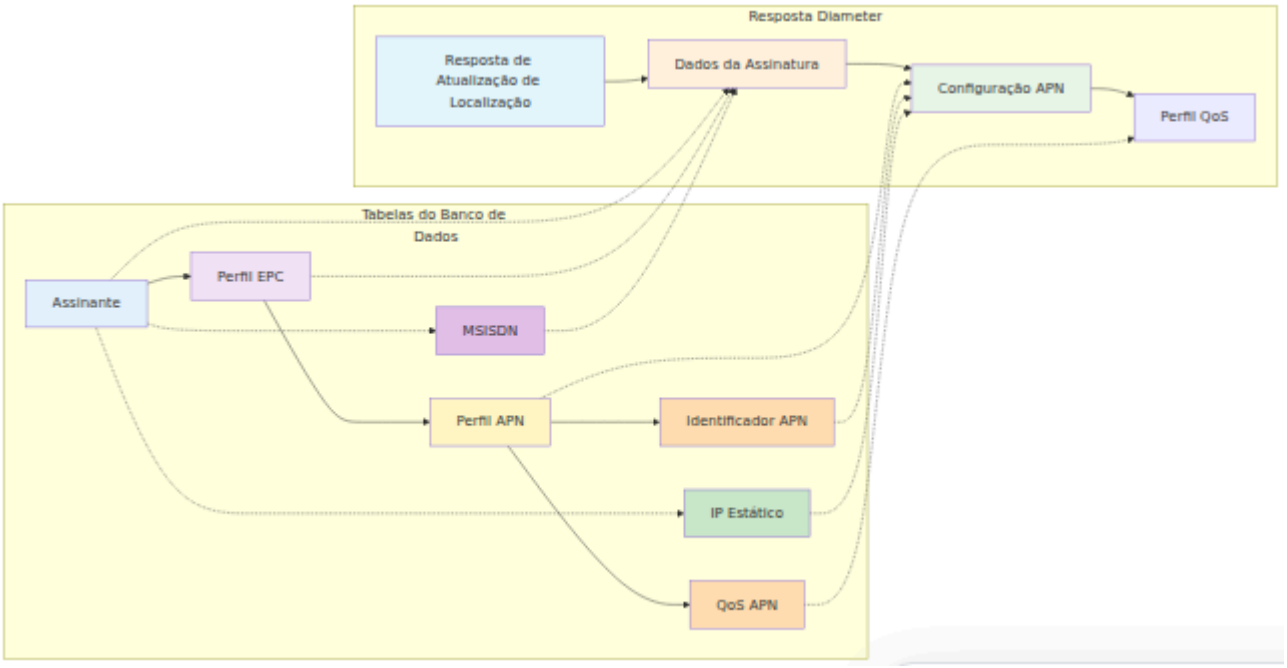
Índice

- Resposta de Atualização de Localização (S6a ULA)
- Resposta de Informação de Autenticação (S6a AIA)
- Resposta de Atribuição de Servidor (Cx SAA)
- Resposta de Controle de Crédito (Gx CCA)
- Resposta de Dados do Usuário (Sh UDA)
- Resposta de Verificação de Identidade do ME (S13 ECA)

Resposta de Atualização de Localização (S6a ULA)

A Resposta de Atualização de Localização é enviada pelo HSS para o MME durante os procedimentos de anexação LTE. Este diagrama mostra o fluxo completo de dados das tabelas do banco de dados para os AVPs Diameter.

Mapeamento da Fonte de Dados



Mapeamento Detalhado de Campos

Fonte do Banco de Dados	Campo	D
subscriber.enabled	`true/false`	Su St
msisdn.msisdn	`'14155551234'`	MS
epc_profile.ue_ambr_ul_kbps	`50000`	Ma Re Ba UL
epc_profile.ue_ambr_dl_kbps	`100000`	Ma Re Ba DL
epc_profile.network_access_mode	`'packet_only'`	Ne Ac Mo
apn_identifier.apn	`'internet'`	Se Se
apn_identifier.ip_version	`'ipv4v6'`	PD
apn_qos_profile.qci	`9`	Qo Id

Fonte do Banco de Dados	Campo	D
apn_qos_profile.allocation_retention_priority	`8`	Pr Le
apn_qos_profile.pre_emption_capability	`false`	Pr en Ca
apn_qos_profile.pre_emption_vulnerability	`true`	Pr en Vu
apn_qos_profile.apn_ambr_ul_kbps	`25000`	AP UL
apn_qos_profile.apn_ambr_dl_kbps	`50000`	AP DL
static_ip.ipv4_static_ip	`'100.64.1.1'`	Se Pa Ac (IP
static_ip.ipv6_static_ip	`'2606:4700::1111'`	Se Pa Ac (IP

Transformações Chave:

1. **Largura de banda AMBR**: O banco de dados armazena em kbps, Diameter espera bps (multiplicar por 1000)
2. **Codificação da Versão IP**: `0=IPv4, 1=IPv6, 2=IPv4v6, 3=IPv4_or_IPv6`
3. **Status do Assinante**: `enabled: true → 0 (SERVICE_GRANTED)`, `enabled: false → 1 (OPERATOR_DETERMINED_BARRING)`

4. **Context-Identifier**: Numeração sequencial (0, 1, 2...) para cada APN no perfil
5. **IP Estático**: Incluído somente se atribuído via relacionamento muitos-para-muitos `static_ips`

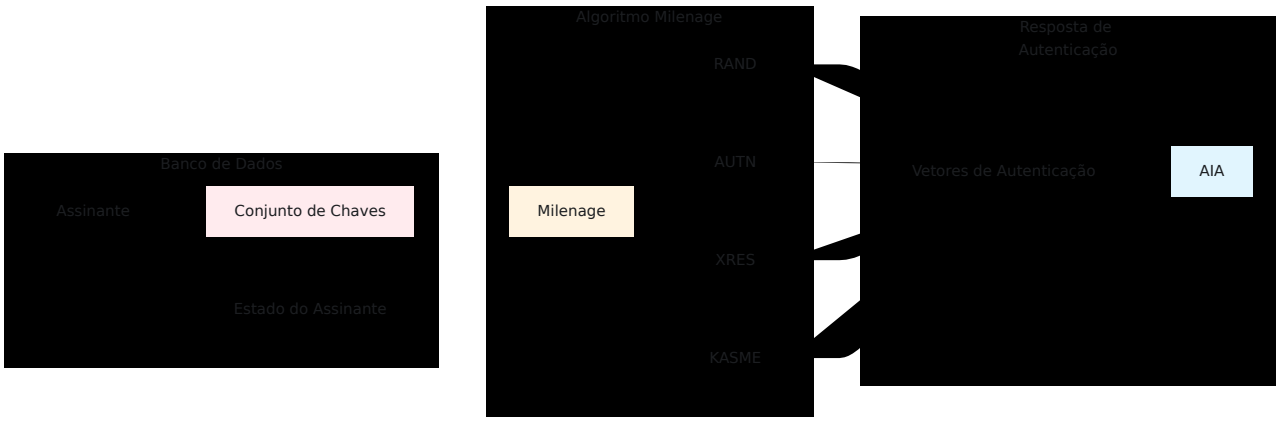
Validação da Lógica de Negócios:

- Verificação de Roaming: Combinar PLMN visitado com `roaming_profile.roaming_rules`
- Verificação de Assinante habilitado: `subscriber.enabled == true`
- Filtrar APNs: Pode excluir APNs IMS se a política de roaming negar IMS

Resposta de Informação de Autenticação (S6a AIA)

A Resposta de Informação de Autenticação fornece vetores de autenticação para assinantes LTE/EPC.

Mapeamento da Fonte de Dados



Componentes Chave:

1. **Chaves Criptográficas**: Todas as chaves armazenadas como strings hexadecimais na tabela `key_set`
2. **Gerenciamento de SQN**: Número de sequência incrementado após cada geração de vetor de autenticação (previne ataques de repetição)

3. **Algoritmo Milenage**: 3GPP TS 35.206 - gera vetores de autenticação
4. **Derivação de KASME**: Chave derivada de CK||IK usando KDF conforme TS 33.401

Características de Segurança:

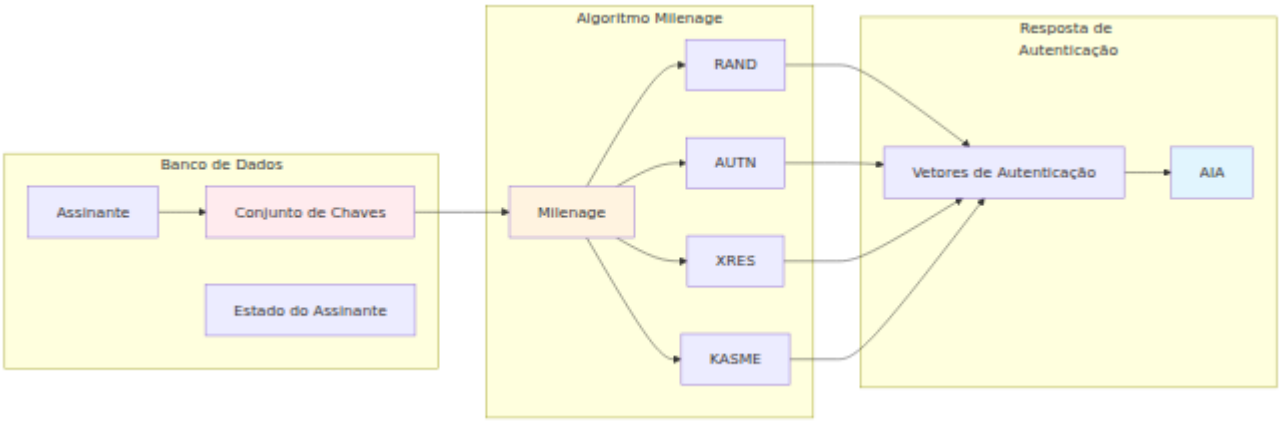
- SQN armazenado por assinante (não global)
- Ki/OPc nunca saem do HSS (somente valores derivados transmitidos)
- AUTN inclui número de sequência (SQN) e AMF para autenticação de rede
- O algoritmo Milenage fornece autenticação mútua entre UE e rede

---

Resposta de Atribuição de Servidor (Cx SAA)

A Resposta de Atribuição de Servidor é enviada pelo HSS para o S-CSCF durante o registro IMS.

Mapeamento da Fonte de Dados



Características Chave:

1. **Template IFC**: Template XML armazenado em `ims_profile.ifc_template`
2. **Substituição Dinâmica**: Substitui `{{msisdn}}`, `{{imsi}}`, `{{impu}}` em tempo de execução
3. **Atribuição S-CSCF**: Armazena S-CSCF atribuído em `subscriber_state.assigned_scscf`

4. **Identidade Pública IMS**: Formato: `sip:+{msisdn}@{ims_domain}` ou `tel:+{msisdn}`

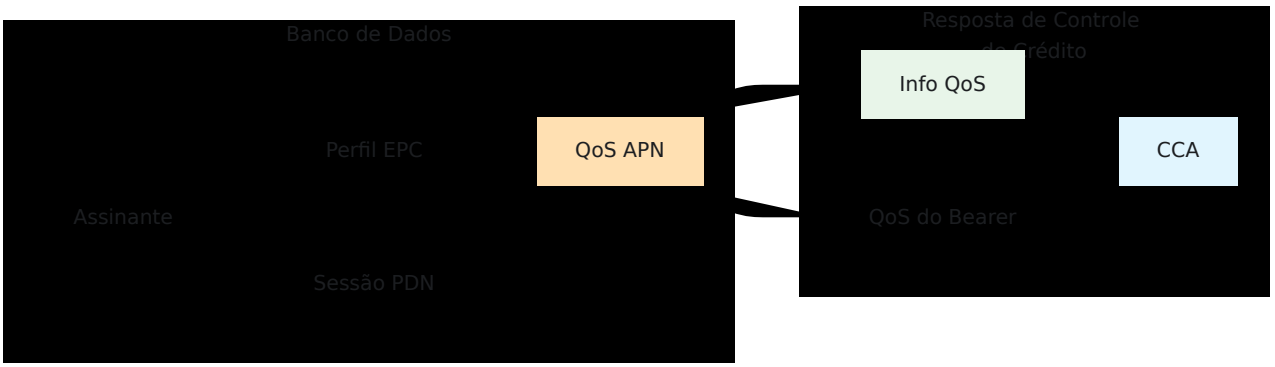
Parâmetros do Template IFC:

- `{{msisdn}}` - Primeiro MSISDN do assinante
- `{{imsi}}` - IMSI do assinante
- `{{impu}}` - Identidade Pública do Usuário IMS (do estado do assinante)
- `{{impi}}` - Identidade Privada do Usuário IMS (tipicamente IMSI@realm)

Resposta de Controle de Crédito (Gx CCA)

A Resposta de Controle de Crédito é enviada pela função PCRF para o PGW durante o estabelecimento do bearer.

Mapeamento da Fonte de Dados



Características Chave:

1. **Rastreamento de Sessão**: Cria/atualiza registro `pdn_session` para cada bearer
2. **Imposição de QoS**: Fornece QCI e limites de largura de banda do perfil QoS APN
3. **Regras de Cobrança**: Retorna regras de cobrança padrão para integração de faturamento
4. **CC-Request-Type**: Trata INITIAL (1), UPDATE (2), TERMINATION (3)

Gerenciamento de Estado da Sessão:

- `INITIAL_REQUEST`: Cria novo registro de sessão PDN
- `UPDATE_REQUEST`: Atualiza sessão PDN existente
- `TERMINATION_REQUEST`: Deleta registro de sessão PDN

Resposta de Dados do Usuário (Sh UDA)

A Resposta de Dados do Usuário é enviada pelo HSS para o AS (Servidor de Aplicação) via interface Sh.

Mapeamento da Fonte de Dados



Características Chave:

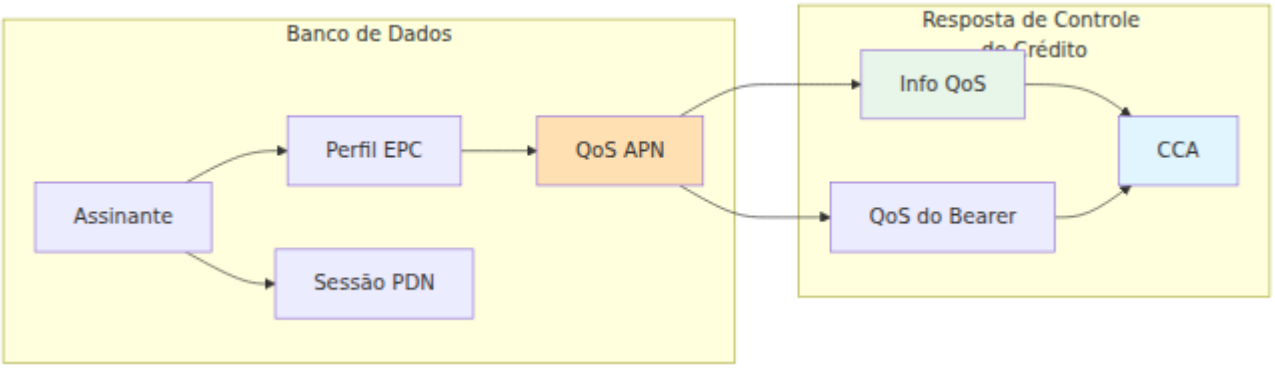
1. **Dados do Repositório**: Pode armazenar XML personalizado em `subscriber_state.sh_repository_data`
2. **Indicação de Serviço**: Filtra dados pelo serviço solicitado (por exemplo, presença, mensagens)
3. **Identidades Públicas**: Retorna todas as identidades públicas IMS para o assinante
4. **Dados de Referência vs Transparente**: Suporta modos de dados de referência e transparente

---

Resposta de Verificação de Identidade do ME (S13 ECA)

A Resposta de Verificação de Identidade do ME é enviada pela função EIR para o MME para validação do IMEI.

Mapeamento da Fonte de Dados



Características Chave:

1. **Combinação Regex**: As regras são expressões regulares testadas contra o IMEISV, depois o IMEI, depois o IMSI se a solicitação carregar um. Dentro de cada etapa, as regras são avaliadas em ordem de `precedência` e a primeira combinação vence. Veja Lógica de Combinação de Equipamentos e Precedência de Regras
2. **Regras baseadas em TAC**: Pode combinar o Código de Alocação de Tipo (primeiros 8 dígitos)
3. **Comportamento Padrão**: `unknown_equipment_behaviour` seleciona lista branca, lista negra, lista cinza ou rejeitar (Resultado Experimental 5422) para equipamentos que não combinam com nenhuma regra. Veja Configuração EIR
4. **Valores de Status do Equipamento**:
   - `0` = LISTA BRANCA (explicitamente permitido)
   - `1` = LISTA NEGRA (roubado/bloqueado)
   - `2` = LISTA CINZA (permitido, mas monitorado)
   - Sem combinação: o `unknown_equipment_behaviour` configurado seleciona um dos acima, ou com `:reject_unknown_equipment` o ECA carrega o Código de Resultado Experimental 5422

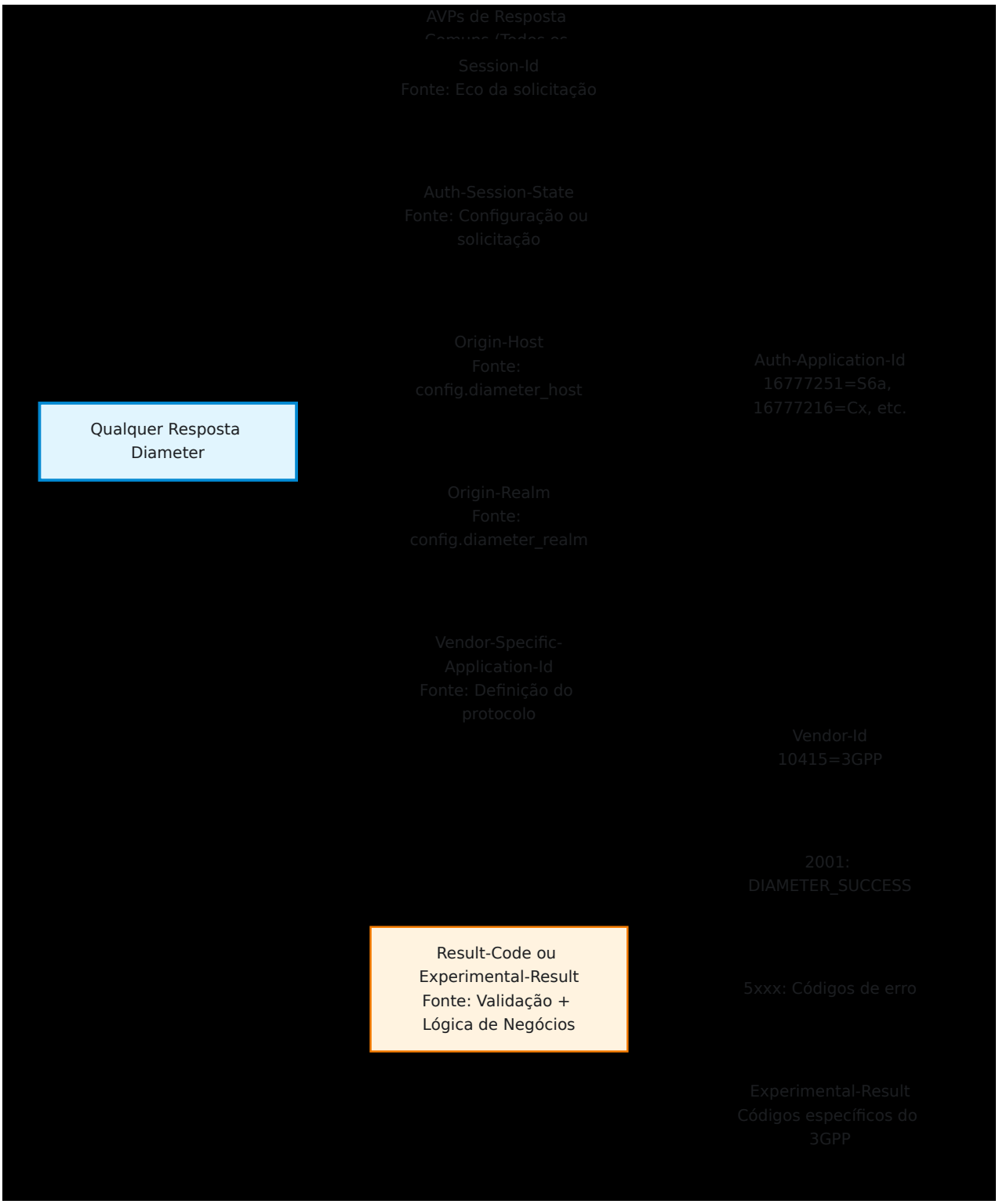
(DIAMETER_ERROR_EQUIPMENT_UNKNOWN) em vez de um Status do Equipamento

Casos de Uso:

- Bloquear dispositivos roubados por IMEI exato
- Bloquear modelos de dispositivos por padrão TAC
- Apenas dispositivos aprovados na lista branca
- Rastrear dispositivos do mercado cinza

Elementos Comuns de Resposta

Todas as respostas Diameter compartilham estes AVPs comuns:

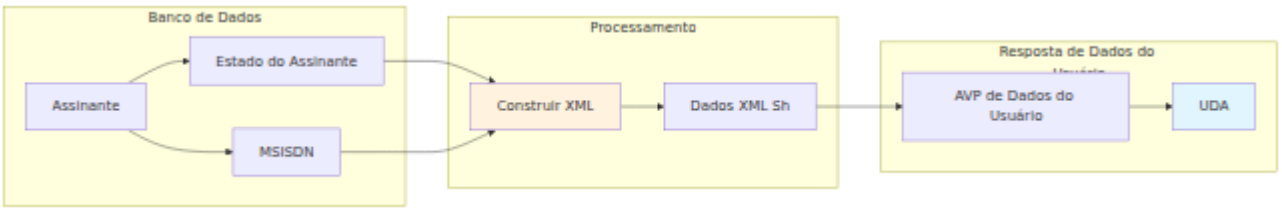


Exemplo de Configuração:

```
config :diameter_ex,
  diameter_host: "hss",
  diameter_realm: "example.com",
  diameter_service_name: "OmniHSS"
```

Resumo do Fluxo de Dados

Pipeline de Processamento de Solicitação



Notas de Implementação

Manipuladores de Protocolo

O sistema implementa manipuladores para os seguintes protocolos Diameter:

- **S6a** - Interface LTE/MME para autenticação e atualizações de localização
- **Cx** - Interface IMS/CSCF para registro IMS e atribuição de servidor
- **Sh** - Interface IMS/AS para recuperação de dados do assinante
- **Gx** - Interface PCRF para controle de políticas e cobrança
- **Rx** - Interface IMS/AF para autorização de mídia
- **S13** - Interface EIR para validação de IMEI
- **SWx** - Interface WiFi/IMS para autenticação de acesso não-3GPP

Modelos de Dados

O esquema do banco de dados inclui as seguintes entidades principais:

- **Assinante** - Registro principal do assinante com IMSI
- **Conjunto de Chaves** - Chaves criptográficas para autenticação
- **Perfil EPC** - Configuração de serviço LTE
- **Perfil APN** - Configuração do ponto de acesso
- **Perfil IMS** - Configuração de serviço IMS com templates IFC
- **Perfil de Roaming** - Regras e restrições de roaming
- **Estado do Assinante** - Rastreamento dinâmico de sessão e estado

- **Sessão PDN** - Rastreamento de sessão de bearer ativa
- **IP Estático** - Atribuições de endereço IP estático
- **Regra EIR** - Regras de identidade de equipamento (regex contra IMEISV, IMEI ou IMSI, avaliadas em ordem de precedência)

← Voltar ao Guia de Operações | Referência da API → | Fluxos de Protocolo →

Guia de Métricas e Monitoramento do OmniHSS

← Voltar ao Guia de Operações

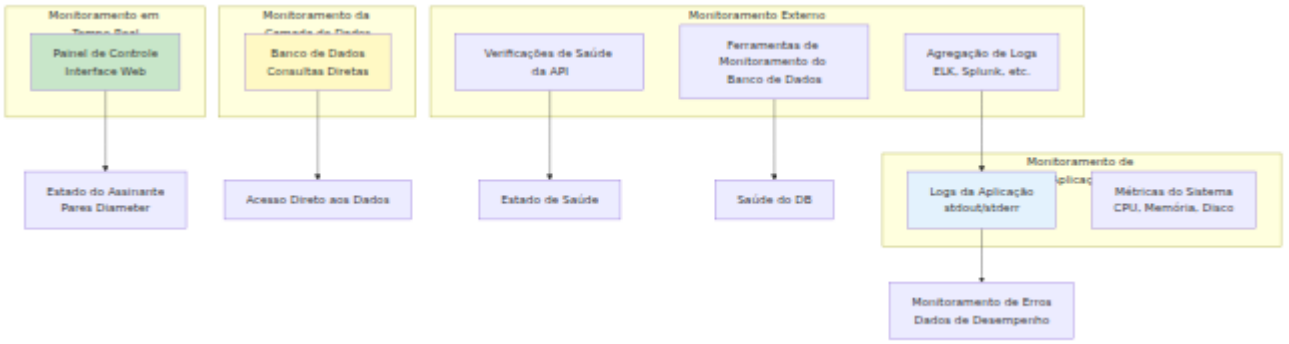
Índice

- Visão Geral do Monitoramento
- Monitoramento do Painel de Controle
- Monitoramento do Banco de Dados
- Monitoramento de Logs
- Integração de Monitoramento Externo
- Indicadores-Chave de Desempenho
- Estratégias de Alerta

Visão Geral do Monitoramento

OmniHSS fornece vários mecanismos para monitorar a saúde do sistema, desempenho e atividade dos assinantes. A equipe de operações deve usar uma combinação dessas ferramentas para total visibilidade.

Camadas de Monitoramento



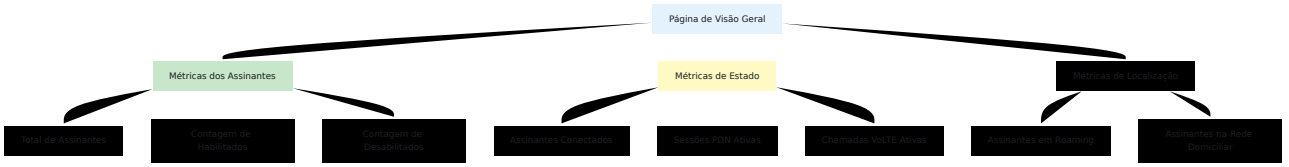
Monitoramento do Painel de Controle

O Painel de Controle fornece a interface principal de monitoramento em tempo real.

Monitoramento da Página de Visão Geral

URL: `https://[hostname]:7443/overview`

Principais Métricas Disponíveis



Estados dos Assinantes Monitorados

Estado	Indicador	O Que Significa
Inativo	Sem informações de localização	Assinante desligado ou fora de cobertura
Conectado	MME presente	Assinante registrado na rede
PDN Ativo	Contagem de sessões PDN > 0	Conexão de dados ativa
Registrado IMS	S-CSCF atribuído	Serviços de voz prontos
Em Chamada	Contagem de chamadas ativas > 0	Chamada VoLTE em andamento

Extraindo Métricas da Visão Geral

O Painel de Controle não exporta métricas diretamente. Você pode:

1. **Contar linhas visíveis** para total de assinantes
2. **Procurar por marcas de verificação verdes** para contar assinantes habilitados
3. **Rever detalhes expandidos** para informações de estado
4. **Anotar timestamps da última visualização** para responsividade

Monitoramento da Página Diameter

URL: `https://[hostname]:7443/diameter`

Principais Métricas



Monitoramento de Pares Críticos

Identifique pares críticos e monitore seu status:

Tipo de Par	Criticidade	Impacto se Desconectado
MME	Alto	Nenhum novo anexo LTE
P-GW	Alto	Sem sessões de dados
S-CSCF	Alto	Sem registros IMS
P-CSCF	Alto	Sem chamadas VoLTE
I-CSCF	Médio	Problemas de roteamento IMS
AS	Baixo-Médio	Serviço específico indisponível

Monitoramento da Página da Aplicação

URL: `https://[hostname]:7443/application`

Principais Métricas

Métrica	Descrição	Faixa Normal	Limite de Ação
Contagem de Processos	Processos Erlang ativos	Varia conforme a carga	> 90% do limite
Uso de Memória	Memória total consumida	< 80%	> 90%
Tempo de Atividade	Tempo desde a última reinicialização	N/A	Monitorar para estabilidade

---

Monitoramento do Banco de Dados

Consultas Diretas ao Banco de Dados

Conecte-se ao Banco de Dados SQL para extrair métricas detalhadas:

Contagens de Assinantes

Consulte o banco de dados para recuperar:

- Contagem total de todos os assinantes
- Contagem de assinantes habilitados
- Contagem de assinantes habilitados para IMS

Estatísticas de Sessão

Consulte o banco de dados para recuperar:

- Contagem de sessões PDN ativas
- Contagem de chamadas VoLTE ativas
- Distribuição de sessões PDN por perfil APN

Estatísticas de Localização

Consulte o banco de dados para recuperar:

- Contagem de assinantes agrupados por rede visitada (combinação MCC-MNC)
- Contagem de assinantes atualmente em roaming (não na PLMN domiciliar 001-001)
- Distribuição de assinantes em diferentes redes visitadas

Atividade Recente

Consulte o banco de dados para recuperar:

- Contagem de assinantes vistos na última hora
- Distribuição de assinantes por MME atendente

- Análise de timestamps da última atividade do assinante

Monitoramento da Saúde do Banco de Dados

Monitore a saúde do banco de dados consultando:

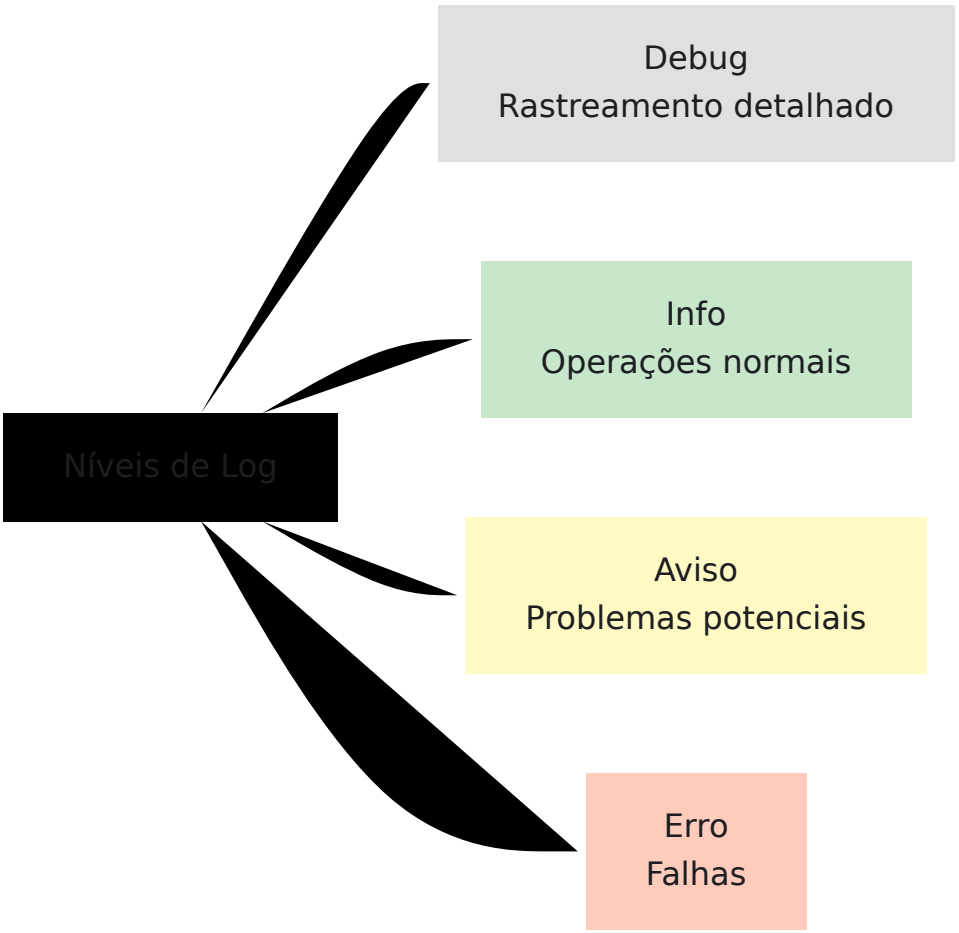
- Tamanho total do banco de dados e tendências de crescimento
- Tamanhos de tabelas individuais e contagens de linhas
- Contagem atual de conexões ao banco de dados
- Desempenho de consultas e uso de recursos

Monitoramento de Logs

Saída de Logs

OmniHSS gera logs para **stdout/stderr**, que devem ser capturados pelo seu gerenciador de processos.

Níveis de Log



Padrões de Log Chave a Monitorar

Eventos de Par Diameter:

```
[info] Par Diameter conectado: mme01.epc.example.com
[warn] Par Diameter desconectado: pgw01.epc.example.com
[error] Falha na conexão do par Diameter: timeout
```

Eventos de Banco de Dados:

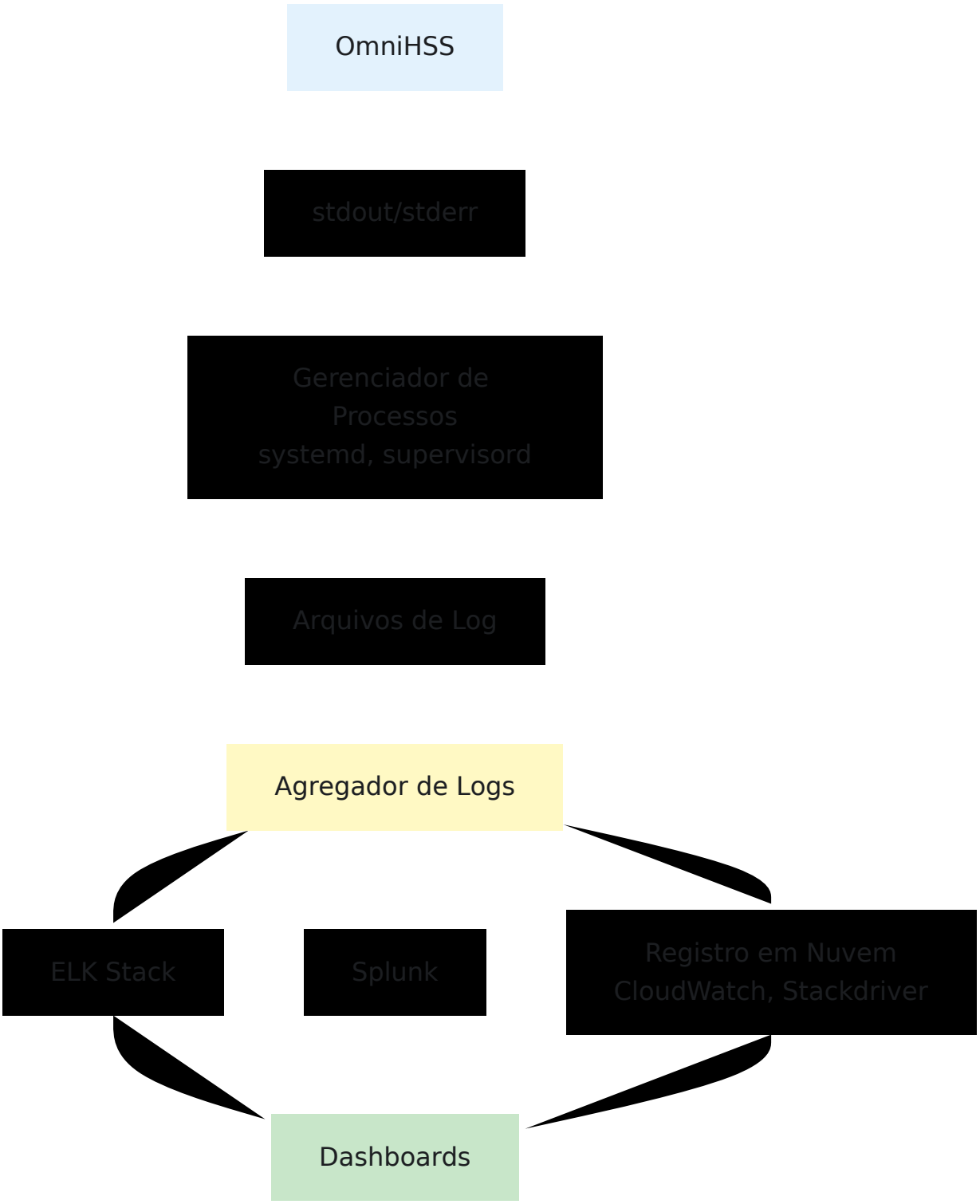
```
[info] Conexão com o banco de dados estabelecida
[error] Conexão com o banco de dados perdida: timeout
[error] Consulta ao banco de dados falhou: deadlock detectado
```

Eventos de Autenticação:

```
[info] Autenticação bem-sucedida: IMSI 001001123456789
[warn] Falha na autenticação: IMSI 001001123456789, vetor inválido
[error] Roaming negado: IMSI 001001123456789, MCC 310 MNC 410
```

Agregação de Logs

Para implantações em produção, implemente agregação de logs:


OmniHSS
stdout/stderr
Gerenciador de Processos
systemd, supervisord
Arquivos de Log
Agregador de Logs
ELK Stack
Splunk
Registro em Nuvem
CloudWatch, Stackdriver
Dashboards

Integração de Monitoramento Externo

Endpoint de Verificação de Saúde

Verificação de Saúde da API: `GET /api/status`

```
curl -k https://hss.example.com:8443/api/status
```

Resposta Esperada:

```
{"status": "ok"}
```

Status HTTP: 200 OK

Integração com Ferramentas de Monitoramento

Exemplo Nagios/Icinga

```
#!/bin/bash
# check_omnihss.sh

API_URL="https://hss.example.com:8443/api/status"

response=$(curl -k -s -o /dev/null -w "%{http_code}" "$API_URL" --
max-time 5)

if [ "$response" = "200" ]; then
    echo "OK - API OmniHSS respondendo"
    exit 0
else
    echo "CRÍTICO - API OmniHSS não respondendo (HTTP $response)"
    exit 2
fi
```

Integração com Prometheus

OmniHSS expõe um endpoint nativo de métricas do Prometheus. Coleta `http://<host>:9568/metrics`. A porta é definida por `prometheus_metrics_port` e o padrão é `9568`.

O endpoint inclui métricas de tempo de execução BEAM, métricas de pool de banco de dados, status de pares Diameter, contadores de mensagens Diameter por aplicação e o contador Cx abaixo.

Contador de Abandono de Autenticação S-CSCF

Métrica: `hss_cx_auth_abandon_total` **Tipo**: Contador **Descrição**: Conta SARs Cx que encerram um desafio IMS-AKA sem um registro bem-sucedido. Este é um desafio abandonado ou falhado. O S-CSCF envia um SAR `AUTHENTICATION_TIMEOUT` quando seu `auth_vector_timeout` expira em um desafio abandonado, ou um SAR `AUTHENTICATION_FAILURE` em uma resposta ruim. Veja Estado de Atribuição S-CSCF. **Rótulos**:

- `sar_type` - O Tipo de Atribuição do Servidor: `authentication_timeout` ou `authentication_failure`.
- `outcome` - `name_cleared` quando o nome do S-CSCF foi liberado, ou `kept_registered` quando um vínculo registrado foi preservado. O HSS libera o nome quando o assinante não estava registrado, e também quando o SAR vem de um S-CSCF diferente do que está registrado. O segundo caso significa que o I-CSCF re-dividiu o REGISTER porque não conseguiu alcançar o nó registrado, então esse nome está obsoleto.
- `origin_host` - O S-CSCF que enviou o SAR.

Consultas de exemplo:

```
# Taxa de desafio abandonado/falhado por segundo
rate(hss_cx_auth_abandon_total[5m])

# Dividir por causa
sum by (sar_type) (rate(hss_cx_auth_abandon_total[5m]))

# Principais S-CSCFs por desafios abandonados
topk(5, sum by (origin_host)
(rate(hss_cx_auth_abandon_total[5m])))
```

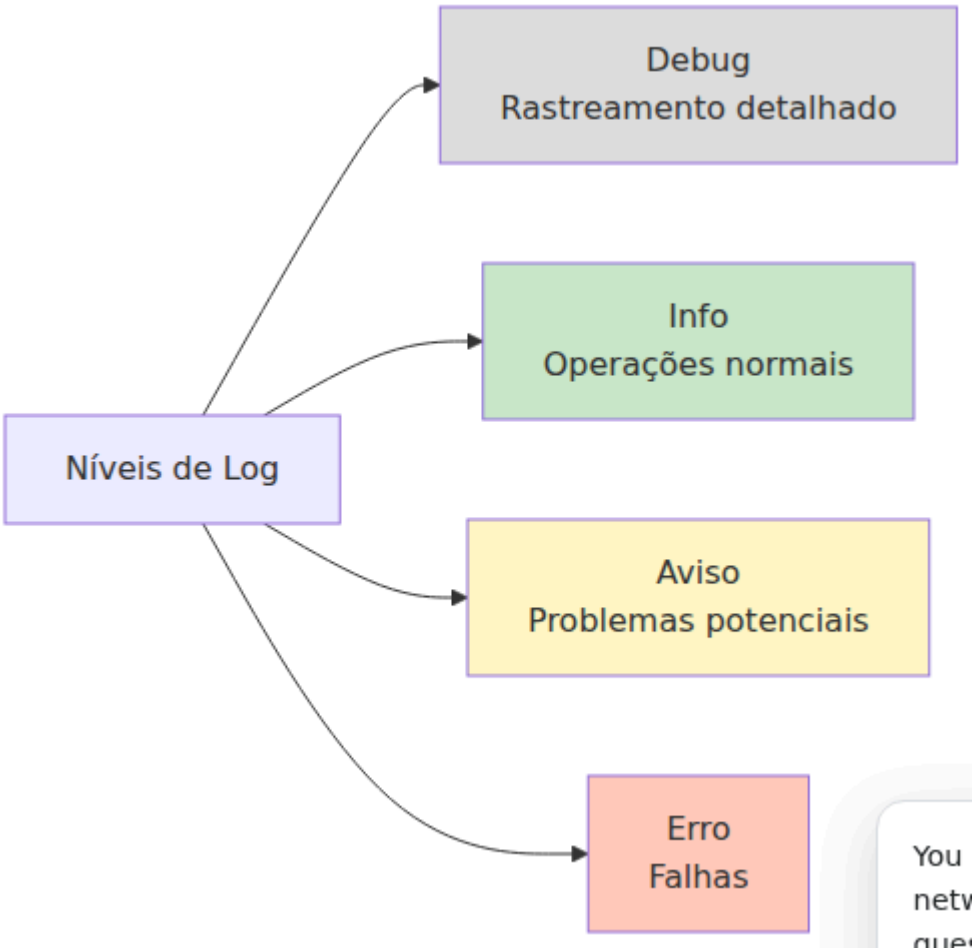
Uma alta taxa em um `origin_host` pode indicar um S-CSCF que está falhando em desafios, ou assinantes que abandonam o registro nesse nó.

Integração SNMP

Para monitoramento baseado em SNMP, scripts de extensão SNMP personalizados podem consultar o banco de dados ou API para métricas e retornar valores via OIDs SNMP.

Indicadores-Chave de Desempenho

KPIs Operacionais



Limites Recomendados para KPIs

KPI	Alvo	Aviso	Crítico
Tempo de Atividade do Sistema	99.99%	< 99.95%	< 99.9%
Tempo de Atividade do Par Diameter	99.9%	< 99.5%	< 99%
Taxa de Sucesso de Autenticação	> 99%	< 99%	< 95%
Tempo de Resposta Diameter	< 100ms	> 200ms	> 500ms
Tempo de Consulta ao Banco de Dados	< 50ms	> 100ms	> 500ms
Taxa de Erro	< 0.1%	> 0.5%	> 1%

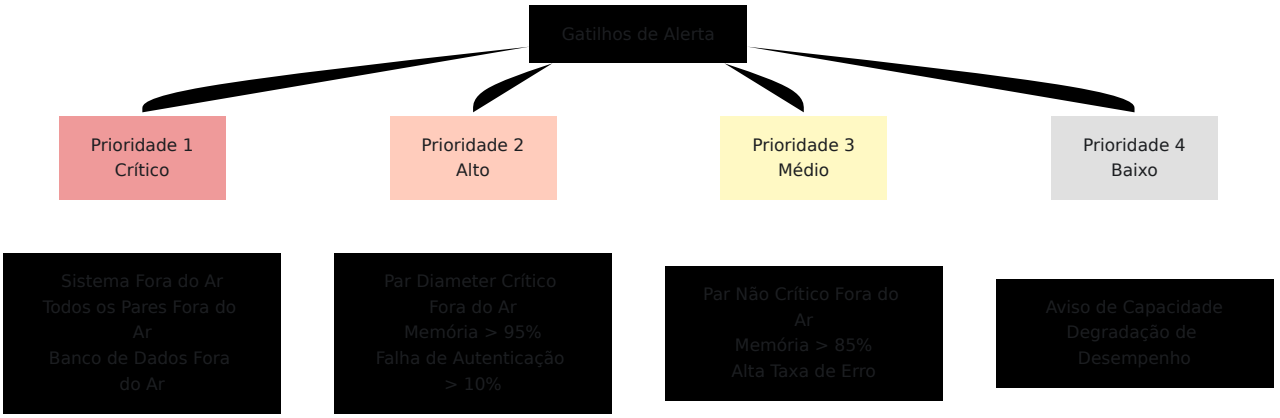
KPIs de Capacidade

Métrica	Monitorar	Planejar Ação em
Total de Assinantes	Contagem atual	80% da capacidade esperada
Sessões PDN Concorrentes	Sessões ativas	70% do máximo esperado
Tamanho do Banco de Dados	MB usados	80% do armazenamento alocado
Conexões ao Banco de Dados	Conexões ativas	80% do tamanho do pool

---

Estratégias de Alerta

Prioridades de Alerta



Definições de Alerta

Alertas Críticos (P1)

Sistema Indisponível:

- Falha na verificação de saúde da API
- Painel de Controle inacessível
- Falha na conexão com o banco de dados
- Ação: Investigação imediata e escalonamento

Todos os Pares Diameter Desconectados:

- Zero pares conectados
- Ação: Verificar rede, reiniciar se necessário

Banco de Dados Fora do Ar:

- Não é possível conectar ao Banco de Dados SQL
- Ação: Investigar servidor de banco de dados, reiniciar se necessário

Alertas de Alta Prioridade (P2)

Par Diameter Crítico Fora do Ar:

- MME primário desconectado
- P-GW primário desconectado
- S-CSCF primário desconectado
- Ação: Investigar conectividade do par em 15 minutos

Uso Elevado de Memória:

- Memória > 95%
- Ação: Investigar vazamento de memória, planejar reinicialização

Alta Taxa de Falha de Autenticação:

- > 10% das solicitações de autenticação falham
- Ação: Verificar provisionamento de assinantes, investigar causa

Alertas de Média Prioridade (P3)

Par Não Crítico Fora do Ar:

- Par secundário desconectado
- Servidor de Aplicação desconectado
- Ação: Investigar em 1 hora

Uso Elevado de Memória:

- Memória > 85%
- Ação: Monitorar tendência, planejar atualização de capacidade

Taxa Elevada de Erro:

- Taxa de erro > 1%
- Ação: Revisar logs, identificar causa raiz

Alertas de Baixa Prioridade (P4)

Aviso de Capacidade:

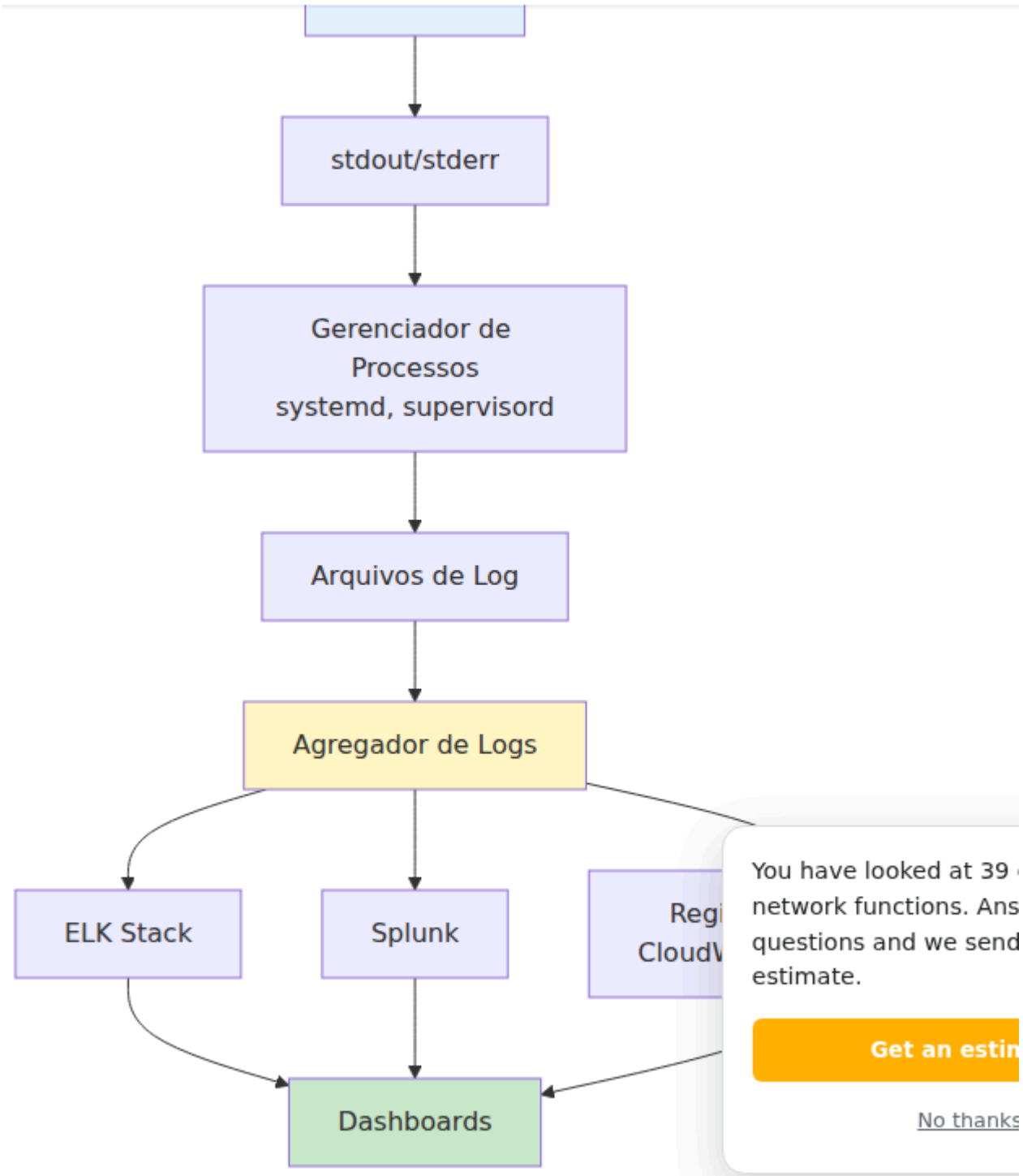
- Assinantes > 80% da capacidade
- Banco de Dados > 80% do armazenamento alocado

- Ação: Planejar expansão de capacidade

Degradação de Desempenho:

- Tempos de resposta elevados, mas aceitáveis
- Ação: Monitorar e otimizar consultas

Canais de Notificação de Alerta



You have looked at 39
network functions. Ans
questions and we send
estimate.
Get an estim
No thanks

Checklist de Monitoramento

Verificações Diárias

- [ ] Revisar Visão Geral do Painel de Controle - contagens de assinantes normais
- [ ] Revisar página Diameter - todos os pares críticos conectados
- [ ] Revisar página da Aplicação - memória e processos dentro dos limites
- [ ] Verificar logs de erro - sem erros críticos nas últimas 24 horas
- [ ] Verificar se o backup foi concluído com sucesso

Verificações Semanais

- [ ] Revisar tendências de capacidade - crescimento de assinantes
- [ ] Revisar tendências de desempenho - tempos de resposta
- [ ] Revisar tamanho do banco de dados - taxa de crescimento aceitável
- [ ] Revisar taxas de erro - identificar padrões
- [ ] Testar notificações de alerta - garantir funcionamento

Verificações Mensais

- [ ] Revisão de planejamento de capacidade - projetar 6 meses à frente
- [ ] Revisão de otimização de desempenho - identificar consultas lentas
- [ ] Revisão de segurança - expiração de certificados, vulnerabilidades
- [ ] Revisão de documentação - atualizar runbooks
- [ ] Teste de recuperação de desastres - verificar se os backups são restaurados corretamente

---

← Voltar ao Guia de Operações | Próximo: Multi-Funcionalidades →

Recursos Multi-MSISDN e Multi-IMSI do OmniHSS

← Voltar ao Guia de Operações

Índice

- Visão Geral
- Multi-MSISDN: Vários Números de Telefone
- SIM Multi-IMSI: Múltiplas Identidades de Rede
- Cenários Combinados
- Resolução de MSISDN Através de Interfaces
- Identidade Pública de Usuário Compartilhada
- Exemplos de Configuração
- Procedimentos Operacionais

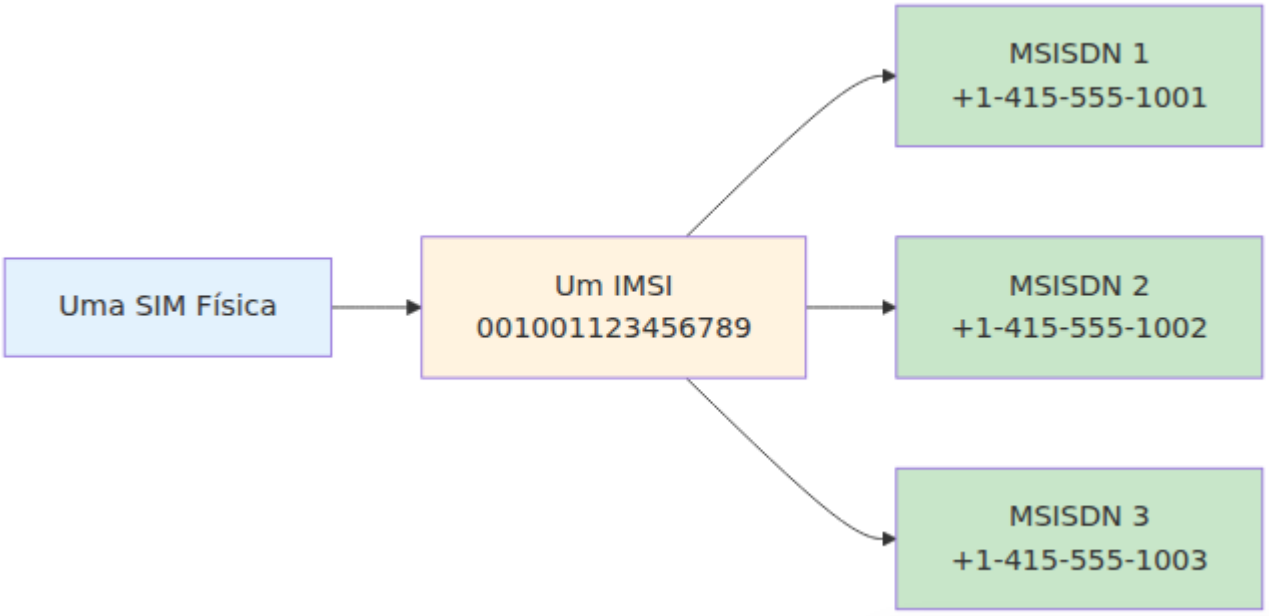
Visão Geral

O OmniHSS suporta capacidades avançadas de provisionamento que permitem configurações de serviço flexíveis:

Suporte a Multi-MSISDN

Um IMSI → Vários Números de Telefone

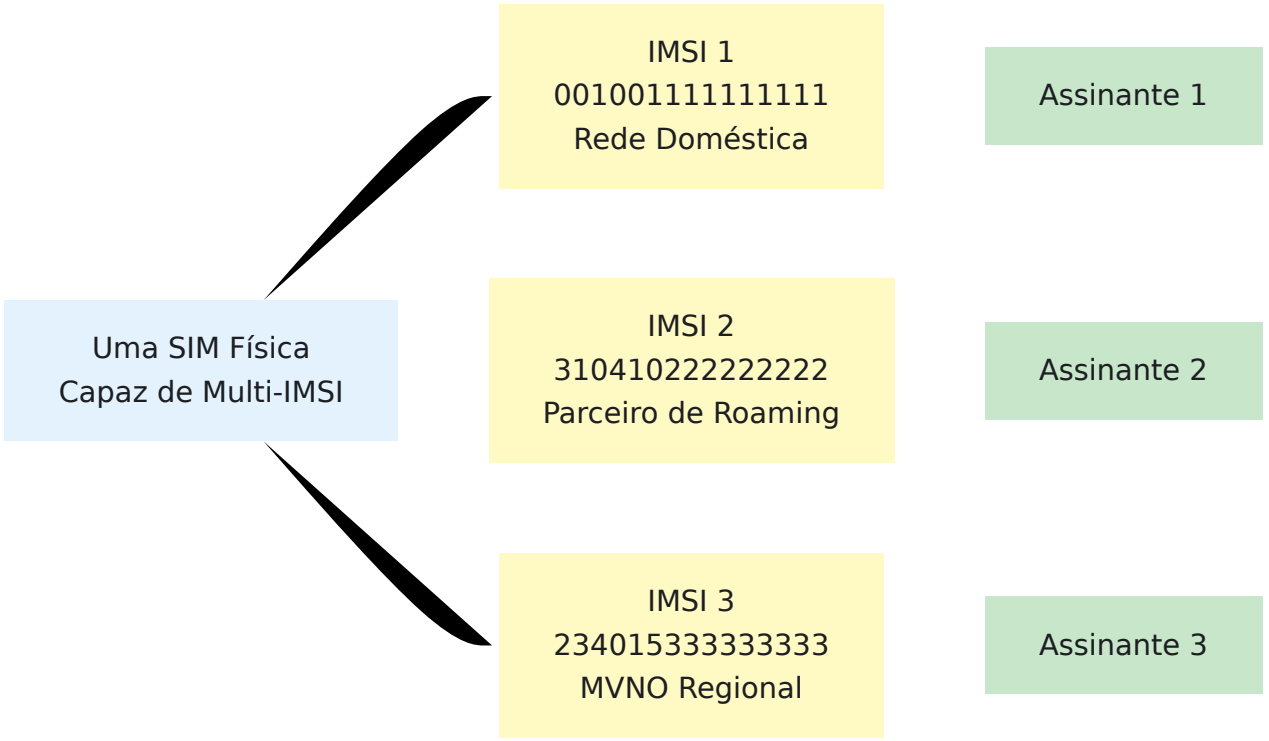
Um único assinante (identificado por um IMSI) pode ter vários MSISDNs (números de telefone) atribuídos. Todos os números tocam no mesmo dispositivo e compartilham os mesmos perfis de serviço.



Suporte a SIM Multi-IMSI

Uma SIM → Vários IMSIs

Um único cartão SIM físico pode conter vários IMSIs, permitindo que o dispositivo se conecte a diferentes redes usando diferentes identidades de rede. Isso é útil para roaming internacional e cenários de MVNO.



Multi-MSISDN: Vários Números de Telefone

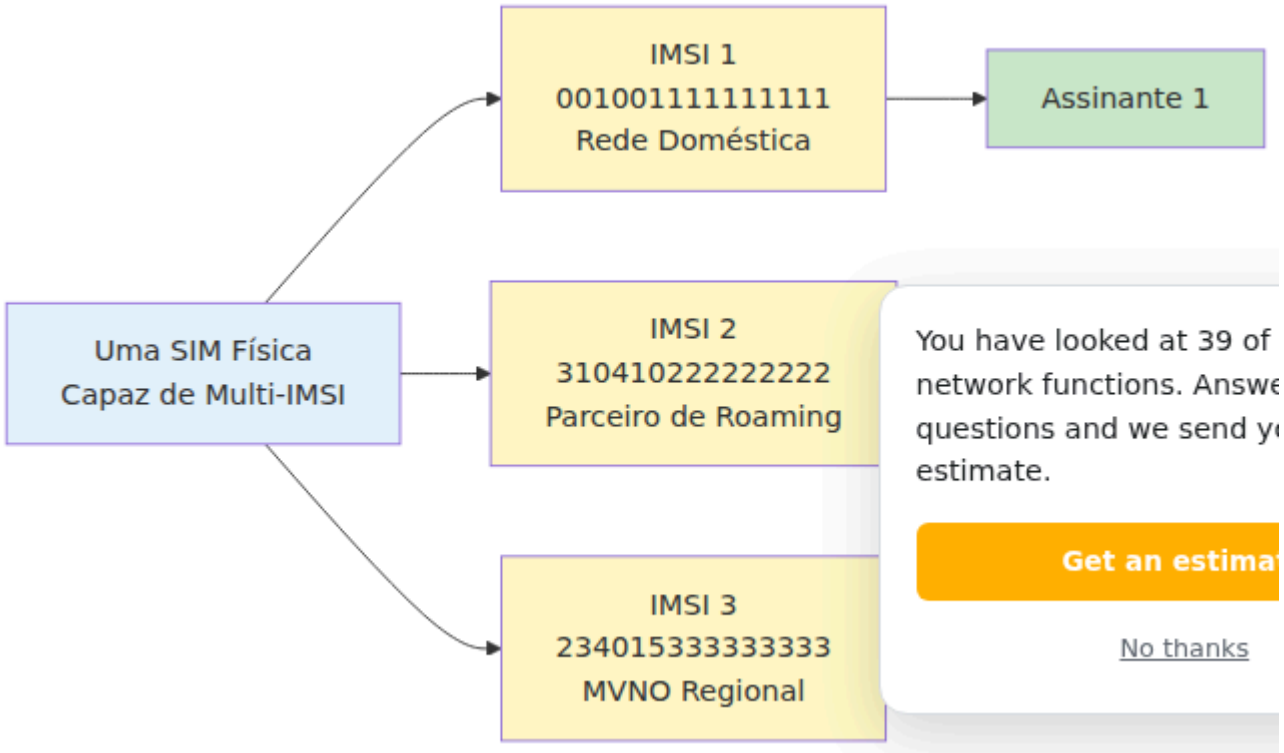
Como Funciona

Um registro de assinante no HSS possui vários MSISDNs vinculados através de uma tabela de junção. Quando o assinante se registra no IMS, todos os MSISDNs são incluídos no perfil IMS, permitindo que chamadas recebidas para qualquer número cheguem ao dispositivo.

Características Principais

- **Um IMSI** - O assinante possui um único IMSI vinculado ao seu cartão SIM
- **Vários MSISDNs** - O assinante pode ter vários números de telefone
- **Integração IMS** - Todos os MSISDNs estão registrados no IMS
- **Serviço Compartilhado** - Todos os números compartilham os mesmos perfis de serviço (EPC, IMS, Roaming)

Modelo de Dados

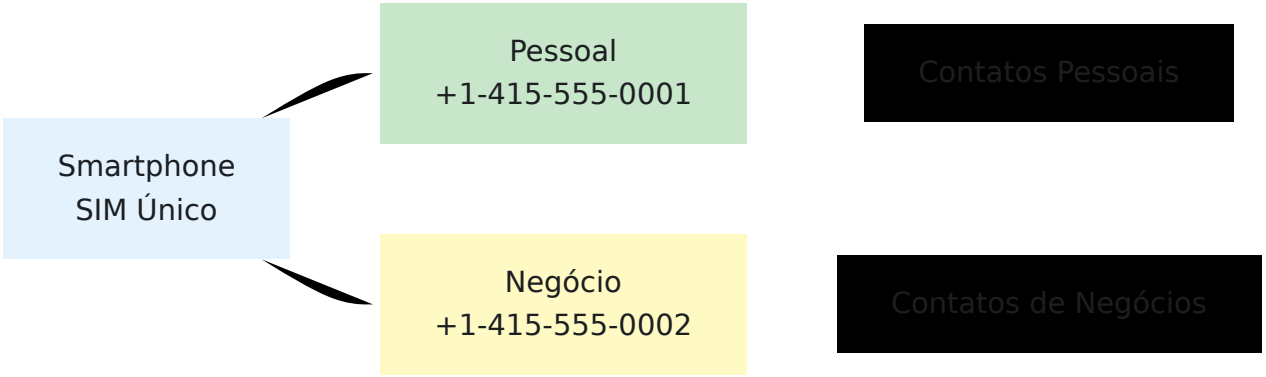


Importante: Um MSISDN pode ser atribuído a UM único assinante por vez. No entanto, um assinante pode ter MUITOS MSISDNs.

Casos de Uso

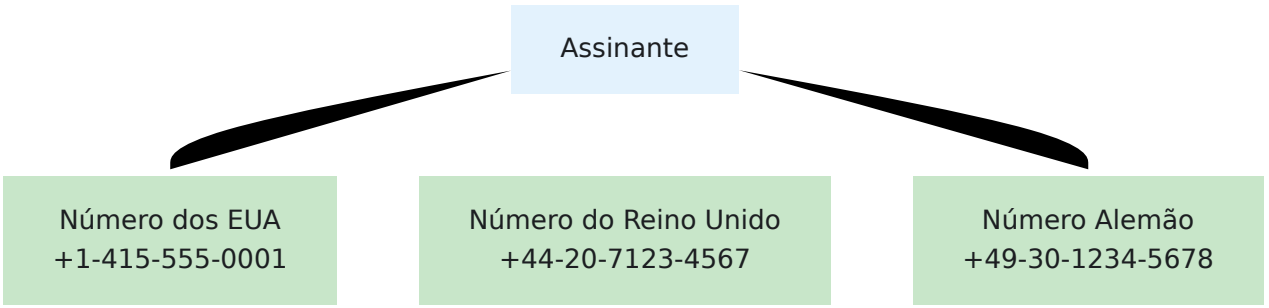
1. Linhas de Negócios e Pessoais

Um assinante possui números de telefone de negócios e pessoais no mesmo dispositivo:



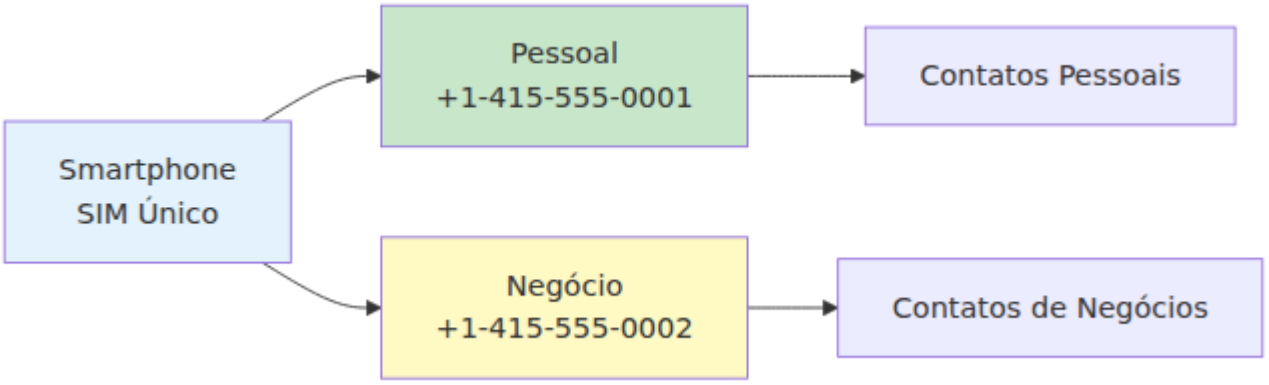
2. Números Internacionais

Um assinante que viaja com frequência possui números em vários países:



3. Planos Familiares

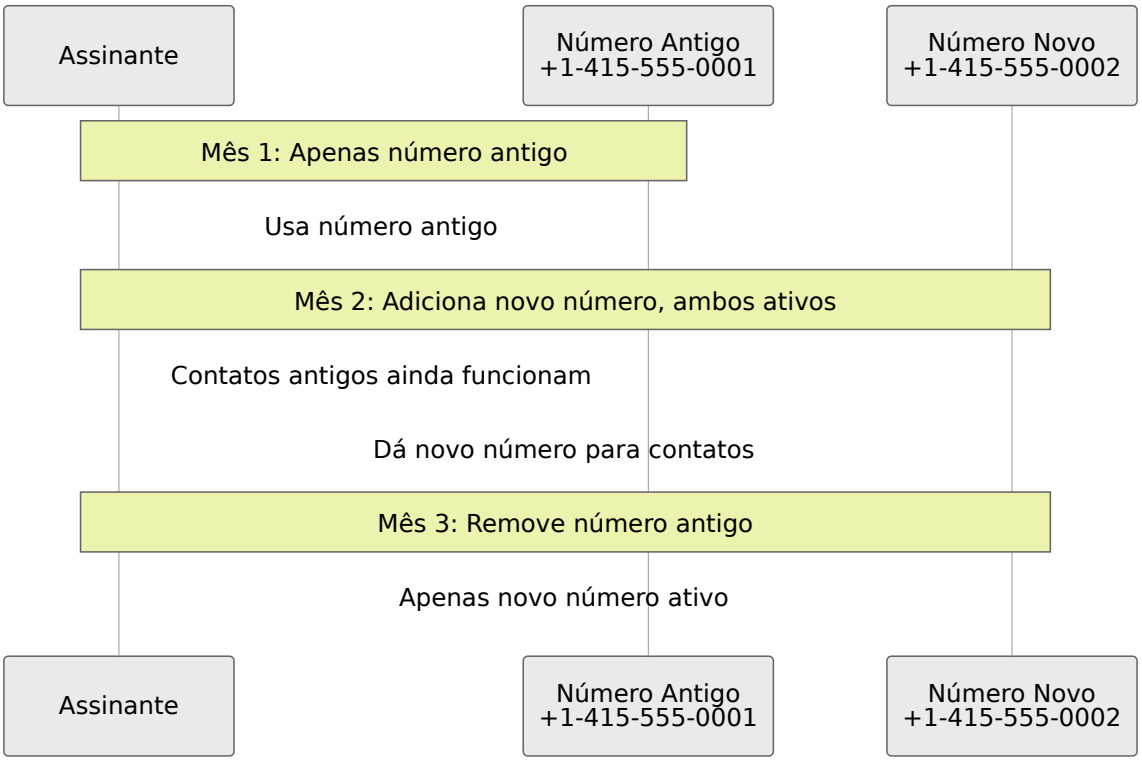
Um dos pais gerencia vários números de membros da família:



Nota: No OmniHSS, isso exigiria múltiplos assinantes (um por SIM/IMSI), cada um potencialmente tendo múltiplos MSISDNs.

4. Portabilidade de Linha Legada

Quando um assinante muda de número, mas deseja manter o número antigo ativo durante a transição:



Configuração

Criando MSISDNs

Os MSISDNs devem ser criados antes de serem atribuídos aos assinantes.

```
# Criar primeiro MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551001"}}'

# Criar segundo MSISDN
curl -k -X POST https://hss.example.com:8443/api/msisdn \
  -H "Content-Type: application/json" \
  -d '{"msisdn": {"msisdn": "14155551002"}}'
```

Atribuindo MSISDNs a Assinantes

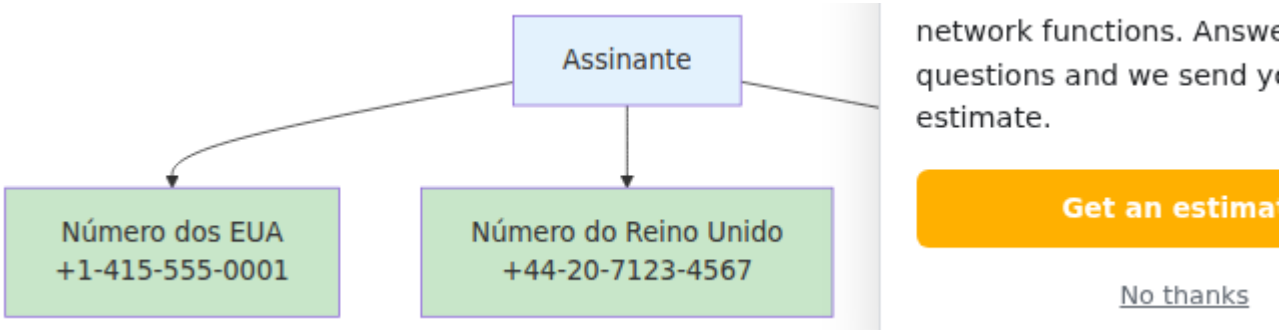
A atribuição é feita através da tabela de junção no banco de dados.

Método de Banco de Dados:

1. Consultar o banco de dados para obter o ID do assinante para o IMSI alvo
2. Consultar o banco de dados para obter os IDs dos MSISDNs para os números de telefone
3. Inserir registros na tabela de junção vinculando subscriber_id a cada msisdn_id

Isso cria a relação muitos-para-muitos entre o assinante e seus números de telefone.

Fluxo de Trabalho de Provisionamento



Verificando Atribuição

Consultar o banco de dados para recuperar o assinante juntamente com todos os MSISDNs vinculados por:

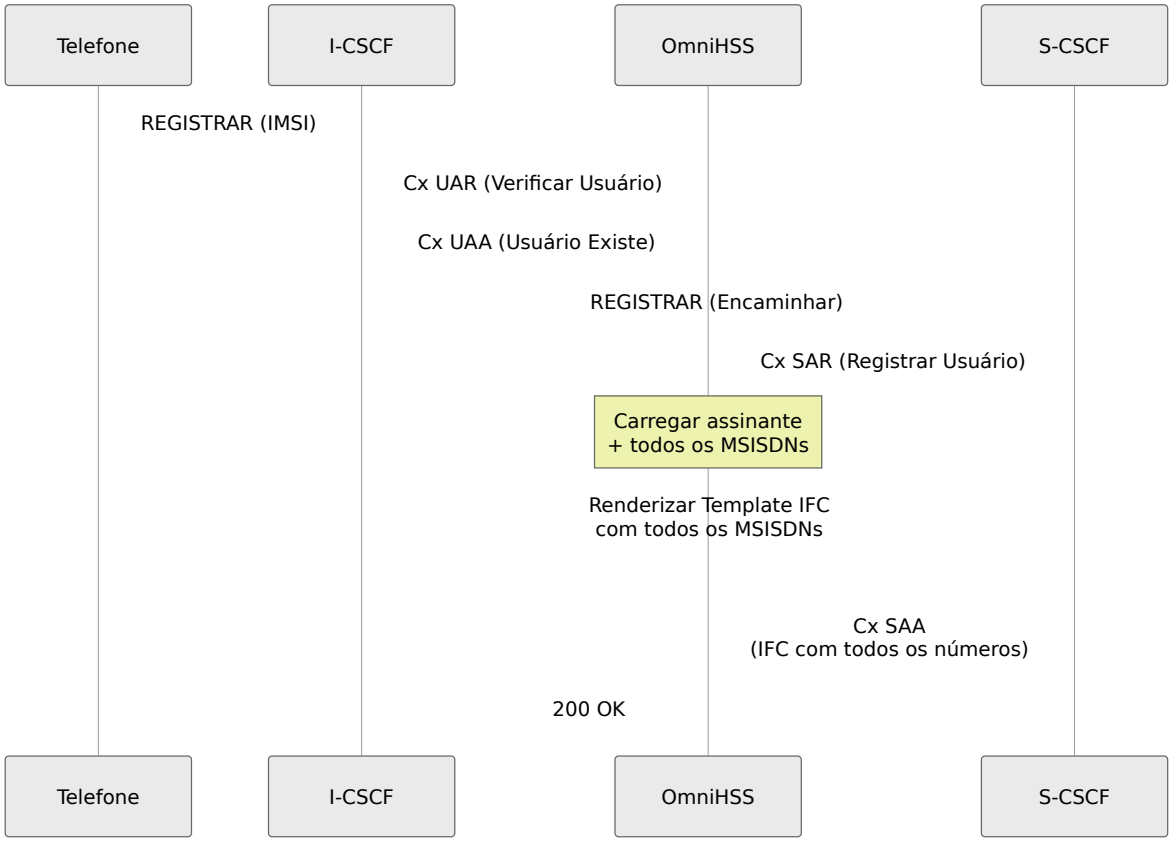
- Juntando a tabela de assinantes com a tabela de junção
- Juntando a tabela de junção com a tabela de msisdn
- Agrupando resultados por assinante para ver todos os números de telefone juntos

Isso mostrará o ID do assinante, IMSI e uma lista de todos os MSISDNs atribuídos.

Integração IMS

Registro IMS

Quando um assinante se registra no IMS, **todos os MSISDNs atribuídos são incluídos** no perfil IMS enviado ao S-CSCF.



Renderização do Template IFC

O template IFC do IMS pode referenciar todos os MSISDNs usando a variável `{{msisdns}}`.

Exemplo de Template IFC:

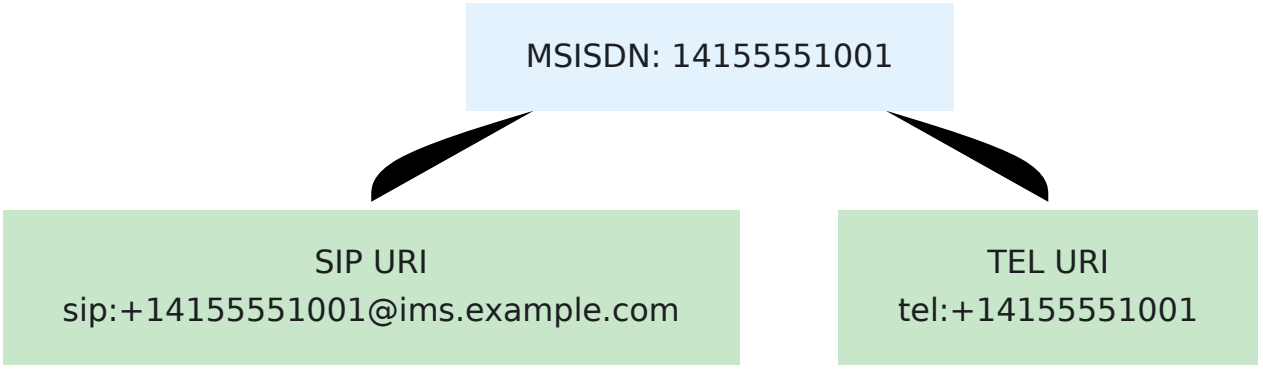
```
<ServiceProfile>
  <PublicIdentity>
    <Identity>sip:
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>
  </PublicIdentity>
  <!-- Repetir para cada MSISDN -->
  <PublicIdentity>
    <Identity>sip:+14155551001@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551001</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>sip:+14155551002@ims.example.com</Identity>
  </PublicIdentity>
  <PublicIdentity>
    <Identity>tel:+14155551002</Identity>
  </PublicIdentity>
  <!-- ... -->
</ServiceProfile>
```

Variável do Template:

- `{{msisdns}}` - Lista de todos os MSISDNs atribuídos ao assinante

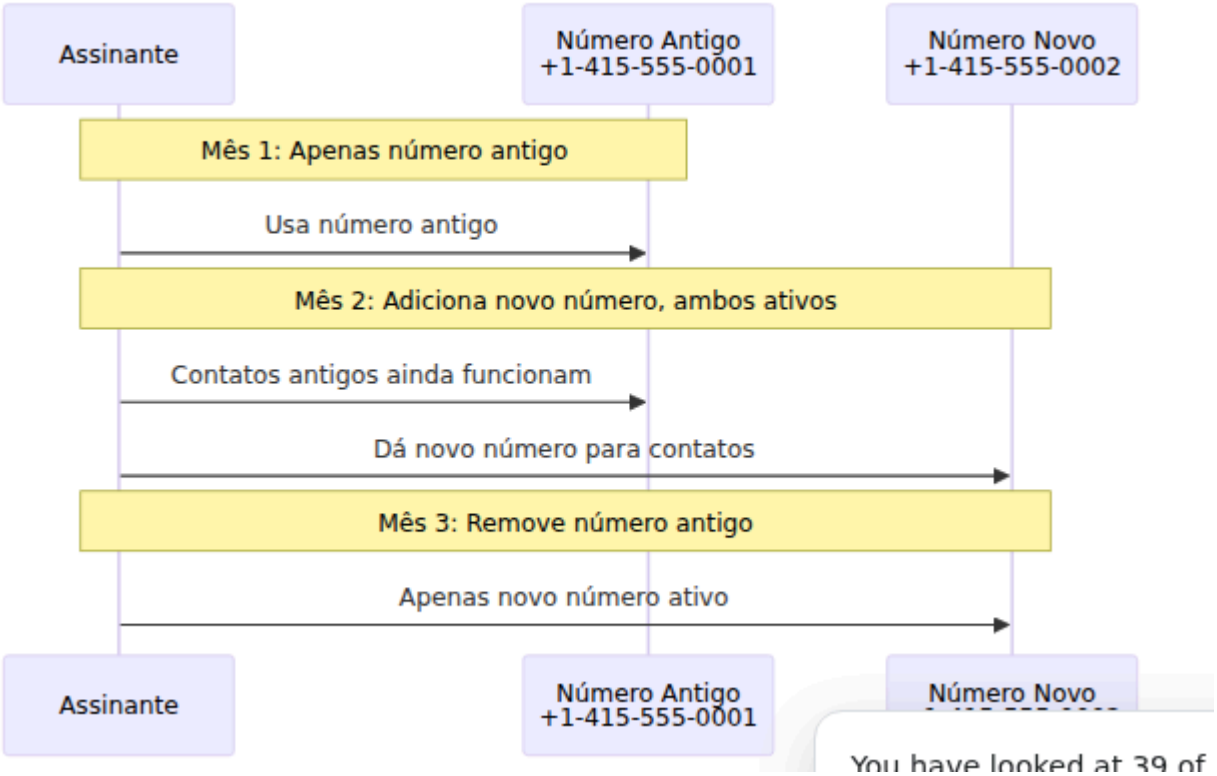
Identidades Públicas

Cada MSISDN normalmente resulta em duas identidades públicas IMS:



Roteamento de Chamadas Inbound

Quando alguém liga para um dos números do assinante, a rede IMS roteia para o SIP URI correto:



Apresentação de Chamadas Outbound

O telefone pode escolher qual número apresentar como ID do chamador para chamadas de saída.

Exemplo SIP INVITE:

```
INVITE sip:+15105551234@ims.example.com SIP/2.0
From: "+14155551002" <sip:+14155551002@ims.example.com>;tag=123
To: <sip:+15105551234@ims.example.com>
P-Asserted-Identity: <sip:+14155551002@ims.example.com>
```

Os cabeçalhos `From` e `P-Asserted-Identity` indicam qual dos números do assinante está sendo usado.

Solução de Problemas Multi-MSISDN

Problema: MSISDN Não Aparece no Registro IMS

Sintomas:

- S-CSCF mostra apenas uma identidade pública
- Chamadas para o segundo número falham

Passos de Solução de Problemas:

1. **Verificar Atribuição de MSISDN no Banco de Dados:**
    - Consultar o banco de dados para recuperar todos os MSISDNs vinculados ao IMSI do assinante
    - Verificar a tabela de junção para garantir que as relações existam
2. **Verificar Template do Perfil IMS:**
    - Verificar se o template inclui a variável `{{msisdns}}`
    - Confirmar que a sintaxe do template é um XML válido
3. **Revisar Logs do HSS:**
    - Procurar mensagens de registro IMS (Cx SAR)
    - Verificar se todos os MSISDNs estão incluídos na resposta
4. **Testar Registro IMS:**
    - Acionar re-registro no telefone
    - Verificar logs do S-CSCF para identidades públicas registradas

Problema: Não é Possível Atribuir MSISDN ao Assinante

Sintomas:

- Falha na inserção do banco de dados
- Erro: "Entrada duplicada" ou "Restrição de chave estrangeira"

Causas Possíveis:

1. **MSISDN Já Atribuído:**
    - Consultar o banco de dados para verificar se o MSISDN já está vinculado a outro assinante

- **Solução:** Remover a atribuição existente primeiro, depois criar a nova atribuição

2. **MSISDN Não Existe:**
   - Consultar o banco de dados para verificar se o registro do MSISDN existe
   - **Solução:** Criar o registro do MSISDN primeiro via API ou inserção no banco de dados

Problema: Chamadas para Um Número Funcionam, Outro Não

Sintomas:

- Chamadas para o número principal funcionam
- Chamadas para o número secundário falham ou roteiam incorretamente

Passos de Solução de Problemas:

1. **Verificar Ambos os Números no Registro IMS:**
   - Verificar identidades públicas registradas no S-CSCF
   - Confirmar que ambos os SIP URIs estão presentes
2. **Verificar Regras de Roteamento IMS:**
   - Verificar se as regras de roteamento do template IFC se aplicam a todas as identidades
   - Verificar se um número específico precisa de roteamento especial
3. **Testar Ambos os Números:**

```
# Testar a partir do cliente SIP
sip:+14155551001@ims.example.com  # Deve funcionar
sip:+14155551002@ims.example.com  # Também deve funcionar
```

Problema: Consulta API por MSISDN Retorna Assinante Errado

Sintomas:

- Consulta API `/api/subscriber/msisdn/:msisdn` retorna assinante inesperado

Verificação:

Consultar o banco de dados para descobrir a qual assinante o MSISDN está atribuído. Isso deve retornar exatamente um assinante. Se retornar múltiplos ou o assinante errado, a tabela de junção tem dados incorretos que precisam ser corrigidos.

Melhores Práticas

Ordem de Provisionamento

1. Criar todos os MSISDNs primeiro
2. Criar assinante
3. Atribuir MSISDNs ao assinante
4. Verificar atribuição antes da ativação

Gestão de MSISDN

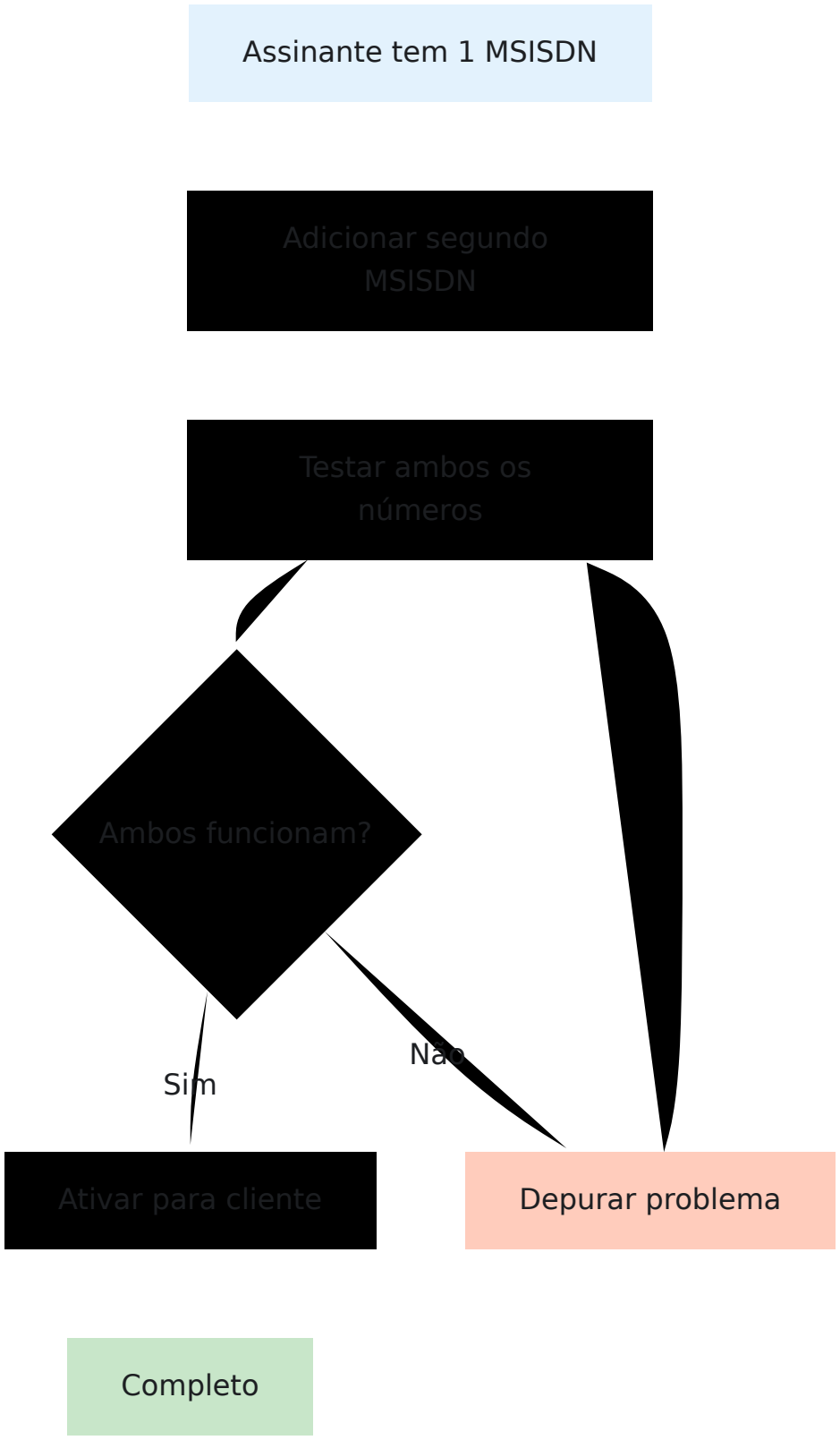
- **Documentar números primários vs secundários** em custom_attributes do assinante
- **Portar números sequencialmente** ao portar para evitar interrupção de serviço
- **Testar todos os números** após o provisionamento antes de entregar ao cliente

Configuração IMS

- Garantir que o template IFC lida corretamente com múltiplas identidades públicas
- Testar roteamento inbound para todos os números
- Verificar apresentação de ID do chamador para chamadas de saída

Migração

Ao migrar de um único para multi-MSISDN:


Assinante tem 1 MSISDN
Adicionar segundo MSISDN
Testar ambos os números
Ambos funcionam?
Não
Sim
Ativar para cliente
Depurar problema
Completo

SIM Multi-IMSI: Múltiplas Identidades de Rede

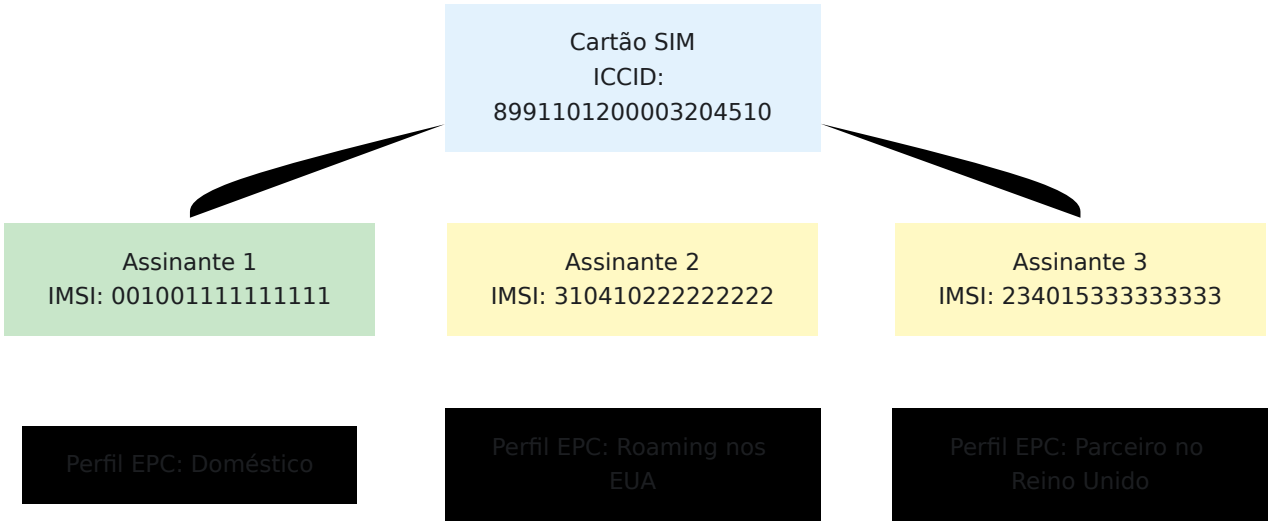
Como Funciona

Um SIM multi-IMSI contém vários perfis completos de assinante, cada um com seu próprio IMSI, chaves e credenciais. O dispositivo pode alternar entre IMSIs para se conectar a diferentes redes, muitas vezes automaticamente com base na localização ou disponibilidade da rede.

Importante: Apenas **um IMSI pode estar ativo a qualquer momento**. Quando um dispositivo muda para um IMSI diferente no mesmo cartão SIM, o HSS automaticamente desregistra o IMSI previamente ativo.

Implementação do OmniHSS

No OmniHSS, cada IMSI em um SIM multi-IMSI é provisionado como um **registro de assinante separado**, mas todos referenciam o **mesmo cartão SIM**:



Casos de Uso

1. Otimização de Roaming Internacional

- IMSI Doméstico: 001-001 (taxas de rede doméstica)

- IMSI de Roaming nos EUA: 310-410 (taxas locais nos EUA)
- IMSI de Roaming na UE: 234-015 (taxas locais na UE)
- Dispositivo alterna IMSI com base na localização

2. Serviço MVNO

- IMSI Primário: rede MVNO (revendedor)
- IMSI de Fallback: rede hospedeira (operadora mãe)
- Failover automático se a cobertura MVNO não estiver disponível

3. IoT/M2M Multi-Rede

- IMSI 1: Operadora primária
- IMSI 2: Operadora de backup para redundância
- IMSI 3: Fallback de emergência/custo baixo
- Dispositivos críticos mantêm conectividade

4. Conformidade Regulamentar

- Diferentes IMSIs para diferentes zonas regulatórias
- Cumprir requisitos locais de residência de dados
- Usar identidade de rede local por jurisdição

Recursos Multi-IMSI

Autenticação Independente

- Cada IMSI tem seu próprio Ki, OPC e conjunto de chaves
- Vetores de autenticação separados por IMSI
- Diferentes credenciais de segurança por rede

Perfis de Serviço Separados

- Diferentes perfis EPC (largura de banda, APNs)
- Diferentes perfis IMS (serviços de voz)
- Diferentes regras de roaming por IMSI

Identidade Física Compartilhada

- Todos os IMSIs referenciam o mesmo SIM (via sim_id)
- Mesmo ICCID em todos os registros de assinante
- Agrupamento lógico via cartão SIM

Seleção de Rede

- O dispositivo ou cartão SIM decide qual IMSI usar
- Com base nas redes disponíveis, localização, política
- O HSS autentica qualquer IMSI que o dispositivo apresentar

Configuração

```
# 1. Criar cartão SIM (capaz de multi-IMSI)
SIM_ID=$(curl -k -X POST https://hss.example.com:8443/api/sim \
  -d '{"sim": {"iccid": "8991101200003204510", "is_esim": false}}'
\
  | jq -r '.data.id')

# 2. Criar conjunto de chaves para IMSI 1 (rede doméstica)
KEYSET1=$(curl -k -X POST https://hss.example.com:8443/api/key_set
\
  -d '{"key_set": {"ki": "0123456789ABCDEF...", "opc":
"FEDCBA9876..."}}' \
  | jq -r '.data.id')

# 3. Criar assinante 1 (IMSI doméstico)
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d "{\"subscriber\": {
    \"imsi\": \"001001111111111\",
    \"sim_id\": \"$SIM_ID\",
    \"key_set_id\": \"$KEYSET1\",
    \"epc_profile_id\": \"e58ed763-928c-4155-bee9-fdbaaadc15f3\"
  }}"

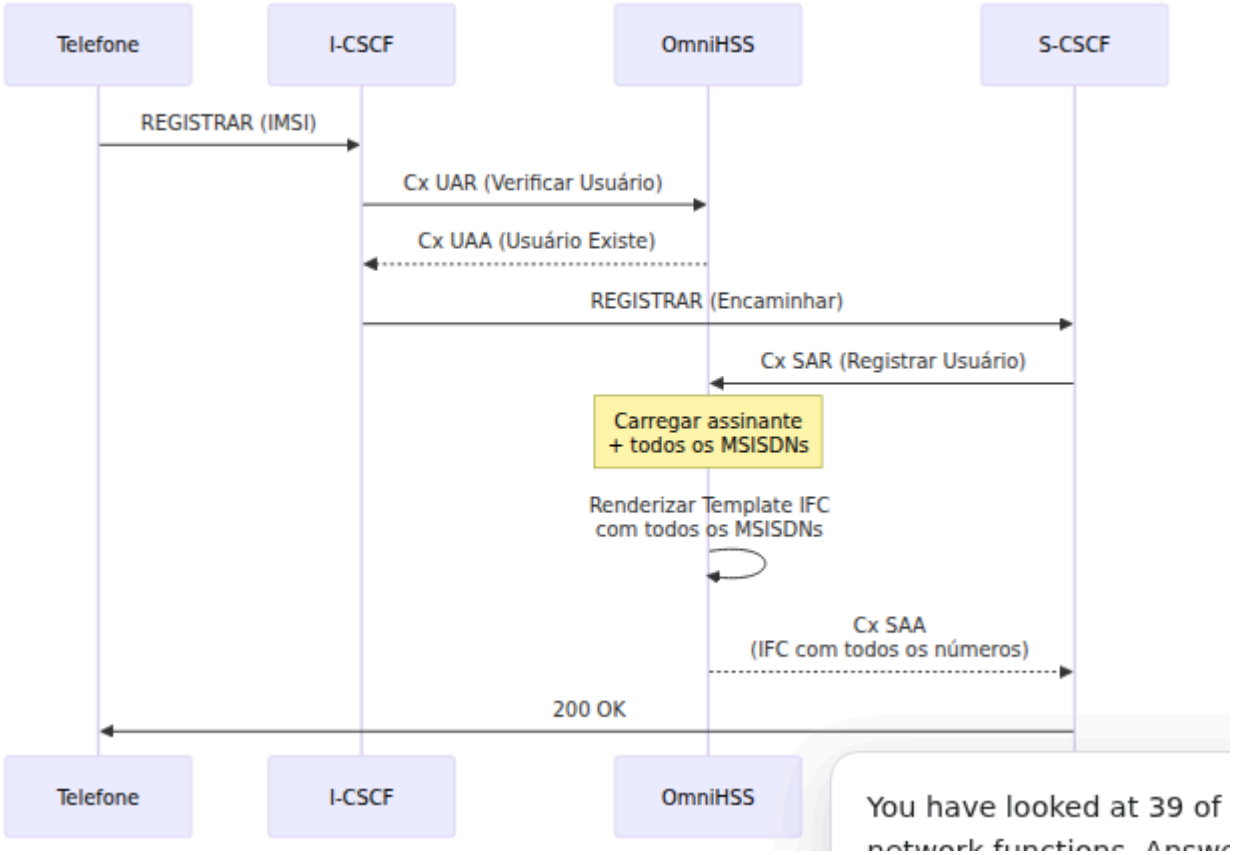
# 4. Criar conjunto de chaves para IMSI 2 (parceiro de roaming)
KEYSET2=$(curl -k -X POST https://hss.example.com:8443/api/key_set
\
  -d '{"key_set": {"ki": "1111111111111111...", "opc":
"2222222222..."}}' \
  | jq -r '.data.id')

# 5. Criar assinante 2 (IMSI de roaming)
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -d "{\"subscriber\": {
    \"imsi\": \"310410222222222\",
    \"sim_id\": \"$SIM_ID\",
    \"key_set_id\": \"$KEYSET2\",
    \"epc_profile_id\": \"f47ac10b-58cc-4372-a567-0e02b2c3d479\"
  }}"

# 6. Repetir para IMSIs adicionais no SIM...
```

Fluxo de Autenticação

Quando um dispositivo multi-IMSI se conecta:



O HSS não precisa saber que é um SIM multi-IMSI. Ele autentica qualquer IMSI que o dispositivo apresentar.

Mudança de IMSI e Desregistro Automático

Quando um SIM multi-IMSI muda de um IMSI para outro, apenas um IMSI pode estar registrado na rede a qualquer momento. O OmniHSS lida automaticamente com isso enviando uma **Solicitação de Cancelamento de Localização (CLR)** para desregistrar o IMSI previamente ativo quando um novo IMSI do mesmo cartão SIM se registra.

Regra de IMSI Único Ativo

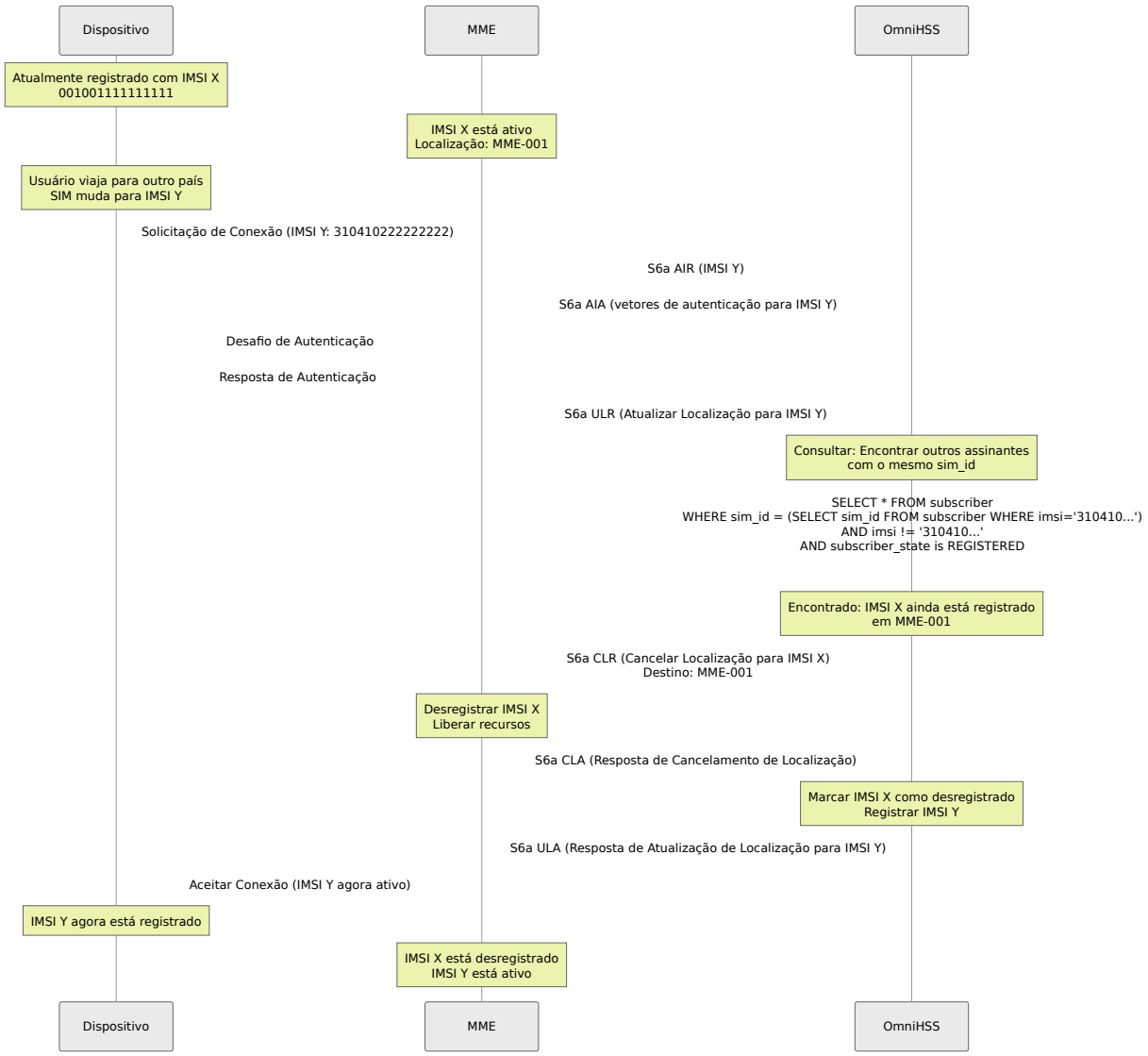
Conceito Chave: Apenas um assinante (IMSI) por cartão SIM pode estar ativo a qualquer momento.

- Se um assinante está registrado em um MME usando **IMSI X**

- E o HSS recebe uma Solicitação de Atualização de Localização para **IMSI Y** (no mesmo SIM que IMSI X)
- O HSS envia automaticamente uma **Solicitação de Cancelamento de Localização** para desregistrar **IMSI X**

Isso garante uma transição limpa entre IMSIs e previne conflitos na rede.

Fluxo de Mudança de IMSI



Por Que Isso É Importante

Integridade da Rede:

- Previne registros duplicados do mesmo SIM físico
- Garante que os recursos da rede sejam liberados corretamente

- Mantém dados precisos de localização do assinante

Precisão de Cobrança:

- Apenas um IMSI é cobrado pelo acesso à rede de cada vez
- Limites de sessão claros entre as trocas de IMSI
- Geração precisa de CDR (Registro Detalhado de Chamadas)

Gerenciamento de Recursos:

- Recursos do MME para o antigo IMSI são liberados
- Contextos PDP e portadores são limpos
- O rastreamento de localização permanece preciso

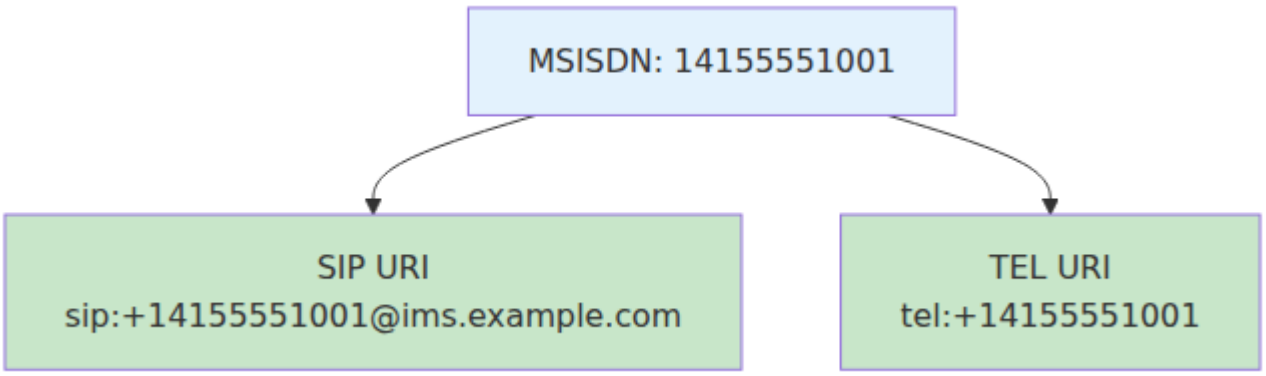
Gatilhos de Troca de IMSI

O dispositivo/SIM decide quando trocar IMSIs com base em:

1. **Disponibilidade da Rede**
    - Rede IMSI de origem não disponível
    - Trocar para IMSI de parceiro de roaming
2. **Seleção Manual**
    - O usuário seleciona manualmente a rede
    - O SIM troca para o IMSI correspondente
3. **Baseado em Políticas**
    - O cartão SIM possui regras internas (por exemplo, preferir IMSI local em certos países)
    - Troca automática com base em MCC/MNC
4. **Otimização de Custos**
    - Trocar para IMSI com tarifas de roaming mais baixas
    - Usar IMSI local para evitar cobranças de roaming

Considerações sobre IMS

O mesmo comportamento de Cancelar Solicitação de Localização se aplica ao registro IMS:



Impacto Operacional

Para a Equipe de Operações:

1. **Assinante aparece offline:** Quando o IMSI troca, o antigo IMSI aparecerá como "desregistrado" no HSS. Isso é um comportamento normal.

2. **Dois registros de assinante para um SIM:** SIMs Multi-IMSI terão vários registros de assinante compartilhando o mesmo `sim_id`. Apenas um estará no estado "registrado" por vez.

3. **Rastreamento de localização:** A tabela `subscriber_state` rastreia com qual MME/SGSN cada IMSI está registrado. Quando o IMSI troca, a localização antiga é limpa.

4. **Solução de Problemas:** Se um dispositivo não puder ser alcançado:

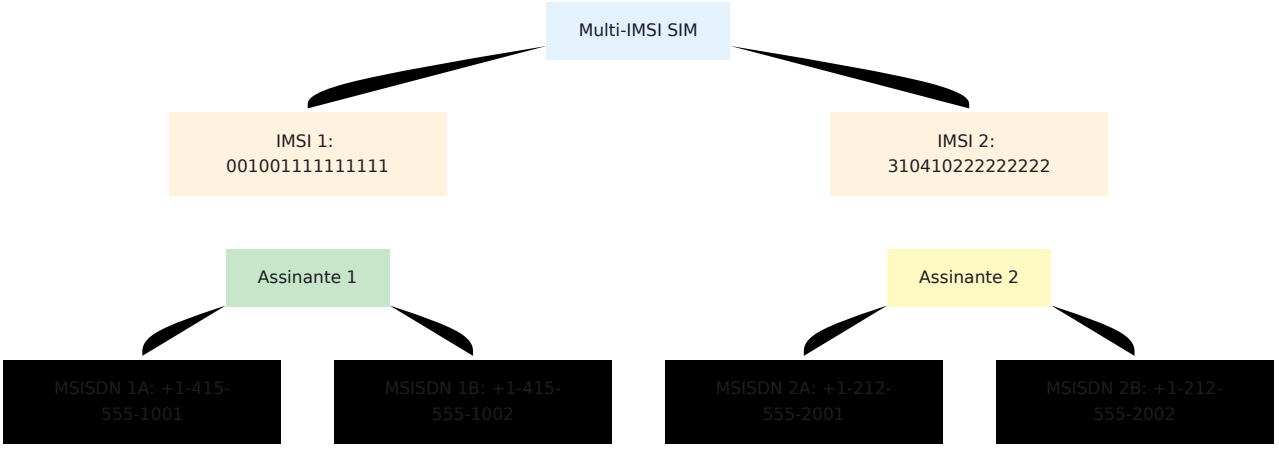
    - Verifique qual IMSI está atualmente registrado
    - Verifique se o IMSI correto está sendo usado para a rede atual
    - Confirme que apenas um IMSI por SIM está no estado registrado

---

Cenários Combinados

Multi-IMSI + Multi-MSISDN

Você pode combinar ambas as funcionalidades: múltiplos IMSIs em um SIM, cada um com múltiplos MSISDNs.



Exemplo de Caso de Uso:

- **Rede de Origem (IMSI 1):**
  - Número pessoal: +1-415-555-1001
  - Número comercial: +1-415-555-1002
- **Rede de Roaming nos EUA (IMSI 2):**
  - Número pessoal: +1-212-555-2001
  - Número comercial: +1-212-555-2002

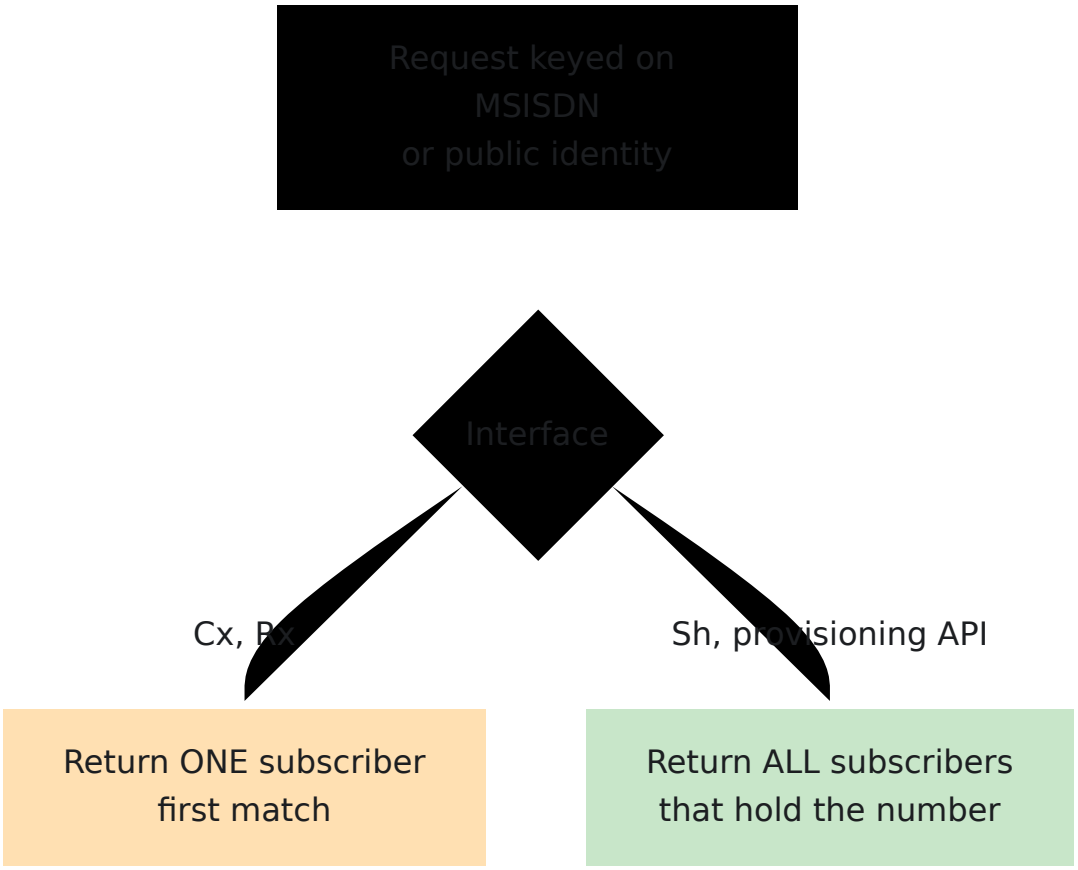
Quando o dispositivo está em território de origem, usa IMSI 1 com seus MSISDNs. Quando está em roaming nos EUA, troca para IMSI 2 com diferentes MSISDNs otimizados para a rede dos EUA.

Resolução de MSISDN Através de Interfaces

Um MSISDN é uma entidade compartilhada no modelo de dados. Um assinante pode ter muitos MSISDNs (um IMSI, muitos números). Um MSISDN pode ser mantido por muitos assinantes (muitos IMSIs, um número, por exemplo, um telefone e um smartwatch). Veja as entidades Assinante e MSISDN para o modelo de dados.

Como um número não é sempre exclusivo para um assinante, cada interface deve decidir como resolver uma solicitação que chega com base em um MSISDN ou em uma identidade pública que contém um MSISDN. O OmniHSS usa duas regras de resolução diferentes, e você deve saber qual regra uma interface usa.

Duas regras de resolução



- **Retornar um.** A pesquisa retorna um único assinante (o primeiro correspondente). As interfaces **Cx** e **Rx** usam esta regra.

- **Retornar todos.** A pesquisa retorna todos os assinantes que possuem o número. A interface **Sh** e a API de provisionamento usam esta regra.

Comportamento por interface

Interface	Baseado em	Multi-MSISDN por IMSI	Multi-IMSI po MSISDN (número compartilhad
Cx UAR, MAR	Identidade Privada (IMSI)	Correto. IMSI é exato.	Correto. Cada IMSI é separad
Cx SAR	Identidade Privada ou Identidade Pública	Correto. O IFC lista todos os MSISDNs do assinante.	Resolve para ur assinante se baseado no número compartilhado. Use a Identidad Privada (IMSI) para selecionar um assinante específico.
Cx LIR	Identidade Pública	Correto.	Resolve para ur assinante **registrado** (un com um S-CSCF atribuído) quando o número é compartilhado, então uma solicitação de terminação retorna um nó de serviço alcançável.

Interface	Baseado em	Multi-MSISDN por IMSI	Multi-IMSI por MSISDN (número compartilhado
Sh UDR, PUR, SNR	Identidade do Usuário (MSISDN ou Identidade Pública)	Correto. Os dados são por assinante.	Retorna todos os assinantes que possuem o número. Um UDR agrega seus dados. Um PUR atualiza todos eles.
Rx AAR	Assinante da sessão de dados	O assinante pode ter muitos números, mas a resposta carrega um MSISDN como Subscription-Id (o protocolo permite um).	Correto. O número é selecionado da sessão de dados, não por pesquisa reversa.
API `GET /subscriber/msisdn/:msisdn`	MSISDN	Retorna o assinante.	Retorna todos os assinantes que possuem o número.

O que isso significa na prática

- **Registro e autenticação estão sempre corretos.** O caminho de registro IMS (Cx MAR, SAR, UAR) é baseado na Identidade Privada, que contém o IMSI. Cada dispositivo se registra com seu próprio IMSI, então um telefone e um smartwatch que compartilham um número se registram e autenticam de forma independente.
- **Um assinante mantém todos os seus números.** Para um IMSI com muitos MSISDNs, o Cx SAR User-Data lista cada número como uma identidade pública (`sip:` e `tel:`).
- **Um número compartilhado resolve para um assinante em Cx e para todos os assinantes compartilhados em Sh.** Esta é a diferença de resolução na tabela acima. No caminho de terminação (Cx LIR), o HSS prefere um assinante compartilhado **registrado**, então retorna o S-CSCF de serviço de um dispositivo que é alcançável. Como a chamada é então entregue aos dispositivos registrados é uma questão do núcleo IMS, não do HSS.
- **Para agir sobre um assinante específico, use o IMSI ou o ICCID.** Uma pesquisa reversa por número não é única quando o número é compartilhado.

Especificações

Os dois modelos têm diferentes status de padrão. Saiba qual você usa.

Multi-MSISDN por IMSI é padronizado. A 3GPP define mais de um número para uma assinatura:

- **Opção de Multinúmeros** (3GPP TS 23.008, cláusula 2.1.3). O HLR/HSS armazena um MSISDN Básico mais MSISDNs adicionais. Um indicador de MSISDN Básico marca o número que o VLR e o SGSN usam.
- **MSISDN Adicional (A-MSISDN)** (3GPP TS 23.003, cláusula 18.9). Um A-MSISDN é "atribuído a um usuário com assinatura PS além dos MSISDN(s) já atribuídos" e "deve ser atribuído exclusivamente por IMSI".

Multi-IMSI por MSISDN é um padrão de implementação, não um único padrão de rede. É realizado no SIM por um applet multi-IMSI que armazena vários IMSIs e alterna entre eles (termo da indústria: Multiple-IMSI-Multiple-

MSISDN, MIMM). A orientação relacionada da GSMA é GSMA IR.73 Steering of Roaming, e para eSIM as especificações do perfil SGP.2x/SGP.32. No lado da rede, a 3GPP não padroniza um MSISDN em várias assinaturas como um serviço; em vez disso, a restringe:

- **MSISDN de Correlação (C-MSISDN)** (3GPP TS 23.003, cláusula 18.7). Um C-MSISDN "é atribuído exclusivamente por IMSI e Identidade Privada do Usuário IMS". Apenas um MSISDN "que não é compartilhado por mais de uma Identidade Privada IMS" pode servir como um C-MSISDN. Portanto, um número compartilhado não pode ser usado para correlação de sessão SRVCC. Planeje SRVCC e fallback CS com um número por IMSI, não o número compartilhado.

Especificações de interface:

- Cx (MAR, SAR, UAR, LIR): 3GPP TS 29.228 e 3GPP TS 29.229.
- Sh (UDR, PUR, SNR): 3GPP TS 29.328 e 3GPP TS 29.329.

Modelo	Status	Referência
Multi-MSISDN por IMSI	Padronizado	TS 23.008 §2.1.3 (multinúmeros); TS 23.003 §18.9 (A-MSISDN, único por IMSI)
Multi-IMSI por MSISDN	Padrão de applet SIM (MIMM); restrito na rede	GSMA IR.73; TS 23.003 §18.7 (um MSISDN compartilhado não pode ser um C-MSISDN)

Identidade Pública do Usuário Compartilhada (um n��mero, vários dispositivos)

Para fazer um número tocar em vários dispositivos (por exemplo, um telefone e um smartwatch acompanhante em SIMs separados), agrupe os dispositivos em

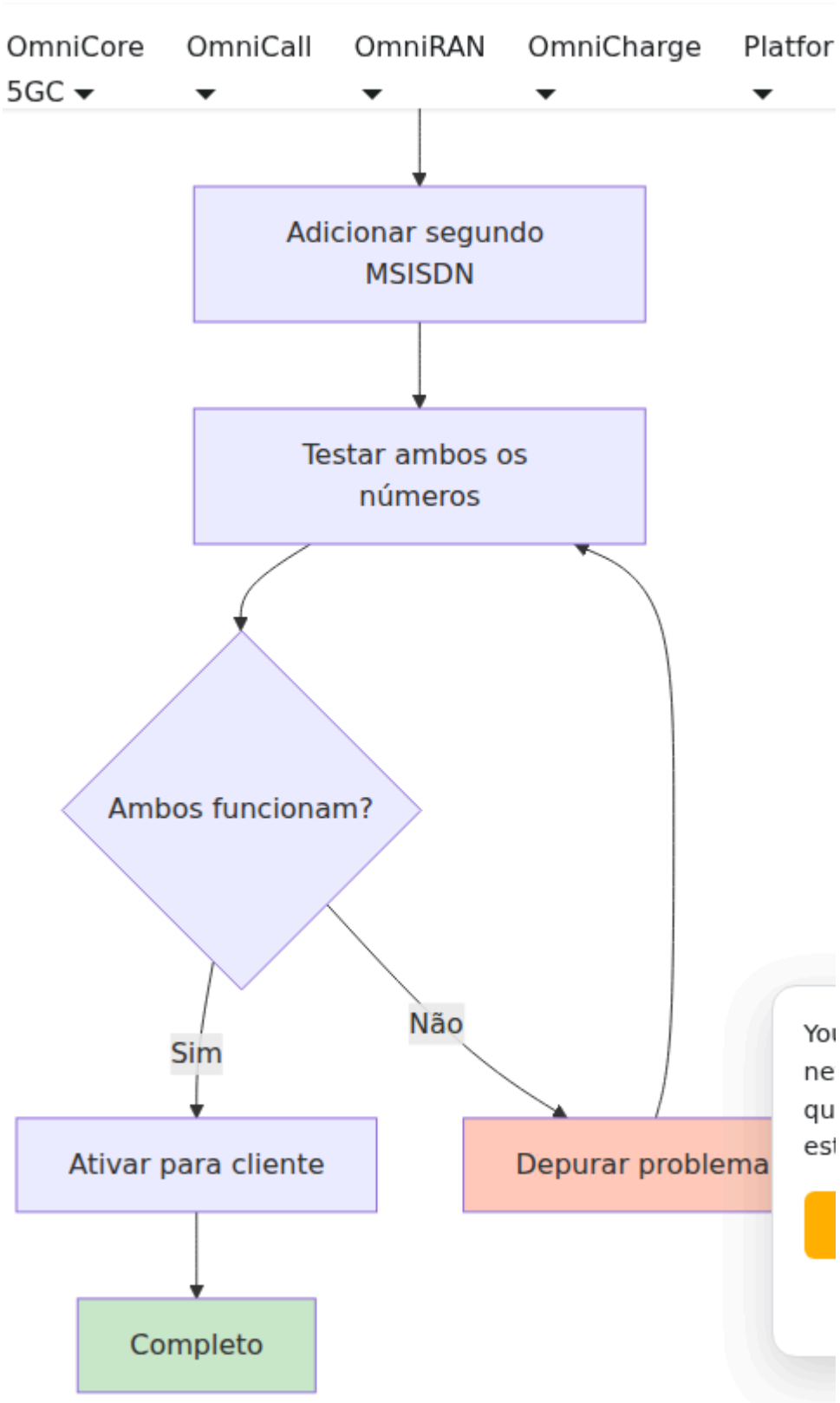
uma **assinatura IMS**. Este é o modelo 3GPP de uma Identidade Pública do Usuário compartilhada (3GPP TS 23.228): várias Identidades Privadas do Usuário (IMPIs, uma por IMSI) compartilham uma identidade pública e, criticamente, um único S-CSCF de serviço. O S-CSCF então bifurca uma chamada de terminação para cada dispositivo registrado.

Sem um grupo, dois assinantes que apenas compartilham um número são duas assinaturas IMS separadas. Eles podem ser atribuídos a diferentes S-CSCFs, então uma chamada de terminação atinge apenas os dispositivos em um deles. O grupo `ims_subscription` remove essa lacuna, dando aos membros um S-CSCF comum.

Como Funciona

- Um `ims_subscription` mantém o S-CSCF compartilhado do grupo (`assigned_scscf`).
- Cada assinante membro referencia o grupo (`ims_subscription_id`).
- Quando o primeiro membro se registra, o HSS seleciona um S-CSCF. O HSS ancora o grupo a esse nó quando o Cx SAR do nó confirma o registro. Cada membro posterior é direcionado para o **mesmo** S-CSCF, independentemente de qual S-CSCF seu registro reivindica.
- A âncora sempre mantém um nó S-CSCF real. Com `selection_method::pool_uri`, o HSS retorna o nome do pool no primeiro REGISTER, mas ancora o nó que o DNS escolheu, não o nome do pool. Veja Um S-CSCF de Serviço Por Assinatura.
- Um novo registro retorna o nó ancorado, nunca o nome do pool.
- O S-CSCF compartilhado é liberado apenas quando o **último** membro desregistra.
- Em uma solicitação de terminação (Cx LIR), o HSS retorna o S-CSCF compartilhado enquanto qualquer membro estiver registrado.
- O Cx SAR User-Data lista cada IMPI membro (o conjunto de registro implícito), então o S-CSCF registra todo o conjunto. Um perfil de serviço compartilhado opcional no grupo renderiza uma iFC para as identidades compartilhadas.
- Em uma solicitação Sh para a identidade pública compartilhada (Data-Reference `IMSPublicIdentity`), o HSS inclui um

`<IMSPrivateUserIdentity>` por IMSI membro nos dados do usuário. Um Servidor de Aplicação (por exemplo, o MMTel TAS) conta esses para detectar um número compartilhado: uma contagem maior que um marca um número compartilhado multi-IMSI, então o AS permite que a chamada de terminação bifurque para cada dispositivo. Um número comum retorna um único `<IMSPrivateUserIdentity>` e é entregue a um contato.



Provisionamento

Crie o grupo, anexe o assinante de cada dispositivo a ele e atribua o número compartilhado a cada membro.

```
# 1. Crie a assinatura IMS (o grupo). Um optional ims_profile_id
define um
#    Perfil de Serviço compartilhado para as identidades
compartilhadas do grupo.
GROUP_ID=$(curl -k -X POST
https://hss.example.com:8443/api/ims_subscription \
  -H "Content-Type: application/json" \
  -d '{"name": "Família Smith - número compartilhado"}' | jq -r
'.response.id')

# 2. Anexe cada assinante membro ao grupo e dê a ele o MSISDN
compartilhado.
curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$PHONE_SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{\"ims_subscription_id\": \"$GROUP_ID\", \"msisdns\":
[\"$SHARED_MSISDN_ID\"]}"

curl -k -X PATCH
https://hss.example.com:8443/api/subscriber/$WATCH_SUBSCRIBER_ID \
  -H "Content-Type: application/json" \
  -d "{\"ims_subscription_id\": \"$GROUP_ID\", \"msisdns\":
[\"$SHARED_MSISDN_ID\"]}"
```

Parâmetros:

Parâmetro	Tipo	Necessário	Descrição
`name`	String	Sim	Nome legível por humanos para a assinatura IMS (único).
`ims_profile_id`	UUID	Não	Perfil de Serviço Compartilhado (iFC) para as identidades compartilhadas do grupo. Quando ausente, cada membro renderiza seu próprio perfil IMS.
`ims_subscription_id`	UUID	Não	Em um assinante: o grupo ao qual pertence. Quando ausente, o assinante é autônomo (seu próprio S-CSCF por assinante, o padrão).

O `assigned_scscf` e `last_seen_scscf_timestamp` no grupo são estado de tempo de execução escrito pelo HSS; você não os define.

Notas

- Um assinante sem `ims_subscription_id` é autônomo e se comporta exatamente como antes.
- Dê aos membros o mesmo MSISDN para que o número compartilhado resolva para o grupo. Os membros podem manter seus próprios números adicionais.
- Excluir um grupo não exclui seus assinantes; seu `ims_subscription_id` é limpo e eles se tornam autônomos.

Procedimentos Operacionais

Gerenciando Assinantes Multi-MSISDN

Veja todos os MSISDNs para um assinante:

```
Consulta via API: GET /api/subscriber/imsi/:imsi
```

A resposta inclui todos os MSISDNs vinculados.

Solução de Problemas Multi-IMSI

Dispositivo não se conectando com o segundo IMSI:

1. Verifique se o segundo registro de assinante existe para esse IMSI
2. Verifique se o key_set está configurado corretamente para esse IMSI
3. Verifique se o perfil EPC está atribuído
4. Confirme se as regras de roaming permitem a conexão

Dispositivo trocando IMSIs inesperadamente:

- Isso é controlado pela lógica do dispositivo/SIM, não pelo HSS
- O HSS autentica qualquer IMSI apresentado
- Verifique as configurações de seleção de IMSI do dispositivo

Solução de Problemas Multi-MSISDN

Segundo número não toca:

1. Verifique se o MSISDN está vinculado na tabela de junção
2. Verifique se o modelo de perfil IMS inclui a variável `{{msisdns}}`
3. Confirme se o registro IMS inclui todas as identidades públicas
4. Revise os logs do S-CSCF para identidades registradas

Chamadas de saída mostram apenas um número:

- O dispositivo seleciona qual número apresentar como ID do chamador
- Isso é configuração do dispositivo, não do HSS
- O HSS fornece todas as identidades; o dispositivo escolhe

Resumo dos Benefícios

Benefícios do Multi-MSISDN

✓ Um SIM, múltiplos números de telefone
✓ Linhas comerciais e pessoais separadas
✓ Presença local internacional
✓ Gerenciamento simplificado de dispositivos
✓ Todos os números compartilham o mesmo serviço de dados
✓ Cobrança centralizada por IMSI

Benefícios do SIM Multi-IMSI

✓ Custos de roaming otimizados
✓ Seleção automática de rede
✓ Redundância e failover
✓ Identidade de rede local
✓ Conformidade regulatória
✓ Continuidade de serviço entre redes

Benefícios Combinados

✓ Máxima flexibilidade
✓ Diferentes conjuntos de números por rede
✓ Otimizado para cada caso de uso
✓ Cenários de negócios complexos
✓ Otimização internacional e local

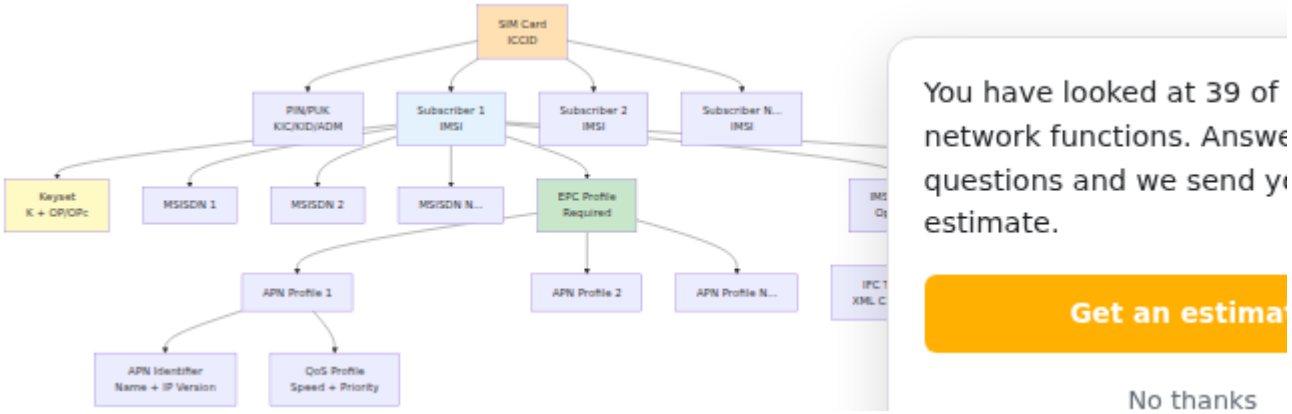
← Voltar ao Guia de Operações

Gerenciamento de Perfis OmniHSS

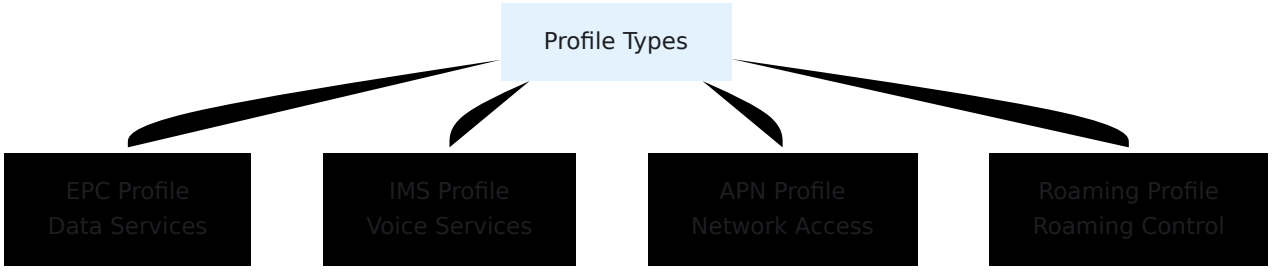
← Voltar ao Guia de Operações

Visão Geral

OmniHSS usa **perfis** para definir características de serviço para assinantes. Os perfis permitem criar modelos de serviço reutilizáveis que podem ser atribuídos a vários assinantes, simplificando o provisionamento e garantindo consistência.



Tipos de Perfis



Perfis EPC

Os Perfis EPC (Evolved Packet Core) definem características de serviço de dados para assinantes LTE.

Parâmetros Chave

Parâmetro	Descrição	Valores Típicos
`ue_ambr_dl_kbps`	Limite de velocidade de download	10.000 - 1.000.000 Kbps
`ue_ambr_ul_kbps`	Limite de velocidade de upload	5.000 - 500.000 Kbps
`network_access_mode`	Tipo de serviço	"packet_only" ou "packet_and_circui
`tracking_area_update_interval_seconds`	Timer TAU	54 segundos (típico)

Criando Perfis EPC

```
curl -k -X POST https://hss.example.com:8443/api/epc/profile \
  -H "Content-Type: application/json" \
  -d '{
    "apn_profiles": [],
    "name": "Premium 100Mbps",
    "network_access_mode": "packet_only",
    "tracking_area_update_interval_seconds": 600,
    "ue_ambr_dl_kbps": 100000,
    "ue_ambr_ul_kbps": 50000
  }'
```

Modelos Comuns de Perfis EPC

Internet Básica:

- Download: 10 Mbps (10.000 Kbps)
- Upload: 5 Mbps (5.000 Kbps)

Padrão:

- Download: 50 Mbps (50.000 Kbps)
- Upload: 25 Mbps (25.000 Kbps)

Premium:

- Download: 100 Mbps (100.000 Kbps)
- Upload: 50 Mbps (50.000 Kbps)

Ilimitado:

- Download: 1 Gbps (1.000.000 Kbps)
- Upload: 500 Mbps (500.000 Kbps)

Perfis IMS

Os Perfis IMS definem características de serviço de voz, principalmente através de modelos IFC (Initial Filter Criteria).

Modelos IFC

Os modelos IFC são documentos XML que definem regras de roteamento de chamadas para o S-CSCF.

Variáveis do Modelo:

- `{{imsi}}` - IMSI do assinante
- `{{msisdns}}` - Lista de números de telefone
- `{{mcc}}` - Código do país de origem

- `{{mnc}}` - Código da rede de origem

Criando Perfis IMS

```
curl -k -X POST https://hss.example.com:8443/api/ims/profile \
  -H "Content-Type: application/json" \
  -d '{
    "ims_profile": {
      "name": "Standard VoLTE",
      "ifc_template": "<InitialFilterCriteria>...
</InitialFilterCriteria>"
    }
  }'
```

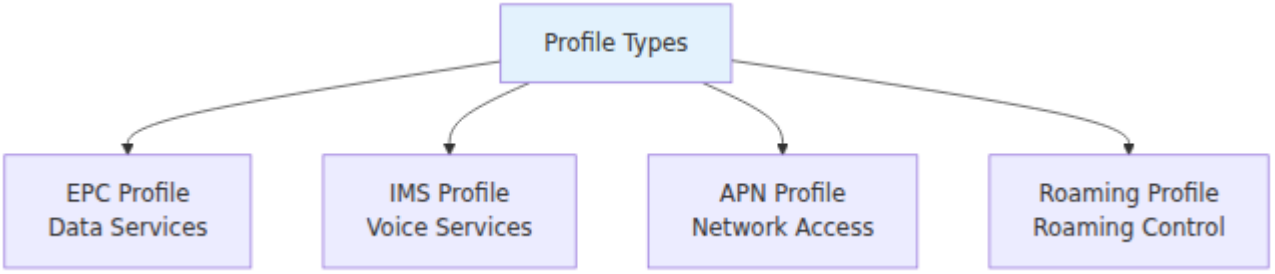
Exemplo de Modelo IFC

```
<ServiceProfile>
  <PublicIdentity>
    <Identity>sip:
{{imsi}}@ims.mnc{{mnc}}.mcc{{mcc}}.3gppnetwork.org</Identity>
  </PublicIdentity>
  <InitialFilterCriteria>
    <Priority>0</Priority>
    <TriggerPoint>
      <ConditionTypeCNF>0</ConditionTypeCNF>
      <SPT>
        <ConditionNegated>0</ConditionNegated>
        <Group>0</Group>
        <Method>INVITE</Method>
      </SPT>
    </TriggerPoint>
    <ApplicationServer>
      <ServerName>sip:as.ims.example.com</ServerName>
      <DefaultHandling>0</DefaultHandling>
    </ApplicationServer>
  </InitialFilterCriteria>
</ServiceProfile>
```

Perfis APN

Os Perfis APN (Access Point Name) definem pontos de acesso à rede para conexões de dados.

Componentes APN



Identificador APN

Define o nome do APN e o suporte ao protocolo IP.

APNs Comuns:

- `internet` - Acesso geral à internet
- `ims` - Sinalização IMS/VoLTE
- `mms` - Mensagens multimídia
- `vzwadmin` - Específico do operador

Opções de Versão IP:

- `"ipv4"`: Apenas IPv4
- `"ipv6"`: Apenas IPv6
- `"ipv4v6"`: IPv4v6 (pilha dupla)
- `"ipv4_or_ipv6"`: IPv4 ou IPv6 (escolha da rede)

Perfil QoS APN

Define parâmetros de qualidade de serviço.

Valores de QCI (QoS Class Identifier):

QCI	Tipo	Caso de Uso	Prioridade
1	GBR	Voz conversacional	Mais Alto
2	GBR	Vídeo conversacional	Alto
4	GBR	Streaming de vídeo	Alto
5	Non-GBR	Sinalização IMS	Médio
9	Non-GBR	Internet (padrão)	Mais Baixo

Método de Cobrança em Nível de Sessão (opcional):

O perfil QoS APN também pode definir o método de cobrança padrão para a sessão IP-CAN, enviado como os AVPs `Online`/`Offline` de nível superior no Gx CCA-I.

Campo	Tipo	Padrão	Descrição
`online_charging_enabled`	boolean	`null`	Quando definido, força o AVP `Online` em nível de sessão (`true` = habilitar, `false` = desabilitar). Quando `null`, o PCRF ecoa o valor que o PGW enviou no CCR-I.
`offline_charging_enabled`	boolean	`null`	Quando definido, força o AVP `Offline` em nível de sessão (`true` = habilitar, `false` = desabilitar). Quando `null`, o PCRF ecoa o valor que o PGW enviou no CCR-I.

Veja PCRF: Cobrança Online/Offline em Nível de Sessão para o comportamento completo e referência de valores AVP.

Criando Configuração Completa de APN

```
# 1. Criar Identificador APN
APN_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/identifier \
  -H "Content-Type: application/json" \
  -d '{"apn": "internet", "ip_version": "ipv4v6"}' \
  | jq -r '.response.id')

# 2. Criar Perfil QoS APN
QOS_ID=$(curl -k -X POST
https://hss.example.com:8443/api/apn/qos_profile \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Best Effort",
    "allocation_retention_priority": 8,
    "apn_ambr_dl_kbps": 50000,
    "apn_ambr_ul_kbps": 25000,
    "pre_emption_capability": false,
    "pre_emption_vulnerability": true,
    "qci": 9
  }' | jq -r '.response.id')

# 3. Criar Perfil APN
curl -k -X POST https://hss.example.com:8443/api/apn/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"apn_identifier_id\": \"$APN_ID\",
    \"apn_qos_profile_id\": \"$QOS_ID\",
    \"name\": \"Internet APN\"
  }"
```

Atribuindo APNs ao Perfil EPC

APNs estão vinculados aos Perfis EPC através da tabela `join_epc_profile_to_apn_profile`.

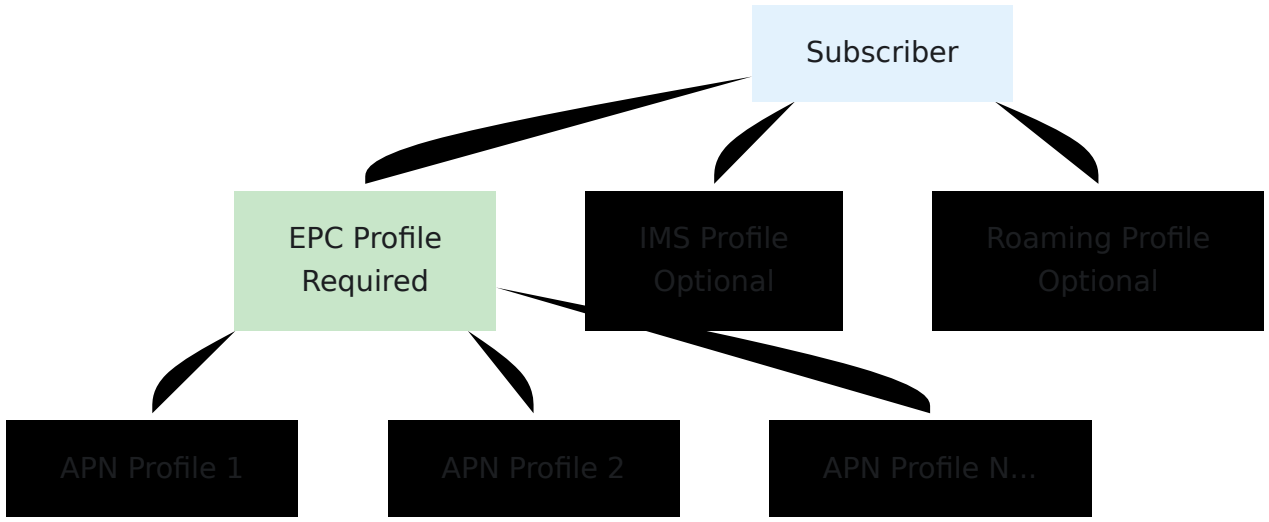
Insira registros na tabela de junção para vincular IDs de perfil APN ao ID do perfil EPC. Vários perfis APN podem ser atribuídos a um perfil EPC.

Perfis de Roaming

Veja a documentação detalhada no Guia de Controle de Roaming.

Atribuição de Perfis

Relações de Perfil de Assinante



Atribuindo Perfis a Assinantes

```
# Atribuir perfis EPC e IMS durante a criação do assinante
curl -k -X POST https://hss.example.com:8443/api/subscriber \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "imsi": "001001123456789",
      "key_set_id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "epc_profile_id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "ims_profile_id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "roaming_profile_id": "9c858901-8a57-4791-81fe-4c455b099bc9"
    }
  }'

# Atualizar perfil do assinante
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-4b01-
90e6-d701748f0851 \
  -H "Content-Type: application/json" \
  -d '{
    "subscriber": {
      "epc_profile_id": "f47ac10b-58cc-4372-a567-0e02b2c3d479"
    }
  }'
```

Melhores Práticas de Gerenciamento de Perfis

Princípios de Design

1. **Criar Perfis Padrão** - Definir níveis de serviço comuns (Básico, Padrão, Premium)
2. **Reutilizar Perfis** - Atribuir o mesmo perfil a vários assinantes
3. **Documentar Mudanças** - Rastrear modificações de perfil

4. **Testar Antes da Produção** - Verificar se o perfil funciona primeiro com um assinante de teste

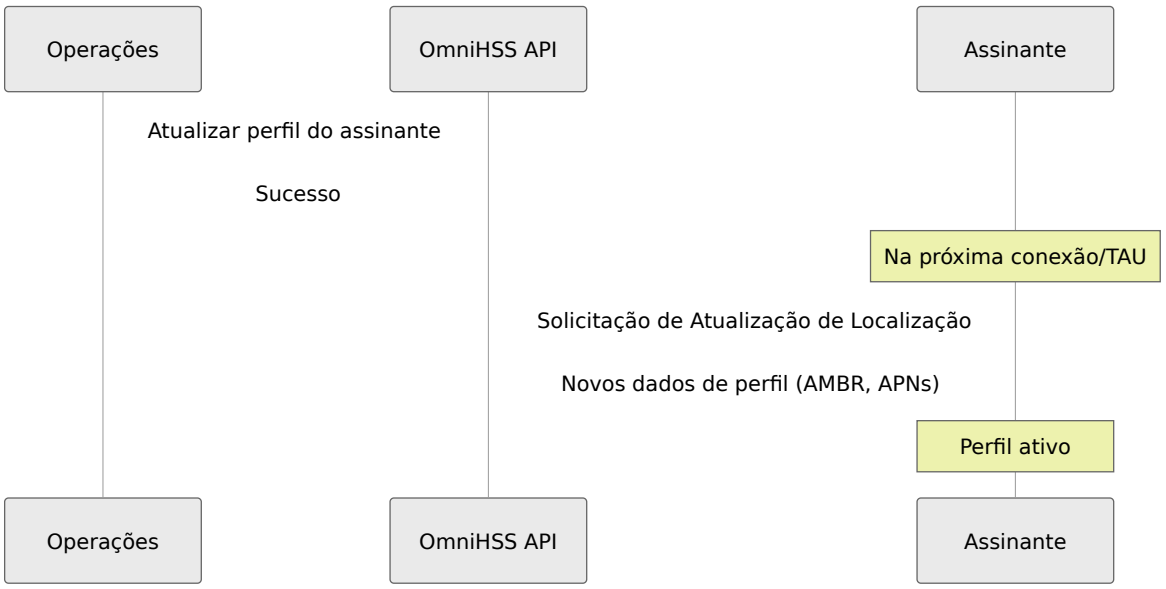
Convenção de Nomenclatura de Perfis

```
[Nível de Serviço]-[Velocidade]-[Recursos]

Exemplos:
- "Básico-10Mbps-Internet"
- "Premium-100Mbps-VoLTE"
- "Empresarial-1Gbps-MultiAPN"
```

Migração de Perfis

Ao alterar o perfil de um assinante:



Importante: Mudanças de perfil entram em vigor na próxima:

- Atualização de Área de Rastreamento (TAU)
- Conexão
- Registro IMS (para mudanças de perfil IMS)

Solução de Problemas de Perfis

Assinante não está recebendo a velocidade esperada:

1. Verifique os valores AMBR do perfil EPC atribuído
2. Verifique os valores AMBR do perfil QoS APN
3. Verifique se o MME/P-GW está aplicando QoS corretamente
4. Verifique se há congestionamento na rede

Registro IMS falha:

1. Verifique se o perfil IMS está atribuído
2. Verifique a validade do XML do modelo IFC
3. Revise os logs do S-CSCF para erros de processamento IFC
4. Confirme a configuração de seleção do S-CSCF

APN não disponível:

1. Verifique se o perfil APN está vinculado ao perfil EPC
2. Verifique se o identificador APN corresponde à solicitação da rede
3. Revise a solicitação de conectividade PDN do UE

← Voltar ao Guia de Operações | Próximo: Controle de Roaming →

Fluxos do Protocolo OmniHSS

← Voltar ao Guia de Operações

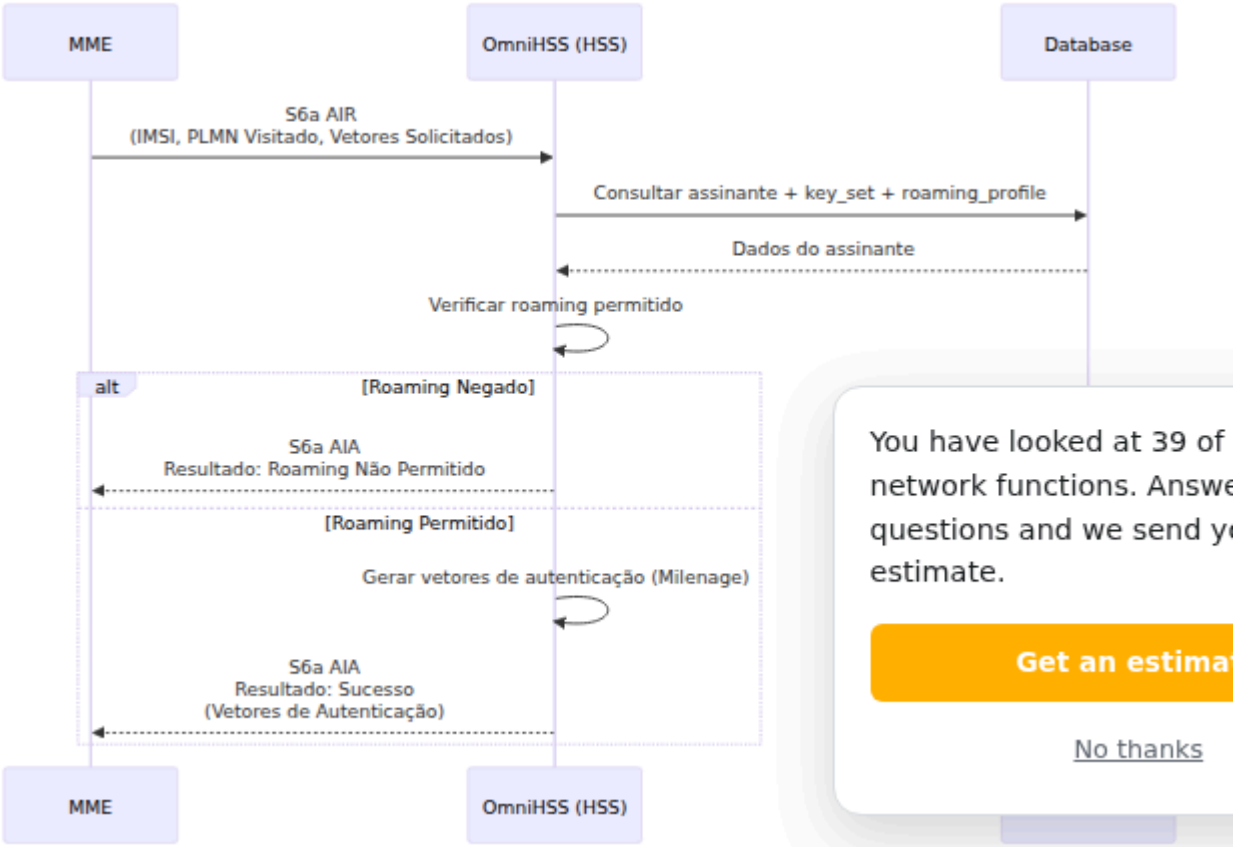
Visão Geral

Este documento detalha os fluxos de mensagens do protocolo Diameter suportados pelo OmniHSS. Compreender esses fluxos é essencial para solução de problemas e operações.

Interface S6a (LTE/EPC)

Solicitação de Informação de Autenticação (AIR/AIA)

O MME solicita vetores de autenticação para o assinante.

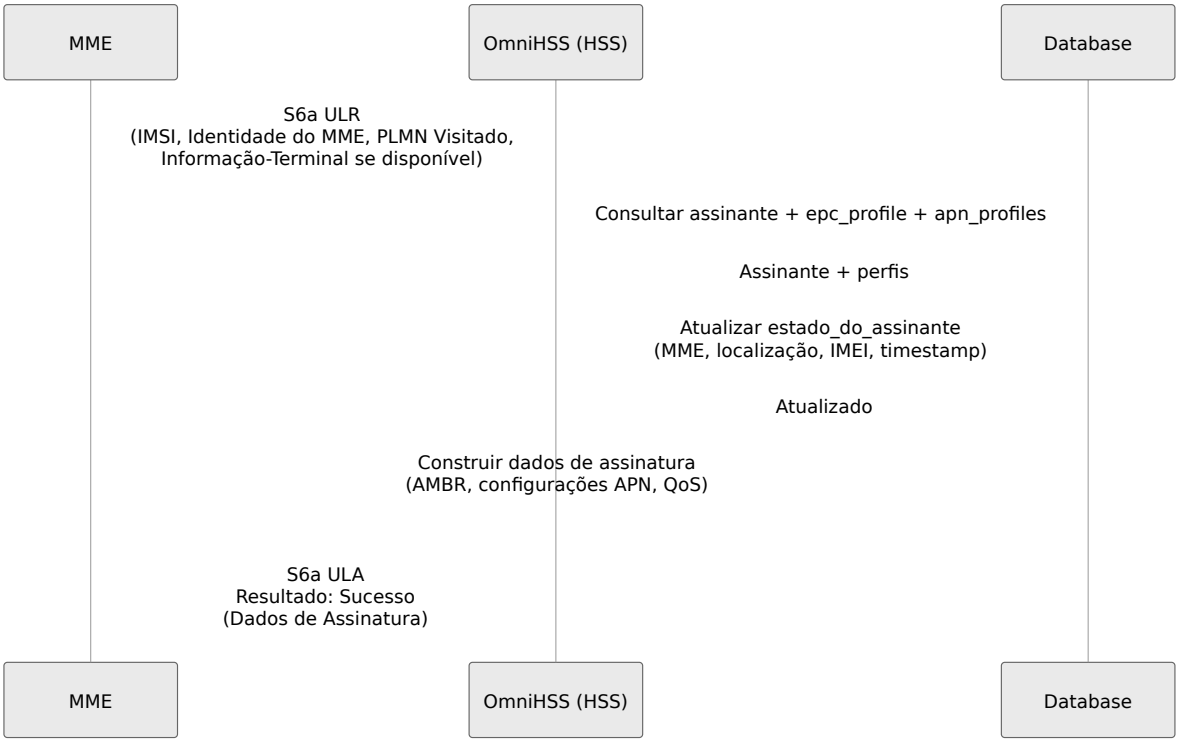


AVPs Chave:

- Solicitação: User-Name (IMSI), Visited-PLMN-Id, Número de Vetores Solicitados
- Resposta: Authentication-Info (RAND, AUTN, XRES, KASME)

Solicitação de Atualização de Localização (ULR/ULA)

O MME notifica o HSS sobre a localização do assinante e recupera dados de assinatura.

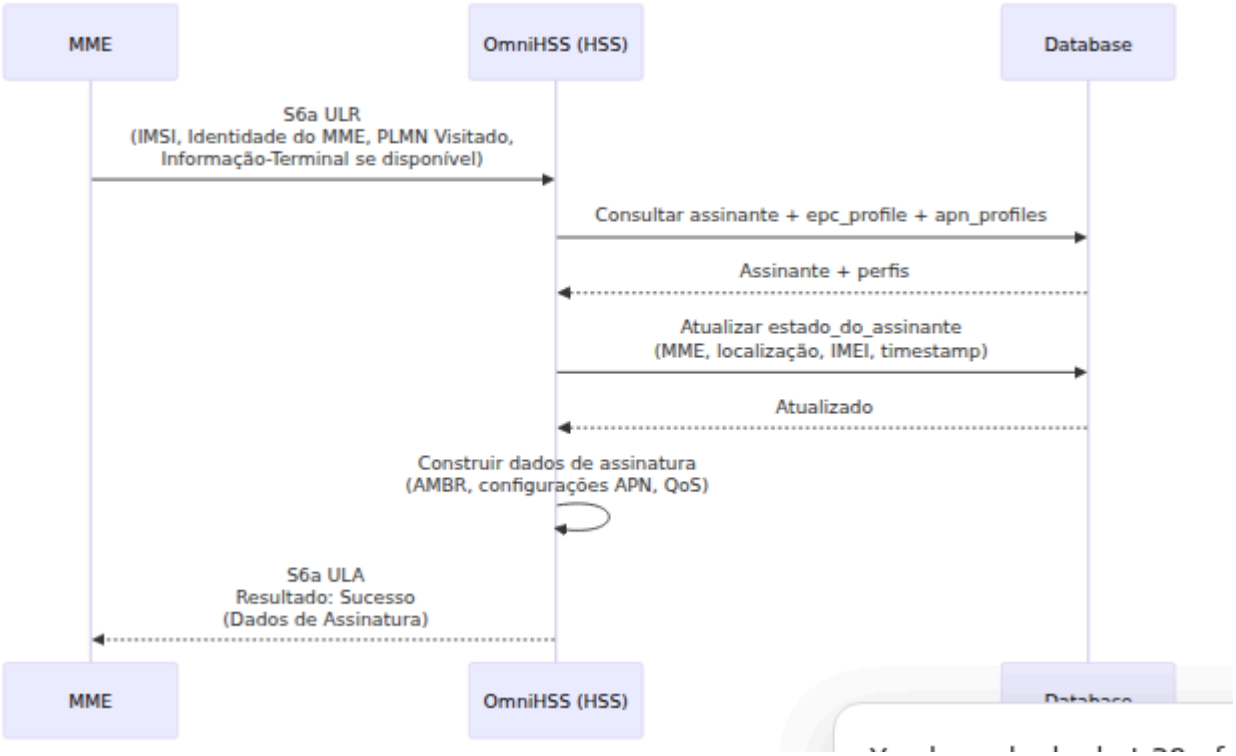


AVPs Chave:

- Solicitação: User-Name (IMSI), RAT-Type, ULR-Flags, Visited-PLMN-Id, UE-SRVCC-Capability, Terminal-Information (IMEI, Software-Version; opcional)
- Resposta: Subscription-Data (AMBR, APN-Configuration, Network-Access-Mode)

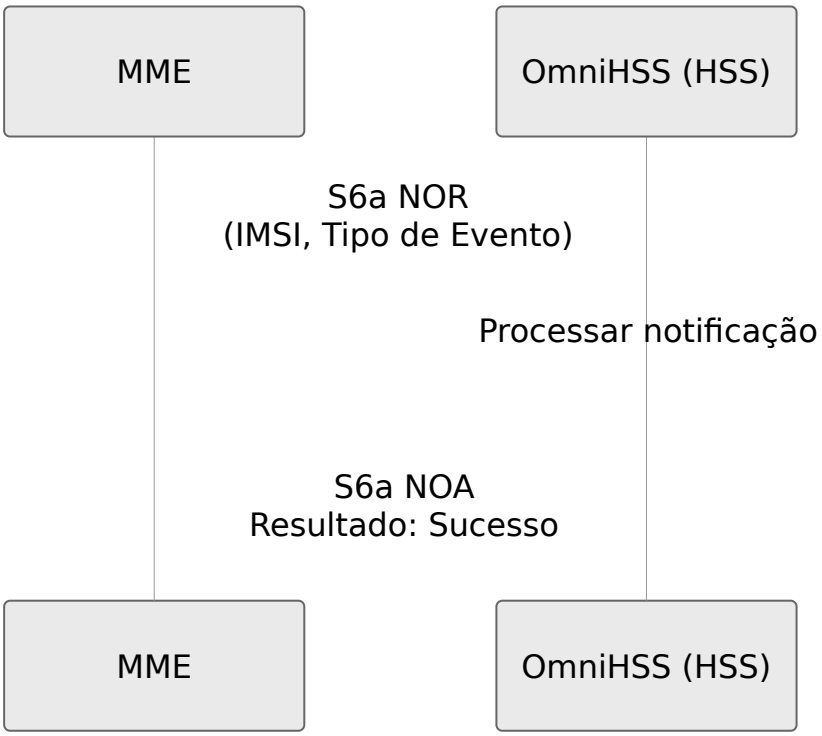
Solicitação de Purga de UE (PUR/PUA)

O MME notifica o HSS quando o contexto do assinante é excluído.



Solicitação de Notificação (NOR/NOA)

O MME informa o HSS sobre vários eventos.

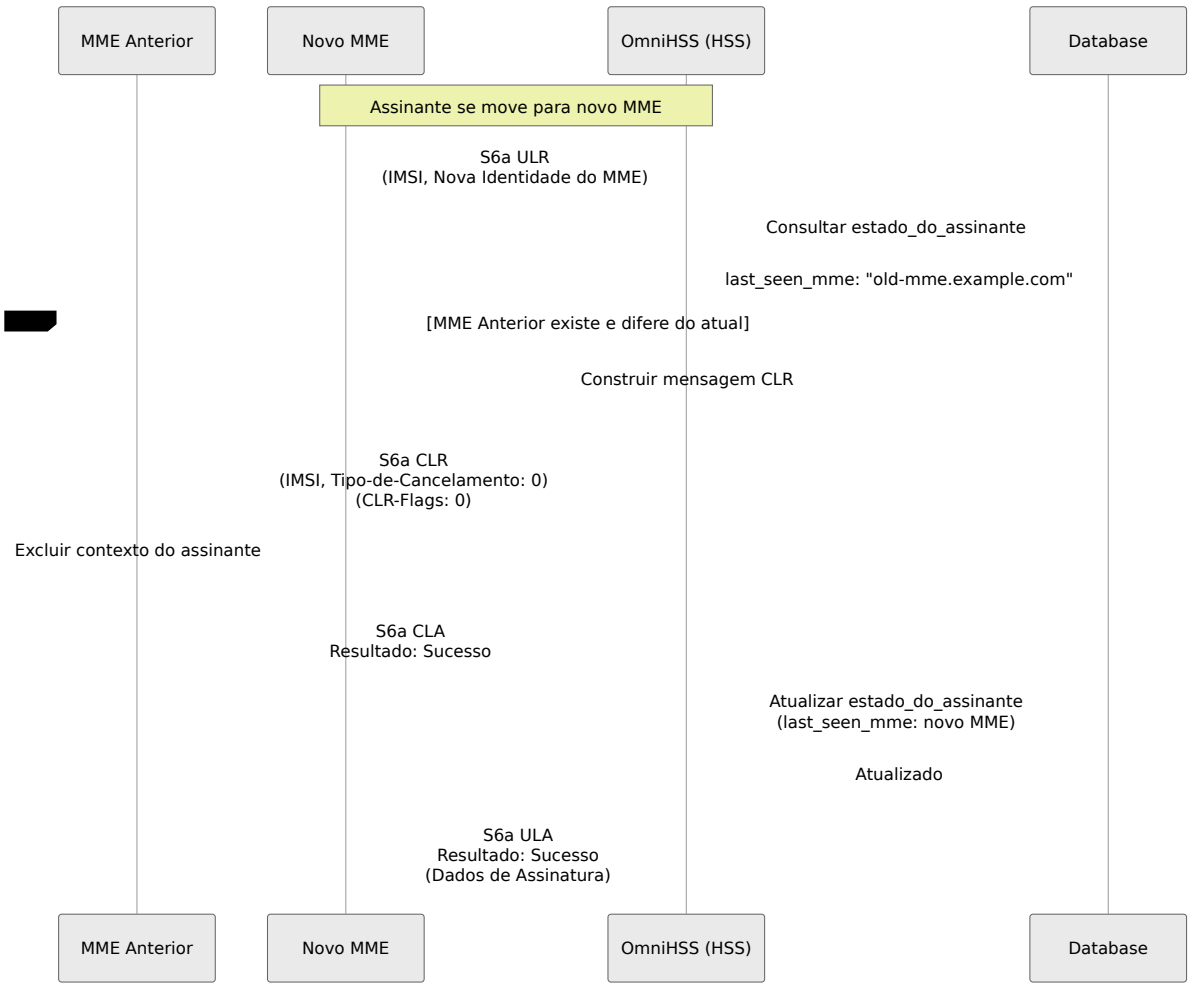


Solicitação de Cancelamento de Localização (CLR/CLA)

O HSS inicia o cancelamento de localização para informar ao MME que o assinante deve ser desconectado. O OmniHSS suporta tanto o envio automático quanto programático de CLR.

CLR Automático (Transferência de MME)

Quando um assinante realiza uma Solicitação de Atualização de Localização de um novo MME, o OmniHSS envia automaticamente um CLR para o MME anterior para limpar registros obsoletos.



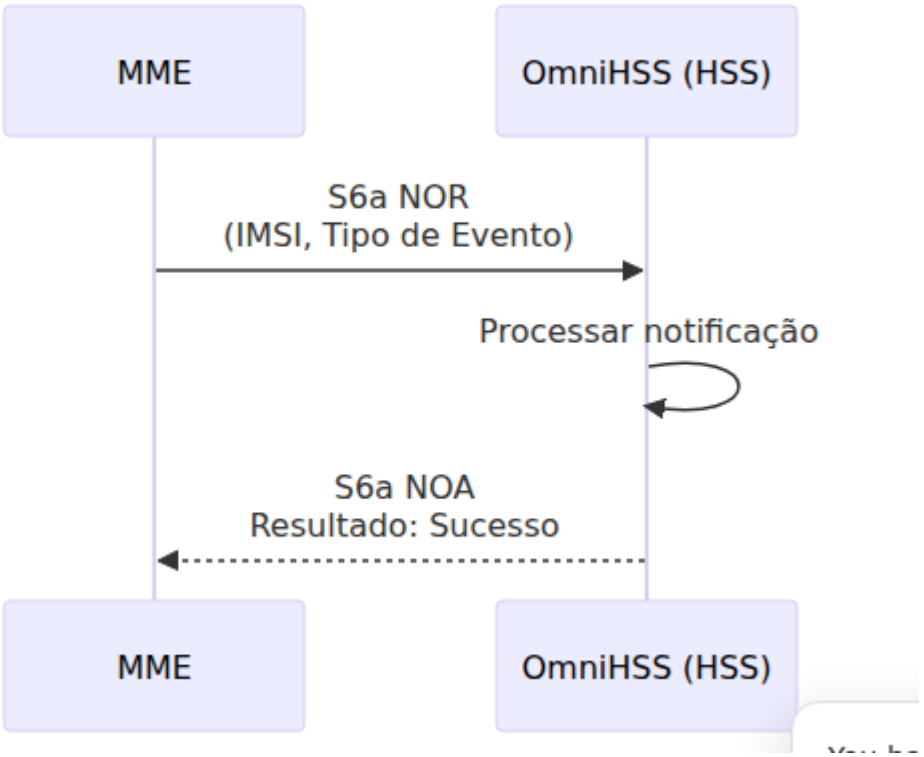
AVPs Chave (CLR Automático):

- User-Name: IMSI do assinante
- Destination-Host: Nome do host do MME anterior
- Destination-Realm: Reino do MME anterior

- Cancellation-Type: 0 (Procedimento de Atualização do MME)
- CLR-Flags: 0
- Subscription-Data: Perfil de assinatura completo

CLR Programático (Acionado por API)

Administradores podem acionar CLR via a API programática para desconectar assinantes à força (por exemplo, para retirada de assinatura, prevenção de fraudes ou ações administrativas).



AVPs Chave (CLR Programático):

- User-Name: IMSI do assinante
- Destination-Host: Nome do host do último MME visto
- Destination-Realm: Reino do último MME visto
- Cancellation-Type: `:subscription_withdrawal` (codificado como inteiro conforme 3GPP TS 29.272)
- CLR-Flags:
  - s6a_indicator: 1 (indica interface S6a)
  - reattach_required: 1 (UE deve re-autenticar para se reconectar)

Tipos de Cancelamento

O OmniHSS suporta múltiplos tipos de cancelamento conforme 3GPP TS 29.272:

Tipo	Valor	Descrição	Caso de Uso
Procedimento de Atualização do MME	0	Mudança normal de MME	Automático durante ULR de novo MME
Procedimento de Atualização do SGSN	1	Transferência de SGSN	Cenários de transferência 3G/2G
Retirada de Assinatura	2	Término administrativo	Desconexão manual via API
Procedimento de Atualização IWF	3	Atualização da função de interconexão	Interoperabilidade de rede legada
Procedimento de Anexação Inicial	4	Registro fresco	Forçar re-autenticação

CLR-Flags

O AVP CLR-Flags é uma máscara de bits com os seguintes campos:

Flag	Bit	Descrição
Indicador S6a/S6d	0	1 = interface S6a utilizada
Reanexação Necessária	1	1 = UE deve realizar novo anexo

Exemplo de Configuração de CLR-Flags:

```
clr_flags: %{
  s6a_indicator: 1,        # Usando interface S6a
  reattach_required: 1     # Forçar re-autenticação
}
```

Cenários Multi-IMSI

O OmniHSS rastreia o registro do MME **por assinante (IMSI)**, não por MSISDN. Isso é crítico para entender o comportamento do CLR em cenários multi-IMSI:

Cenário 1: Múltiplos MSISDNs, Um Único IMSI

```
Assinante A:
  - IMSI: 999000123456789
  - MSISDNs: ["+1234567890", "+9876543210"]
  - last_seen_mme: "mme01.operator.com"
```

Quando este assinante se move para um novo MME:

- **Um CLR enviado** para "mme01.operator.com" com IMSI 999000123456789
- Ambos os MSISDNs são afetados (mesmo assinante, mesmo SIM)
- O AVP User-Name contém o IMSI, não os MSISDNs

Cenário 2: Múltiplos Assinantes (IMSI Diferentes), Mesmo MSISDN

O OmniHSS aplica a **restrição de MSISDN único** (um MSISDN não pode pertencer a múltiplos assinantes simultaneamente). No entanto, durante portabilidade/migração:

```
Assinante A:
  - IMSI: 999000111111111
  - MSISDN: "+1234567890"
  - last_seen_mme: "mme01.operator.com"

Assinante B (após portabilidade):
  - IMSI: 999000222222222
  - MSISDN: "+1234567890"  # Mesmo MSISDN, SIM/IMSI diferente
  - last_seen_mme: "mme02.operator.com"
```

Quando o Assinante B se registra:

- **Nenhum CLR enviado** (IMSI diferente = assinante diferente)
- O Assinante A permanece registrado no mme01
- O Assinante B se registra no mme02
- Ambos podem estar ativos simultaneamente (dispositivos físicos diferentes)

Cenário 3: CLR Programático para Assinante Multi-MSISDN

Resultado:

- **Um CLR enviado** para o last_seen_mme do assinante
- **Todos os MSISDNs** associados a esse IMSI são efetivamente desconectados
- O IMSI é a chave primária para rastrear o registro do MME

Notas Importantes

1. **IMSI é a Chave:** As operações de CLR são sempre **por IMSI**, nunca por MSISDN. A tabela `subscriber_state` rastreia `last_seen_mme` por assinante (IMSI).

2. **Operação Atômica:** Cada assinante pode estar registrado em apenas um MME por vez. O CLR automático garante isso limpando o registro antigo.

3. **Sem CLR se Não Houver MME Anterior:** Se `last_seen_mme` for `nil` (assinante nunca registrado), nenhum CLR é enviado durante ULR.

4. **Dados de Assinatura Incluídos:** O CLR automático (durante ULR) inclui o AVP completo `Subscription-Data` para ajudar o MME antigo a limpar o contexto adequadamente.

5. **Assíncrono:** O CLR é enviado de forma assíncrona (fire-and-forget). A resposta ULA para o novo MME não espera pelo CLA do MME antigo.

6. **Tratamento de CLA:** O OmniHSS recebe respostas CLA, mas atualmente as descarta. Isso evita loops de mensagens e é um comportamento padrão do HSS.

Interface SWx (Acesso Não-3GPP / Chamadas WiFi)

A interface SWx permite chamadas WiFi conectando o ePDG (Gateway de Dados de Pacote Evoluído) ao HSS para autenticação e autorização de assinantes em redes de acesso não-3GPP.

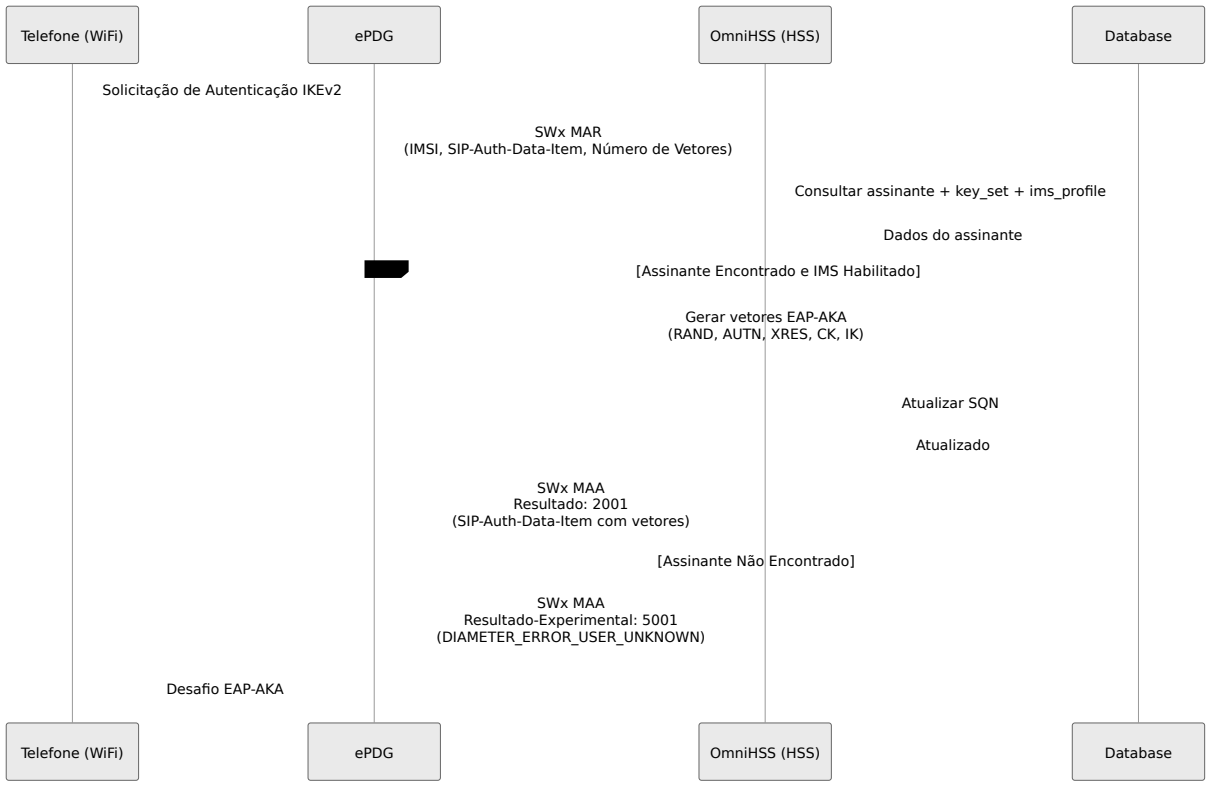
Solicitação de Autenticação Multimídia (MAR/MAA)

O ePDG solicita vetores de autenticação EAP-AKA para registro de chamadas WiFi. Um TWAG (WLAN confiável) solicita vetores EAP-AKA' pela mesma interface.

O `SIP-Authentication-Scheme` na solicitação seleciona o tipo de vetor (cláusula 8.1.2.1 do 3GPP TS 29.273):

Esquema	ANID	Vetor retornado
`EAP-AKA`	Ignorado	Bit de separação AMF 0. `Confidentiality-Key` e `Integrity-Key` transportam CK e IK.
`EAP-AKA'`	Requerido	Bit de separação AMF 1. `Confidentiality-Key` e `Integrity-Key` transportam CK' e IK', derivados de CK, IK, SQN xor AK e o ANID conforme o Anexo A.2 do 3GPP TS 33.402. CK e IK não saem do HSS.

Uma solicitação com `EAP-AKA'` e sem `ANID` é respondida com `Result-Code` 5005 (DIAMETER_MISSING_AVP). Qualquer outro esquema é respondido com 5004 (DIAMETER_INVALID_AVP_VALUE). O ANID para uma rede de acesso WLAN é `WLAN` (cláusula 8.1.1.2 do 3GPP TS 24.302).



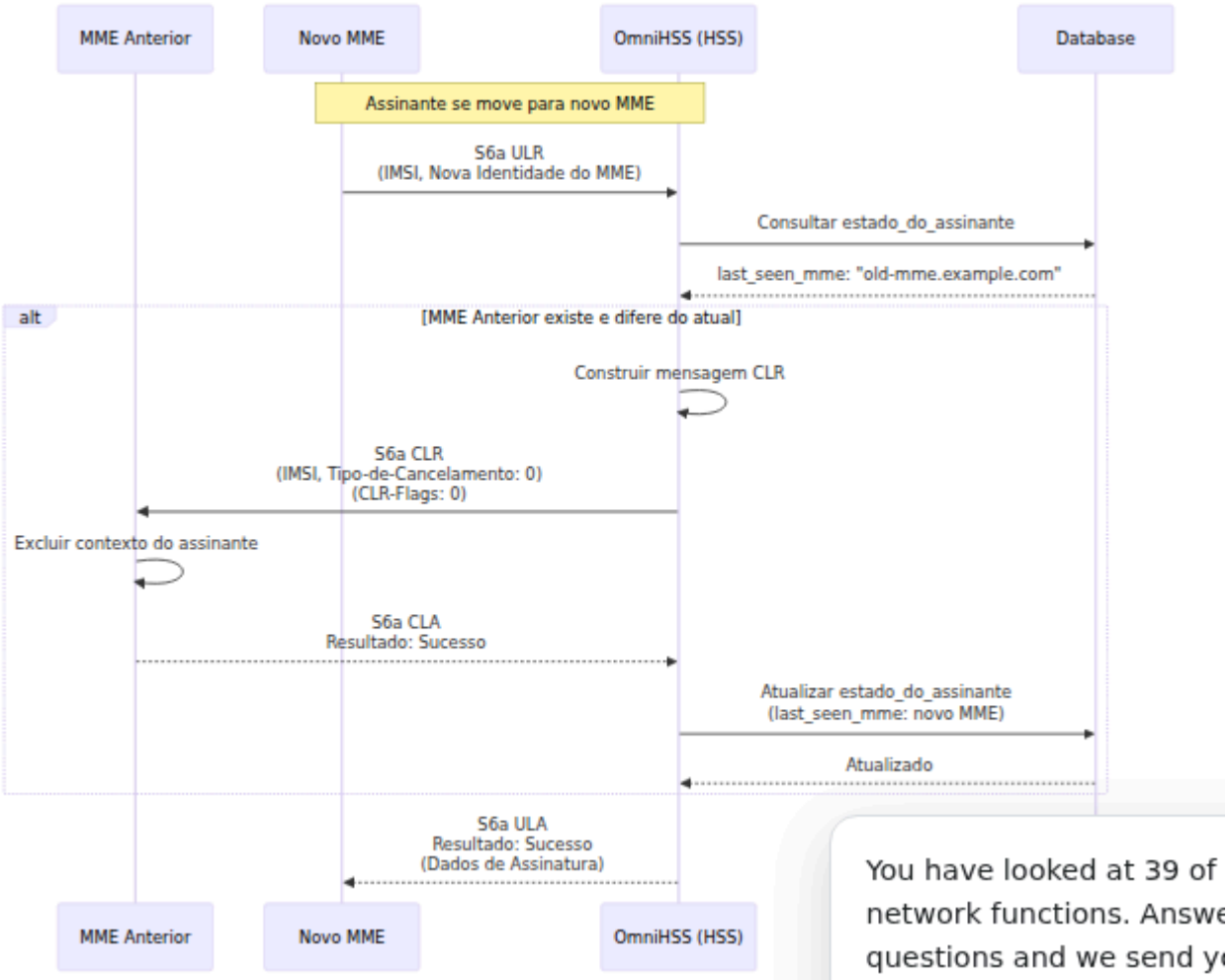
AVPs Chave:

- Solicitação: User-Name (IMSI ou IMSI@realm), SIP-Auth-Data-Item (esquema de autenticação), SIP-Number-Auth-Items
- Resposta: SIP-Auth-Data-Item (SIP-Authenticate, SIP-Authorization, Confidentiality-Key, Integrity-Key)

Resincronização de SQN: Quando o UE detecta uma falha de autenticação devido a um desvio de SQN, o ePDG inclui dados de ressincronização no AVP SIP-Authorization. O OmniHSS extrai AUTS deste campo e recalcula o SQN usando computação ciente do comprimento do IND conforme o 3GPP TS 33.102.

Solicitação de Atribuição de Servidor (SAR/SAA)

O ePDG registra ou desregistra o assinante para serviços de chamadas WiFi.



You have looked at 39 of
network functions. Answe
questions and we send y

Tipos de Atribuição de Servidor:

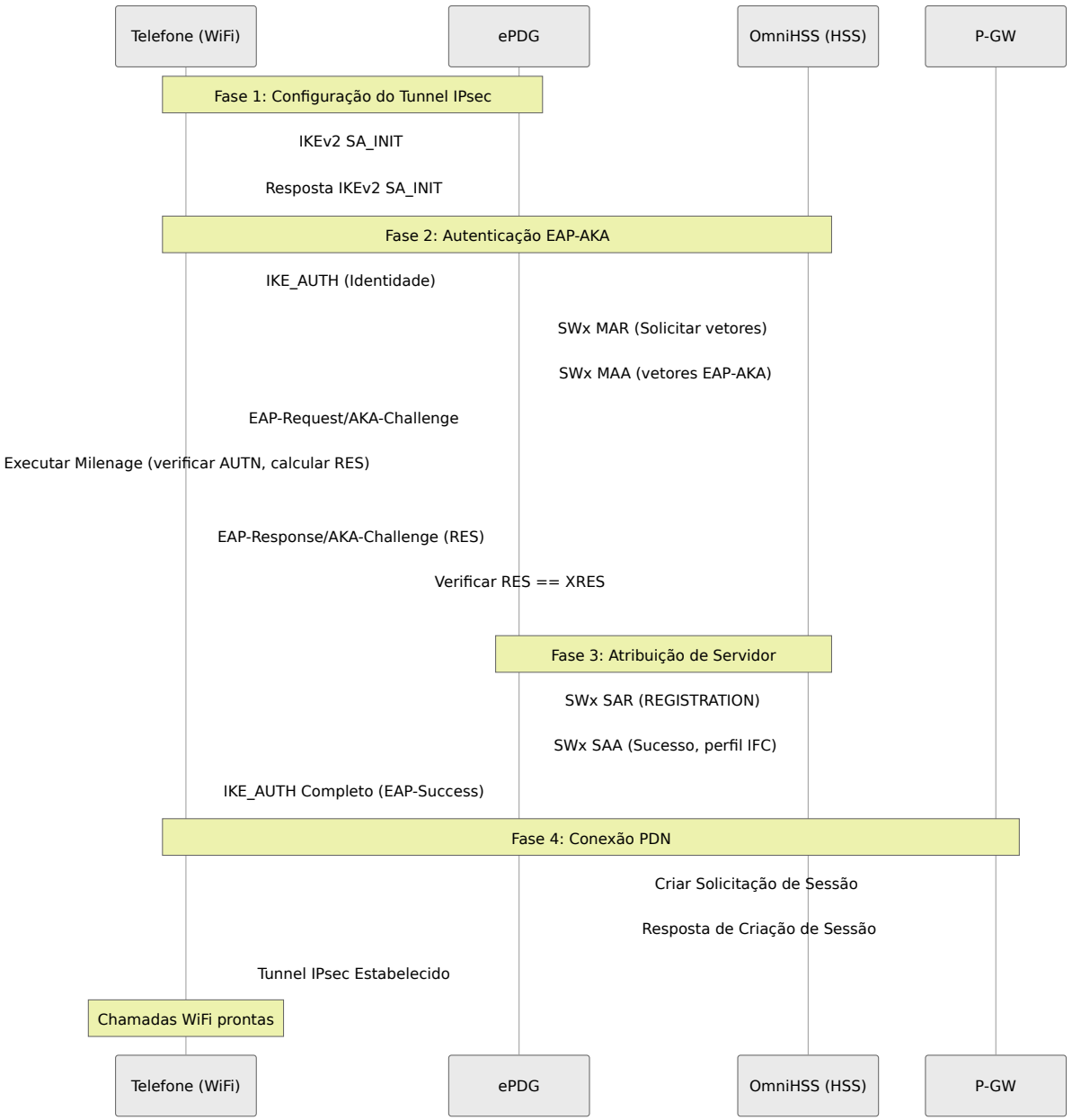
Tipo	Valor	Descrição
NO_ASSIGNMENT	0	Apenas consulta, sem mudança de estado
REGISTRATION	1	Registro inicial de chamadas WiFi
RE_REGISTRATION	2	Atualização de registro
UNREGISTERED_USER	3	Serviços para usuário não registrado
TIMEOUT_DEREGISTRATION	4	Timeout de registro
USER_DEREGISTRATION	5	Desregistro iniciado pelo usuário
AUTHENTICATION_FAILURE	7	Falha de autenticação, limpar estado
ADMINISTRATIVE_DEREGISTRATION	8	Desregistro iniciado por administrador

AVPs Chave:

- Solicitação: User-Name (IMSI), Tipo-de-Atribuição-Servidor, Nome-do-Servidor (identidade do ePDG), Service-Selection (APN)
- Resposta: User-Data (perfil IFC XML), Non-3GPP-User-Data

Fluxo de Registro de Chamadas WiFi

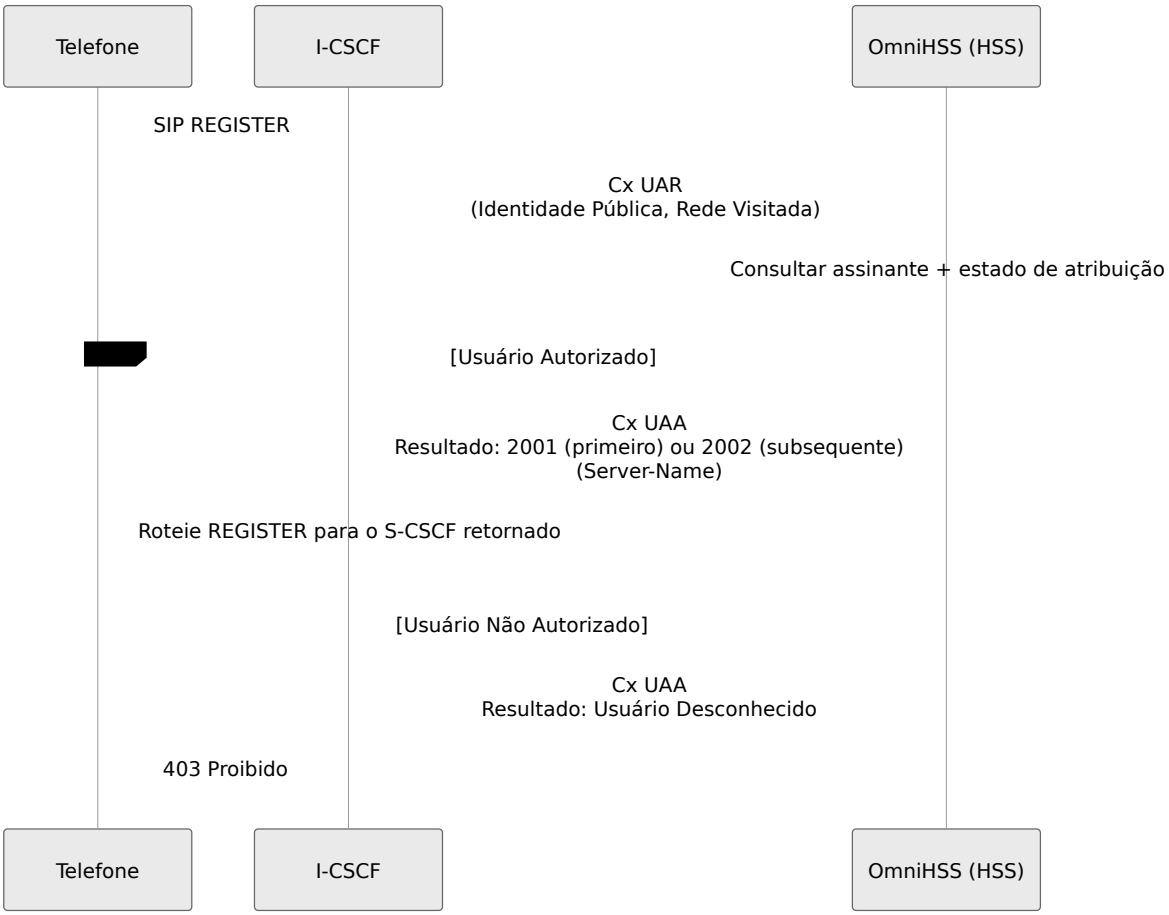
Registro completo de chamadas WiFi de ponta a ponta.



Interface Cx (IMS)

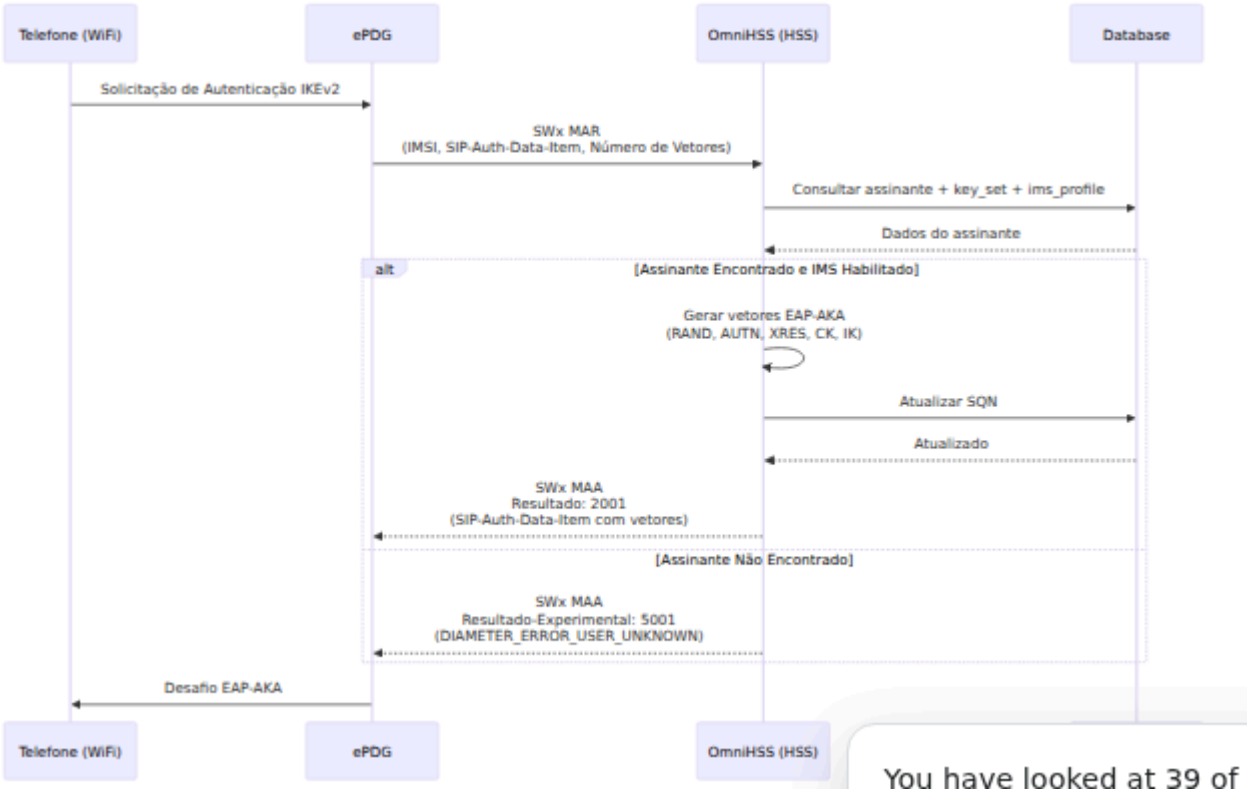
Solicitação de Autorização de Usuário (UAR/UAA)

O I-CSCF pergunta se o usuário pode se registrar e qual S-CSCF usar. O HSS retorna um nome de S-CSCF (um `Server-Name`), não um conjunto de capacidades. O nome e o código de resultado seguem as regras do Estado de Atribuição do S-CSCF.



Solicitação de Atribuição de Servidor (SAR/SAA)

O S-CSCF registra/desregistra o usuário e recupera o perfil IMS.



You have looked at 39 of

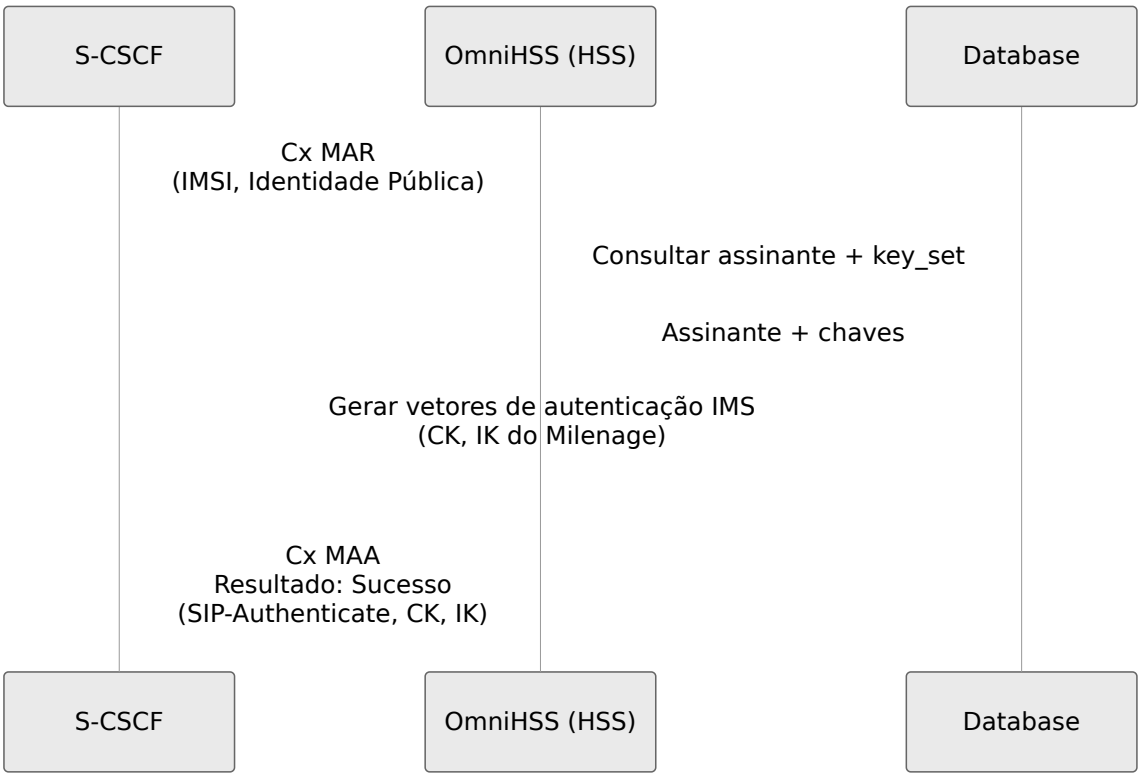
Renderização do Template IFC:

- `{{imsi}}` → IMSI real
- `{{msisdns}}` → Lista de números de telefone
- `{{mcc}}, {{mnc}}` → Códigos PLMN de origem

Para como o SAR atualiza o estado de registro e o estado de autenticação pendente, e como lida com `AUTHENTICATION_FAILURE` e `AUTHENTICATION_TIMEOUT`, veja Estado de Atribuição do S-CSCF.

Solicitação de Autenticação Multimídia (MAR/MAA)

O S-CSCF solicita vetores de autenticação para registro IMS.



Estado de Atribuição do S-CSCF

O HSS mantém duas peças separadas de estado IMS para cada assinante. O estado de registro e o estado de autenticação pendente são independentes, conforme 3GPP TS 29.228 §3.1.

Campo de estado	Definido em	Limpo em	Significado
`assigned_scscf`	Sucesso do SAR (registro)	Desregistro do SAR, falha/timeout de autenticação quando Não Registrado, ou falha/timeout de autenticação que retira um nome obsoleto	O assinante está registrado neste S-CSCF.
`auth_pending_scscf`	MAR	SAR (sucesso ou falha/timeout de autenticação), desregistro UAR, ou expiração do TTL	Um desafio IMS-AKA está em andamento neste S-CSCF.
`auth_pending_since`	MAR	Com `auth_pending_scscf`	O tempo em que o desafio começou. O HSS o usa para expirar um desafio que nunca é concluído.

O MAR armazena o nome do S-CSCF e define o estado de autenticação pendente, conforme TS 29.228 §6.3. O SAR confirma ou limpa isso, conforme TS 29.228 §6.1.2.

Ordem de seleção do S-CSCF UAA. Para cada UAR, o HSS retorna um nome de S-CSCF e um código de resultado com esta precedência:

1. Um S-CSCF de autenticação pendente ativo está registrado: retorne-o, com `DIAMETER_SUBSEQUENT_REGISTRATION` (2002). Um desafio está ativo

enquanto `auth_pending_scscf` estiver definido e `auth_pending_since` estiver dentro de `auth_pending_ttl_seconds`.

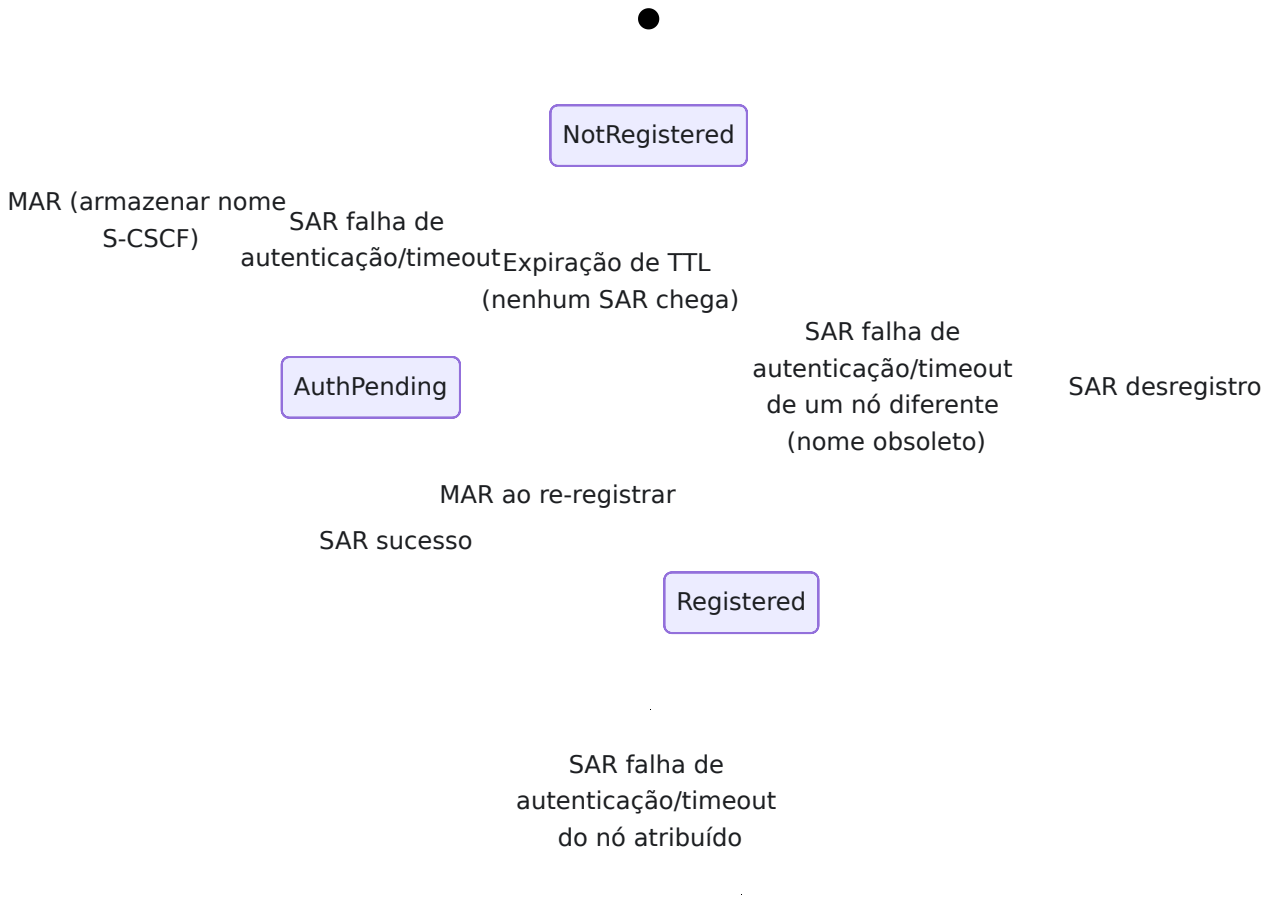
2. `assigned_scscf` contém um nó S-CSCF real: retorne-o, com `DIAMETER_SUBSEQUENT_REGISTRATION` (2002).
3. Nenhum dos dois se aplica: selecione um novo S-CSCF (veja Configuração de Seleção do S-CSCF), com `DIAMETER_FIRST_REGISTRATION` (2001).

A Regra 2 testa intencionalmente a presença de um nó real. Um URI de pool é um nome para todo o pool, portanto nunca identifica o nó que atende o assinante. Se um URI de pool estiver registrado de uma versão anterior, o HSS trata o assinante como não atendido e responde 2001. O próximo SAR então armazena o nó real.

O re-registro sempre retorna o S-CSCF que está atendendo. A Regra 2 torna a atribuição fixa. Depois que um nó assume o registro, cada UAR de re-registro e cada LIR de término retornam esse nó. O HSS retorna um URI de pool apenas para um assinante que nenhum nó atende ainda.

O HSS prefere o S-CSCF de autenticação pendente sobre `assigned_scscf`. Isso faz com que um ciclo de desafio em andamento converja para o nó que o emitiu. Para um assinante registrado saudável, o nó que desafia é o nó atribuído, então a preferência não tem efeito. Isso importa apenas durante a Recuperação de Falha do S-CSCF.

O estado de autenticação pendente expira. Um ciclo de desafio é concluído em segundos, e cada resultado envia um SAR que limpa o estado. Se o S-CSCF desafiador nunca enviar esse SAR, o nome pendente permanece registrado. O HSS ignora um nome pendente que é mais antigo que `auth_pending_ttl_seconds` (padrão de 30 segundos). Sem o TTL, um MAR perdido ou abandonado superaria a vinculação registrada para sempre. O tráfego de origem iria para um nó que não possui nada, enquanto o tráfego de término ainda iria para `assigned_scscf`.



Códigos de resultado Cx:

Código	Nome	Significado	Referê
2001	DIAMETER_FIRST_REGISTRATION	Nenhum S-CSCF registrado. O nome retornado é uma seleção.	TS 29.2
2002	DIAMETER_SUBSEQUENT_REGISTRATION	Um S-CSCF está registrado. O HSS retorna o nome armazenado.	TS 29.2

Valores relevantes de Server-Assignment-Type (SAR):

Valor	Nome	Ação do HSS
0	NO_ASSIGNMENT	Limpar o nome S-CSCF e o registro.
1	REGISTRATION	Definir o estado de registro como registrado. Limpar autenticação pendente.
2	RE_REGISTRATION	Manter o registro. Limpar autenticação pendente.
4	TIMEOUT_DEREGISTRATION	Limpar o nome S-CSCF e o registro.
5	USER_DEREGISTRATION	Limpar o nome S-CSCF e o registro.
6	TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME	Encerrar o registro. Manter o nome S-CSCF registrado.
7	USER_DEREGISTRATION_STORE_SERVER_NAME	Encerrar o registro.

Valor	Nome	Ação do HSS
		Manter o nome S-CSCF registrado.
8	ADMINISTRATIVE_DEREGISTRATION	Limpar o nome S-CSCF e o registro.
9	AUTHENTICATION_FAILURE	Manter um vínculo registrado, a menos que o nome registrado esteja obsoleto. Limpar o nome S-CSCF quando Não Registrado.
10	AUTHENTICATION_TIMEOUT	Manter um vínculo registrado, a menos que o nome registrado esteja obsoleto. Limpar o nome S-CSCF quando Não Registrado.

Valor	Nome	Ação do HSS
11	DEREGISTRATION_TOO_MUCH_DATA	Limpar o nome S-CSCF e o registro.

Os tipos 6 e 7 encerram o registro, mas mantêm o nome S-CSCF, conforme TS 29.228 §6.1.2. Esse nó continua a atender o estado não registrado do assinante, então um UAR posterior ainda se resolve para ele.

Detecção de nome obsoleto em uma falha de autenticação. Um `AUTHENTICATION_FAILURE` ou `AUTHENTICATION_TIMEOUT` SAR normalmente mantém o vínculo de um assinante registrado. Há uma exceção. Se o SAR vier de um S-CSCF diferente do que está registrado, o I-CSCF deve ter re-dividido o REGISTER no pool, porque não conseguiu alcançar o nó registrado. O HSS então limpa esse nome como obsoleto. Sem essa regra, Recuperação de Falha do S-CSCF não converge: o HSS retornaria o nó inativo em todos os UAR posteriores, e apenas um desafio que o UE completa antes de qualquer SAR de timeout recuperaria o assinante.

Veja a métrica hss_cx_auth_abandon_total para visibilidade de desafios abandonados ou falhados.

Um S-CSCF Servindo Por Assinatura

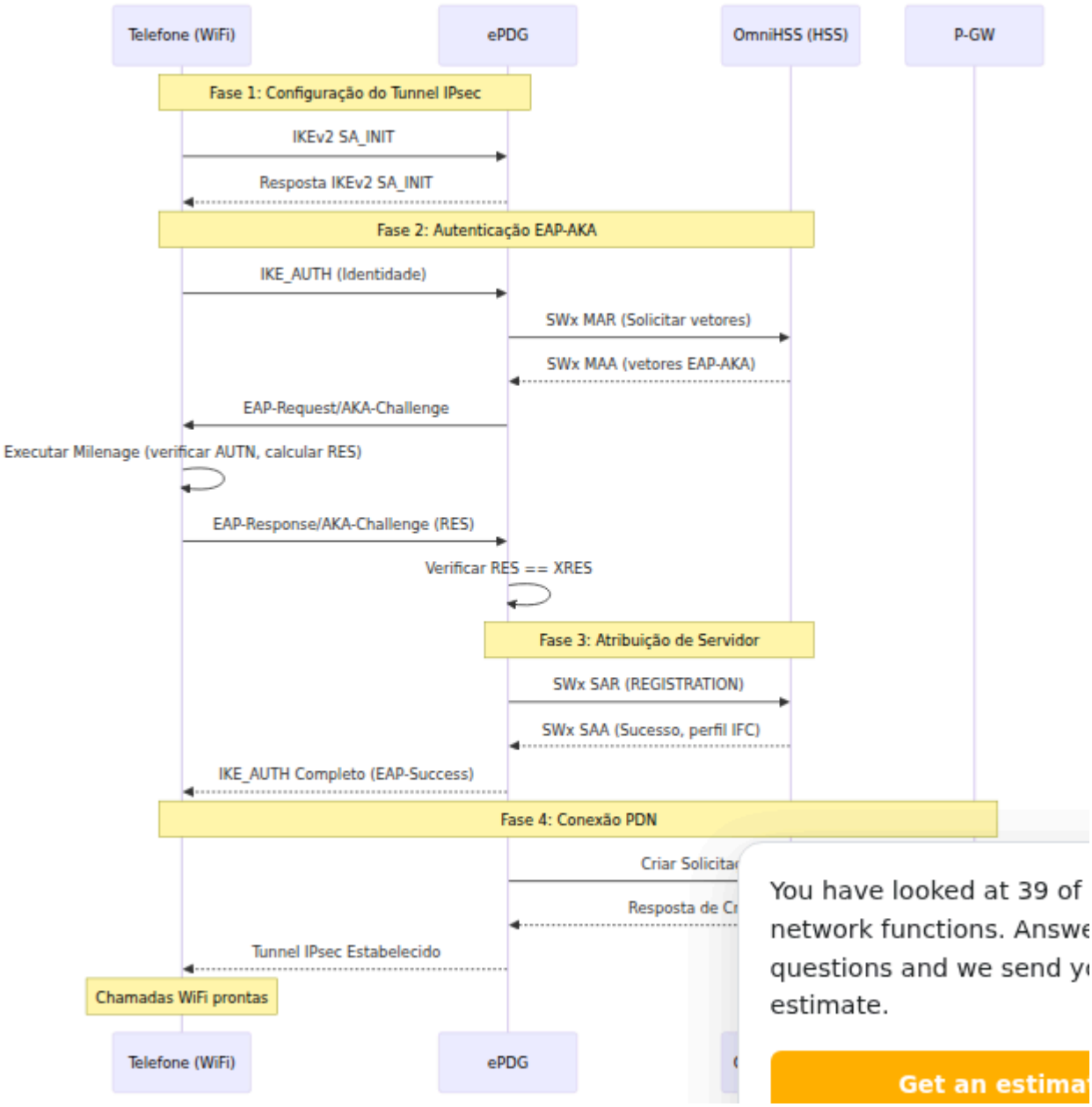
Uma assinatura IMS compartilhada agrupa vários IMSIs ou vários MSISDNs sob uma única identidade (veja Recursos Multi-MSISDN e Multi-IMSI). Exatamente um S-CSCF deve atender toda a assinatura. Se dois membros se registrarem em nós diferentes, uma chamada de terminação alcançaria um nó que não possui vínculo para a parte chamada.

O HSS mantém um nome S-CSCF compartilhado na assinatura IMS, chamado de âncora:

1. O primeiro membro se registra. Nenhuma âncora é definida, então o UAA retorna uma seleção (uma URI de pool ou um nó sob `:random_peer`).

2. Um nó real aceita o registro e envia seu SAR. O HSS ancla a assinatura a esse nó.
3. Cada membro posterior lê a âncora. Seu UAA retorna o nó ancorado, então cada membro chega ao mesmo S-CSCF.
4. A âncora é liberada quando o último membro se desregistra.

O HSS nunca ancla uma URI de pool. Uma URI de pool nomeia o pool, não um nó, então o HSS a mantém apenas o tempo suficiente para que o I-CSCF a resolva. Se uma âncora mantiver uma URI de pool de uma liberação anterior, o próximo SAR a substituirá pelo nó real que está servindo.



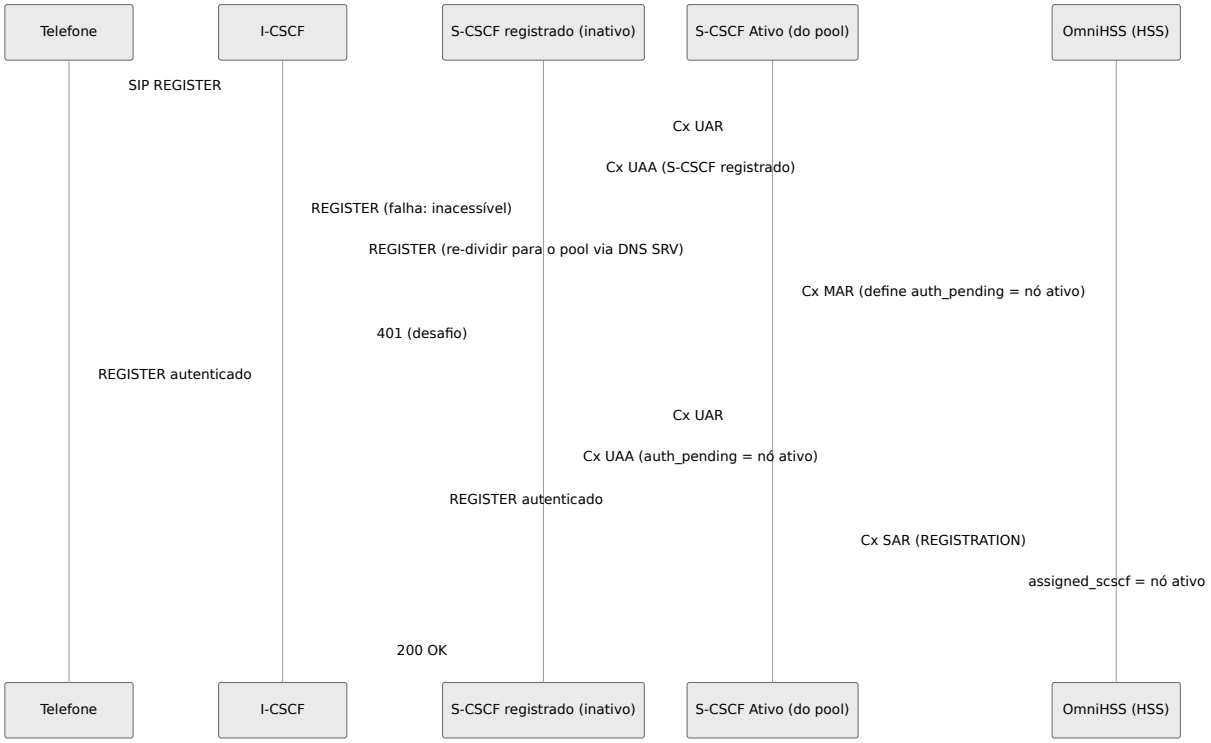
You have looked at 39 of
network functions. Answe
questions and we send y
estimate.
Get an estima

Recuperação de Falha do S-CSCF

Se o S-CSCF registrado não for acessível, o I-CSCF reenvia o REGISTER para o pool de S-CSCF. Um S-CSCF ativo então atende o assinante. Isso requer a rota de fallback do pool do I-CSCF e a preferência `auth_pending_scscf` do HSS juntas.

Existem dois modos de falha:

Modo de falha	Como se apresenta	Recuperação
Nome S-CSCF irreconhecível (um nome desativado ou obsoleto nos dados do HSS)	DNS retorna NXDOMAIN. O encaminhamento do I-CSCF falha antes de criar um ramo.	O I-CSCF reenvia para a URI do pool. DNS SRV seleciona um S-CSCF ativo.
S-CSCF reconhecível, mas inativo (o serviço S-CSCF está fora do ar)	Conexão TCP recusada. O ramo falha de forma assíncrona.	O I-CSCF reenvia para a URI do pool a partir de sua rota de falha. `use_dst_blocklist` e `use_dns_failover` fazem o DNS ignorar o nó inativo.



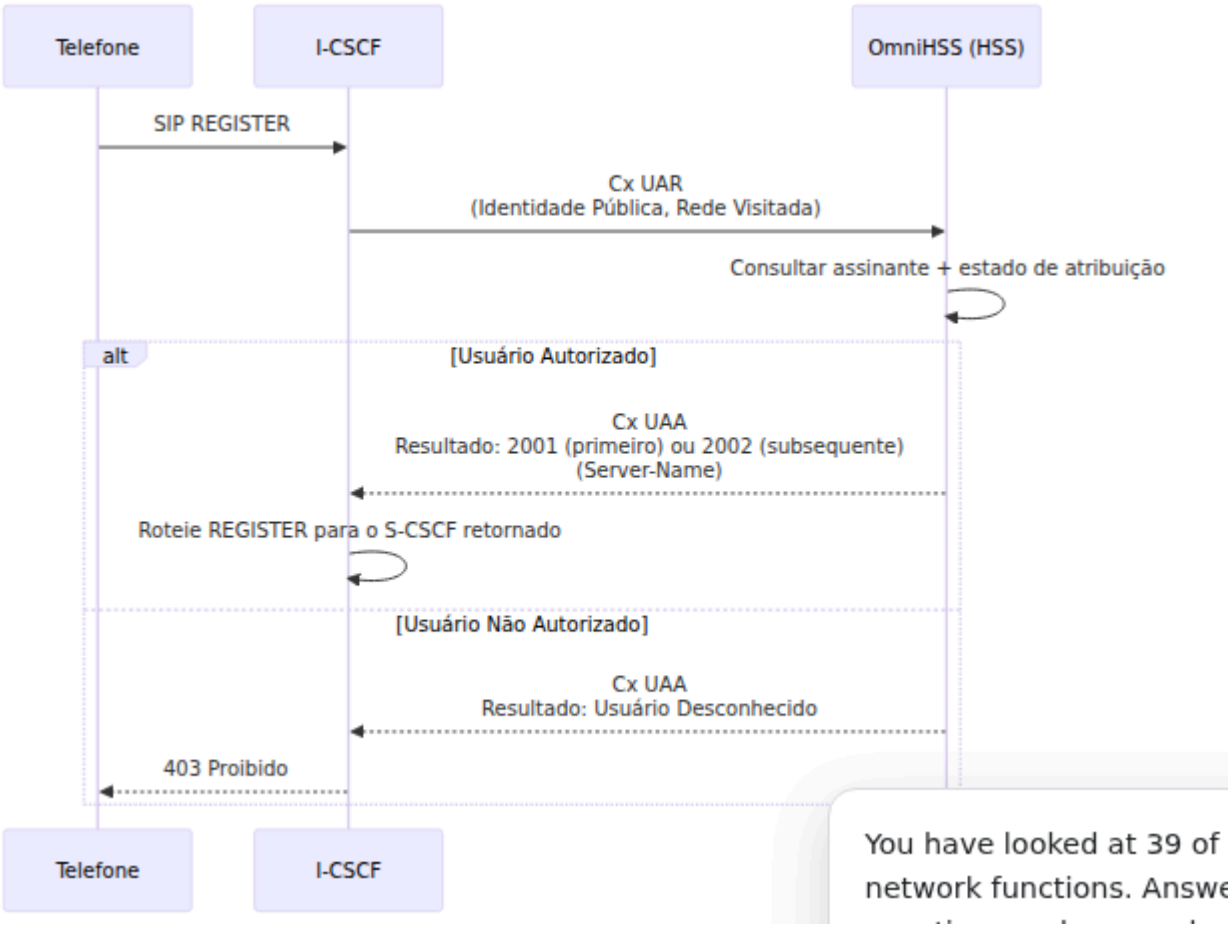
O REGISTER autenticado deve alcançar o mesmo S-CSCF ativo que emitiu o desafio. A preferência `auth_pending_scscf` no UAA faz isso acontecer: após o nó ativo enviar seu MAR, o HSS retorna esse nó no próximo UAR, não o nó inativo registrado. Sem a preferência, os dois REGISTERs iriam para membros diferentes do pool e o ciclo não convergiria.

Limitação. Se o S-CSCF for encerrado durante o ciclo de desafio IMS-AKA, o UE é re-desafiado pelo novo S-CSCF com um novo nonce. O UE se recupera em seu próximo registro. Isso é inerente a uma mudança de S-CSCF no meio do desafio, porque o vetor de autenticação em andamento não pode ser transferido para um S-CSCF diferente.

Para habilitar essa recuperação, defina `selection_method: :pool_uri` (veja Configuração de Seleção do S-CSCF) e implemente a rota de fallback do I-CSCF.

Solicitação de Informação de Localização (LIR/LIA)

O I-CSCF consulta qual S-CSCF está atendendo o usuário.

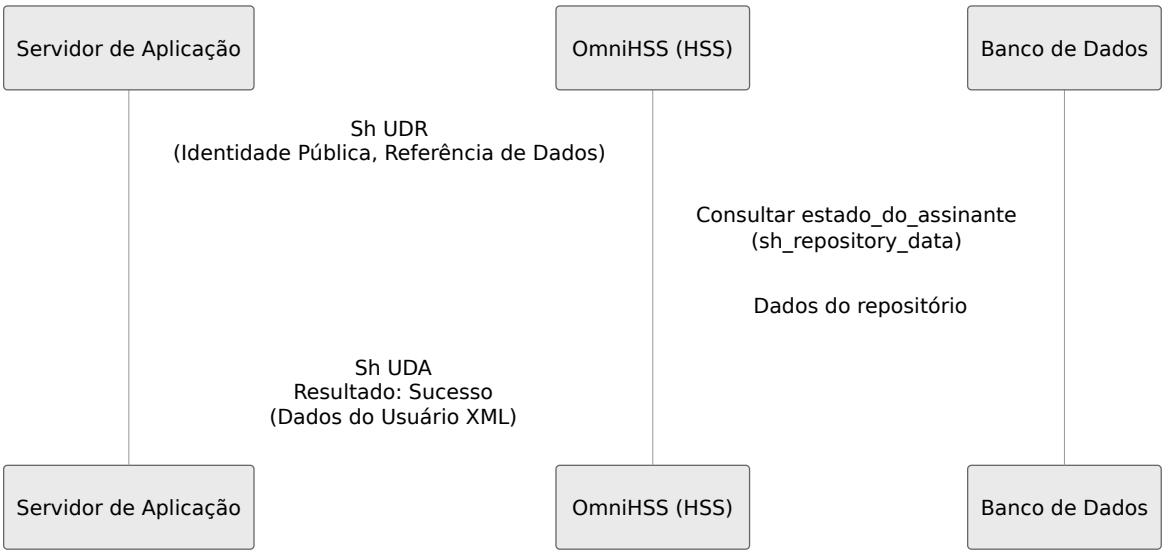


Interface Sh (Dados de Perfil IMS)

Veja Sh RepositoryData (Dados Transparentes) para o modelo de dados, regras de SequenceNumber/concorrência e a API de provisionamento REST para documentos de MMTEL-Services / IMS-ODB-Information.

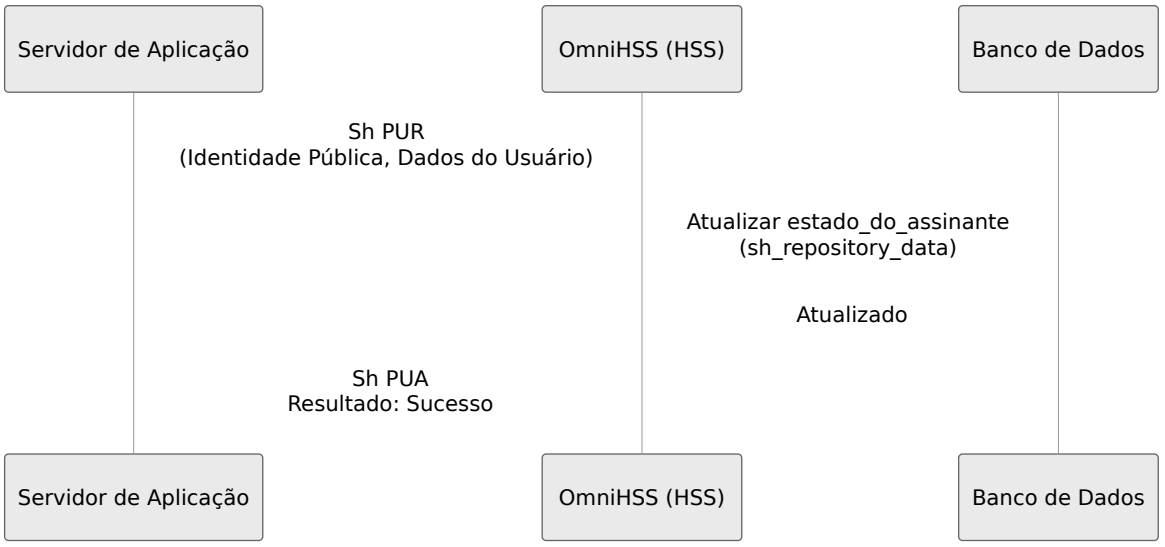
Solicitação de Dados do Usuário (UDR/UDA)

O Servidor de Aplicação solicita dados do perfil do assinante.



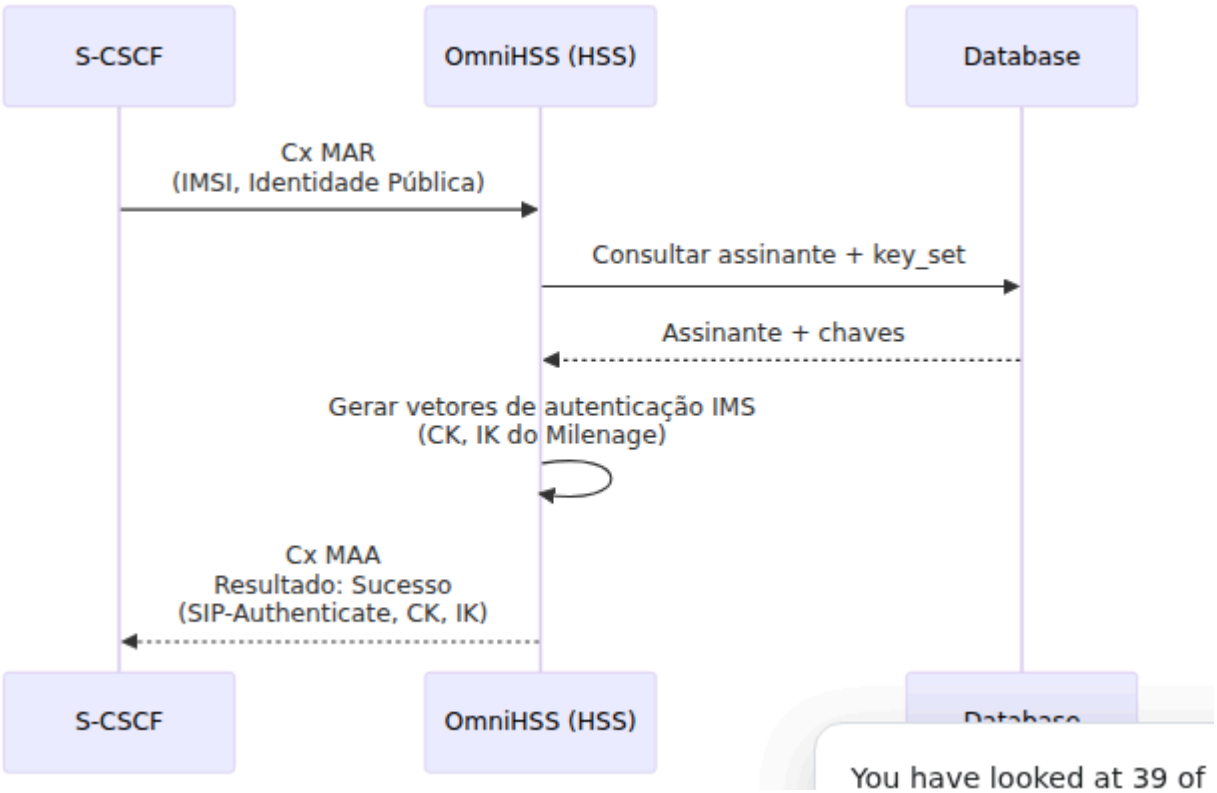
Solicitação de Atualização de Perfil (PUR/PUA)

O Servidor de Aplicação atualiza os dados do perfil do assinante.



Solicitação de Notificações de Assinatura (SNR/SNA)

O Servidor de Aplicação se inscreve para mudanças de perfil.



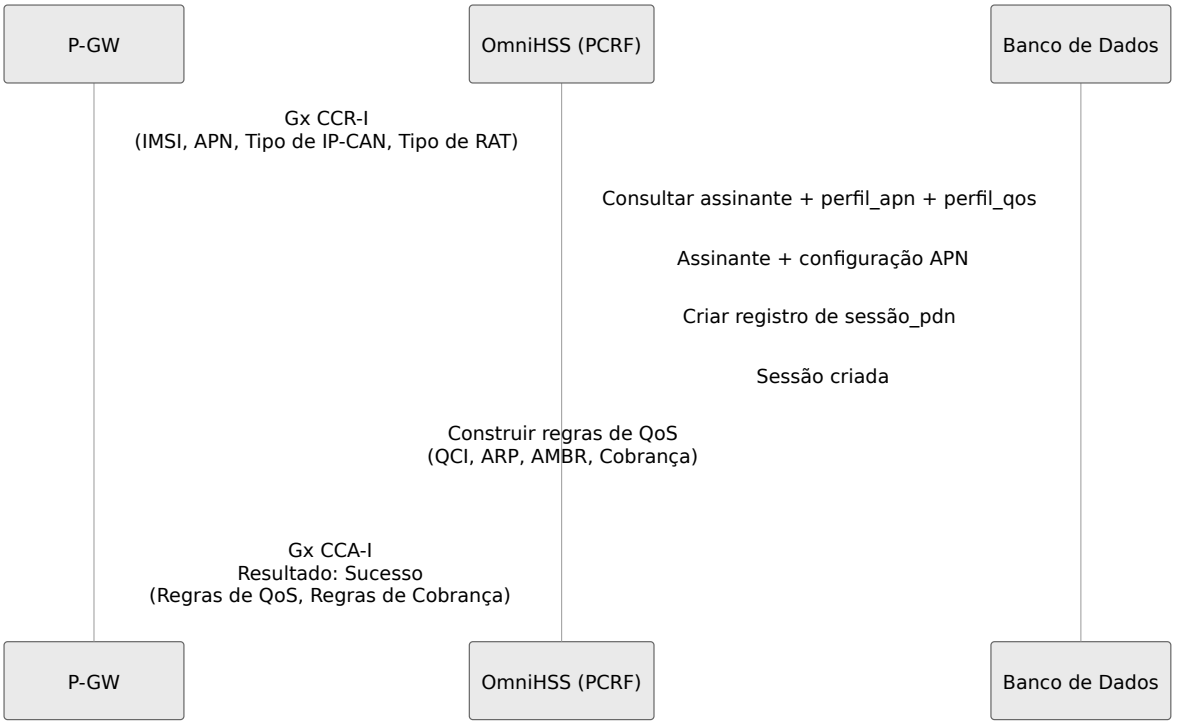
Interface Gx (Controle de Políticas)

OmniHSS funciona como o PCRF (Função de Regras de Políticas e Cobrança) através da interface Gx.

Veja Documentação do PCRF para arquitetura detalhada, configuração de políticas e gerenciamento de QoS.

Solicitação de Controle de Crédito - Inicial (CCR-I/CCA-I)

O P-GW solicita regras de políticas quando a sessão PDN é estabelecida.

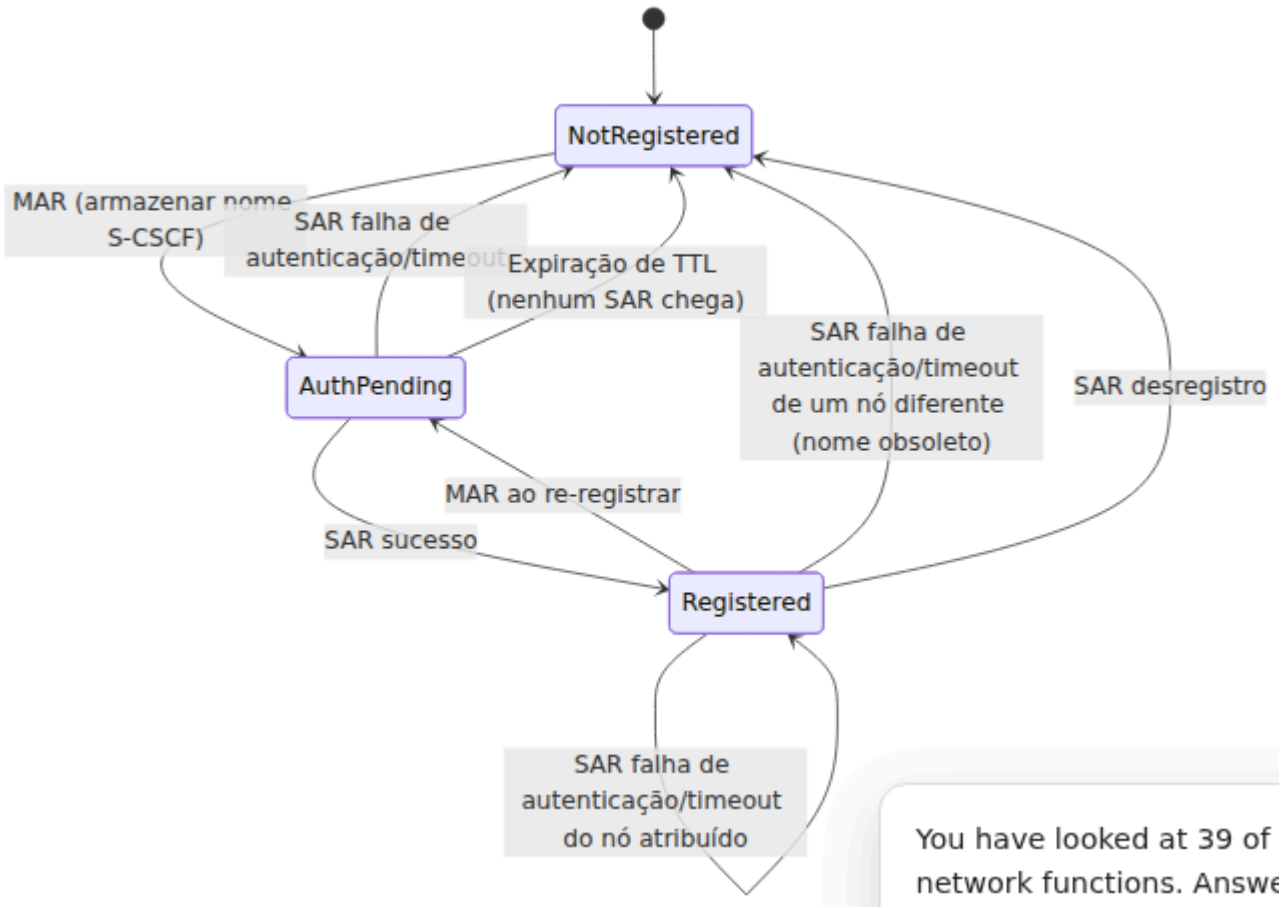


AVPs Chave:

- Solicitação: Subscription-Id (IMSI), Called-Station-Id (APN), RAT-Type, IP-CAN-Type
- Resposta: QoS-Information (QCI, ARP, AMBR), Charging-Rule-Install

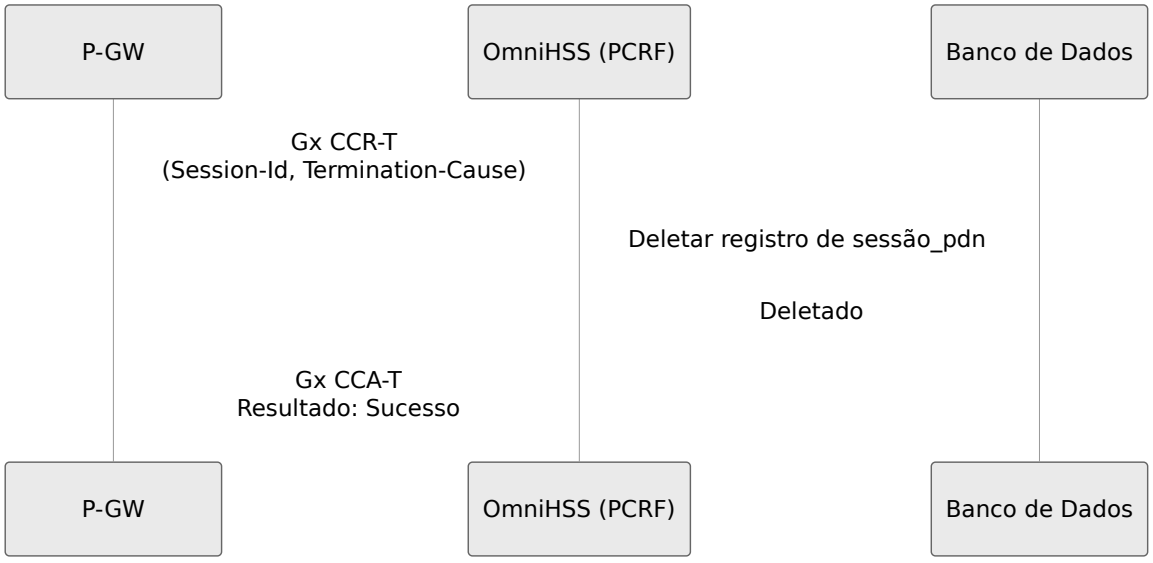
Solicitação de Controle de Crédito - Atualização (CCR-U/CCA-U)

O P-GW notifica sobre mudanças na sessão.



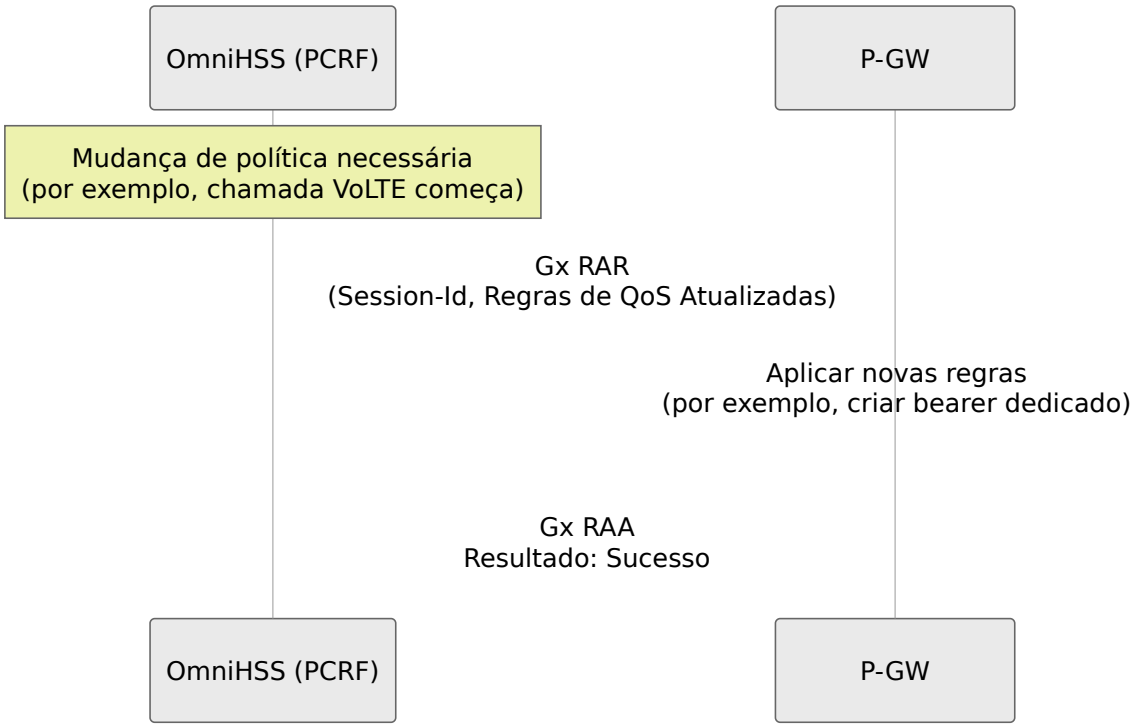
Solicitação de Controle de Crédito - Terminar (CCR-T/CCA-T)

O P-GW notifica quando a sessão PDN termina.



Solicitação de Reautenticação (RAR/RAA)

OmniHSS (PCRF) inicia a atualização de políticas para o P-GW.



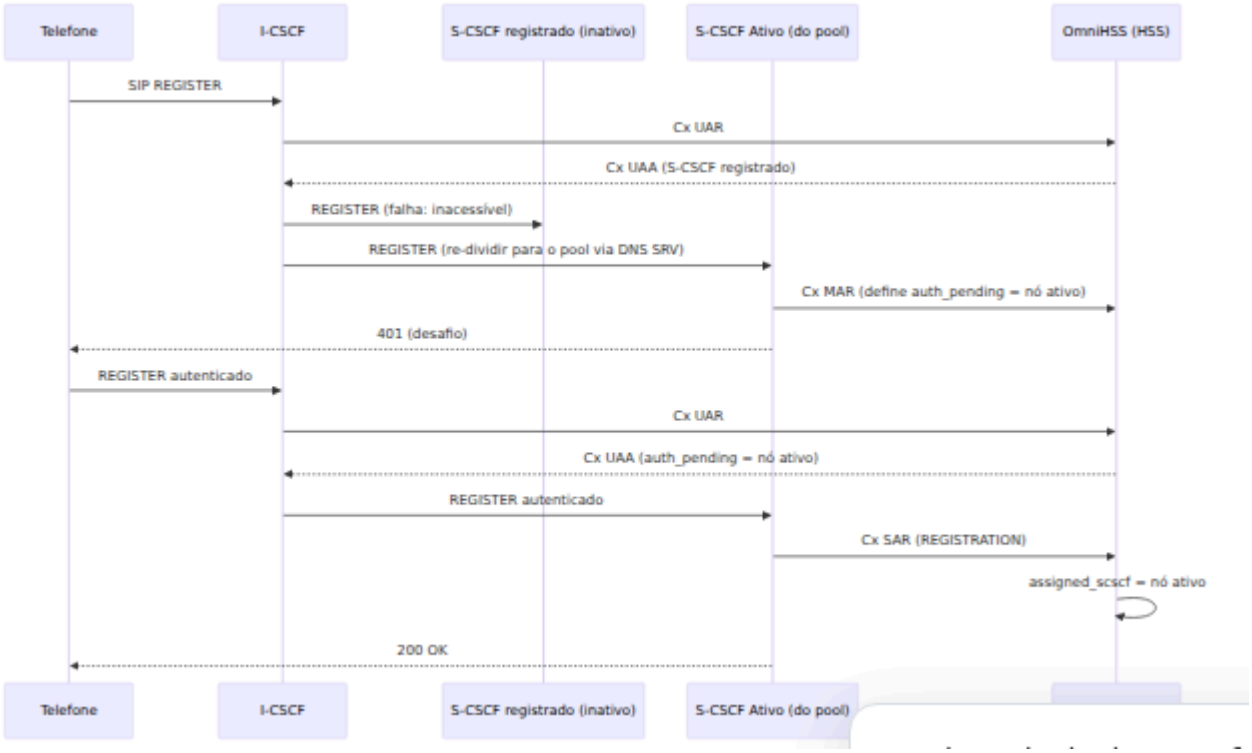
Interface Rx (Política de Mídia IMS)

OmniHSS funciona como o PCRF através da interface Rx para autorização de mídia IMS.

Veja Documentação do PCRF para fluxos de chamadas VoLTE detalhados e autorização de mídia.

Solicitação AA (AAR/AAA)

O P-CSCF solicita autorização de mídia para a sessão IMS.

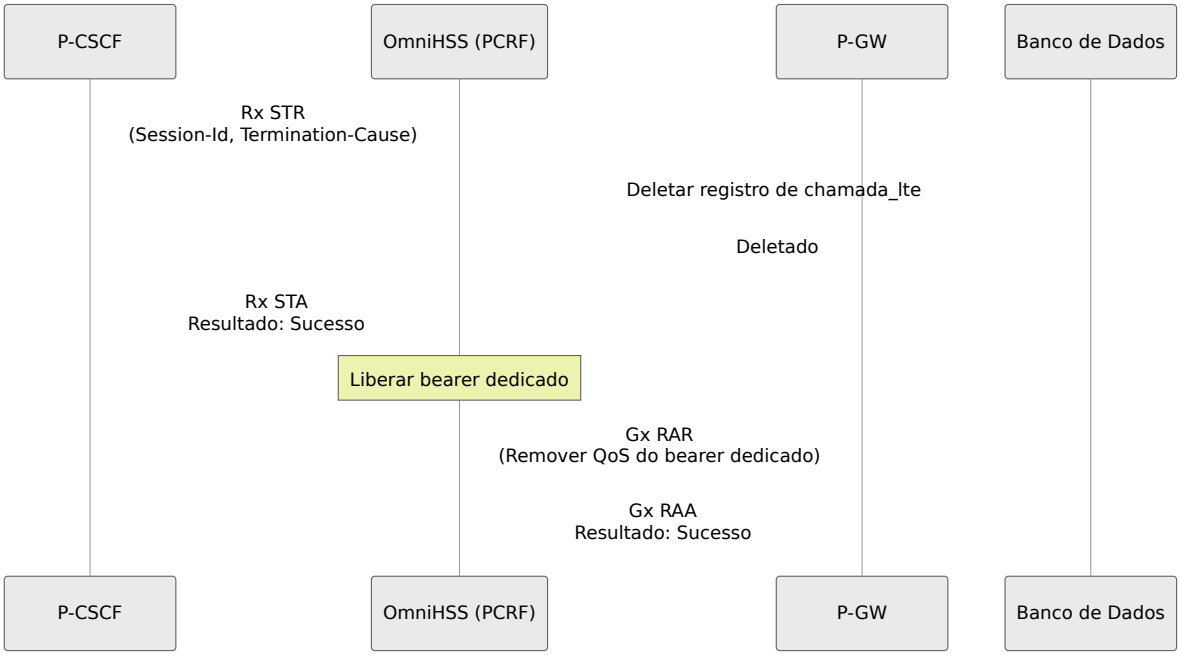


Informações Chave:

- Analisar SDP para determinar codec e largura de banda
- Calcular largura de banda necessária (UL/DL)
- Criar filtros SDF para fluxos de mídia
- Acionar bearer dedicado via Gx RAR

Solicitação de Término de Sessão (STR/STA)

O P-CSCF notifica quando a sessão IMS termina.

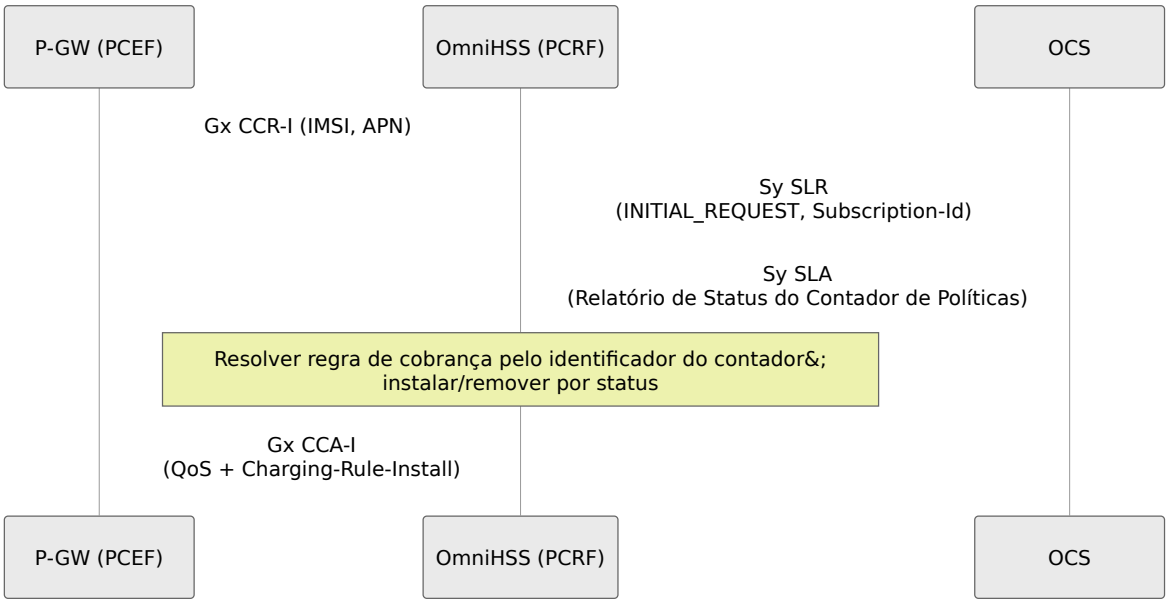


Interface Sy (Controle de Limite de Gastos)

OmniHSS, atuando como o PCRF, usa a interface Sy para obter o status do limite de gastos do assinante de um OCS externo (Sistema de Cobrança Online) e aplicar políticas baseadas em gastos sobre Gx. Sy usa o ID da aplicação Diameter `16777302` (3GPP TS 29.219). O HSS é o cliente Sy; o OCS é o servidor. Veja Interface Sy para a referência completa (mapeamento de contadores de políticas, AVPs, configuração).

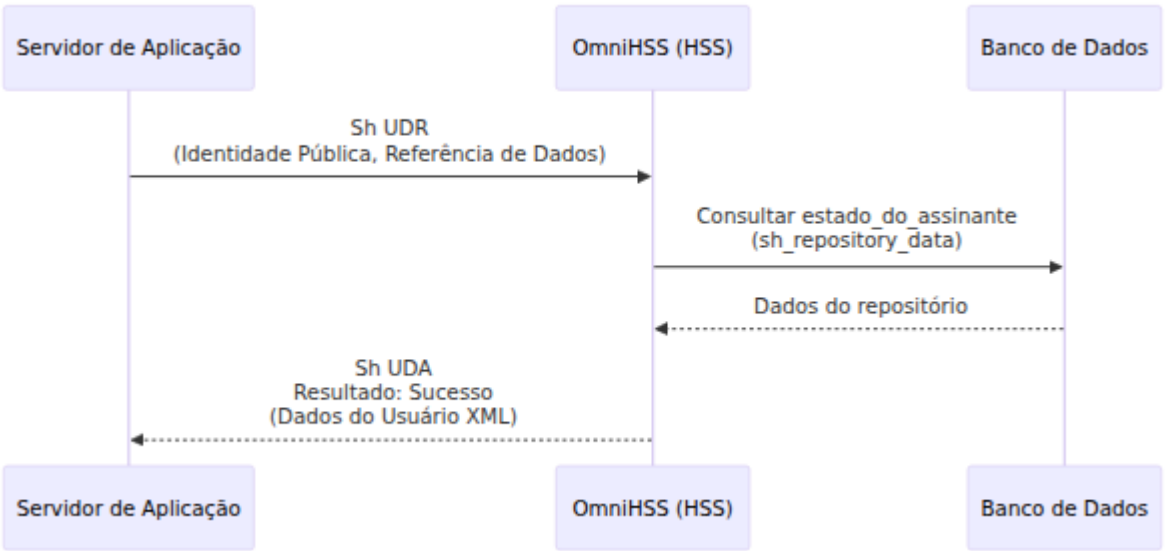
Solicitação de Limite de Gastos (SLR/SLA)

O PCRF abre uma sessão Sy por assinante para buscar o status atual dos contadores de políticas do OCS, e então incorpora as regras resultantes no Gx CCA que já retorna.



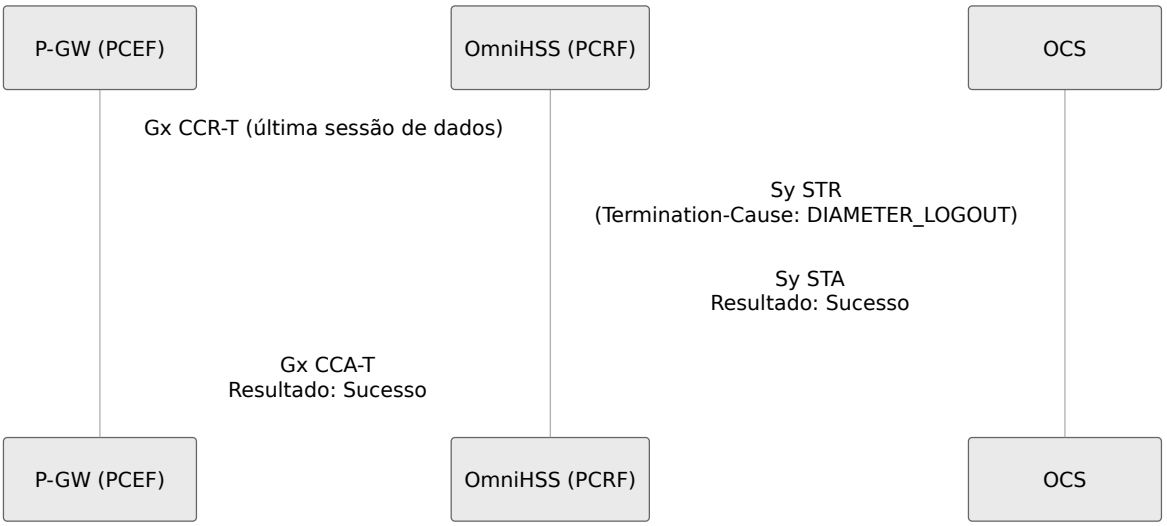
Notificação de Status de Gastos (SNR/SNA)

Quando o status de um contador muda (por exemplo, uma cota de dados é esgotada), o OCS notifica o PCRF, que empurra a mudança de política para o PGW com um Gx RAR.



Término de Sessão (STR/STA)

Quando a última sessão de dados do assinante termina, o PCRF fecha a sessão Sy.



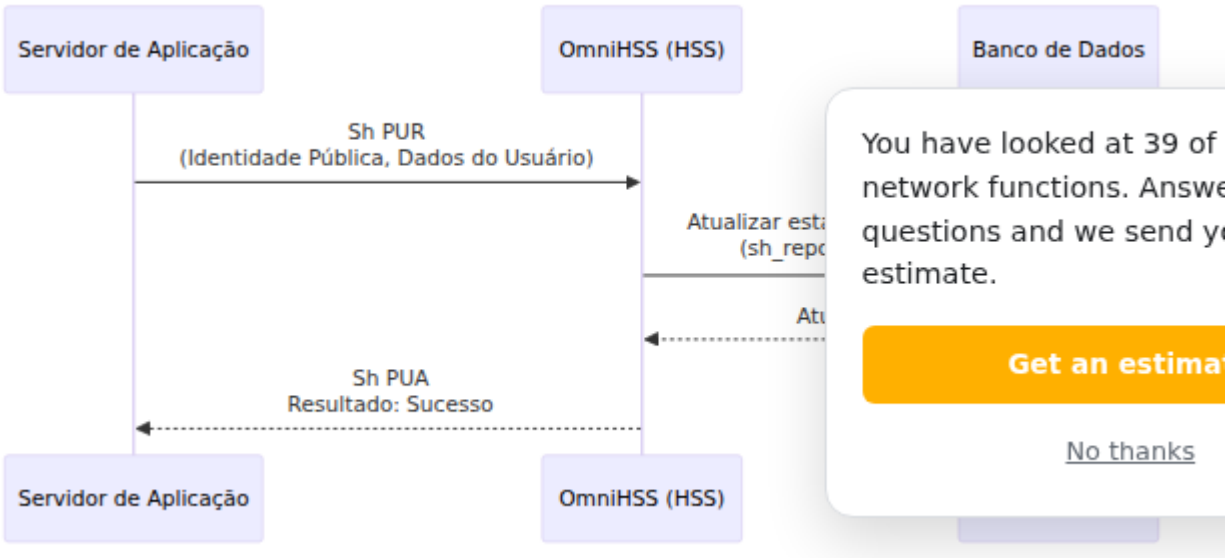
Interface S13 (EIR)

OmniHSS funciona como o EIR (Registro de Identidade de Equipamento) através da interface S13.

Veja Documentação do EIR para verificação detalhada de identidade de equipamentos, validação de IMEI, gerenciamento de listas negras e precedência de regras.

Solicitação de Verificação de Identidade do ME (ECR/ECA)

Cliente EIR externo (ou MME) solicita validação de equipamentos.



Valores de Status do Equipamento:

- **Na Lista Branca (0)** - Dispositivo permitido
- **Na Lista Negra (1)** - Dispositivo bloqueado
- **Na Lista Cinza (2)** - Dispositivo permitido, mas monitorado

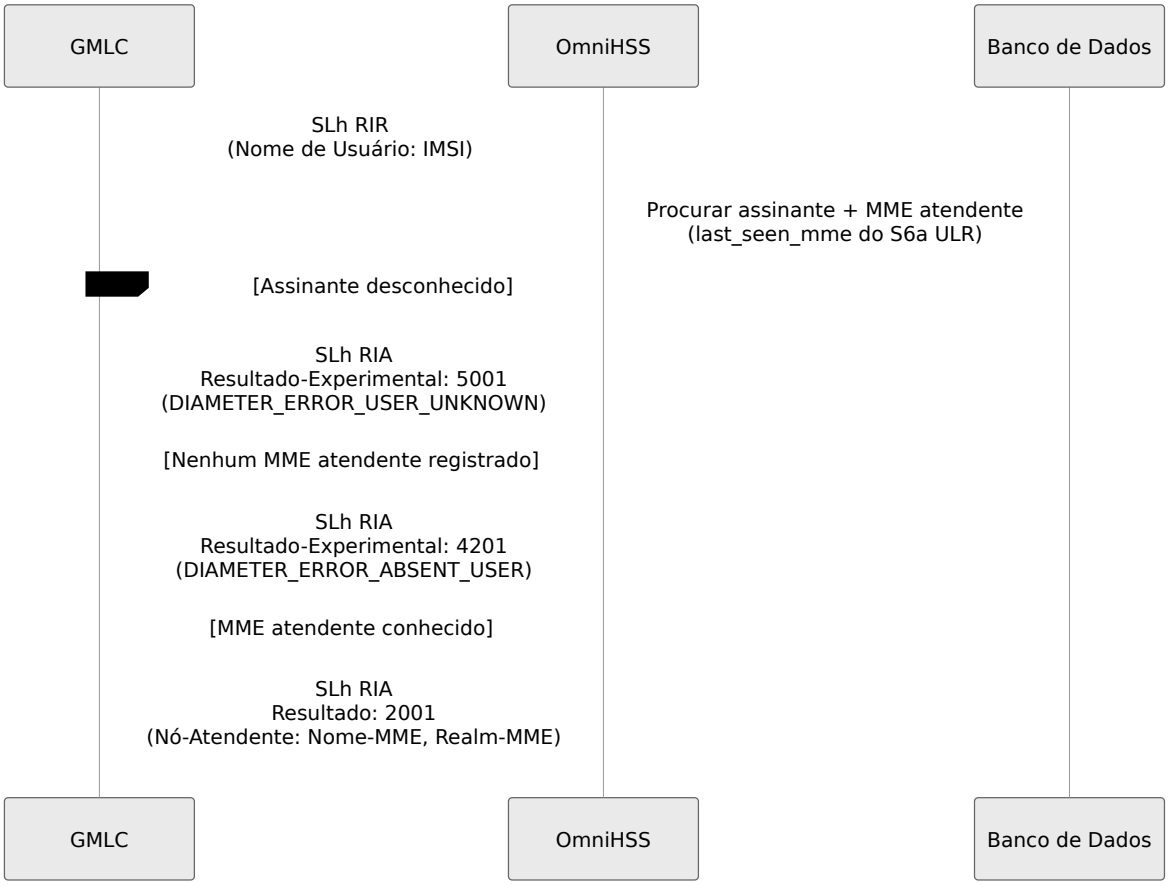
Se o IMSI na solicitação for desconhecido ou desativado, ou nenhuma regra corresponder e `unknown_equipment_behaviour` for `:reject_unknown_equipment`, o ECA carrega o Código de Resultado Experimental 5422 (DIAMETER_ERROR_EQUIPMENT_UNKNOWN) em vez de um Status do Equipamento.

Interface SLh (LCS / Serviços de Localização)

OmniHSS responde ao `LCS-Routing-Info-Request` (RIR) do SLh para que um GMLC possa descobrir qual MME atualmente atende um assinante antes de solicitar sua localização via SLg. O MME que atende é o que está registrado na atualização de registro S6a (Update-Location). O SLh usa o ID da aplicação Diameter `16777291` (3GPP TS 29.173).

Solicitação de Informação de Roteamento LCS (RIR/RIA)

GMLC solicita o nó que está atendendo um assinante.



Notas:

- O IMSI é retirado de `User-Name`; uma consulta MSISDN é usada como fallback.
- O `Nó-Atendente` retornado carrega o `Nome-MME`/`Realm-MME` registrado pela última vez pelos manipuladores S6a AIR/ULR (`subscriber_state.last_seen_mme` / `last_seen_realm`) - nenhum provisionamento extra é necessário.
- Um GMLC normalmente segue uma RIA bem-sucedida com um SLg Provide-Location-Request para o MME retornado.

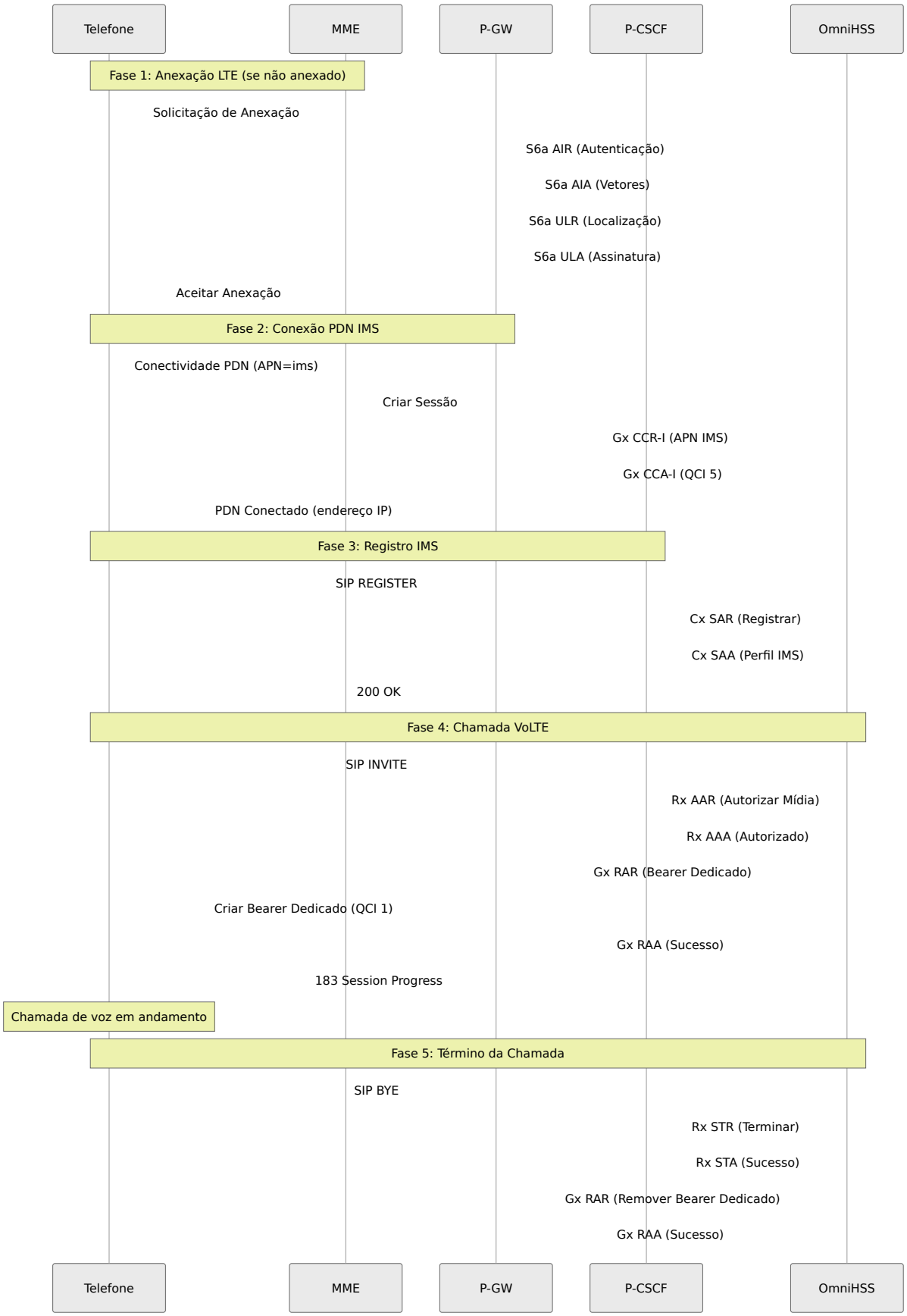
Códigos de Resultado:

- **2001** - Sucesso; `Nó-Atendente` retornado.
- **5001** - `DIAMETER_ERROR_USER_UNKNOWN`; IMSI/MSISDN não provisionado.

- **4201** - `DIAMETER_ERROR_ABSENT_USER`; assinante conhecido, mas nenhum MME atendente registrado (nunca anexado ou localização cancelada).

Fluxo de Chamada Completo: Chamada VoLTE

Configuração de chamada VoLTE de ponta a ponta mostrando várias interfaces.

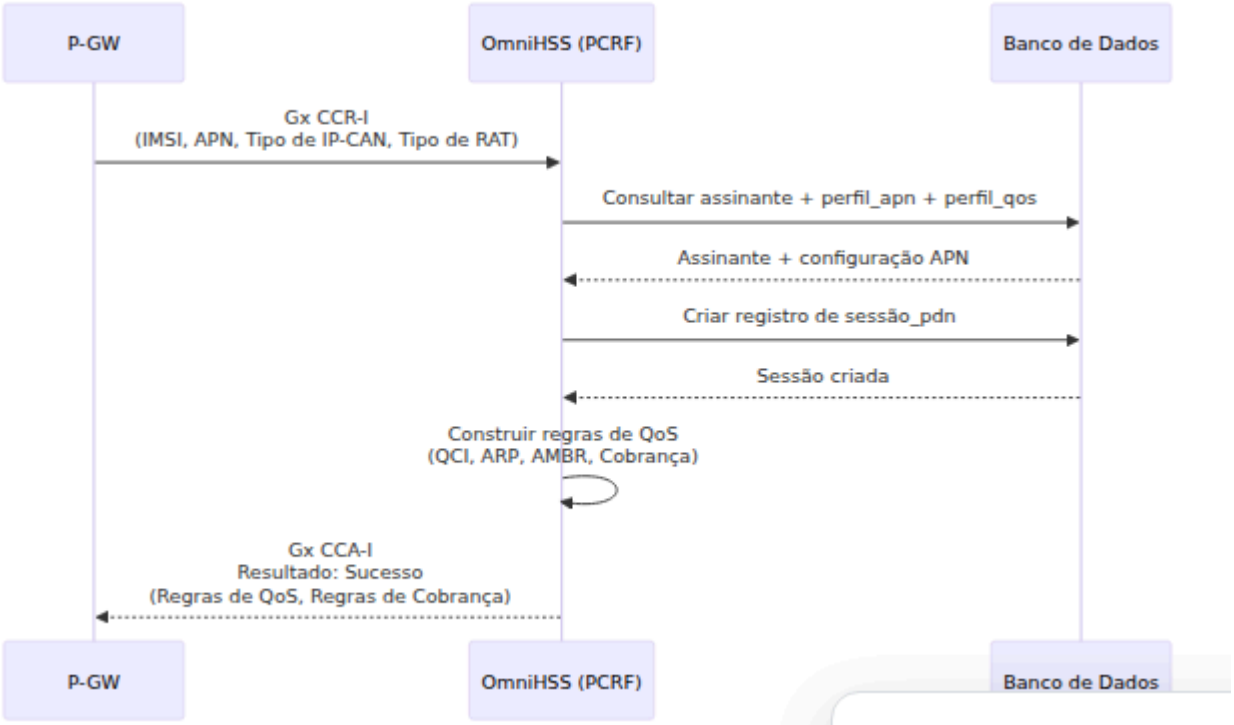

Telefone
MME
P-GW
P-CSCF
OmniHSS
Fase 1: Anexação LTE (se não anexado)
Solicitação de Anexação
S6a AIR (Autenticação)
S6a AIA (Vetores)
S6a ULR (Localização)
S6a ULA (Assinatura)
Aceitar Anexação
Fase 2: Conexão PDN IMS
Conectividade PDN (APN=ims)
Criar Sessão
Gx CCR-I (APN IMS)
Gx CCA-I (QCI 5)
PDN Conectado (endereço IP)
Fase 3: Registro IMS
SIP REGISTER
Cx SAR (Registrar)
Cx SAA (Perfil IMS)
200 OK
Fase 4: Chamada VoLTE
SIP INVITE
Rx AAR (Autorizar Mídia)
Rx AAA (Autorizado)
Gx RAR (Bearer Dedicado)
Criar Bearer Dedicado (QCI 1)
Gx RAA (Sucesso)
183 Session Progress
Chamada de voz em andamento
Fase 5: Término da Chamada
SIP BYE
Rx STR (Terminar)
Rx STA (Sucesso)
Gx RAR (Remover Bearer Dedicado)
Gx RAA (Sucesso)
Telefone
MME
P-GW
P-CSCF
OmniHSS

Interface Zh (Arquitetura de Inicialização Genérica)

A interface Zh habilita a Arquitetura de Inicialização Genérica (GBA) conectando a Função do Servidor de Inicialização (BSF) ao HSS para autenticação de assinantes e derivação de chaves.

Solicitação de Autenticação Multimídia (MAR/MAA)

O BSF solicita vetores de autenticação GBA para inicialização.



AVPs Chave:

- Solicitação: Nome de Usuário (IMPI), Identidade Pública, Item de Dados de Autenticação SIP (esquema de autenticação), Reino-Destino
- Resposta: Item de Dados de Autenticação SIP (SIP-Authenticate, SIP-Authorization, Chave de Confidencialidade, Chave de Integridade), GBA-UserSecSettings

Casos de Uso do GBA:

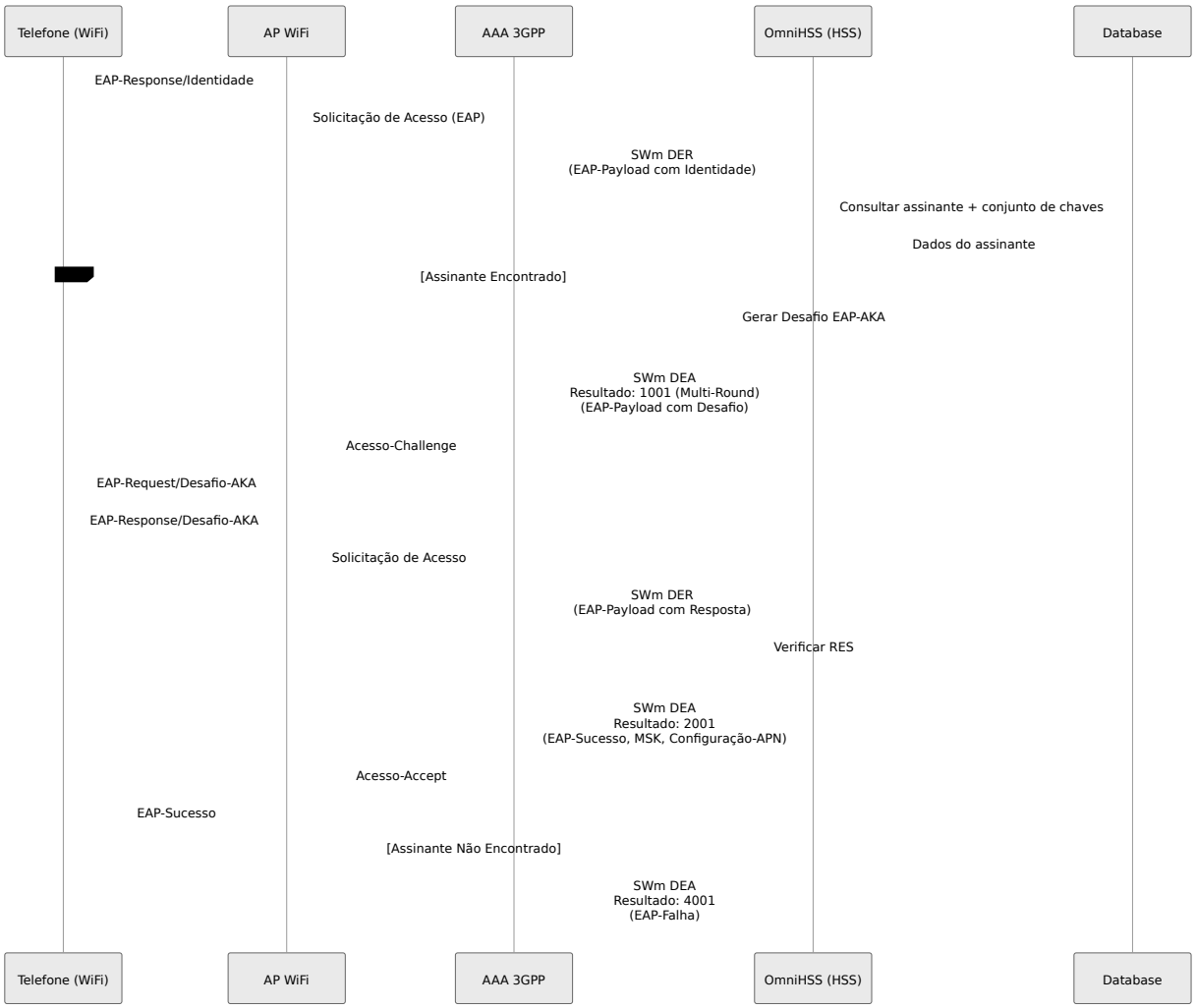
Aplicativo	Descrição
MBMS	Gerenciamento de chaves do Serviço de Multicast de Transmissão Móvel
HTTPS	Provisionamento de certificados baseado em GBA
NAF	Autenticação da Função de Aplicação de Rede Genérica
Presença	Autenticação do serviço de presença segura

Interface SWm (Acesso Não Confiável Não-3GPP)

A interface SWm habilita a autenticação EAP-AKA para redes de acesso não confiáveis não-3GPP (por exemplo, WiFi via ePDG) conectando o Servidor AAA 3GPP ao HSS.

Solicitação Diameter-EAP (DER/DEA)

Troca de autenticação EAP para acesso WiFi.

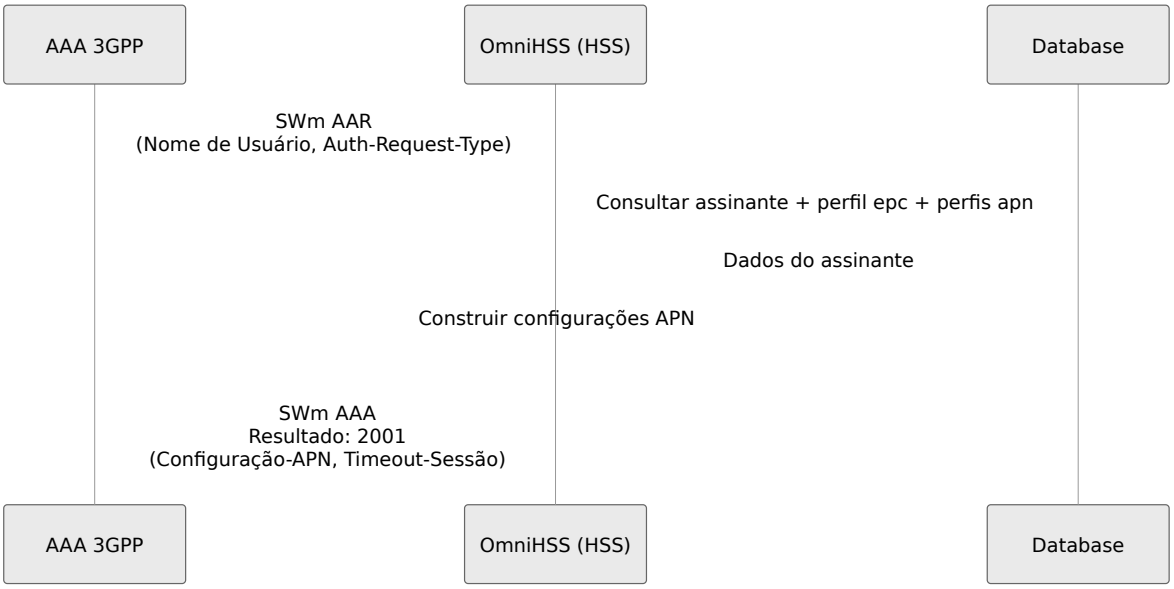


AVPs Chave:

- Solicitação: Session-Id, Auth-Application-Id, EAP-Payload, Auth-Request-Type, Nome de Usuário (opcional)
- Resposta: EAP-Payload, Chave-Mestre-EAP (MSK), Configuração-APN, Timeout-Sessão

Solicitação AA (AAR/AAA)

Autorização após autenticação EAP bem-sucedida.

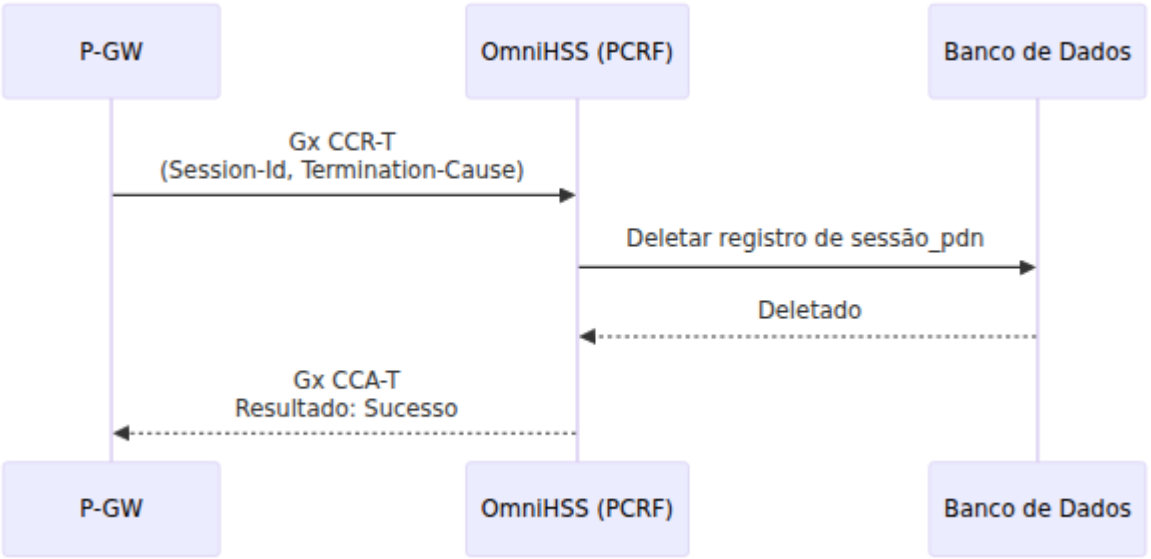


AVPs Chave:

- Solicitação: Session-Id, Auth-Application-Id, Nome de Usuário, Auth-Request-Type, AAR-Flags
- Resposta: Configuração-APN, Características-de-Cobrança-3GPP, Timeout-Sessão

Solicitação de Término de Sessão (STR/STA)

Notificação de término de sessão WiFi.



Causas de Término:

Causa	Valor	Descrição
DIAMETER_LOGOUT	1	Logout do usuário
DIAMETER_SERVICE_NOT_PROVIDED	2	Serviço indisponível
DIAMETER_BAD_ANSWER	3	Erro de protocolo
DIAMETER_ADMINISTRATIVE	4	Término administrativo
DIAMETER_LINK_BROKEN	5	Conexão perdida
DIAMETER_AUTH_EXPIRED	6	Timeout de autenticação
DIAMETER_USER_MOVED	7	Usuário movido para outro acesso
DIAMETER_SESSION_TIMEOUT	8	Timeout de sessão

Solução de Problemas de Protocolos

Falhas de Autenticação (S6a AIR)

Verifique:

1. Conjunto de chaves configurado corretamente (Ki, OPC, AMF)
2. Sincronização do SQN (se falhas repetidas)
3. Regras de roaming permitem rede visitada

Falhas de Atualização de Localização (S6a ULR)

Verifique:

1. Perfil EPC existe e tem APNs configurados
2. Roaming permitido para serviços de dados
3. Formato da identidade do MME correto

Falhas de Registro IMS (Cx SAR)

Verifique:

1. Perfil IMS atribuído ao assinante
2. Modelo IFC XML válido
3. Seleção de S-CSCF configurada
4. MSISDNs atribuídos se usados no modelo

Falhas de Conexão PDN (Gx CCR-I)

Verifique:

1. APN existe na lista de APNs do perfil EPC
2. Perfil QoS do APN configurado
3. Tabela de sessão PDN não está cheia (se limites existirem)

← Voltar ao Guia de Operações

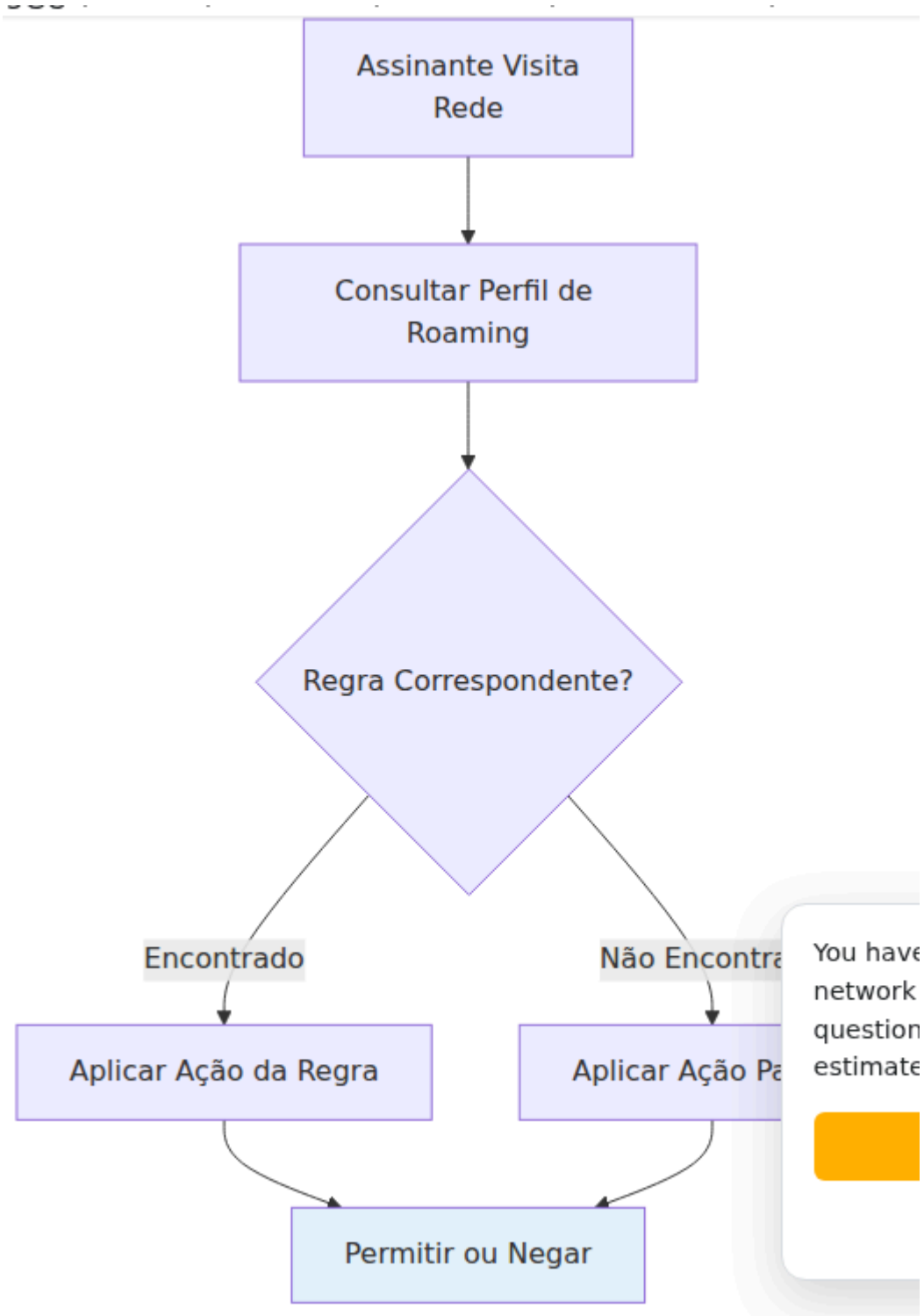
Controle de Roaming OmniHSS

← Voltar ao Guia de Operações

Visão Geral

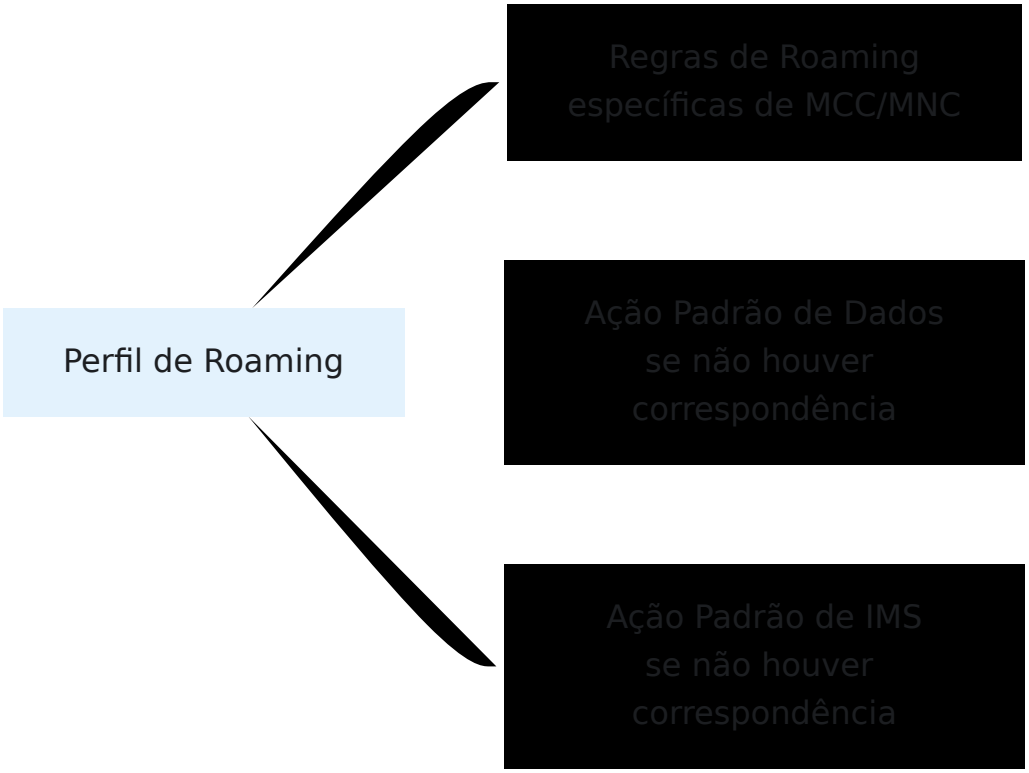
OmniHSS fornece controle de roaming granular, permitindo que você defina quais redes os assinantes podem acessar para serviços de dados e IMS ao roaming.

Fluxo de Controle de Roaming



Estrutura do Perfil de Roaming

Componentes



Regra de Roaming

Cada regra especifica a ação para uma rede específica (combinação MCC/MNC).

Campos:

- `name` - Nome descritivo
- `mcc` - Código do País Móvel (3 dígitos)
- `mnc` - Código da Rede Móvel (2-3 dígitos)
- `data_action` - "permitir" ou "negar"
- `ims_action` - "permitir" ou "negar"

Perfil de Roaming

Define o comportamento padrão e vincula às regras.

Campos:

- `name` - Nome do perfil
- `data_action_if_no_rules_match` - "permitir" ou "negar"
- `ims_action_if_no_rules_match` - "permitir" ou "negar"

Exemplos de Configuração

Permitir Todo o Roaming

```
# Criar perfil que permite tudo
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_profile": {
      "name": "Permitir Tudo",
      "data_action_if_no_rules_match": "allow",
      "ims_action_if_no_rules_match": "allow",
      "roaming_rules": []
    }
  }'
```

Negar Todo o Roaming

```
# Criar perfil que bloqueia tudo
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_profile": {
      "name": "Sem Roaming",
      "data_action_if_no_rules_match": "deny",
      "ims_action_if_no_rules_match": "deny",
      "roaming_rules": []
    }
  }'
```

Permitir Redes Específicas (Lista Branca)

```
# Criar regra da AT&T
RULE1=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Permitir AT&T",
      "mcc": "310",
      "mnc": "410",
      "data_action": "allow",
      "ims_action": "allow"
    }
  }' | jq -r '.response.id')

# Criar regra da Verizon
RULE2=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Permitir Verizon",
      "mcc": "311",
      "mnc": "480",
      "data_action": "allow",
      "ims_action": "allow"
    }
  }' | jq -r '.response.id')

# Criar perfil com negação por padrão e vincular regras
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"roaming_profile\": {
      \"name\": \"Apenas Operadoras dos EUA\",
      \"data_action_if_no_rules_match\": \"deny\",
      \"ims_action_if_no_rules_match\": \"deny\",
      \"roaming_rules\": [$RULE1, $RULE2]
    }
  }"
```

Permitir Dados, Bloquear Voz

```
# Criar regra que permite dados mas bloqueia IMS
curl -k -X POST https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Apenas Dados - T-Mobile",
      "mcc": "310",
      "mnc": "260",
      "data_action": "allow",
      "ims_action": "deny"
    }
  }'
```

Bloquear Redes Específicas (Lista Negra)

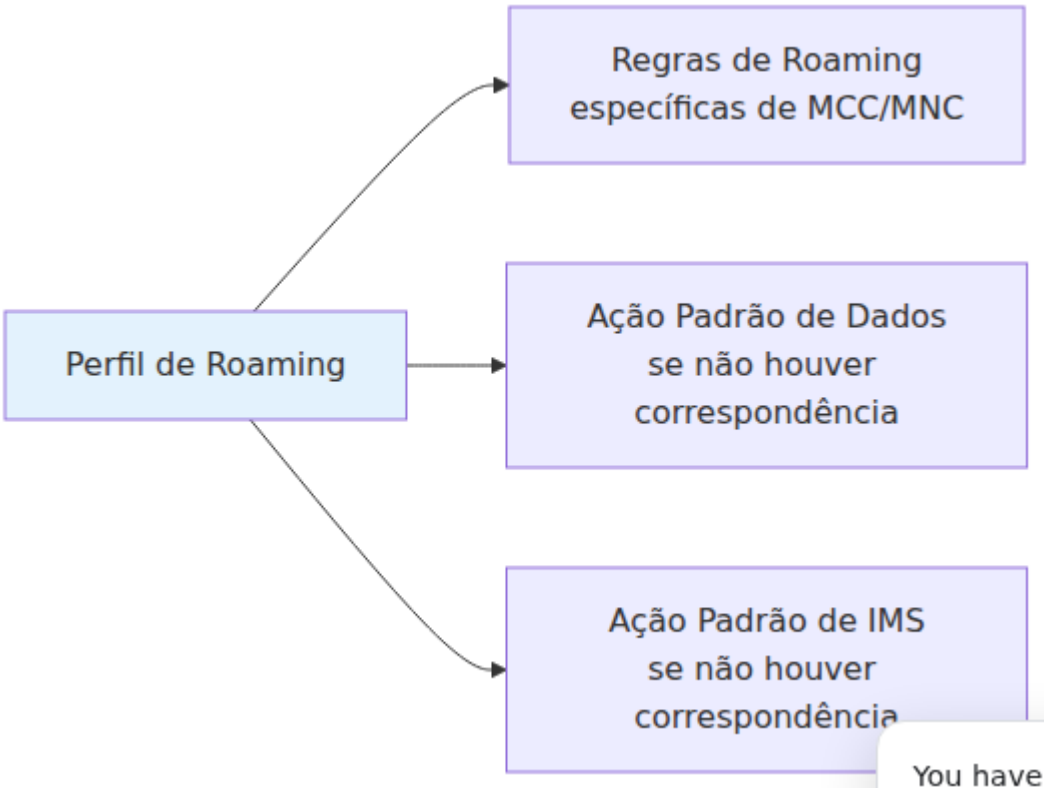
```
# Criar regra de bloqueio de rede cara
RULE=$(curl -k -X POST
https://hss.example.com:8443/api/roaming/rule \
  -H "Content-Type: application/json" \
  -d '{
    "roaming_rule": {
      "name": "Bloquear Rede Cara",
      "mcc": "206",
      "mnc": "01",
      "data_action": "deny",
      "ims_action": "deny"
    }
  }' | jq -r '.response.id')

# Criar perfil com permissão por padrão
curl -k -X POST https://hss.example.com:8443/api/roaming/profile \
  -H "Content-Type: application/json" \
  -d "{
    \"roaming_profile\": {
      \"name\": \"Bloquear Redes Caras\",
      \"data_action_if_no_rules_match\": \"allow\",
      \"ims_action_if_no_rules_match\": \"allow\",
      \"roaming_rules\": [$RULE]
    }
  }"
```

Cenários Comuns de Roaming

Cenário 1: Apenas Roaming Doméstico

O assinante pode roaming dentro do país de origem, mas não internacionalmente.

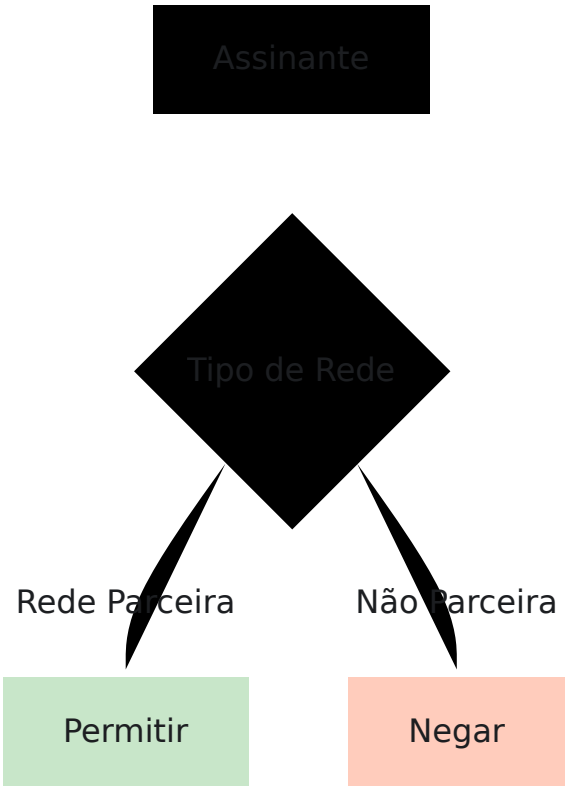


Configuração:

- Padrão: Negar todos
- Regras: Permitir todos os códigos MCC dos EUA (310, 311, 312, 313, 314, 315, 316)

Cenário 2: Apenas Parceiros de Roaming

O assinante só pode roaming em redes com acordos comerciais.

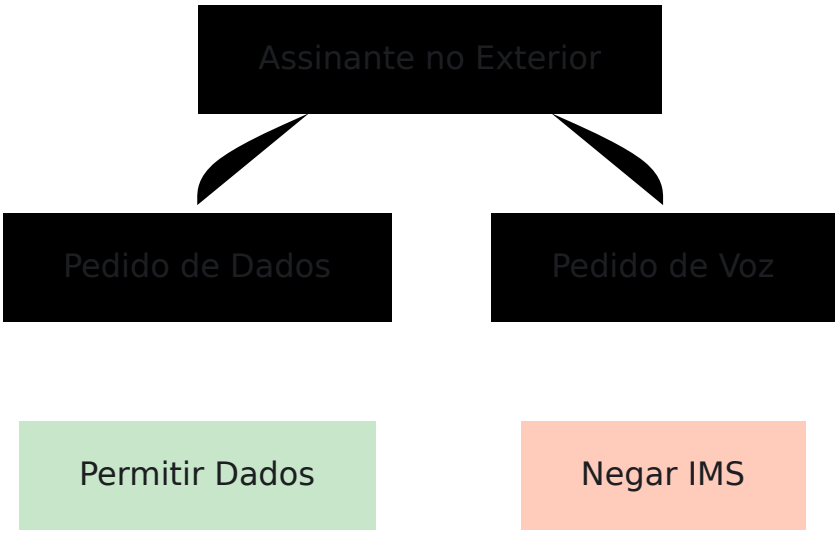


Configuração:

- Padrão: Negar todos
- Regras: Permitir cada rede parceira explicitamente (por MCC/MNC)

Cenário 3: Roaming de Dados, Sem Roaming de Voz

O assinante pode usar dados no exterior, mas deve usar WiFi para chamadas de voz.



Configuração:

- Regras: `data_action: "allow"`, `ims_action: "deny"`

Cenário 4: Acesso a Serviços de Emergência

Sempre permitir serviços de emergência, mesmo que o roaming esteja bloqueado.

Nota: O tratamento de serviços de emergência é tipicamente feito no nível do MME/rede. As regras de roaming do OmniHSS se aplicam a serviços normais.

Referência MCC/MNC

Códigos de País Comuns (MCC)

MCC	País	Redes
310-316	EUA	AT&T, Verizon, T-Mobile, etc.
302	Canadá	Rogers, Bell, Telus
234-235	Reino Unido	Vodafone, O2, EE
262	Alemanha	Deutsche Telekom, Vodafone
208	França	Orange, SFR, Bouygues
222	Itália	TIM, Vodafone, Wind
214	Espanha	Movistar, Vodafone

Operadoras Comuns dos EUA (MCC 310-316)

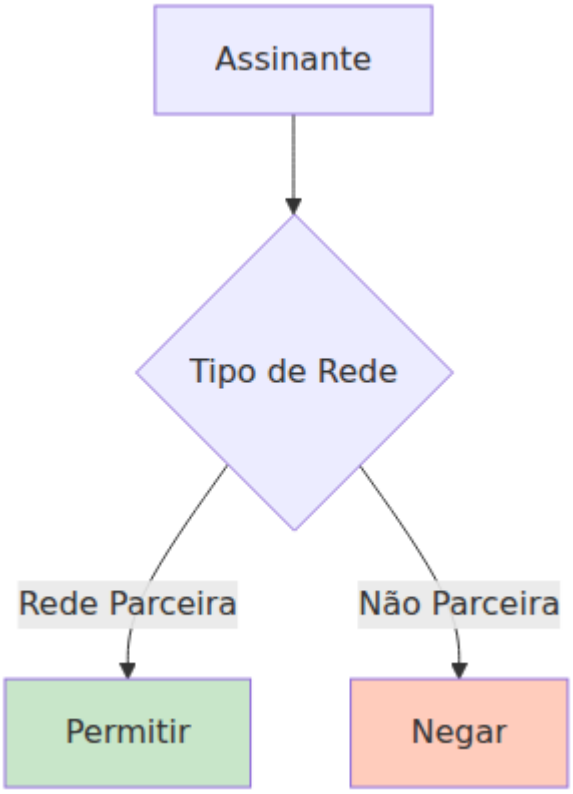
MCC	MNC	Operadora
310	410	AT&T
311	480	Verizon
310	260	T-Mobile
310	120	Sprint
313	380	(Rede de teste de exemplo)

Listas Completas: Veja ITU-T E.212 ou bancos de dados MCC/MNC

Pontos de Aplicação de Roaming

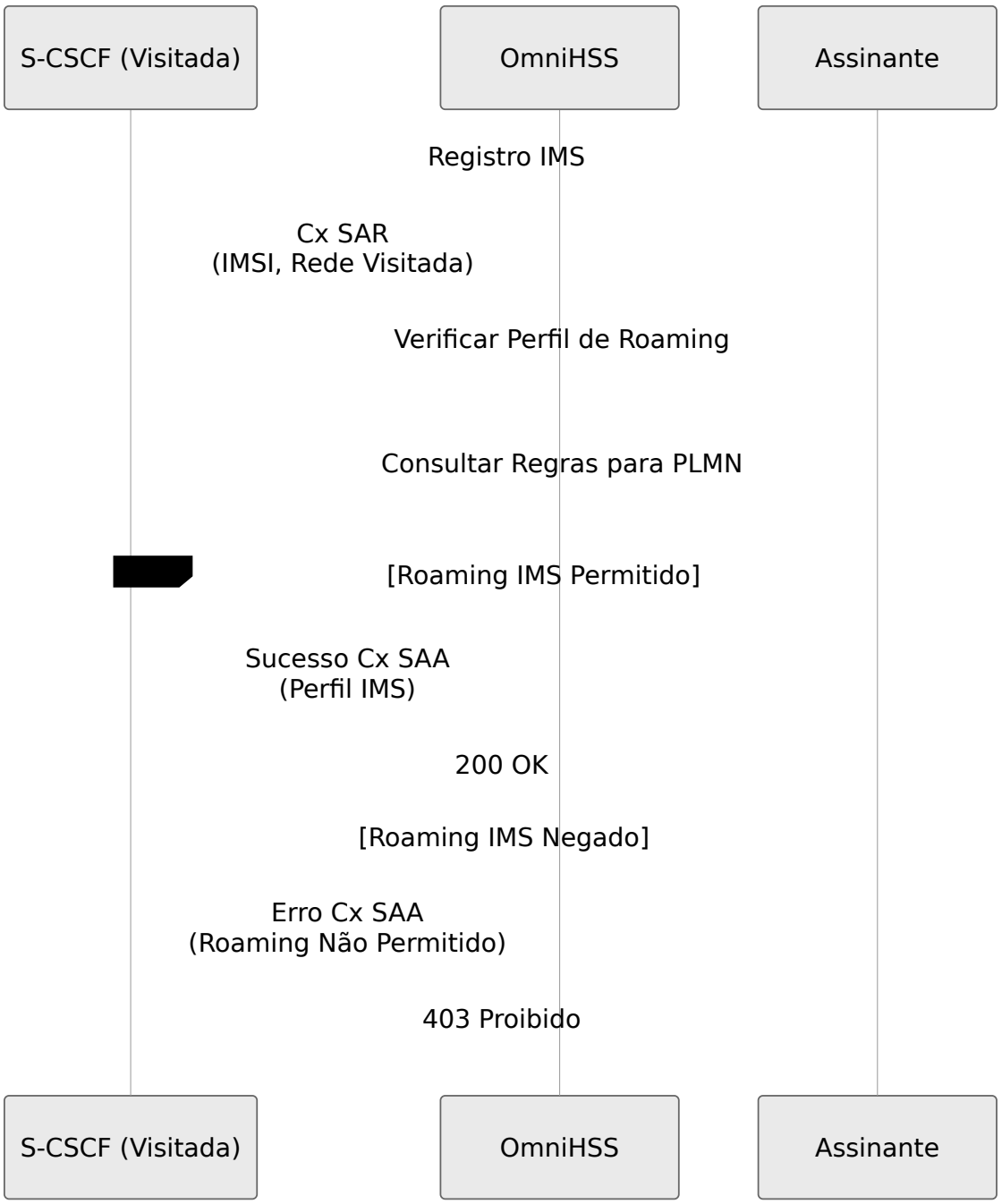
Interface S6a (Dados)

Quando o assinante se conecta à rede visitada:



Interface Cx (IMS)

Quando o assinante se registra no IMS na rede visitada:



Solução de Problemas de Roaming

Assinante Não Consegue Conectar na Rede Visitada

Verifique a atribuição do perfil de roaming:

- Consulte o banco de dados para visualizar o perfil de roaming atribuído ao assinante
- Verifique o nome do perfil e as configurações de ação padrão

Verifique se a regra existe para a rede visitada:

- Consulte o banco de dados para regras de roaming correspondentes ao MCC/MNC da rede visitada
- Verifique se uma regra existe para o perfil de roaming do assinante
- Verifique o valor de data_action para aquela rede específica

Assinante Consegue Conectar, Mas Não Registrar IMS

Verifique a ação IMS separadamente:

- Consulte as regras de roaming para a rede visitada
- Verifique os valores de data_action e ims_action
- Procure casos onde os dados são permitidos, mas IMS é negado

Comportamento de Roaming Inesperado

Revise os logs para verificações de roaming:

```
[info] Verificação de roaming: IMSI 001001123456789, PLMN Visitado
310-410
[info] Regra de roaming correspondente: "Permitir AT&T"
[info] Ação de dados: permitir, Ação IMS: permitir
```

Melhores Práticas

Design de Perfil

1. **Comece restritivo** - Padrão negar, permitir explicitamente parceiros

2. **Teste minuciosamente** - Verifique regras em laboratório antes da produção
3. **Documente regras** - Mantenha lista de redes permitidas e por quê
4. **Revise regularmente** - Atualize conforme os acordos de roaming mudam

Gestão de Regras

1. **Use nomes descritivos** - "Permitir-ATT-Apenas-Dados" não "Regra1"
2. **Verifique MCC/MNC** - Verifique códigos contra bancos de dados oficiais
3. **Considere ambos os serviços** - Pense em dados e IMS separadamente
4. **Monitore o uso** - Acompanhe quais redes os assinantes realmente visitam

Procedimentos Operacionais

1. **Mudanças de Emergência** - Tenha um procedimento para habilitar/desabilitar rapidamente o roaming
2. **Atualizações em Massa** - Planeje para atualizar os perfis de roaming de múltiplos assinantes
3. **Relatórios** - Acompanhe o uso de roaming e tentativas negadas
4. **Comunicação com o Cliente** - Notifique os clientes sobre mudanças na política de roaming

---

← Voltar ao Guia de Operações | Próximo: Fluxos de Protocolo →

Sh RepositoryData (Dados Transparentes)

OmniHSS atua como um **repositório transparente** para dados de serviço que os Servidores de Aplicação (AS) armazenam contra um assinante através da interface Sh. Documentos como **MMTEL-Services** (serviços suplementares / desvio de chamadas, espera de chamadas, OIP/OIR, bloqueio) e **IMS-ODB-Information** (bloqueio determinado pelo operador) são escritos pelo AS, mantidos pelo HSS e lidos sob demanda.

"Transparente" tem um significado preciso em 3GPP TS 29.328: o HSS entende esses dados **sintaticamente, mas não semanticamente**. É XML opaco de propriedade do AS. OmniHSS armazena e retorna a carga útil `ServiceData` **byte por byte**. Ele nunca reanalisa ou reserializa a carga útil. Um documento que um AS etags ou checksums (por exemplo, XCAP/MMTel) é retornado exatamente como foi escrito.

Os mesmos documentos também podem ser gerenciados fora de banda por sistemas de provisionamento/CRM através da REST Provisioning API.

Índice

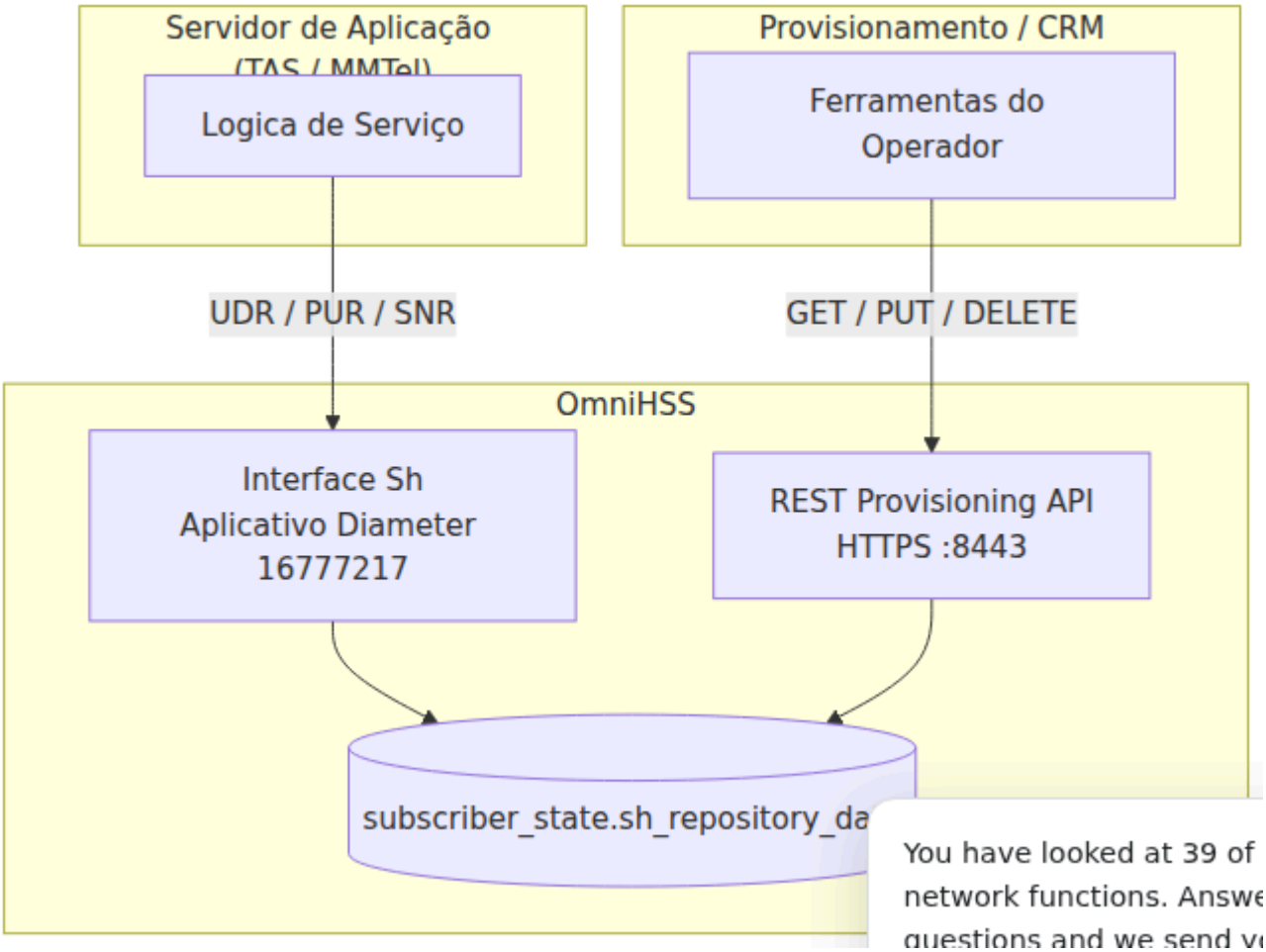
- Conceitos
- Modelo de Dados
- Fluxos de Mensagens Sh
- SequenceNumber e Concorrência
- REST Provisioning API
- Tabelas de Referência
- Solução de Problemas

Conceitos

Um assinante pode ter qualquer número de **documentos transparentes** independentes, cada um identificado por uma **Service-Indication** (o "nome do arquivo"). Um documento tem três partes:

Elemento	Proprietário	Significado
Service-Indication	Gerenciado pelo HSS	Nomeia o documento (por exemplo, `MMTEL-Services`, `IMS-ODB-Information`).
SequenceNumber	Gerenciado pelo HSS	Contador de versão usado para concorrência otimista. Veja SequenceNumber e Concorrência.
ServiceData	De propriedade do AS (opaco)	A carga útil XML em si. Armazenada e retornada literalmente.

O HSS apenas interpreta Service-Indication e SequenceNumber. O corpo `ServiceData` é tratado como bytes opacos.



Modelo de Dados

Todos os documentos transparentes de um assinante são mantidos em um único campo, `subscriber_state.sh_repository_data`, como um objeto JSON indexado por Service-Indication. Cada entrada armazena a versão e a carga útil literal:

```
{
  "MMTEL-Services":      { "sequence_number": "0", "service_data":
"<MMTelServices …>…</MMTelServices>" },
  "IMS-ODB-Information": { "sequence_number": "1", "service_data":
"<OdbForImsOrientedServices xmlns=\"odb.mmtel.omnitou.ch\"/>" }
}
```

Campo	Descrição
`sequence_number`	A versão atual deste documento, conforme aceito pela última vez pelo HSS. Retornado em cada `UDA` e usado para validar atualizações.
`service_data`	O corpo XML opaco, armazenado exatamente como o AS enviou (sem normalização de namespace, ordem de atributos ou espaços em branco). Um documento sem elemento `ServiceData` (os *dados de repositório vazios* da especificação) omite este campo.

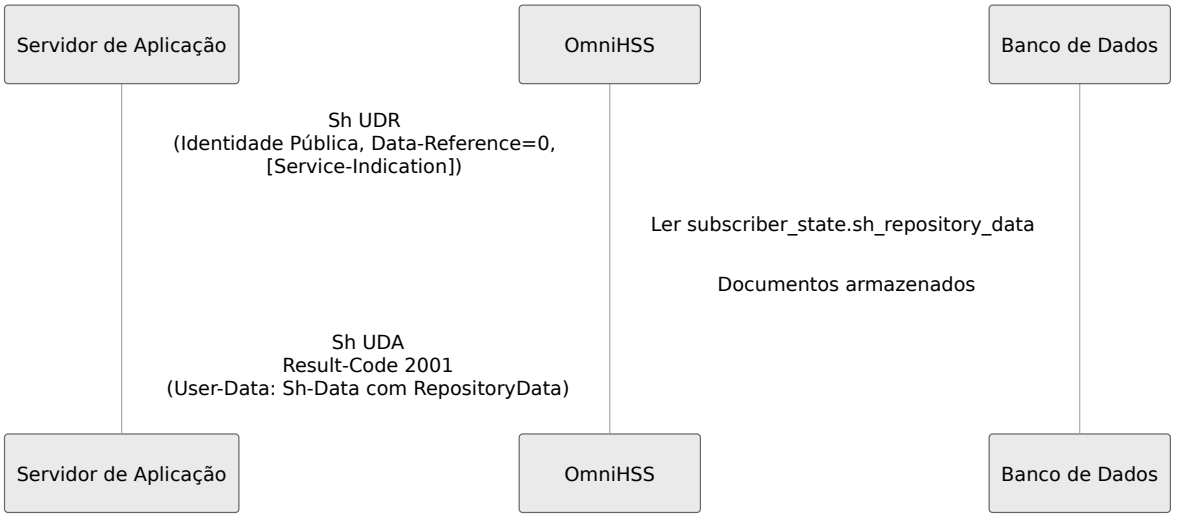
Esse modelo significa que um único assinante pode ter muitos documentos distintos, cada um versionado de forma independente, sem um esquema fixo imposto pelo HSS.

Fluxos de Mensagens Sh

RepositoryData é transportado no AVP `User-Data` como um documento XML `Sh-Data`. RepositoryData é selecionado com **valor Data-Reference 0**. Um AVP **Service-Indication** opcional restringe uma solicitação a um único documento.

Solicitação de Dados do Usuário (UDR/UDA): Leitura

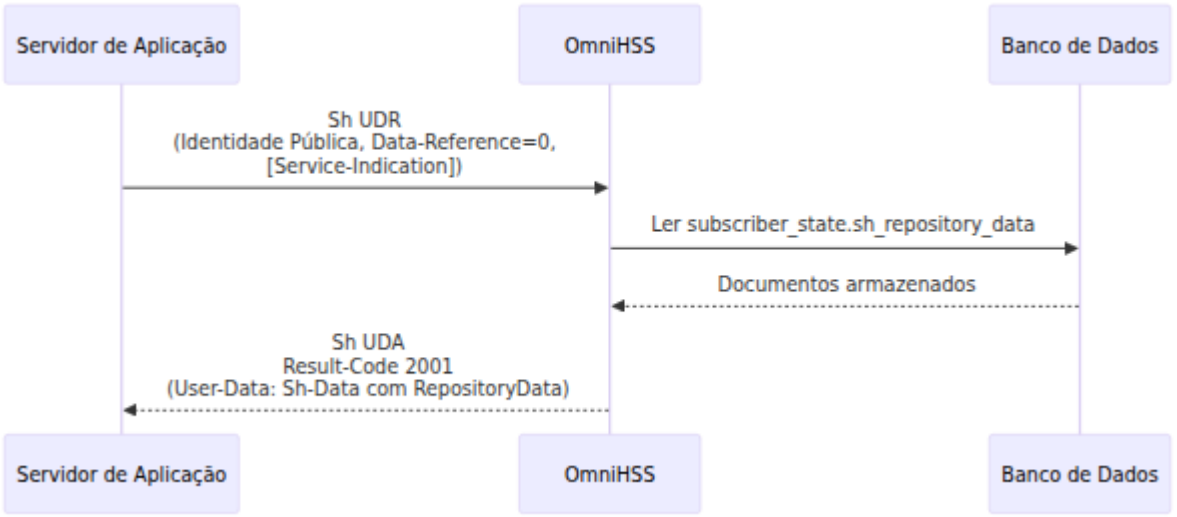
O AS lê um ou todos os documentos transparentes. Com uma Service-Indication, apenas o documento correspondente é retornado; sem uma, todos os documentos armazenados são retornados, cada um como seu próprio elemento `<RepositoryData>`.



Um assinante desconhecido ou não-IMS é respondido com **Experimental-Result 5001 (DIAMETER_ERROR_USER_UNKNOWN)**.

Solicitação de Atualização de Perfil (PUR/PUA): Escrita

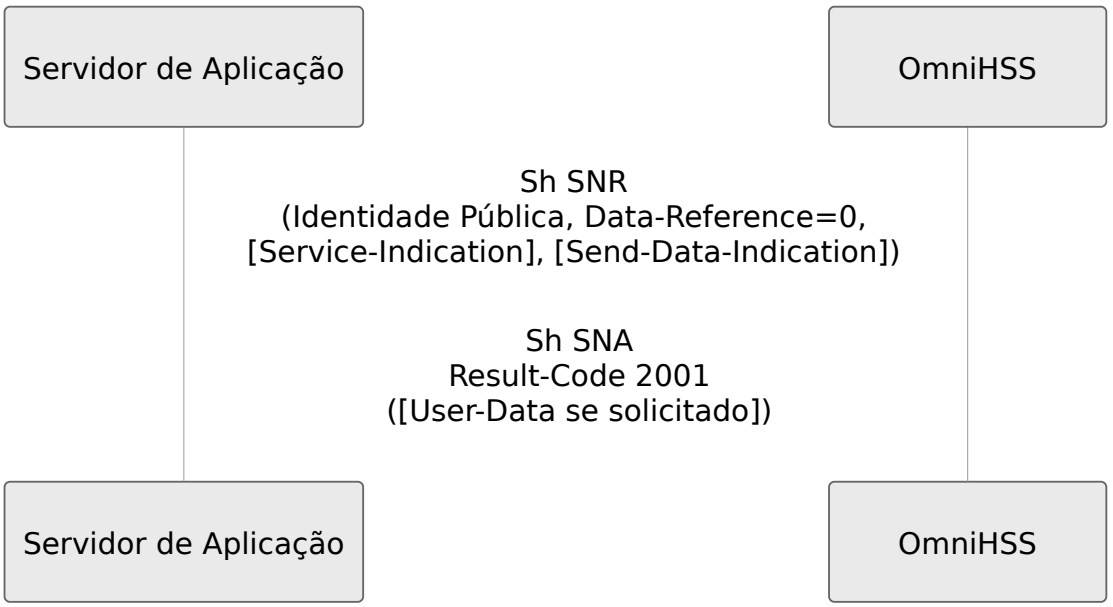
O AS cria ou atualiza um único documento. A carga útil `ServiceData` é armazenada literalmente. A atualização é aceita apenas se seu SequenceNumber for válido (veja abaixo).



Quando um PUR se resolve em múltiplos assinantes (uma Identidade Pública com curinga), a atualização é aplicada de forma atômica: se algum documento estiver fora de sincronia, nenhum é persistido.

Solicitação de Notificações de Inscrição (SNR/SNA): Inscrever

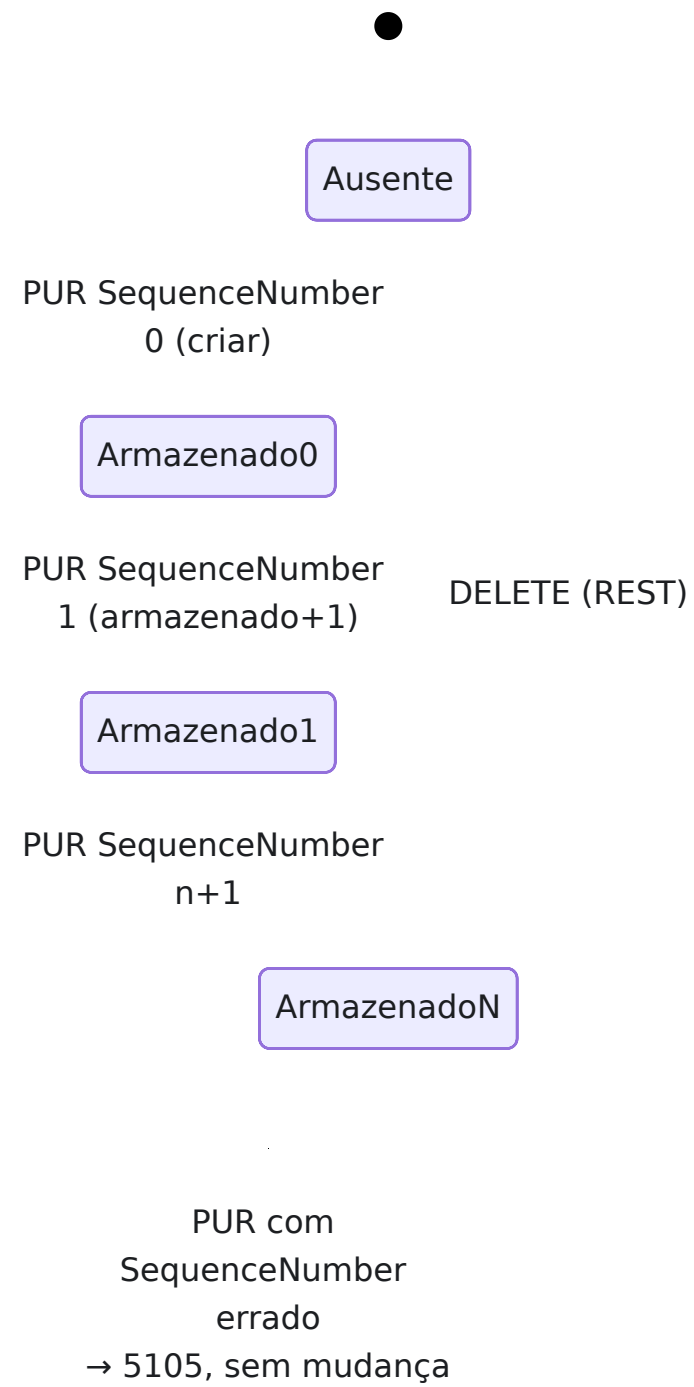
O AS se inscreve nos dados de um assinante e pode solicitar que o valor atual seja retornado imediatamente (`Send-Data-Indication`). OmniHSS responde ao SNR e, quando solicitado, inclui o RepositoryData atual.



SequenceNumber e Concorrência

O SequenceNumber fornece controle de concorrência otimista para que dois Servidores de Aplicação não possam sobrescrever silenciosamente um ao outro, conforme 3GPP TS 29.328.

- **Criar** um novo documento usa **SequenceNumber 0** (o valor `0` é reservado exclusivamente para dados recém-adicionados).
- **Atualizar** um documento existente deve carregar o **sucessor imediato** do SequenceNumber armazenado (armazenado + 1).
- O contador **volta de 65535 para 1** (não 0, uma vez que 0 é reservado).
- Uma atualização cujo SequenceNumber não é o sucessor esperado é rejeitada com **Experimental-Result 5105 (DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC)** e o documento armazenado permanece inalterado.



Exemplo: um documento é armazenado no SequenceNumber `0`. Um PUR com SequenceNumber `5` é rejeitado com `5105` e nada muda. Um PUR com SequenceNumber `1` é aceito e o documento avança para a versão `1`.

REST Provisioning API

Os mesmos documentos podem ser lidos e escritos diretamente por sistemas de provisionamento/CRM. O assinante é endereçado pelo IMSI no caminho; o

documento é identificado por `service_indication` (no corpo da solicitação para atualizações, ou na string de consulta para leituras/exclusões). O opaco `service_data` é transferido byte por byte, exatamente como na Sh.

Todas as respostas usam o envelope padrão da API:

```
{ "status": "success", "response": { } }
```

Ao contrário do caminho Sh, a API REST é um acesso de provisionamento autoritativo e **não** aplica a verificação de SequenceNumber fora de sincronia. Uma escrita define o documento diretamente.

Listar Todos os Documentos

Retorna todos os documentos armazenados para o assinante.

Endpoint: `GET /api/subscriber/repository_data/:imsi`

```
curl -k
https://hss.example.com:8443/api/subscriber/repository_data/001001123
```

```
{
  "status": "success",
  "response": {
    "MMTEL-Services":      { "sequence_number": "0",
"service_data": "<MMTelServices …/>" },
    "IMS-ODB-Information": { "sequence_number": "1",
"service_data": "<OdbForImsOrientedServices
xmlns=\"odb.mmtel.omnitou.ch\"/>" }
  }
}
```

Obter um Único Documento

Retorna um documento por Service-Indication.

Endpoint: `GET /api/subscriber/repository_data/:imsi?service_indication=:service_indication`

```
curl -k
"https://hss.example.com:8443/api/subscriber/repository_data/00100112
service_indication=MMTEL-Services"
```


```
{
  "status": "success",
  "response": { "sequence_number": "0", "service_data": "
<MMTelServices …/>" }
}
```


Criar ou Atualizar um Documento

Cria o documento se ausente, ou o sobrescreve.

Endpoint: `PUT /api/subscriber/repository_data/:imsi`

```
curl -k -X PUT
https://hss.example.com:8443/api/subscriber/repository_data/001001123
\
  -H "Content-Type: application/json" \
  -d '{
        "service_indication": "MMTEL-Services",
        "service_data": "<MMTelServices version=\"1\"/>",
        "sequence_number": "0"
      }'
```

Campo	Tipo	Obrigatório	Padrão	Descrição
`service_indication`	String	Sim	-	O nome do documento a ser criado/atualizado (por exemplo, `MMTEL-Services`).
`service_data`	String	Não	-	A carga útil XML opaca, armazenada literalmente. Omitir para armazenar um documento vazio (apenas Service-Indication e SequenceNumber)
`sequence_number`	String	Não	`0`	A versão a ser registrada para o documento.

A resposta ecoa o documento armazenado:

```
{
  "status": "success",
  "response": { "sequence_number": "0", "service_data": "
<MMTelServices version=\"1\"/>" }
}
```

Excluir um Documento

Remove um único documento por Service-Indication.

Endpoint: `DELETE /api/subscriber/repository_data/:imsi`

```
curl -k -X DELETE
https://hss.example.com:8443/api/subscriber/repository_data/001001123
\
  -H "Content-Type: application/json" \
  -d '{ "service_indication": "MMTEL-Services" }'
```

Campo	Tipo	Obrigatório	Padrão	Descrição
`service_indication`	String	Sim	-	O documento a ser removido.

A resposta retorna os documentos restantes para o assinante.

Tabelas de Referência

Valores de Data-Reference (RepositoryData)

Valor	Nome	Descrição	Referência
0	RepositoryData	Dados de serviço transparentes identificados por Service-Indication	3GPP TS 29.328 §7.6

Códigos de Resultado e Experimental-Result

Código	Nome	Signif
2001	DIAMETER_SUCCESS	UDR/PUR/ processad sucesso
5001	DIAMETER_ERROR_USER_UNKNOWN	Assinante encontrad não habil para IMS
5105	DIAMETER_ERROR_TRANSPARENT_DATA_OUT_OF_SYNC	Sequence de atualiz não é o su do valor armazena

ID da Aplicação

ID	Interface	Descrição	Referência
16777217	Sh	AS ↔ HSS dados do usuário	3GPP TS 29.329

Service-Indications Comuns

Service-Indication	Conteúdo Típico
`MMTEL-Services`	Serviços suplementares MMTel (desvio de chamadas, espera/retém/bloqueio de comunicação, OIP/OIR, conferência): XML simservs
`IMS-ODB-Information`	Informações de bloqueio determinadas pelo operador para serviços orientados a IMS

> Os valores de Service-Indication são definidos pelo Servidor de Aplicação e seu esquema XML correspondente; o HSS não impõe uma lista fixa.

Solução de Problemas

Servidor de Aplicação recebe 5105 na atualização

Sintomas: Um PUR é respondido com Experimental-Result `5105` e o documento permanece inalterado.

Possíveis causas:

- O AS enviou um SequenceNumber que não é o sucessor imediato do valor armazenado no HSS (sua cópia em cache está obsoleta).
- Uma atualização concorrente de outro AS avançou o SequenceNumber.

Resolução:

1. Leia o documento atual com um UDR para obter o SequenceNumber atualizado.
2. Reemita o PUR com SequenceNumber = armazenado + 1.
3. Para forçar um valor administrativamente, defina-o via a REST API, que não aplica a verificação de concorrência.

ServiceData retornado difere do que foi escrito

Sintomas: O AS acredita que a carga útil foi alterada pelo HSS.

Possíveis causas:

- Isso não deve ocorrer. OmniHSS armazena e retorna `ServiceData` byte por byte. Diferenças aparentes geralmente vêm da própria biblioteca XML do AS reserializando o documento (mudando prefixos de namespace, ordem de atributos ou espaços em branco) antes da comparação.

Resolução:

1. Compare os bytes brutos retornados na resposta `UDA`/REST com os bytes brutos enviados, não uma representação reanalisada.
2. Confirme que o AS não está normalizando o XML em seu lado antes da comparação.

Documento não encontrado via REST

Sintomas: `GET …/repository_data/:imsi?service_indication=…` retorna um erro.

Possíveis causas:

- Nenhum documento com essa Service-Indication está armazenado para o assinante.
- O IMSI não corresponde a um assinante existente.

Resolução:

1. Liste todos os documentos com `GET /api/subscriber/repository_data/:imsi` para ver o que está armazenado.
2. Verifique o IMSI com Obter Assinante por IMSI.

SRVCC - Continuidade de Chamada de Voz de Rádio Único

← Voltar ao Guia de Operações

Visão Geral

SRVCC (Continuidade de Chamada de Voz de Rádio Único) permite que um UE transfira uma chamada VoLTE ativa para um portador CS (Comutação de Circuito) 2G/3G quando a cobertura LTE falha. O MME orquestra a transferência através da **interface Sv** (TS 29.280). Para construir a Solicitação Sv PS-to-CS, o MME precisa de dois valores provenientes do HSS:

AVP	Código	Propósito
`STN-SR`	1433	Número de Transferência de Sessão para SRVCC. O servidor MSC disca este número E.164 para alcançar o TAS/S-CSCF, que executa um SIP REFER para mover a sessão VoLTE ativa para o portador CS.
`C-MSISDN`	1452	MSISDN de Correlação. O TAS usa isso para corresponder a chamada STN-SR recebida à sessão ativa correta quando o assinante tem várias chamadas em andamento.

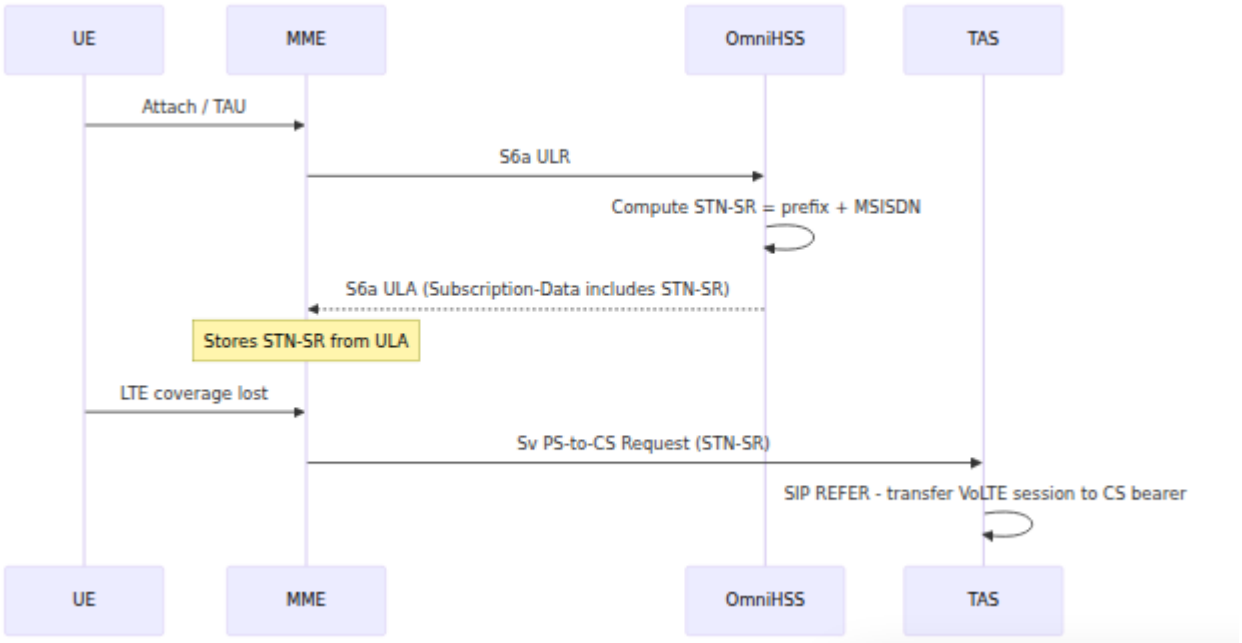
OmniHSS entrega `STN-SR` na **Resposta de Atualização de Localização** (ULA) S6a `Subscription-Data` AVP. O MME lê isso da ULA e o encaminha na Solicitação Sv PS-to-CS.

Especificações de referência:

Especificação	Título
TS 23.216	Procedimento SRVCC PS-to-CS
TS 29.272	Interface S6a (STN-SR §7.3.89, Subscription-Data §7.3.2)
TS 23.003	Formato do número STN-SR (§13.4)

Como Funciona

OmniHSS usa uma abordagem baseada em prefixo. Você configura um único prefixo STN-SR para toda a operadora. O HSS calcula o número completo STN-SR no momento da ULA concatenando o prefixo com o MSISDN do assinante. O resultado é codificado em TBCD e colocado no AVP `STN-SR` dentro de `Subscription-Data`.



Essa abordagem não requer provisionamento por assinante. Todos os assinantes no HSS recebem o mesmo prefixo, então uma única alteração de configuração habilita SRVCC para toda a base de assinantes.

Se o prefixo não estiver configurado, o OmniHSS omite completamente o AVP `STN-SR`. A ULA, de outra forma, não é afetada. Implantações não-SRVCC não

veem alteração no comportamento.

Configuração

Adicione a chave `srvcc` à configuração da aplicação `:hss` em `config/runtime.exs`.

```
config :hss, :srvcc,
  stn_sr_prefix: "9"
```

Parâmetros

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
`stn_sr_prefix`	String	Não	(nenhum)	Prefixo STN-SR atribuído pela operadora que é precedido ao MSISDN de cada assinante para formar o número completo E.164 STN-SR. Quando não definido, o OmniHSS omite o AVP `STN-SR` de todas as respostas ULA.

Formato do Número STN-SR

O número completo STN-SR é:

```
STN-SR = stn_sr_prefix + subscriber MSISDN
```

O número é codificado como **TBCD** (Decimal Codificado em Binário de Telefonia, conforme TS 24.008 §10.5.4.7) e colocado no AVP `STN-SR` (código 1433, fornecedor 10415).

Exemplo:

Entrada	Valor
Prefixo configurado	`9`
MSISDN do assinante	`61412345678`
STN-SR computado	`961412345678`

O servidor MSC disca `961412345678`. A chamada chega ao TAS, que usa a parte MSISDN para localizar a sessão IMS ativa e emite o SIP REFER para transferi-la para o portador CS.

Exemplo de Configuração

SRVCC Habilitado

```
# config/runtime.exs

config :hss, :srvcc,
  # Prefixo discado pelo servidor MSC para alcançar o TAS via
PSTN/CS.
  # STN-SR = stn_sr_prefix + subscriber MSISDN.
  stn_sr_prefix: "9"
```

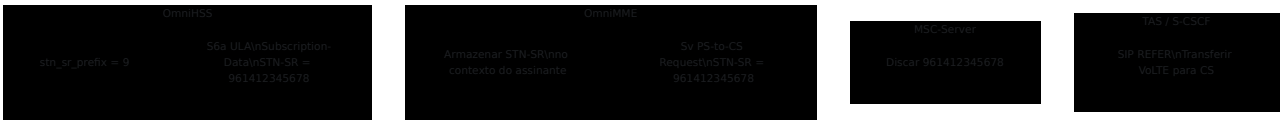
Como funciona: Cada resposta ULA inclui `STN-SR` em `Subscription-Data`. O valor é a codificação TBCD do prefixo concatenado com o MSISDN do assinante. O MME armazena o valor e o encaminha nas Solicitações Sv PS-to-CS quando o SRVCC é acionado.

SRVCC Desabilitado (padrão)

Quando a chave `srvcc` está ausente ou `stn_sr_prefix` não está definido, o OmniHSS omite o AVP `STN-SR`. A ULA é, de outra forma, idêntica. Use isso para implantações que não requerem SRVCC.

Interação com o MME

O MME (OmniMME) lê `STN-SR` da `Subscription-Data` ULA e o armazena no contexto do assinante. Quando o SRVCC é acionado, ele inclui `STN-SR` na Solicitação Sv **PS-to-CS** enviada ao servidor MSC. Veja Omnitouch/omnimme#426 para a implementação do MME.



Solução de Problemas

STN-SR Não Presente na ULA

Sintomas: O MME não consegue preencher a Solicitação Sv PS-to-CS. As transferências SRVCC falham.

Possíveis causas:

- `stn_sr_prefix` não está definido em `config/runtime.exs`.
- O serviço HSS foi iniciado antes que a alteração de configuração tivesse efeito.

Resolução:

1. Verifique se o bloco `srvcc` está presente em `config/runtime.exs` com um `stn_sr_prefix` não vazio.
2. Reinicie o serviço OmniHSS para aplicar a alteração de configuração.

3. Acione uma nova ULR do MME (por exemplo, forçando um TAU). Confirme que os `Subscription-Data` da ULA agora contêm `STN-SR`.

Valor STN-SR Incorreto

Sintomas: O servidor MSC recebe um número malformado ou não roteável.

Possíveis causas:

- O prefixo está incorreto ou faltando dígitos iniciais.
- O MSISDN do assinante armazenado no HSS não está no formato internacional E.164 (apenas dígitos, sem `+`).

Resolução:

1. Verifique o MSISDN do assinante via API (`GET /api/subscriber/imsi/:imsi`).
2. Verifique o STN-SR esperado: concatene `stn_sr_prefix` com o MSISDN armazenado.
3. Confirme que a tabela de roteamento do servidor MSC lida com aquele intervalo de números.

← Voltar ao Guia de Operações

Interface Sy (Controle de Limite de Gastos)

Visão Geral

A interface Sy conecta o HSS (atuando como o **PCRF**, Função de Regras de Política e Cobrança) a um **OCS** (Sistema de Cobrança Online) para **controle de limite de gastos do assinante**. Ela permite que o PCRF baseie suas decisões de política - por exemplo, qual regra de cobrança aplicar, ou se deve limitar um assinante - no status dos **contadores de política** que o OCS mantém para cada assinante (dados consumidos, gastos em relação a um limite mensal, e assim por diante).

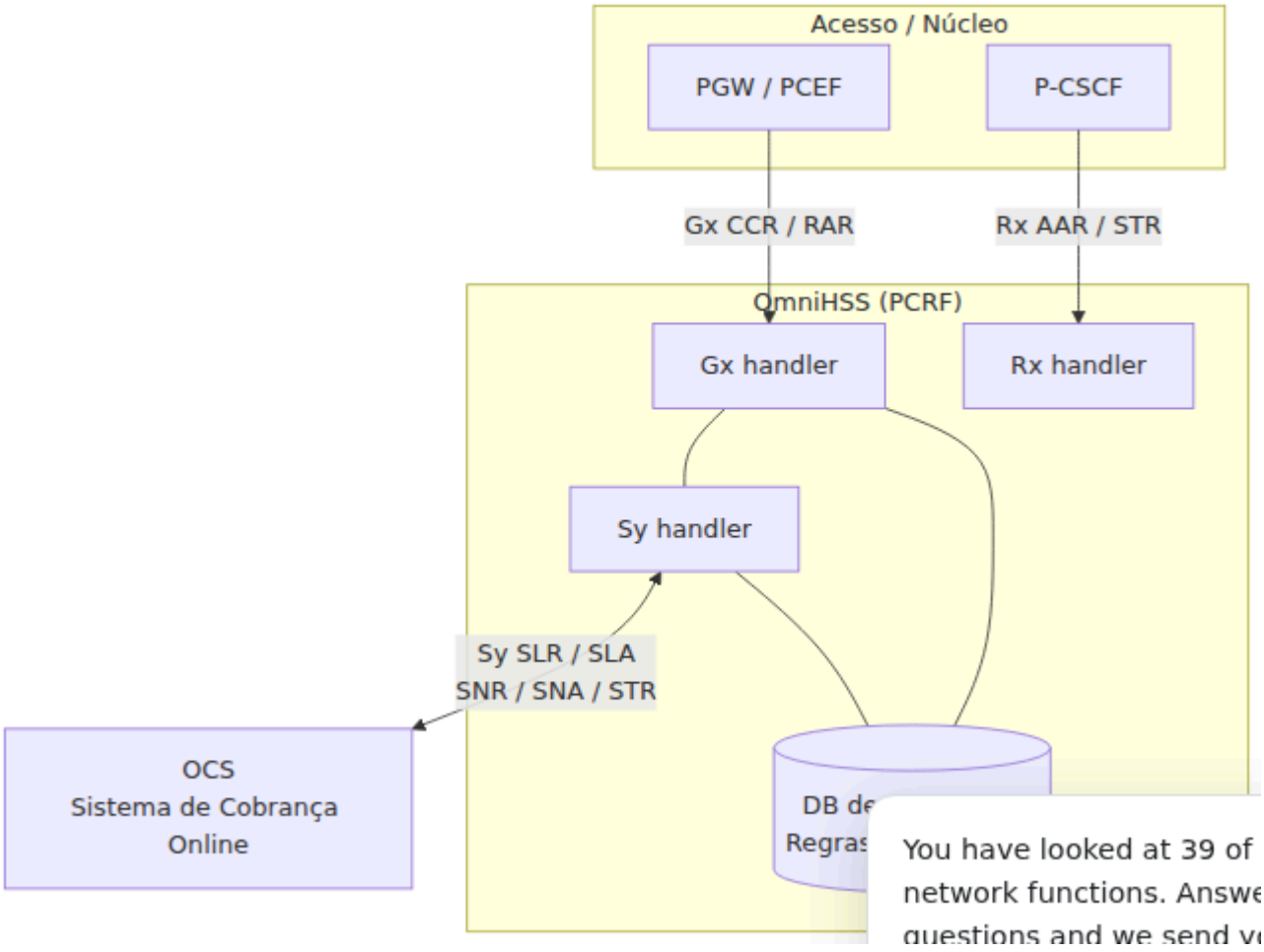
O PCRF OmniHSS instala regras de cobrança e QoS sobre Gx e autoriza mídia IMS sobre Rx. Através do Sy, ele se inscreve no status dos contadores de política de um assinante no OCS e é notificado quando eles mudam, para que essas decisões Gx possam levar em conta o gasto e a cota atuais do assinante.

O Sy é definido pela 3GPP TS 29.219 e usa o ID da aplicação Diameter **16777302**.

Propriedade	Valor
ID da Aplicação	16777302
Especificação	3GPP TS 29.219
Papel do HSS	PCRF (cliente Diameter para SLR/STR, servidor para SNR)
Par	OCS (Sistema de Cobrança Online)
Transporte	TCP ou SCTP, porta 3868

> **Sy vs Gy.** Sy não é cobrança online. Gy (TS 32.299) é onde o PGW pede ao OCS para conceder crédito para tráfego ao vivo. Sy é uma interface separada e leve onde o **PCRF** se inscreve no **status** dos contadores de política do OCS e é notificado quando eles mudam, para que possa ajustar a política. O OCS é o servidor em ambos.

Arquitetura



O HSS é o **cliente Sy**: ele abre uma sessão Sy para o OCS quando precisa pela primeira vez do status de gastos de um assinante e a encerra quando a última sessão de dados do assinante termina. O OCS é o **servidor Sy**: ele responde a solicitações de status e, sempre que o status de um contador muda, notifica o PCRF para que ele possa reaplicar a política sobre o Gx.

Contadores de Política e Regras de Cobrança

O OCS mantém um ou mais **contadores de política** por assinante. Cada contador tem:

- um **Identificador de Contador de Política** (AVP 2901) - um nome estável para o contador, e
- um **Status do Contador de Política** (AVP 2902) - um rótulo descrevendo onde o contador se encontra atualmente (por exemplo, `ativo`, `fora_da_cota`, `limitado`). O conjunto de valores de status não é definido pela 3GPP; é configurado conjuntamente pelo operador tanto no OCS quanto no PCRF (TS 29.219 §3.1).

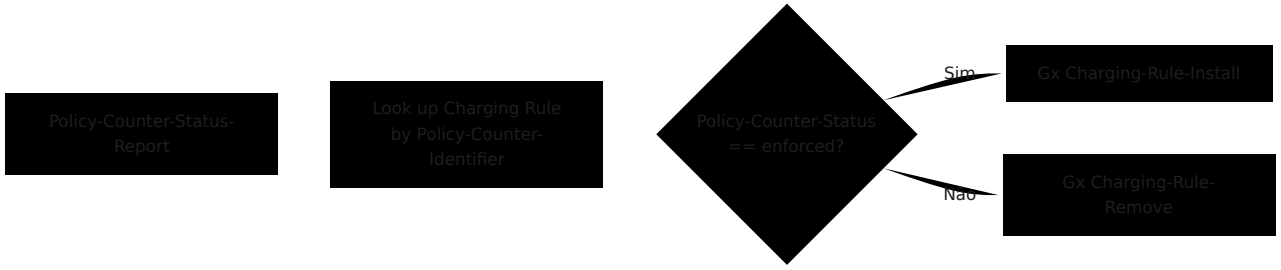
O OmniHSS vincula esses aos regras de cobrança já em seu banco de dados:

AVP Sy	Natureza	Mapeia para	Papel
`Policy-Counter-Identifier` (2901)	Estático - nomeia o contador	Um **nome de Regra de Cobrança** no banco de dados do HSS	Seleciona *qual* regra de cobrança o contador governa
`Policy-Counter-Status` (2902)	Dinâmico - muda em limites	Uma **ação de aplicação**	Decide *o que fazer* com essa regra agora: instalar, remover ou selecionar um nível

O identificador é estável durante a vida do contador, então por si só não pode expressar "dentro da cota" versus "fora da cota". O **status** é o valor que muda quando o assinante ultrapassa um limite, então ele impulsiona a ação:

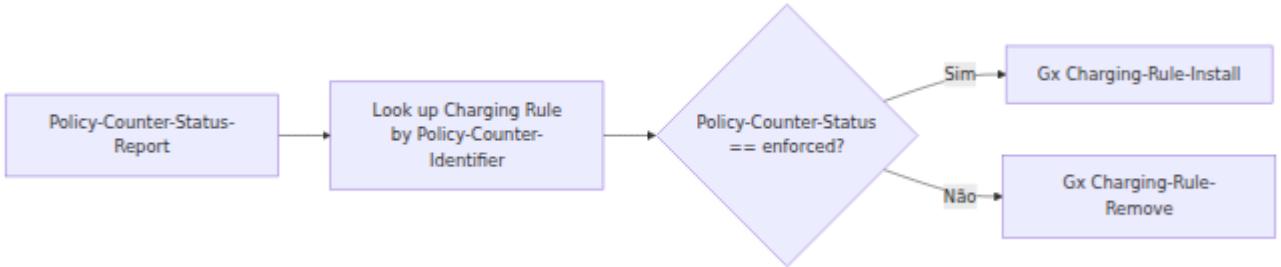
- o status fixo **`enforced`** instala a regra de cobrança nomeada (`Charging-Rule-Install` sobre Gx);
- qualquer outro status a remove (`Charging-Rule-Remove`).

> **Sy não carrega suas próprias regras de cobrança.** Sy não possui AVPs de Regra de Cobrança, QoS ou Fluxo - esses existem apenas no Gx. Portanto, o OCS não pode entregar ao PCRF uma definição de regra; ele pode apenas *nomear* a regra (via o identificador do contador) e sinalizar *quando* aplicá-la (via o status). A definição da regra em si é provisionada no HSS. É por isso que `Policy-Counter-Identifier` deve corresponder a um nome de regra de cobrança, e o OCS deve emitir esses nomes exatos com o status `enforced` quando a regra deve ser aplicada.



Ciclo de Vida da Sessão Sy

Uma sessão Sy é **por assinante**, não por portadora de dados. O PCRF mantém uma sessão Sy (identificada por um único `Session-Id` Diameter) para um assinante enquanto qualquer uma das sessões de dados desse assinante depender de status de gastos (TS 29.219 §4.4.1).

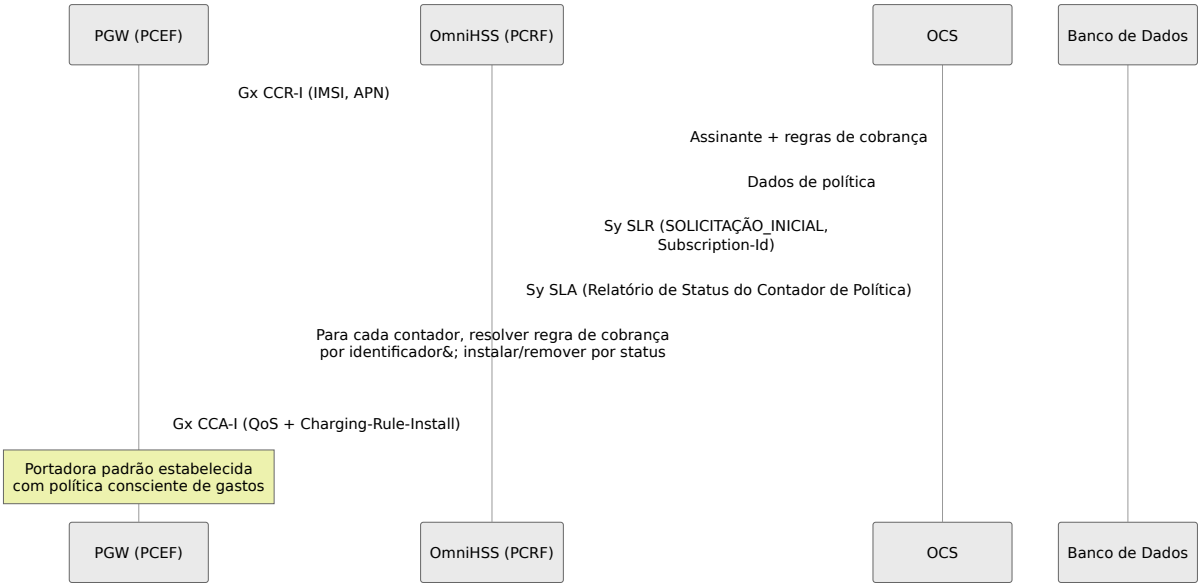


- **Estabelecer (SLR, SOLICITAÇÃO_INICIAL).** Na primeira Gx CCR-I onde a política do assinante depende do status de gastos, o PCRF abre a sessão Sy e solicita os contadores de política relevantes. O OCS responde com o status atual de cada um no SLA.
- **Atualizar (SLR, SOLICITAÇÃO_INTERMEDIÁRIA).** Se o PCRF precisar de contadores adicionais mais tarde, ele reutiliza a mesma sessão Sy com uma solicitação intermediária.

- **Notificar (SNR / SNA).** Quando o status de um contador muda, o OCS envia uma Solicitação de Notificação de Status de Gastos. O PCRF reconhece com um SNA e aplica a mudança sobre o Gx.
- **Encerrar (STR / STA).** Quando a última sessão de dados do assinante termina, o PCRF fecha a sessão Sy com uma Solicitação de Término de Sessão.

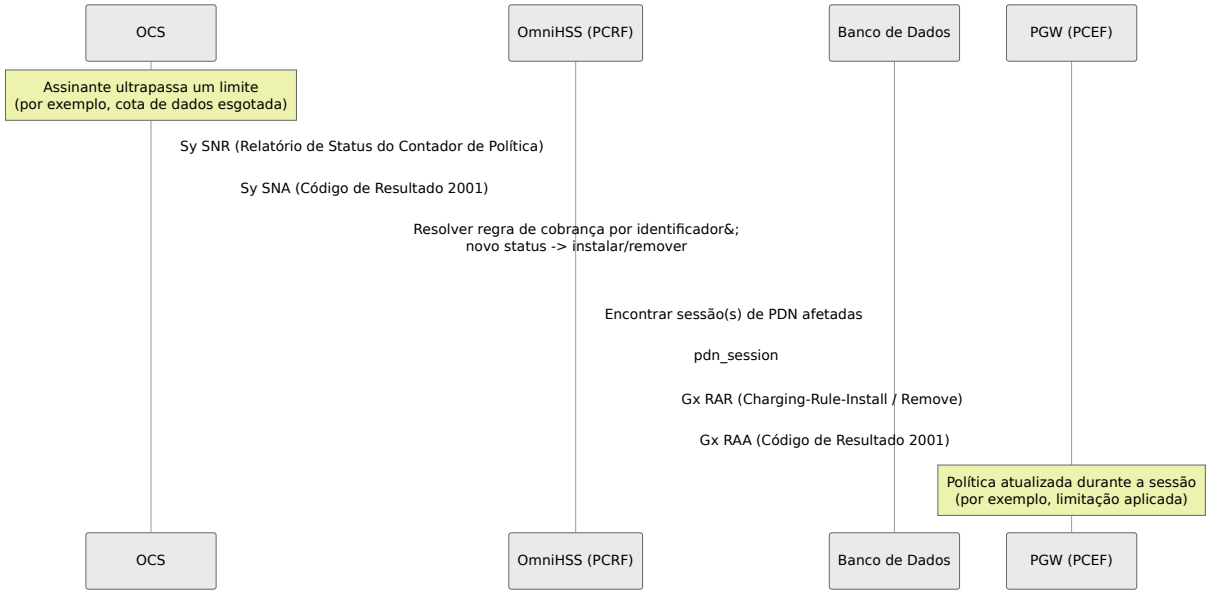
Fluxos de Mensagens

Fluxo 1: Relatório Inicial de Limite de Gastos na Configuração do PDN



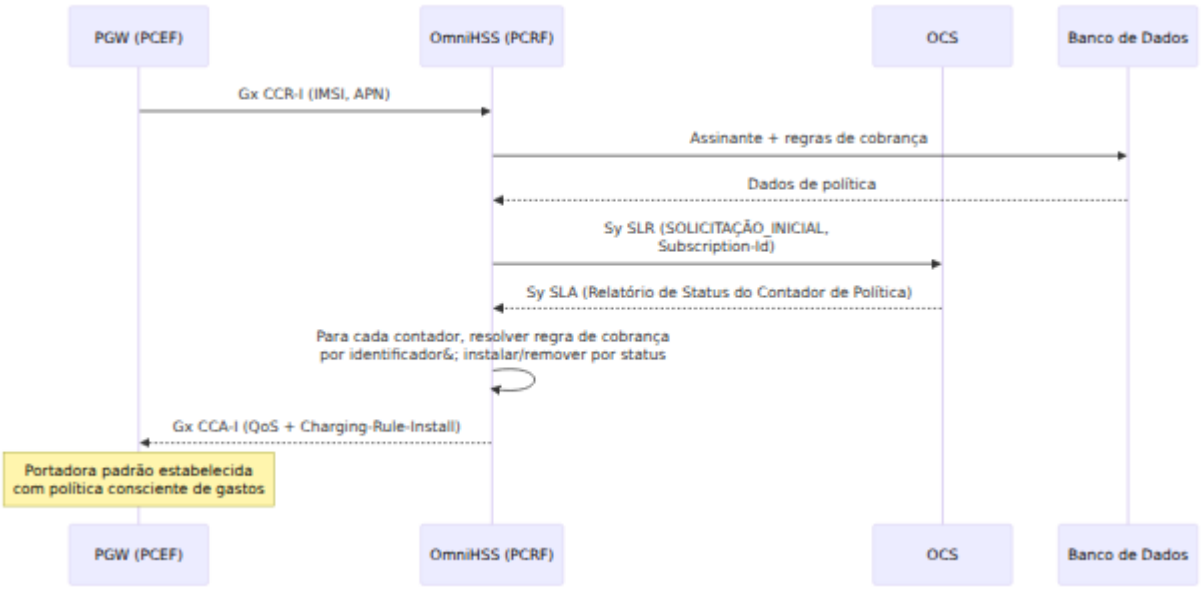
O PCRF incorpora as regras derivadas de gastos na **mesma CCA-I** que já constrói para a sessão Gx. Se o OCS não responder a tempo, o PCRF recorre à política padrão da APN, de modo que uma interrupção do Sy nunca bloqueie a conexão.

Fluxo 2: Mudança de Status de Gastos (OCS -> PCRF -> Gx RAR)



Isso reutiliza a maquinaria existente de Reautenticação Gx do PCRF - o mesmo caminho que instala portadoras dedicadas de VoLTE - para empurrar a mudança de política impulsionada por gastos para o PGW sem esperar pela próxima CCR.

Fluxo 3: Término da Sessão Sy



Configuração

Aplicação Sy

A aplicação Sy é registrada no serviço Diameter do HSS ao lado de Gx e Rx (`config/runtime.exs`):

```
%{
  application_name: :sy,
  application_dictionary: :diameter_gen_3gpp_sy,
  vendor_specific_application_ids: [
    %{vendor_id: 10415, auth_application_id: 16_777_302,
acct_application_id: nil}
  ]
}
```

Parâmetro	Tipo	Necessário	Padrão	
`application_name`	Átomo	Sim	-	Nor inte par
`application_dictionary`	Átomo	Sim	-	Mód Dia `:di` (3G
`vendor_specific_application_ids`	Lista	Sim	-	Anu na t cap aba
`vendor_id`	Inteiro	Sim	-	ID d par
`auth_application_id`	Inteiro	Sim	-	ID d aut `16_`
`acct_application_id`	Inteiro | nil	Não	`nil`	ID d con utili ser

Configurações Sy

O comportamento do Sy é configurado sob a chave `:hss, :sy` (`config/runtime.exs`):

```
config :hss,
  sy: %{
    enabled: false
  }
```

Parâmetro	Tipo	Necessário	Padrão	Descrição
`enabled`	Booleano	Não	`false`	Interruptor mestre. Quando `true`, o PCRF abre uma sessão Sy por assinante na Gx CCR-I. Quando `false`, o Sy é completamente ignorado.

O valor `Policy-Counter-Status` que instala a regra de cobrança nomeada é a convenção fixa **`enforced`**; qualquer outro valor remove a regra. Isso é acordado com o OCS e não é configurável.

Roteamento OCS

O HSS **não** fixa um host ou domínio OCS. SLR e STR originados carregam o **próprio** `Destination-Realm` do HSS e nenhum `Destination-Host`, e são roteados pelo domínio mais o Application-Id Sy (16777302). Qualquer par que anunciar a aplicação Sy nesse domínio recebe a solicitação - tipicamente o **DRA**, que encaminha para um OCS disponível. Isso funciona de maneira idêntica, seja o OCS alcançado através do DRA ou diretamente, e permite que a camada de roteamento escolha entre múltiplos OCSs.

A única exigência é que um par anunciando a aplicação Sy exista no domínio. Em uma implantação DRA, esse é o par DRA já configurado para o HSS; nenhuma entrada específica do par Sy é necessária.

Mapeamento de Contador de Política

Para que a vinculação identificador-regra funcione, o operador configura **ambos** os lados de forma consistente:

- No **OCS**: o `Policy-Counter-Identifier` de cada contador de política é definido como o exato `nome` de uma Regra de Cobrança definida no HSS, e o contador relata o status `enforced` quando essa regra deve ser aplicada.
- No **HSS**: as regras de cobrança referenciadas existem no banco de dados.

Veja Contadores de Política e Regras de Cobrança para o modelo de mapeamento.

Métricas

Sy expõe métricas Prometheus ao lado das outras métricas da interface HSS (veja Métricas & Monitoramento).

Métrica: `hss_sy_messages_total` **Tipo**: Contador **Descrição**: Mensagens Sy processadas, por comando, direção e resultado. **Rótulos**:

- `command` - `slr`, `sla`, `snr`, `sna`, `str`, ou `sta`
- `direction` - `incoming` (recebido do OCS) ou `outgoing` (enviado para o OCS)
- `result` - `sent`, `received`, `success`, ou `error`

Métrica: `hss_sy_policy_updates_total` **Tipo**: Contador **Descrição**: Mudanças de regra de cobrança Gx impulsionadas pelo status de um contador de política Sy, por ação e resultado. **Rótulos**:

- `action` - `install` ou `remove`
- `result` - `success` ou `error` (o resultado do Gx RAR para o PGW)

Consultas de exemplo:

```
# Taxa de notificação Sy do OCS
rate(hss_sy_messages_total{command="snr", direction="incoming"}
[5m])

# Taxa de instalações de regras de limitação/aplicação
impulsionadas por limites de gastos
rate(hss_sy_policy_updates_total{action="install",
result="success"}[5m])

# Razão de falhas do Gx RAR do Sy
sum(rate(hss_sy_policy_updates_total{result="error"}[5m]))
  / sum(rate(hss_sy_policy_updates_total[5m]))
```

Referência

Comandos

Comando	Abreviação	Código	Direção	Propósito
Solicitação de Limite de Gastos	SLR	8388635	PCRF -> OCS	Solicitar status do contador de política; inscrever-se em mudanças
Resposta de Limite de Gastos	SLA	8388635	OCS -> PCRF	Retornar status atual do contador
Solicitação de Notificação de Status de Gastos	SNR	8388636	OCS -> PCRF	Notificar uma mudança de status de contador
Resposta de Notificação de Status de Gastos	SNA	8388636	PCRF -> OCS	Reconhecer uma notificação
Solicitação de Término de Sessão	STR	275	PCRF -> OCS	Fechar a sessão Sy (relatório final)
Resposta de Término de Sessão	STA	275	OCS -> PCRF	Confirmar o fechamento da sessão

Todos os comandos carregam o Application-ID `16777302`, incluindo o STR/STA reutilizado (RFC 6733).

AVPs Chave

AVP	Código	Tipo	Descrição
`Policy-Counter-Identifier`	2901	UTF8String	Nomeia um contador de política. Vinculado a um nome de Regra de Cobrança no OmniHSS.
`Policy-Counter-Status`	2902	UTF8String	Rótulo de status definido pelo operador para o contador. Impulsiona a ação de aplicação.
`Policy-Counter-Status-Report`	2903	Agrupado	Identificador + status de um contador (+ quaisquer status pendentes).
`SL-Request-Type`	2904	Enumerado	`INITIAL_REQUEST (0)` para a primeira solicitação, `INTERMEDIATE_REQUEST (1)` a partir de então.
`Pending-Policy-Counter-Information`	2905	Agrupado	Um status futuro e o momento em que se torna atual.
`Pending-Policy-Counter-Change-Time`	2906	Tempo	Quando um status pendente entra em vigor.
`SN-Request-Type`	2907	Unsigned32	Bitmask em um SNR. Bit 0 = relatório normal, bit 1 = abortar (OCS pede para encerrar a sessão Sy).

AVP	Código	Tipo	Descrição
`Subscription-Id`	443	Agrupado	Identidade do assinante (IMSI e/ou MSISDN) na SLR.

Códigos de Resultado

Código	Nome	Significa
2001	DIAMETER_SUCCESS	Solicitaçã processad com suces
5030	DIAMETER_USER_UNKNOWN	OCS não conhece o assinante; sessão Sy é criada
5570	DIAMETER_ERROR_UNKNOWN_POLICY_COUNTERS	Um ou ma contadores política solicitados desconhec pelo OCS
4241	DIAMETER_ERROR_NO_AVAILABLE_POLICY_COUNTERS	O OCS nã possui contadores política pa assinante

Os códigos 5570 e 4241 são valores de Código de Resultado Experimental específicos do Sy (TS 29.219 §5.5); 5030 é definido pela RFC 4006.

Documentação Relacionada

- PCRF - Controle de política Gx e Rx que o Sy alimenta
- Protocolos Diameter - Fluxos de mensagens em todas as interfaces HSS
- Arquitetura - Arquitetura geral do HSS

troubleshooting

```
---
last_update:
  date: "2026-10-07T06:26:39Z"
---

# Guia de Solução de Problemas do OmniHSS

[← Voltar ao Guia de Operações](../README.md)

---

## Índice

- [Visão Geral da Solução de Problemas](#troubleshooting-overview)
- [Falhas de Autenticação](#authentication-failures)
- [Problemas de Conectividade Diameter](#diameter-connectivity-
issues)
- [Problemas de Banco de Dados](#database-issues)
- [Falhas de Registro EPC](#epc-registration-failures)
- [Falhas de Registro IMS](#ims-registration-failures)
- [Falhas de Chamada VoLTE](#volte-call-failures)
- [Problemas de Roaming](#roaming-issues)
- [Problemas de EIR](#eir-problems)
- [Problemas de Desempenho](#performance-problems)
- [Problemas de Estado do Assinante](#subscriber-state-issues)
- [Problemas de API](#api-issues)
- [Ferramentas e Comandos de Diagnóstico](#diagnostic-tools-and-
commands)

---

## Visão Geral da Solução de Problemas

### Abordagem Geral para Solução de Problemas

```mermaid
graph TB
    START[Problema Reportado] --> IDENTIFY[Identificar Sintomas]
    IDENTIFY --> SCOPE{Escopo do Impacto}

    SCOPE -->|Assinante Único| SINGLE[Verificar Dados do
Assinante]
    SCOPE -->|Múltiplos Assinantes| MULTI[Verificar
```
```

```
Sistema/Perfil]
    SCOPE -->|Todos os Assinantes| SYSTEM[Verificar Saúde do
Sistema]

    SINGLE --> SUB_CHECK[Verificar Configuração do Assinante]
    MULTI --> PROFILE_CHECK[Verificar Configuração do Perfil]
    SYSTEM --> SYS_CHECK[Verificar Diameter/DB/Sistema]

    SUB_CHECK --> LOGS[Revisar Logs]
    PROFILE_CHECK --> LOGS
    SYS_CHECK --> LOGS

    LOGS --> ROOT[Identificar Causa Raiz]
    ROOT --> FIX[Aplicar Correção]
    FIX --> VERIFY[Verificar Resolução]
    VERIFY --> DOC[Documentar Solução]

    style START fill:#e3f2fd
    style ROOT fill:#fff9c4
    style FIX fill:#c8e6c9
```

Informações a Coletar

Antes de solucionar qualquer problema, colete:

1. **Informações do Assinante** (se específico do assinante)
    - IMSI
    - MSISDN (número de telefone)
    - Último estado conhecido
    - Mensagens de erro do dispositivo
2. **Informações de Tempo**
    - Quando o problema começou?
    - É intermitente ou constante?
    - Hora da última operação bem-sucedida
3. **Escopo do Impacto**

- Assinante único ou múltiplos?
- Rede específica ou todas as redes?
- Serviço específico (dados/vóz) ou ambos?

4. **Estado do Sistema**
   - Verifique o Painel de Controle para o status do sistema
   - Revise o status do par Diameter
   - Verifique a conectividade do banco de dados

Falhas de Autenticação

Sintomas

- Assinante não consegue se conectar à rede
- Erros de "Autenticação rejeitada"
- Tentativas de autenticação repetidas

Causas e Soluções Comuns

Causa 1: Conjunto de Chaves Incorreto

Sintomas:

- Falha de autenticação consistente para assinante específico
- Funciona para outros assinantes com o mesmo perfil

Passos de Diagnóstico:

1. Consulte o assinante para verificar o key_set_id:

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. Verifique se o conjunto de chaves existe e possui os valores corretos:

```
curl -k https://hss.example.com:8443/api/key_set/[KEY_SET_ID]
```

3. Compare os valores de Ki e OPC com a documentação do cartão SIM

Solução:

- Atualize o assinante com o conjunto de chaves correto
- Se as chaves estiverem corretas, o cartão SIM pode estar com defeito

Causa 2: SQN Fora de Sincronização

Sintomas:

- A autenticação falha após ter funcionado anteriormente
- Erro: "Falha de sincronização SQN"
- Funciona intermitentemente

Passos de Diagnóstico:

1. Verifique o estado do assinante para o valor SQN no banco de dados
2. Procure por erros relacionados ao SQN nos logs
3. Verifique o valor SQN do conjunto de chaves do assinante

Solução:

- O SQN será automaticamente re-sincronizado após o assinante enviar AUTS
- Se persistir, redefina o SQN para 0 no conjunto de chaves (requer reanexação do assinante)

Aviso: Redefinir o SQN pode causar problemas de segurança. Faça isso apenas durante a manutenção.

Causa 3: Assinante Desativado

Sintomas:

- Autenticação rejeitada imediatamente
- Nenhum vetor de autenticação gerado

Passos de Diagnóstico:

1. Verifique o status de habilitação do assinante:

```
curl -k https://hss.example.com:8443/api/subscriber/imsi/[IMSI]
```

2. Verifique se o campo `enabled` é `true`

Solução:

- Habilitar assinante:

```
curl -k -X PUT https://hss.example.com:8443/api/subscriber/[ID]
\
  -H "Content-Type: application/json" \
  -d '&#123;"subscriber": &#123;"enabled": true&#125;&#125;'
```

Causa 4: Perfil EPC Ausente

Sintomas:

- A consulta ao assinante é bem-sucedida, mas a autenticação falha
- Erro: "Nenhum perfil EPC atribuído"

Passos de Diagnóstico:

1. Verifique o campo `epc_profile_id` do assinante
2. Verifique se o perfil EPC existe:

```
curl -k
https://hss.example.com:8443/api/epc/profile/[PROFILE_ID]
```

Solução:

- Atribua um perfil EPC válido ao assinante

Fluxograma de Solução de Problemas de Autenticação

Lexical error on line 2. Unrecognized text. ... --> CHECK_ENABLED&fl°
°123¶ßAssinante Hab ----------^

Tentar novamente

Problemas de Conectividade Diameter

Sintomas

- Pares Diameter mostrando como desconectados no Painel de Controle
- Erros de "Sem rota para o host"
- Serviços falhando para todos os assinantes

Causas e Soluções Comuns

Causa 1: Conectividade de Rede

Sintomas:

- O par nunca se conecta
- Erros de tempo limite de conexão
- Ping falha para o par

Passos de Diagnóstico:

1. Verifique a conectividade de rede do OmniHSS para o par:

   ```
   ping [PEER_IP]
   ```

2. Verifique se a porta Diameter é acessível:

```
telnet [PEER_IP] 3868
```

3. Verifique se as regras de firewall permitem tráfego Diameter (porta 3868)

Solução:

- Corrija o roteamento de rede
- Atualize as regras de firewall
- Verifique se o par está em execução e ouvindo

Causa 2: Configuração Diameter Incorreta

Sintomas:

- Tentativas de conexão falham
- Troca CER/CEA falha
- O par rejeita a conexão

Passos de Diagnóstico:

1. Revise a configuração Diameter em runtime.exs:
    - Verifique se o origin_host do par corresponde ao valor esperado
    - Verifique a configuração do origin_realm
    - Verifique se o endereço IP do par está correto
2. Verifique os logs para erros CER/CEA
3. Verifique se a configuração do par espera o origin_host do OmniHSS

Solução:

- Atualize runtime.exs com a configuração Diameter correta
- Reinicie o OmniHSS após a alteração da configuração
- Coordene com o administrador do par para verificar as configurações

Causa 3: Problemas de Certificado (TLS Diameter)

Sintomas:

- A conexão falha durante o handshake TLS
- Erros de validação de certificado
- Erros de "Certificado expirado" ou "Certificado inválido"

Passos de Diagnóstico:

1. Verifique se os arquivos de certificado existem em `priv/cert/`
2. Verifique a expiração do certificado:

   ```
   openssl x509 -in priv/cert/diameter.crt -noout -dates
   ```

3. Verifique se a cadeia de certificados está completa
4. Verifique o certificado do par se TLS mútuo

Solução:

- Renove certificados expirados
- Instale a cadeia de certificados correta
- Atualize os arquivos de certificado e reinicie o OmniHSS

Causa 4: Incompatibilidade de Suporte a Aplicativos do Par

Sintomas:

- O par se conecta, mas não suporta os aplicativos necessários
- A troca de capacidades é bem-sucedida, mas as operações falham
- Erros de "Aplicativo não suportado"

Passos de Diagnóstico:

1. Verifique a página Diameter do Painel de Controle para aplicativos do par
2. Verifique se o par suporta o aplicativo necessário (S6a, Cx, Sh, etc.)
3. Revise a troca CER/CEA nos logs

Solução:

- Verifique se a configuração do par inclui os aplicativos Diameter necessários
- Verifique se o tipo de par corresponde à funcionalidade esperada:
  - O MME deve suportar S6a (16777251)
  - O S-CSCF deve suportar Cx (16777216)
  - O P-GW deve suportar Gx (16777238)

Fluxograma de Solução de Problemas Diameter

Lexical error on line 2. Unrecognized text. ...onectado] --> PING&fl° °123¶ßRede Acessíve ----------^

Tentar novamente

Problemas de Banco de Dados

Sintomas

- A API retorna erros 500
- O Painel de Controle não carrega
- Erros de "Falha na conexão com o banco de dados"
- Desempenho de consulta lento

Causas e Soluções Comuns

Causa 1: Servidor de Banco de Dados Offline

Sintomas:

- Todas as chamadas de API falham
- O Painel de Controle mostra erro
- Erros de "Conexão recusada"

Passos de Diagnóstico:

1. Teste a conectividade do banco de dados:

```
# Conectar ao banco de dados
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME]
```

2. Verifique o status do serviço de banco de dados no servidor de banco de dados

3. Verifique a conectividade de rede com o servidor de banco de dados

Solução:

- Inicie o serviço de banco de dados
- Corrija problemas no servidor de banco de dados
- Verifique o roteamento de rede para o servidor de banco de dados

Causa 2: Credenciais de Banco de Dados Incorretas

Sintomas:

- Erros de "Autenticação falhou"
- OmniHSS não consegue conectar na inicialização

Passos de Diagnóstico:

1. Revise a configuração do banco de dados em runtime.exs
2. Teste as credenciais manualmente com o cliente de banco de dados
3. Verifique as permissões do usuário do banco de dados

Solução:

- Atualize a configuração do banco de dados em runtime.exs
- Conceda as permissões corretas ao usuário do banco de dados
- Reinicie o OmniHSS após a alteração da configuração

Causa 3: Pool de Conexões Esgotado

Sintomas:

- Erros 500 intermitentes
- Erros de "Sem conexões disponíveis"
- Períodos de alta carga acionam falhas

Passos de Diagnóstico:

1. Verifique a contagem atual de conexões no banco de dados
2. Revise o tamanho do pool de banco de dados em runtime.exs
3. Monitore o uso de conexões durante a carga máxima

Solução:

- Aumente o tamanho do pool na configuração de runtime.exs
- Investigue vazamentos de conexão se o pool se esgotar repetidamente
- Considere escalar o banco de dados se a carga for consistentemente alta

Causa 4: Consultas Lentas

Sintomas:

- Respostas da API muito lentas
- Timeouts em consultas de assinantes
- CPU do banco de dados alta

Passos de Diagnóstico:

1. Consulte o banco de dados para o log de consultas lentas
2. Identifique consultas lentas específicas
3. Verifique se há índices ausentes
4. Verifique a contagem de assinantes e tamanhos das tabelas

Solução:

- Otimize consultas lentas
- Adicione índices ausentes
- Considere o ajuste de desempenho do banco de dados

- Planeje a escalabilidade do banco de dados, se necessário

Fluxograma de Solução de Problemas de Banco de Dados

Lexical error on line 2. Unrecognized text. ...Dados] --> CONNECT&fl° °123¶ßConsegue Cone ----------^

Tentar novamente

Falhas de Registro EPC

Sintomas

- Assinante não consegue se conectar à rede LTE
- O MME rejeita a conexão
- Nenhuma sessão PDN estabelecida

Causas e Soluções Comuns

Causa 1: Roaming Negado

Sintomas:

- Assinante funciona na rede doméstica, mas falha ao roaming
- Erros de "Roaming não permitido"
- Funciona para algumas redes, mas não para outras

Passos de Diagnóstico:

1. Verifique o `roaming_profile_id` do assinante
2. Consulte o perfil de roaming e as regras
3. Verifique o MCC/MNC da rede visitada
4. Verifique se a regra de roaming existe para essa rede

Solução:

- Adicione uma regra de roaming para o MCC/MNC da rede visitada
- Ou atualize a ação padrão do perfil de roaming para permitir
- Veja a Documentação de Roaming para configuração

Causa 2: Configuração APN Ausente

Sintomas:

- A conexão é bem-sucedida, mas a sessão PDN falha
- Erros de "APN desconhecido" do MME
- Assinante não consegue obter conexão de dados

Passos de Diagnóstico:

1. Verifique se o perfil EPC tem perfis APN vinculados
2. Verifique se o identificador APN corresponde ao que o dispositivo solicita
3. Consulte a configuração do perfil APN

Solução:

- Vincule os perfis APN ao perfil EPC do assinante
- Certifique-se de que o nome APN corresponda à configuração do dispositivo
- Verifique se o perfil QoS APN existe

Causa 3: MME Não Conectado

Sintomas:

- Todos os assinantes falham ao se conectar
- Sem comunicação com o MME
- Par Diameter offline

Passos de Diagnóstico:

1. Verifique a página Diameter do Painel de Controle
2. Verifique se o status do par MME está "Conectado"
3. Verifique se o MME suporta a aplicação S6a

Solução:

- Solucione a conectividade Diameter
- Verifique a configuração do MME
- Contate o administrador do MME

Causa 4: Corrupção do Estado do Assinante

Sintomas:

- Assinante aparece como conectado, mas não consegue se conectar novamente
- O estado não corresponde à realidade
- A desconexão e reconexão falham

Passos de Diagnóstico:

1. Consulte o estado do assinante no banco de dados
2. Verifique se há atribuições de MME desatualizadas
3. Verifique o timestamp da última atualização

Solução:

- Limpe o estado do assinante (procedimento de desconexão)
- Redefina o MME que atende no estado do assinante
- Pode ser necessário reiniciar o assinante

Fluxograma de Solução de Problemas de Registro EPC

Lexical error on line 2. Unrecognized text. ...stro EPC] --> HOME&fl°°123¶ßNa Rede Domés ----------^

Tentar novamente

Falhas de Registro IMS

Sintomas

- Assinante não consegue registrar para VoLTE
- "Falha no registro IMS" no dispositivo
- Dados funcionam, mas voz não

Causas e Soluções Comuns

Causa 1: IMS Desativado para o Assinante

Sintomas:

- Assinante tem dados, mas sem IMS
- Registro rejeitado imediatamente

Passos de Diagnóstico:

1. Consulte o assinante e verifique o campo `ims_enabled`
2. Verifique se o assinante tem `ims_profile_id` atribuído

Solução:

- Habilitar IMS para o assinante
- Atribuir perfil IMS

Causa 2: S-CSCF Não Conectado

Sintomas:

- Todos os registros IMS falham
- Sem tráfego Diameter relacionado ao IMS

Passos de Diagnóstico:

1. Verifique a página Diameter do Painel de Controle
2. Verifique se o par S-CSCF está conectado

3. Verifique se o S-CSCF suporta a aplicação Cx

Solução:

- Corrija a conectividade Diameter para o S-CSCF
- Verifique a configuração do S-CSCF

Causa 3: Modelo IFC Ausente ou Inválido

Sintomas:

- Registro falha durante a Resposta de Autorização do Usuário
- Erros relacionados ao IFC nos logs

Passos de Diagnóstico:

1. Consulte o perfil IMS do assinante
2. Verifique se o modelo IFC está presente
3. Verifique a sintaxe XML do IFC

Solução:

- Atualize o perfil IMS com um modelo IFC válido
- Veja a Documentação de Perfis para exemplos de IFC

Causa 4: Roaming Negado para IMS

Sintomas:

- IMS funciona na rede doméstica
- Falha ao roaming
- Roaming de dados funciona, mas não IMS

Passos de Diagnóstico:

1. Verifique a ação IMS do perfil de roaming
2. Verifique se as regras de roaming têm a `ims_action` correta

Solução:

- Atualize as regras de roaming para permitir IMS
- Ou atualize a ação padrão do perfil de roaming para IMS

Fluxograma de Solução de Problemas de Registro IMS

Lexical error on line 2. Unrecognized text. ...o IMS] --> ENABLED&fl°
°123¶ßIMS Habilitad ----------^

Tentar novamente

Falhas de Chamada VoLTE

Sintomas

- Registro IMS bem-sucedido, mas chamadas falham
- Áudio unidirecional
- Chamada cai imediatamente
- Erro de "Chamada falhou" no dispositivo

Causas e Soluções Comuns

Causa 1: P-CSCF Não Conectado

Sintomas:

- Registro funciona, mas chamadas falham
- Autorização de mídia falha

Passos de Diagnóstico:

1. Verifique a página Diameter do Painel de Controle
2. Verifique se o par P-CSCF está conectado
3. Verifique se o P-CSCF suporta a aplicação Rx (função PCRF do OmniHSS)

Solução:

- Corrija a conectividade Diameter para o P-CSCF
- Verifique se a configuração do P-CSCF aponta para o OmniHSS para Rx

Causa 2: Autorização de Mídia Ausente

Sintomas:

- A configuração da chamada começa, mas falha
- Troca AAR/AAA falha
- Erros na interface Rx

Passos de Diagnóstico:

1. Verifique os logs para mensagens Diameter Rx
2. Verifique se o AAR (AA-Request) foi recebido
3. Verifique a resposta AAA (AA-Answer)

Solução:

- Verifique se o P-CSCF está enviando AAR para autorização de mídia
- Verifique a configuração da aplicação Rx do OmniHSS
- Verifique se o assinante tem registro IMS ativo

Causa 3: Problemas de QoS/Bearer

Sintomas:

- Chamada conecta, mas sem áudio
- Áudio unidirecional
- Problemas de qualidade

Passos de Diagnóstico:

1. Verifique o perfil QoS APN para o APN de voz
2. Verifique se o QCI está configurado corretamente (tipicamente QCI 1 para voz)
3. Verifique se o P-GW está conectado para Gx (função PCRF)

Solução:

- Verifique o perfil QoS APN para o APN IMS
- Certifique-se de que o QCI 1 está configurado para bearer de voz
- Corrija a conectividade Diameter para o P-GW, se necessário

Fluxograma de Solução de Problemas de Chamada VoLTE

Lexical error on line 2. Unrecognized text. ...ada VoLTE] --> REG&fl°
°123¶ßIMS Registrad ----------^

Tentar novamente

Problemas de Roaming

Sintomas

- Assinante funciona em casa, mas não quando está em roaming
- Algumas redes de roaming funcionam, outras não
- Roaming de dados funciona, mas não voz (ou vice-versa)

Causas e Soluções Comuns

Causa 1: Nenhum Perfil de Roaming Atribuído

Sintomas:
- Roaming falha para o assinante
- Outros assinantes rodam com sucesso

Passos de Diagnóstico:

1. Consultar `roaming_profile_id` do assinante
2. Verificar se o campo é nulo

Solução:
- Atribuir [perfil de roaming](./api-reference.md#roaming-management) ao assinante

Causa 2: Roaming Negado pela Política

Sintomas:
- Roaming falha consistentemente em uma rede específica
- Erro indica rejeição da política

Passos de Diagnóstico:

1. Identificar MCC/MNC da rede visitada a partir do dispositivo do assinante ou MME
2. Consultar o perfil de roaming do assinante
3. Verificar regras de roaming para MCC/MNC correspondente
4. Verificar a ação padrão do perfil

Solução:
- Adicionar [regra de roaming](./api-reference.md#create-roaming-rule) para permitir a rede visitada:
  ```bash
  curl -k -X POST https://hss.example.com:8443/api/roaming/rule \
    -H "Content-Type: application/json" \
    -d '{
      "roaming_rule": {
        "name": "Allow Visited Network",
        "mcc": "310",
        "mnc": "410",
  ```

```
        "data_action": "allow",
        "ims_action": "allow"
      }
    }'
```

Causa 3: Dados Permitidos, mas IMS Negado

Sintomas:

- Roaming de dados funciona
- Roaming de voz/IMS falha
- Disponibilidade de serviço dividida

Passos de Diagnóstico:

1. Consultar regras de roaming para a rede visitada
2. Verificar valores de `data_action` vs `ims_action`
3. Verificar ações padrão do perfil de roaming

Solução:

- Atualizar regra de roaming para permitir IMS:
  - Definir `ims_action: "allow"`
- Ou atualizar `ims_action_if_no_rules_match` do perfil para `"allow"`

Veja a Documentação de Roaming para configuração detalhada.

Problemas de EIR

Sintomas

- Dispositivos bloqueados inesperadamente
- Dispositivos roubados não bloqueados
- Verificação de EIR falhando

Causas e Soluções Comuns

Causa 1: Regex IMEI Incorreto

Sintomas:

- Dispositivos errados bloqueados/permitidos
- Regra corresponde incorretamente

Passos de Diagnóstico:

1. Consultar regras de EIR (`GET /api/eir/rule`). A lista é retornada em ordem de `id`, então classifique-a por `precedence` para ver a ordem de avaliação
2. Identificar qual regra está correspondendo. As regras são testadas contra o IMEISV (IMEI + Versão de Software) primeiro, depois o IMEI, e então o IMSI se a solicitação carregar um. Dentro de cada etapa, a regra correspondente com a menor `precedence` vence, então uma regra IMSI de baixa precedência nunca é alcançada se qualquer regra corresponder ao IMEISV ou IMEI. Veja a Lógica de Correspondência de Equipamentos
3. Testar o padrão regex contra o IMEISV, IMEI e IMSI reais. Padrões não são ancorados, então `35979139461650` também corresponde ao IMEISV `3597913946165008`
4. Verificar a precedência da regra (valores mais baixos são avaliados primeiro)

Solução:

- Atualizar regra de EIR com regex correto
- Testar regex minuciosamente antes de aplicar
- Considerar a precedência da regra (a regra correspondente com a menor precedência vence)

Causa 2: MME Não Enviando Solicitações S13

Sintomas:

- Verificação de EIR nunca acontece
- Todos os dispositivos permitidos independentemente das regras

Passos de Diagnóstico:

1. Verificar se o MME está configurado para usar a interface S13
2. Verificar se o par Diameter do MME está conectado
3. Verificar suporte à aplicação S13
4. Revisar configuração do MME

Solução:

- Configurar o MME para realizar verificações de EIR via S13
- Verificar se o par Diameter suporta a aplicação S13 (16777252)
- Contatar o administrador do MME se necessário

Causa 3: Sem Regra Padrão

Sintomas:

- Dispositivos que não correspondem a nenhuma regra têm comportamento inesperado

Passos de Diagnóstico:

1. Consultar todas as regras de EIR
2. Verificar se existe uma regra catch-all
3. Verificar a precedência da regra

Solução:

- Equipamentos que não correspondem a nenhuma regra recebem a configuração `unknown_equipment_behaviour` de `config :hss, :eir` (veja Configuração de EIR). Verifique isso primeiro
- Se preferir uma regra catch-all explícita, adicione uma regra com regex `.*` e a ação desejada, e dê a ela o maior número de `precedence` para que seja testada por último em cada etapa
- Uma regra `.*` corresponde na primeira etapa que é executada (IMEISV, ou IMEI quando a solicitação não carrega uma Versão de Software), então regras baseadas em IMSI e o padrão configurado nunca são alcançados para solicitações que carregam um IMEI

Problemas de Desempenho

Sintomas

- Respostas lentas da API
- Timeouts de solicitação Diameter
- Alto uso de CPU ou memória
- Painel de Controle lento para carregar

Causas e Soluções Comuns

Causa 1: Alta Carga no Banco de Dados

Sintomas:

- Todas as operações lentas
- CPU do banco de dados alta
- Timeouts de consulta

Passos de Diagnóstico:

1. Verificar uso de recursos do servidor de banco de dados
2. Identificar consultas lentas
3. Verificar índices ausentes
4. Monitorar padrões de consulta

Solução:

- Otimizar consultas lentas
- Adicionar índices ao banco de dados
- Aumentar recursos do banco de dados
- Considerar escalonamento do banco de dados
- Veja Problemas de Banco de Dados

Causa 2: Alto Número de Assinantes

Sintomas:

- Desempenho degradado ao longo do tempo
- Lentidão correlacionada com o crescimento de assinantes
- Operações de lista especialmente lentas

Passos de Diagnóstico:

1. Consultar contagem total de assinantes
2. Verificar tamanhos das tabelas
3. Revisar planos de execução de consultas
4. Monitorar tendências de uso de recursos

Solução:

- Planejar atualização de capacidade
- Otimizar consultas para grandes conjuntos de dados
- Considerar paginação para grandes resultados
- Implementar cache se necessário

Causa 3: Problemas com Pares Diameter

Sintomas:

- Operações Diameter lentas
- Timeouts em par específico
- Alguns pares rápidos, outros lentos

Passos de Diagnóstico:

1. Verificar página Diameter do Painel de Controle
2. Identificar par lento
3. Testar latência de rede para o par
4. Verificar uso de recursos do par

Solução:

- Investigar problemas de desempenho do par
- Verificar caminho de rede para congestionamento
- Considerar adicionar pares redundantes

- Aumentar timeout Diameter se necessário

Causa 4: Problemas de Memória

Sintomas:

- Uso de memória do OmniHSS alto
- Erros de falta de memória
- Desempenho degrada ao longo do tempo

Passos de Diagnóstico:

1. Verificar uso de memória do OmniHSS na página de Aplicação
2. Monitorar tendência de memória
3. Verificar vazamentos de memória
4. Revisar configurações da VM Erlang

Solução:

- Reiniciar o OmniHSS para limpar condição temporária
- Investigar vazamento de memória se o uso crescer continuamente
- Ajustar configurações de memória da VM Erlang em runtime.exs
- Planejar atualização de hardware se consistentemente alto

Problemas de Estado do Assinante

Sintomas

- Assinante aparece como anexado, mas não está
- Informações de estado desatualizadas
- Informações de localização incorretas
- Não é possível desanexar assinante

Causas e Soluções Comuns

Causa 1: Falha/Reinício do MME

Sintomas:

- Assinante mostra MME servindo que não está mais servindo
- Assinante não consegue anexar após reinício do MME
- Estado está desatualizado

Passos de Diagnóstico:

1. Verificar estado do assinante para MME servindo
2. Verificar se o MME reiniciou
3. Verificar o último horário de conexão do MME

Solução:

- Aguardar o assinante anexar novamente (o estado será atualizado)
- Ou limpar manualmente o estado do assinante
- O MME deve enviar Cancel-Location no reinício

Causa 2: Desanexação de Rede Não Recebida

Sintomas:

- Assinante desligado, mas aparece como anexado
- Sessões PDN permanecem no banco de dados
- Localização não é limpa

Passos de Diagnóstico:

1. Verificar timestamp `last_seen` do assinante
2. Verificar se o estado é antigo (horas ou dias)
3. Verificar se o dispositivo do assinante é acessível

Solução:

- O estado será limpo quando o assinante anexar novamente

- Ou aguardar o timeout de estado (se implementado)
- Limpeza manual pode ser necessária para estado muito desatualizado

Causa 3: Corrupção do Banco de Dados

Sintomas:

- Estado inconsistente entre tabelas
- Violações de chave estrangeira
- Estado não faz sentido

Passos de Diagnóstico:

1. Consultar estado do assinante diretamente do banco de dados
2. Verificar registros órfãos
3. Verificar integridade referencial

Solução:

- Identificar e corrigir dados inconsistentes
- Pode exigir limpeza manual do banco de dados
- Contatar suporte se a corrupção for generalizada

Problemas de API

Sintomas

- API retorna erros
- Respostas lentas da API
- Não é possível criar/atualizar entidades
- Erros 500

Causas e Soluções Comuns

Causa 1: Dados de Solicitação Inválidos

Sintomas:

- Erros 400 ou 422
- Mensagens de erro de validação
- Campo rejeitado

Passos de Diagnóstico:

1. Revisar resposta de erro para erros de campo específicos
2. Verificar formato da solicitação da API
3. Verificar se os campos obrigatórios estão presentes
4. Verificar tipos de dados

Solução:

- Corrigir dados da solicitação para corresponder à referência da API
- Garantir que todos os campos obrigatórios estejam incluídos
- Verificar se referências de chave estrangeira existem (IDs de perfil, etc.)

Causa 2: Restrição de Chave Estrangeira

Sintomas:

- Não é possível criar assinante
- Erro: "key_set_id não existe"
- Entidade referenciada não encontrada

Passos de Diagnóstico:

1. Identificar qual chave estrangeira está falhando
2. Verificar se a entidade referenciada existe:
    - key_set_id → conjuntos de chaves
    - epc_profile_id → perfis EPC
    - ims_profile_id → perfis IMS

Solu��ão:

- Criar a entidade referenciada primeiro

- Ou usar o ID da entidade existente
- Seguir fluxo de provisão completo

Causa 3: Conectividade com o Banco de Dados

Sintomas:

- Erros 500
- Todas as chamadas da API falham
- Erros de conexão com o banco de dados

Solução:

- Veja Problemas de Banco de Dados

Ferramentas e Comandos de Diagnóstico

Verificações Rápidas do Painel de Controle

1. **Visão Geral do Sistema**
    - URL: `https://[hostname]:7443/overview`
    - Verificar: Contagens de assinantes, sessões ativas, status do sistema
2. **Status Diameter**
    - URL: `https://[hostname]:7443/diameter`
    - Verificar: Todos os pares críticos conectados
3. **Saúde da Aplicação**
    - URL: `https://[hostname]:7443/application`
    - Verificar: Uso de memória, contagem de processos, tempo de atividade

Comandos de Diagnóstico da API

Verificar Saúde do Sistema:

```
curl -k https://hss.example.com:8443/api/status
```

Consultar Assinante:

```
# Por IMSI
curl -k
https://hss.example.com:8443/api/subscriber/imsi/001001123456789

# Por MSISDN
curl -k
https://hss.example.com:8443/api/subscriber/msisdn/14155551234

# Por ID
curl -k https://hss.example.com:8443/api/subscriber/d290f1ee-6c54-
4b01-90e6-d701748f0851
```

Listar Todos os Assinantes:

```
curl -k https://hss.example.com:8443/api/subscriber
```

Verificar Configuração do Perfil:

```
# Perfil EPC
curl -k https://hss.example.com:8443/api/epc/profile/e58ed763-
928c-4155-bee9-fdbaaadc15f3

# Perfil IMS
curl -k https://hss.example.com:8443/api/ims/profile/7c9e6679-
7425-40de-944b-e07fc1f90ae7

# Perfil de Roaming
curl -k https://hss.example.com:8443/api/roaming/profile/9c858901-
8a57-4791-81fe-4c455b099bc9
```

Comandos de Diagnóstico de Rede

Testar Conectividade da Porta Diameter:

```
telnet [PEER_IP] 3868
```

Verificar Certificado TLS:

```
openssl s_client -connect [hostname]:8443 -showcerts
```

Testar Conectividade com o Banco de Dados:

```
psql -h [DB_HOST] -U [DB_USER] -d [DB_NAME] -c "SELECT COUNT(*)
FROM subscriber;"
```

Análise de Logs

Pesquisar Logs para IMSI Específico:

```
grep "001001123456789" /var/log/omnihss/omnihss.log
```

Encontrar Falhas de Autenticação:

```
grep "authentication.*fail" /var/log/omnihss/omnihss.log
```

Verificar Eventos de Pares Diameter:

```
grep "Diameter peer" /var/log/omnihss/omnihss.log
```

Encontrar Erros de Banco de Dados:

```
grep -i "database.*error" /var/log/omnihss/omnihss.log
```

Diretrizes de Escalação

Quando Escalar

Escalar para suporte de engenharia/fornecedor quando:

1. **Falhas em todo o sistema** que não podem ser resolvidas com procedimentos documentados
2. **Corrupção de dados** ou estado inconsistente do banco de dados
3. **Suspeitas de bugs de software** ou comportamento inesperado
4. **Problemas de desempenho** que não podem ser resolvidos com ajustes
5. **Incidentes de segurança** ou acesso não autorizado
6. **Perguntas sobre comportamento não documentado**

Informações a Fornecer

Ao escalar, incluir:

1. **Sintomas detalhados** - O que está falhando, quando, para quem
2. **Passos tomados** - O que você já fez para solucionar
3. **Logs** - Trechos relevantes de logs mostrando o problema
4. **Configuração** - Porções relevantes de runtime.exs (redigir dados sensíveis)
5. **Ambiente** - Versão do OmniHSS, versão do banco de dados, versão do SO
6. **Impacto** - Quantos assinantes afetados, impacto nos negócios
7. **Exemplos de assinantes** - IMSIs específicos mostrando o problema

Crítico vs Não Crítico

Problemas Críticos (Escalar Imediatamente):

- Sistema completamente fora do ar
- Todos os assinantes incapazes de anexar
- Corrupção de banco de dados

- Violação de segurança

Problemas Não Críticos (Documentar e Escalar Durante o Horário Comercial):

- Problemas de assinante único que podem ser contornados
- Degradação de desempenho que é gerenciável
- Solicitações de melhoria
- Perguntas sobre documentação

Referência de Mensagens de Erro Comuns

Erros de Autenticação

Mensagem de Erro	Causa	Solução
"Falha na geração de vetores de autenticação"	Conjunto de chaves ausente ou inválido	Verificar configuração do conjunto de chaves
"Falha de sincronização SQN"	SQN fora de sincronia	Aguardar ressincronização
"Assinante não encontrado"	IMSI inválido	Verificar IMSI, provisionar assinante
"Assinante desativado"	enabled=false	Habilitar assinante

Erros Diameter

Mensagem de Erro	Causa	Solução
"Timeout de conexão com o par Diameter"	Problema de rede	Verificar conectividade de rede
"Falha na troca CER/CEA"	Desajuste de configuração	Verificar configuração Diameter
"Aplicação não suportada"	Par não suporta o aplicativo requerido	Verificar aplicações do par
"Falha na negociação TLS"	Problema de certificado	Verificar certificados

Erros de Banco de Dados

Mensagem de Erro	Causa	Solução
"Conexão recusada"	Banco de dados fora do ar	Iniciar banco de dados
"Falha na autenticação"	Credenciais erradas	Corrigir credenciais
"Sem conexões disponíveis"	Pool esgotado	Aumentar tamanho do pool
"Timeout de consulta"	Consulta lenta	Otimizar consultas

Erros de API

Mensagem de Erro	Causa	Solução
"key_set_id não existe"	Chave estrangeira inválida	Criar conjunto de chaves primeiro
"IMSI já foi utilizado"	IMSI duplicado	Usar IMSI diferente ou excluir existente
"Erro de validação"	Entrada inválida	Verificar formato e requisitos do campo

---

← Voltar para o Guia de Operações | Próximo: Referência da API →

Integração de Webhook do OmniHSS

← Voltar ao Guia de Operações

Índice

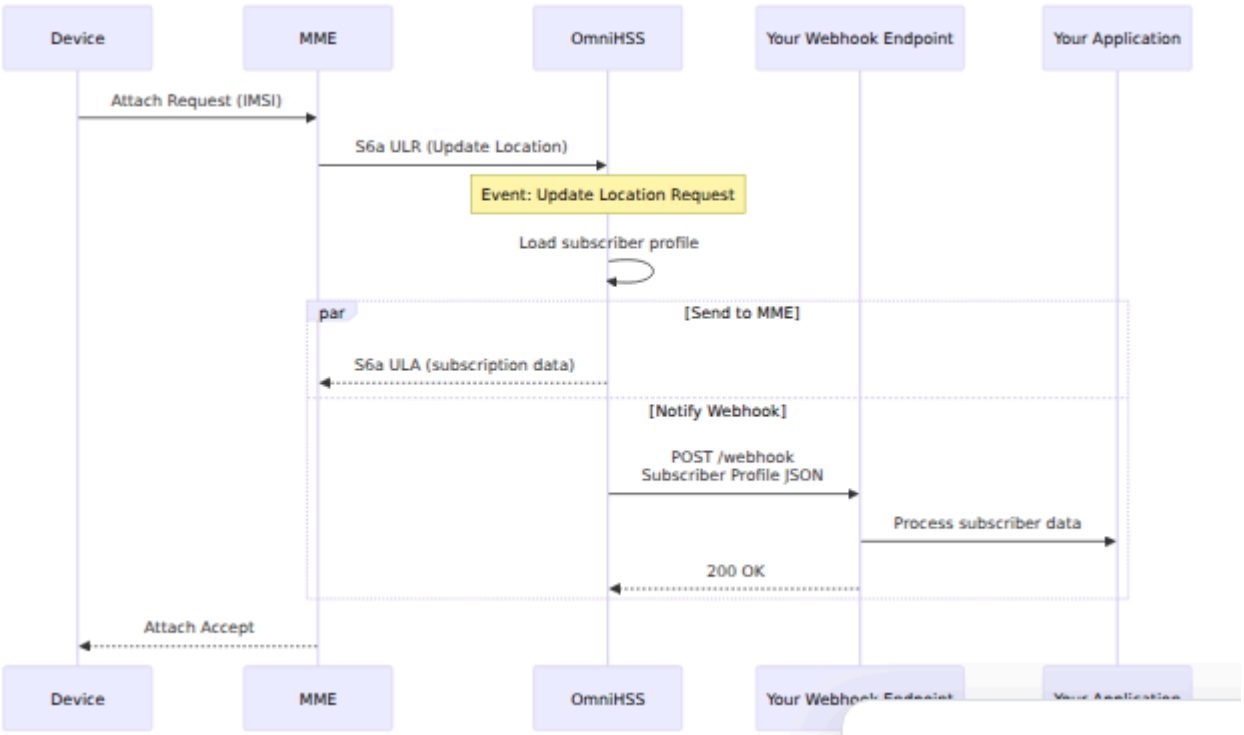
- Visão Geral
- Como Funcionam os Webhooks
- Eventos de Webhook
- Carga Útil do Webhook
- Configuração
- Casos de Uso
- Considerações de Segurança
- Solução de Problemas

Visão Geral

O OmniHSS suporta **webhooks** para notificar sistemas externos sobre eventos de assinantes em tempo real. Quando eventos específicos ocorrem (como atualizações de localização, solicitações de autenticação ou registros IMS), o OmniHSS pode enviar uma solicitação HTTP POST para o seu endpoint de webhook configurado com os dados completos do perfil do assinante.

O Que São Webhooks?

Webhooks são callbacks HTTP que permitem que o OmniHSS envie notificações de eventos para sua aplicação à medida que acontecem, em vez de exigir que sua aplicação consulte a API HSS em busca de alterações.



Principais Benefícios

- **Notificações em tempo real** - Receba atualizações instantâneas quando eventos de assinantes ocorrem
- **Dados completos do assinante** - Cada webhook inclui o perfil completo do assinante (mesmo que `GET /api/subscriber`)
- **Automação orientada a eventos** - Acione fluxos de trabalho, análises ou provisionamento com base em eventos de rede
- **Redução de polling** - Não há necessidade de consultar continuamente a API em busca de alterações de status do assinante
- **Flexibilidade de integração** - Conecte o OmniHSS a sistemas de faturamento, plataformas de análise ou aplicações personalizadas

---

Como Funcionam os Webhooks

Fluxo de Eventos

1. **Evento ocorre** - Um assinante realiza uma ação (anexar, atualização de localização, registro IMS, etc.)

2. **HSS processa o evento** - O OmniHSS lida com a solicitação/resposta Diameter normalmente
3. **Webhook acionado** - Se um webhook estiver registrado para este tipo de evento, o HSS envia um HTTP POST para seu endpoint
4. **Dados do assinante incluídos** - A carga útil do webhook contém o perfil completo do assinante em formato JSON
5. **Sua aplicação responde** - Seu endpoint deve retornar HTTP 200-299 para reconhecer o recebimento

Garantias de Entrega

- **Entrega de melhor esforço** - Webhooks são enviados de forma assíncrona e não bloqueiam operações de rede
- **Timeout** - Solicitações de webhook expiram após 5 segundos
- **Sem tentativas** - Se seu endpoint estiver indisponível ou retornar um erro, o webhook não será tentado novamente
- **Ordem não garantida** - Eventos podem chegar fora de ordem sob alta carga

Importante: Operações de rede (autenticação, atualizações de localização, etc.) **não** dependem da entrega do webhook. Se seu endpoint de webhook estiver fora do ar, o serviço do assinante continua normalmente.

Eventos de Webhook

O OmniHSS pode acionar webhooks para os seguintes eventos:

Eventos EPC/LTE

Evento	Acionador	Descrição
`update_location_request`	S6a ULR	Assinante anexa ou realiza Atualização de Área de Rastreamento
`authentication_information_request`	S6a AIR	A rede solicita vetores de autenticação para o assinante
`purge_request`	S6a PUR	MME remove o contexto do assinante (dispositivo desligado, desconectado)
`cancel_location_answer`	S6a CLA	MME reconhece a desregistro do assinante

Eventos IMS

Evento	Acionador	Descrição
`ims_registration`	Cx SAR	Assinante registra para o serviço IMS/VoLTE
`ims_deregistration`	Cx SAR (des-reg)	Assinante se desregistra do IMS
`ims_profile_request`	Sh UDR	O Servidor de Aplicação solicita o perfil IMS do assinante

Eventos de Política (PCRF)

Evento	Acionador	Descrição
`policy_request`	Gx CCR	P-GW solicita política para a sessão de dados do assinante
`media_authorization`	Rx AAR	P-CSCF solicita autorização de mídia para chamada IMS

Eventos Multi-IMSI

Evento	Acionador	Descrição
`imsi_switch`	ULR para IMSI diferente no mesmo SIM	Dispositivo muda para um IMSI diferente em um SIM multi-IMSI

Carga Útil do Webhook

Formato da Solicitação

Quando um evento ocorre, o OmniHSS envia uma solicitação HTTP POST para a URL do webhook configurada:

```
POST /your-webhook-endpoint HTTP/1.1
Host: your-server.com
Content-Type: application/json
X-OmniHSS-Event: update_location_request
X-OmniHSS-Event-ID: 550e8400-e29b-41d4-a716-446655440000
X-OmniHSS-Timestamp: 2025-01-15T14:30:00Z

{
  "event": "update_location_request",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "timestamp": "2025-01-15T14:30:00Z",
  "subscriber": {
    "id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "imsi": "001001123456789",
    "enabled": true,
    "ims_enabled": true,
    "msisdns": [
      {"id": "16fd2706-8baf-433b-82eb-8c7fada847da", "msisdn":
"14155551001"},
      {"id": "27ae3807-9cbf-4a4c-93fc-9d8afdb958eb", "msisdn":
"14155551002"}
    ],
    "sim": {
      "id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
      "iccid": "8991101200003204510",
      "is_esim": false
    },
    "key_set": {
      "id": "6ba7b810-9dad-41d1-80b4-00c04fd430c8",
      "amf": "8000"
    },
    "epc_profile": {
      "id": "e58ed763-928c-4155-bee9-fdbaaadc15f3",
      "name": "Premium 100Mbps",
      "ue_ambr_dl_kbps": 100000,
      "ue_ambr_ul_kbps": 50000
    },
    "ims_profile": {
      "id": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
      "name": "Standard VoLTE"
    },
    "roaming_profile": {
      "id": "9c858901-8a57-4791-81fe-4c455b099bc9",
```

```
      "name": "International Roaming Allowed"
    },
    "subscriber_state": {
      "mme_host": "mme-01.example.com",
      "mme_realm": "epc.mnc001.mcc001.3gppnetwork.org",
      "visited_plmn": "001001",
      "last_update": "2025-01-15T14:30:00Z"
    },
    "custom_attributes": {
      "account_type": "premium",
      "billing_plan": "unlimited"
    }
  },
  "event_context": {
    "visited_plmn": "310410",
    "mme_host": "mme-roaming.example.com",
    "location_update_type": "initial_attach"
  }
}
```

Estrutura da Carga Útil

Campo	Tipo	Descrição
`event`	string	Tipo de evento (ex: `update_location_request`)
`event_id`	string	UUID único para esta entrega de webhook
`timestamp`	string	Timestamp ISO 8601 quando o evento ocorreu
`subscriber`	object	**Perfil completo do assinante** (mesmo que `GET /api/subscriber/:id`)
`event_context`	object	Dados contextuais adicionais específicos do evento

Campos do Contexto do Evento

O objeto `event_context` contém informações específicas do evento:

Para `update_location_request`:

```
{
  "visited_plmn": "310410",
  "mme_host": "mme-roaming.example.com",
  "mme_realm": "epc.mnc410.mcc310.3gppnetwork.org",
  "location_update_type": "initial_attach"
}
```

Para `imsi_switch`:

```
{
  "previous_imsi": "001001111111111",
  "new_imsi": "310410222222222",
  "sim_id": "3f2504e0-4f89-41d3-9a0c-0305e82c3301",
  "previous_mme_host": "mme-home.example.com",
  "new_mme_host": "mme-roaming.example.com"
}
```

Para `ims_registration`:

```
{
  "scscf_host": "scscf-01.ims.example.com",
  "public_identities": [
    "sip:001001123456789@ims.mnc001.mcc001.3gppnetwork.org",
    "sip:+14155551001@ims.example.com",
    "tel:+14155551001"
  ]
}
```

Cabeçalhos HTTP

Cabeçalho	Descrição	Exemplo
`Content-Type`	Sempre `application/json`	`application/json`
`X-OmniHSS-Event`	Tipo de evento	`update_location_request`
`X-OmniHSS-Event-ID`	Identificador único do evento	UUID
`X-OmniHSS-Timestamp`	Timestamp do evento	Formato ISO 8601
`User-Agent`	Versão do OmniHSS	`OmniHSS/1.0`

Configuração

Registrando Webhooks

Webhooks são configurados via API do OmniHSS.

Registrar um Webhook

```
curl -k -X POST https://hss.example.com:8443/api/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "url": "https://your-server.com/omnihss-webhook",
      "events": [
        "update_location_request",
        "ims_registration",
        "imsi_switch"
      ],
      "enabled": true,
      "description": "Webhook do sistema de faturamento de
produção"
    }
  }'
```

Resposta:

```
{
  "data": {
    "id": "4e5d6c7b-8a9f-4b0c-8d1e-2f3a4b5c6d7e",
    "url": "https://your-server.com/omnihss-webhook",
    "events": [
      "update_location_request",
      "ims_registration",
      "imsi_switch"
    ],
    "enabled": true,
    "description": "Webhook do sistema de faturamento de
produção",
    "created_at": "2025-01-15T14:00:00Z"
  }
}
```

Listar Webhooks

```
curl -k https://hss.example.com:8443/api/webhook
```

Atualizar Webhook

```
curl -k -X PUT https://hss.example.com:8443/api/webhook/4e5d6c7b-
8a9f-4b0c-8d1e-2f3a4b5c6d7e \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "enabled": false
    }
  }'
```

Deletar Webhook

```
curl -k -X DELETE
https://hss.example.com:8443/api/webhook/4e5d6c7b-8a9f-4b0c-8d1e-
2f3a4b5c6d7e
```

Requisitos do Endpoint do Webhook

Seu endpoint de webhook deve:

1. **Aceitar solicitações POST** com `Content-Type: application/json`
2. **Responder rapidamente** - Retornar HTTP 200-299 dentro de 5 segundos
3. **Ser idempotente** - Lidar com entregas duplicadas de forma adequada
4. **Usar HTTPS** - Para segurança, use endpoints TLS/SSL (recomendado)
5. **Validar cargas úteis** - Verificar se a solicitação é do OmniHSS (veja a seção de Segurança)

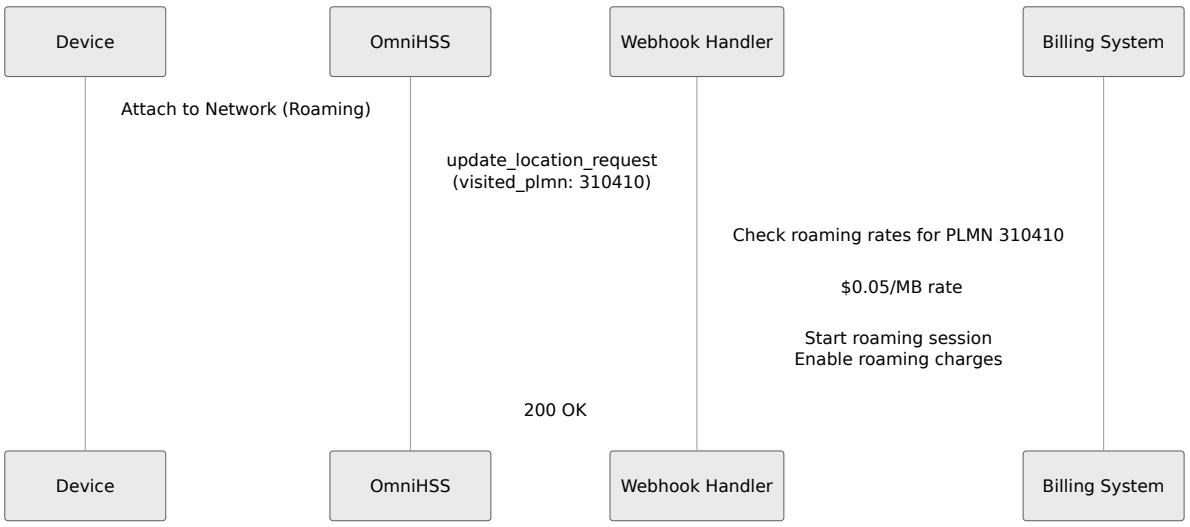
Exemplo de Manipulador de Webhook (Node.js/Express):

```
const express = require('express');
const app = express();

app.post('/omnihss-webhook', express.json(), (req, res) => {
  const { event, subscriber, event_context } = req.body;

  console.log(`Evento recebido: ${event}`);
  console.log(`IMSI do assinante: ${subscriber.imsi}`);

  // Processar os dados do assinante
  // ... sua lógica de negócios aqui ...

  // Responder imediatamente para reconhecer o recebimento
  res.status(200).json({ received: true });

  // Lidar com processamento assíncrono após a resposta
  processWebhook(req.body).catch(console.error);
});

async function processWebhook(payload) {
  // Sua lógica de processamento assíncrono
  // ex: atualizar sistema de faturamento, acionar análises, etc.
}

app.listen(3000);
```

Casos de Uso

1. Faturamento em Tempo Real e Monitoramento de Uso

Rastrear o uso da rede do assinante e acionar eventos de faturamento em tempo real.



Benefícios:

- Detectar instantaneamente quando assinantes estão em roaming internacional
- Aplicar taxas de roaming apropriadas em tempo real
- Rastrear horários de início/fim da sessão com precisão
- Gerar alertas de uso quando os limites são atingidos

2. Análise e Monitoramento

Enviar dados de atividade do assinante para plataformas de análise para painéis e relatórios em tempo real.

Caso de Uso: Rastrear assinantes ativos por região

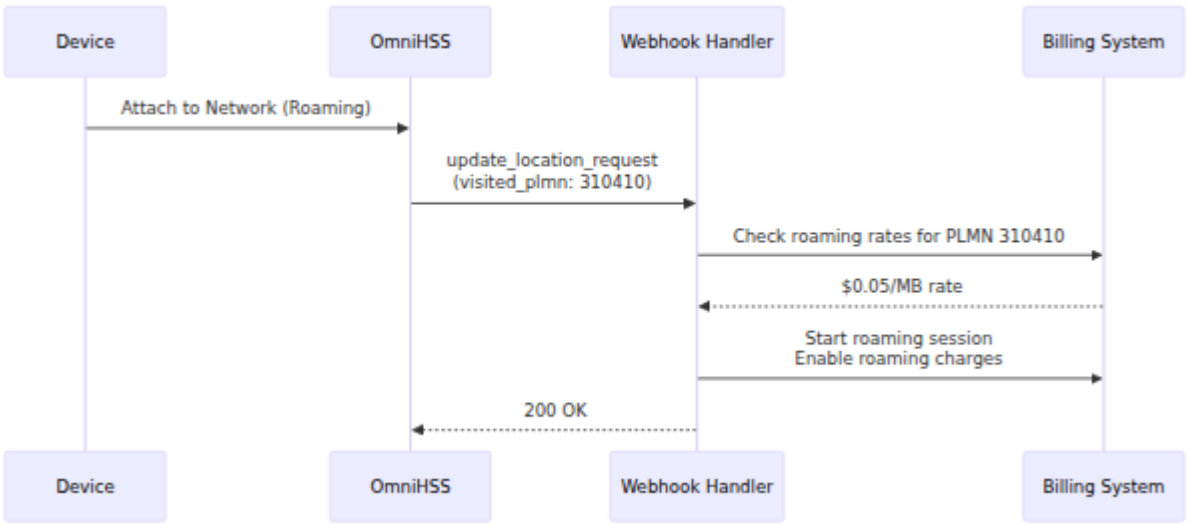
```
// Manipulador de webhook alimentando dados para a plataforma de análise
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'update_location_request') {
    await analytics.track({
      event: 'subscriber_location_update',
      imsi: subscriber.imsi,
      visited_plmn: event_context.visited_plmn,
      timestamp: req.body.timestamp,
      profile: subscriber.epc_profile.name
    });
  }

  res.status(200).send();
});
```

Painel de Análise:

- Assinantes ativos por MME
- Assinantes em roaming por país
- Distribuição de níveis de serviço
- Taxas de sucesso de registro IMS

3. Detecção de Fraude e Segurança

Detectar padrões de atividade suspeita em tempo real e acionar respostas automatizadas.



Cenários de Detecção de Fraude:

1. **Mudanças Rápidas de Localização**
    - Assinante anexa no País A
    - 30 minutos depois, anexa no País B (fisicamente impossível)
    - Ação: Marcar conta, enviar alerta para equipe de segurança
2. **Abuso de Troca de IMSI**
    - Múltiplas trocas rápidas de IMSI no mesmo SIM
    - Possível clonagem de SIM ou uso não autorizado de multi-IMSI
    - Ação: Desabilitar todos os IMSIs no SIM, notificar equipe de fraude
3. **Roaming Não Autorizado**
    - Assinante vai para país bloqueado (sanções, risco de fraude)
    - Ação: Desabilitar assinante automaticamente, bloquear acesso à rede

Exemplo de Implementação:

```python
@app.route('/omnihss-webhook', methods=['POST'])
def webhook_handler():
    data = request.json
    subscriber = data['subscriber']
    event_context = data.get('event_context', {})

    if data['event'] == 'update_location_request':
        visited_plmn = event_context.get('visited_plmn')

        # Verificar países bloqueados
        if visited_plmn in BLOCKED_PLMNS:
            disable_subscriber(subscriber['imsi'])
            alert_security_team(subscriber, 'Roaming para PLMN
bloqueado')

        # Verificar viagem impossível
        if is_impossible_travel(subscriber['imsi'], visited_plmn):
            flag_for_review(subscriber['imsi'])
            alert_fraud_team(subscriber, 'Viagem impossível
detectada')

    return jsonify({'status': 'ok'}), 200
```

4. Automação de Provisionamento

Provisionar ou atualizar automaticamente serviços de assinantes com base em eventos de rede.

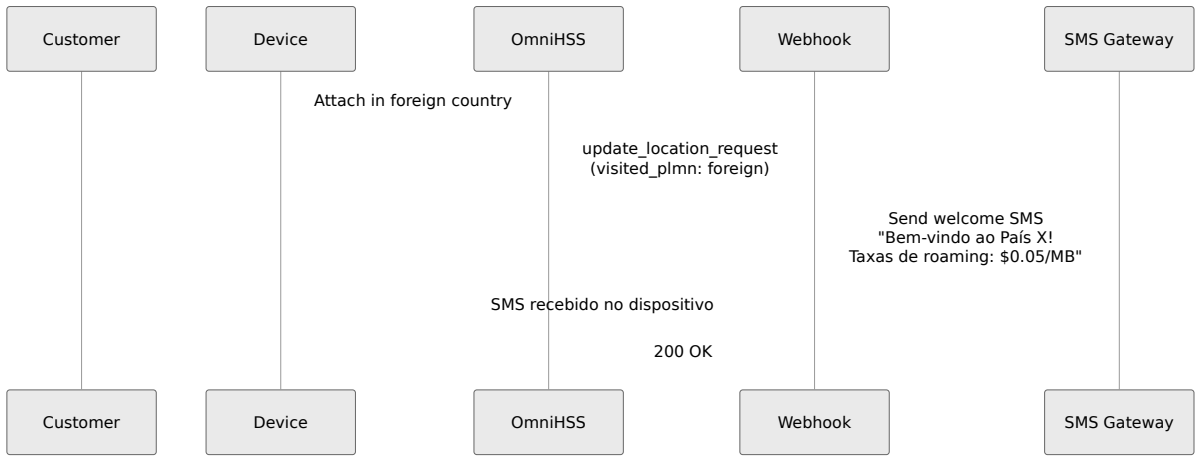
Caso de Uso: Ativar automaticamente IMS quando o assinante usa VoLTE pela primeira vez

```
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber } = req.body;

  if (event === 'ims_registration' && !subscriber.ims_enabled) {
    // Primeiro usuário de IMS - habilitar IMS permanentemente
    await omnihss.updateSubscriber(subscriber.id, {
      ims_enabled: true,
      custom_attributes: {
        ...subscriber.custom_attributes,
        volte_activated_at: new Date().toISOString()
      }
    });

    // Atualizar CRM
    await crm.updateCustomer(subscriber.imsi, {
      features: ['volte']
    });
  }

  res.status(200).send();
});
```

5. Notificações para Clientes

Enviar notificações em tempo real para clientes sobre seus serviços.

Caso de Uso: Mensagem de boas-vindas ao entrar em roaming internacional



Exemplo de Notificações:

- "Bem-vindo ao [País]! Taxas de roaming se aplicam."
- "Você usou 80% da sua cota de dados"
- "Serviço VoLTE agora ativo em seu dispositivo"
- "Sua conta foi atualizada para Premium"

6. Gestão de SIM Multi-IMSI

Rastrear e gerenciar assinantes com SIMs multi-IMSI, recebendo notificações quando eles trocam IMSIs.

```
app.post('/omnihss-webhook', async (req, res) => {
  const { event, subscriber, event_context } = req.body;

  if (event === 'imsi_switch') {
    const { previous_imsi, new_imsi, sim_id } = event_context;

    // Registrar troca de IMSI para análise
    await db.logImsiSwitch({
      sim_id,
      from_imsi: previous_imsi,
      to_imsi: new_imsi,
      timestamp: req.body.timestamp
    });

    // Atualizar sistema de faturamento
    await billing.endSession(previous_imsi);
    await billing.startSession(new_imsi);

    // Alertar se troca excessiva (potencial fraude)
    const switchCount = await db.getSwitchCount(sim_id, '24h');
    if (switchCount > 10) {
      await alertFraudTeam(`Troca excessiva de IMSI: SIM
${sim_id}`);
    }
  }

  res.status(200).send();
});
```

7. Integração com Sistemas Externos

Conectar o OmniHSS a sistemas de terceiros sem polling.

Exemplos de Integrações:

- **Sistemas de CRM** - Atualizar registros de clientes com uso de serviços
- **Monitoramento de Rede** - Alimentar dados de assinantes para plataformas de análise de rede
- **Sistemas de Faturamento** - Acionar cobranças com base em eventos de rede
- **Sistemas de Tickets** - Criar tickets automaticamente para autenticações falhadas
- **Armazéns de Dados** - Transmitir eventos de assinantes para análise de big data

---

Considerações de Segurança

Segredo/Assinatura do Webhook

Para verificar se os webhooks são do OmniHSS, implemente a verificação de assinatura:

```
# Configurar webhook com segredo
curl -k -X POST https://hss.example.com:8443/api/webhook \
  -H "Content-Type: application/json" \
  -d '{
    "webhook": {
      "url": "https://your-server.com/omnihss-webhook",
      "events": ["update_location_request"],
      "secret": "your-secret-key-here"
    }
  }'
```

O OmniHSS incluirá um cabeçalho `X-OmniHSS-Signature`:

```
X-OmniHSS-Signature:
sha256=5d7a8f9b2c1e3a4d6f7e8b9c0a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a
```

Verifique a assinatura:

```javascript
const crypto = require('crypto');

function verifyWebhook(req) {
  const signature = req.headers['x-omnihss-signature'];
  const secret = process.env.WEBHOOK_SECRET;
  const payload = JSON.stringify(req.body);

  const expectedSignature = 'sha256=' +
    crypto.createHmac('sha256', secret)
      .update(payload)
      .digest('hex');

  return crypto.timingSafeEqual(
    Buffer.from(signature),
    Buffer.from(expectedSignature)
  );
}

app.post('/omnihss-webhook', (req, res) => {
  if (!verifyWebhook(req)) {
    return res.status(401).json({ error: 'Assinatura inválida' });
  }

  // Processar webhook...
  res.status(200).send();
});
```

Melhores Práticas

1. **Use HTTPS** - Sempre use TLS para endpoints de webhook
2. **Valide assinaturas** - Verifique assinaturas de webhook para evitar spoofing
3. **Limitação de taxa** - Implemente limitação de taxa em endpoints de webhook

4. **Lista de IPs permitidos** - Restringir o acesso ao webhook aos endereços IP do OmniHSS
5. **Monitore falhas** - Rastrear falhas e erros de entrega de webhook
6. **Sanitize dados** - Validar e sanitizar cargas úteis de webhook antes de processá-las
7. **Credenciais seguras** - Armazenar segredos de webhook em configuração segura (variáveis de ambiente, gerenciador de segredos)

Privacidade de Dados

As cargas úteis do webhook contêm **informações sensíveis do assinante**:

- IMSI (identidade do assinante)
- MSISDNs (números de telefone)
- Dados de localização (PLMN visitado, MME)
- Informações do perfil de serviço

Requisitos de Conformidade:

- **GDPR** - Garantir que os dados do webhook sejam processados em conformidade com o GDPR
- **Retenção de dados** - Implementar políticas de retenção de dados apropriadas
- **Controle de acesso** - Restringir o acesso ao endpoint do webhook
- **Criptografia** - Usar TLS para transporte de webhook
- **Registro de auditoria** - Registrar todas as entregas de webhook para conformidade

---

Solução de Problemas

Webhook Não Recebido

Sintomas:

- Eventos ocorrem, mas o webhook não é acionado

- Endpoint do webhook nunca recebe solicitações

Passos de Solução de Problemas:

1. **Verifique se o webhook está habilitado:**

```
curl -k https://hss.example.com:8443/api/webhook
# Verifique "enabled": true
```

2. **Verifique a configuração dos eventos do webhook:**
   - Certifique-se de que o tipo de evento esteja incluído na lista de `events` do webhook
   - Exemplo: Se você deseja eventos `ims_registration`, verifique se está na matriz de eventos
3. **Revisar logs do HSS:**
   - Verificar erros de entrega do webhook
   - Procurar problemas de conectividade de rede
   - Verificar se não há falhas de resolução de DNS
4. **Testar acessibilidade do endpoint:**

```
curl -X POST https://your-server.com/omnihss-webhook \
  -H "Content-Type: application/json" \
  -d '{"test": true}'
```

Webhook Expirando

Sintomas:

- Logs do HSS mostram erros de timeout do webhook
- Endpoint do webhook recebe a solicitação, mas o HSS marca como falha

Solução:

1. **Responder imediatamente:**

- Retornar HTTP 200 dentro de 5 segundos
- Processar dados de forma assíncrona após responder

2. **Otimizar o desempenho do endpoint:**

```
// RUIM - Processamento síncrono lento
app.post('/webhook', (req, res) => {
  processData(req.body); // Bloqueia por 10 segundos
  res.status(200).send();
});

// BOM - Processamento assíncrono após resposta
app.post('/webhook', (req, res) => {
  res.status(200).send(); // Responder imediatamente
  processData(req.body); // Processar assíncrono
});
```

Webhooks Duplicados

Sintomas:

- Mesmo evento entregue várias vezes
- `event_id` é idêntico para entregas duplicadas

Causa:

- Retentativas de rede (embora o OmniHSS não retente, a infraestrutura de rede pode)
- Múltiplos webhooks registrados para o mesmo evento

Solução:

Implementar idempotência usando `event_id`:

```
const processedEvents = new Set();

app.post('/omnihss-webhook', (req, res) => {
  const eventId = req.body.event_id;

  if (processedEvents.has(eventId)) {
    // Já processado, pular
    return res.status(200).json({ status: 'duplicate' });
  }

  processedEvents.add(eventId);

  // Processar webhook...
  processWebhook(req.body);

  res.status(200).json({ status: 'processed' });
});
```

Webhook Retorna Erro

Sintomas:

- Endpoint retorna HTTP 4xx ou 5xx
- Logs do HSS falha na entrega do webhook

Erros Comuns:

1. **401 Não Autorizado** - Verificação de assinatura falhou
    - Verificar se o segredo do webhook corresponde à configuração
    - Verificar algoritmo de cálculo da assinatura
2. **400 Solicitação Inválida** - Payload inválido
    - Verificar análise do payload do webhook
    - Garantir que o cabeçalho Content-Type seja tratado
3. **500 Erro Interno do Servidor** - Endpoint falhou
    - Revisar logs de erro do endpoint

- Adicionar tratamento de erro e logging

Solução:

Adicionar tratamento de erro abrangente:

```
app.post('/omnihss-webhook', async (req, res) => {
  try {
    // Verificar assinatura
    if (!verifyWebhook(req)) {
      return res.status(401).json({ error: 'Assinatura inválida'
});
    }

    // Validar payload
    if (!req.body.event || !req.body.subscriber) {
      return res.status(400).json({ error: 'Payload inválido' });
    }

    // Processar webhook
    await processWebhook(req.body);

    res.status(200).json({ status: 'ok' });

  } catch (error) {
    console.error('Erro no processamento do webhook:', error);
    // Retornar 200 para evitar retentativa, registrar erro para
investigação
    res.status(200).json({ status: 'error', message: error.message
});
  }
});
```

Dados do Assinante Ausentes

Sintomas:

- Webhook recebido, mas objeto do assinante está incompleto
- Campos esperados estão nulos ou ausentes

Possíveis Causas:

1. **Assinante não totalmente provisionado** - Alguns perfis podem ser opcionais (IMS, roaming)
2. **Condição de corrida de dados** - Assinante atualizado entre o acionamento do evento e o envio do webhook

Solução:

Tratar campos opcionais de forma adequada:

```
const { subscriber } = req.body;

// Verificar campos opcionais
const imsProfile = subscriber.ims_profile || { name: 'Sem IMS' };
const roamingProfile = subscriber.roaming_profile || { name: 'Sem
Roaming' };

// Tratar MSISDNs ausentes
const msisdns = subscriber.msisdns || [];
```

Monitoramento e Observabilidade

Métricas do Webhook

Acompanhar o desempenho e a confiabilidade do webhook:

Métricas a Monitorar:

- Taxa de entrega do webhook (bem-sucedida vs. falhada)
- Latência do webhook (tempo do evento até a resposta do endpoint)
- Tempos de resposta do endpoint
- Taxas de erro por endpoint
- Eventos por segundo

Exemplo de Consulta de Dashboard (Prometheus/Grafana):

```
# Taxa de sucesso do webhook
rate(omnihss_webhook_success_total[5m]) /
rate(omnihss_webhook_attempts_total[5m])

# Latência do webhook
histogram_quantile(0.95, omnihss_webhook_duration_seconds)
```

Logs do Webhook

Ativar logging detalhado do webhook para solução de problemas:

Formato de Log:

```
{
  "timestamp": "2025-01-15T14:30:00Z",
  "level": "info",
  "component": "webhook",
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "webhook_id": "4e5d6c7b-8a9f-4b0c-8d1e-2f3a4b5c6d7e",
  "event_type": "update_location_request",
  "subscriber_imsi": "001001123456789",
  "endpoint": "https://your-server.com/omnihss-webhook",
  "http_status": 200,
  "duration_ms": 145,
  "error": null
}
```

← Voltar ao Guia de Operações | Próximo: Referência da API →