

OmniTWAG 無線LAN接続

Omnitouch 無線LAN接続

無線LAN接続は、OmniTWAG 無線LAN接続

無線LAN接続

- 無線LAN接続
- 無線LAN WiFi 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN 2.0 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続
- 無線LAN接続

無線LAN接続

OmniTWAG無線LAN WiFi 無線LAN接続は、3GPP TWAG 無線LAN接続は、無線LAN接続は、無線LAN WiFi 無線LAN接続は、SIM 無線LAN接続

TWAG 無線LAN EAP-AKA無線LAN接続 - 無線LAN接続は、SIM 無線LAN WiFi 無線LAN接続は、無線LAN WiFi 無線LAN接続は、WiFi 無線LAN接続は、WiFi 無線LAN接続

無線LAN接続

無線LAN接続

- 電話番号を SIM に登録
- 電話番号をロック解除
- 電話番号を WiFi WPA2
- 電話番号 SIM 番号

□□□□□□□□

- 00000000000000000000
- 00000000000000000000
- 00000000WiFi 0000000000
- 000000WiFi 000000000000
- 000000WiFi 000000000 IMSI
- 0000000000000000 WiFi 000000

□□□□/□□□

- 00000000000000000000
- 00000000000000000000
- 0000000000 WiFi 00

WiFi

[illegible]

TWAG ☐ ☐ ☐ ☐ ☐ ☐

TWAG □□□□□□□□

- **WiFi** 無線 LAN RADIUS 接続
- 無線 LAN HSS/HLR 接続 Diameter SWx 接続

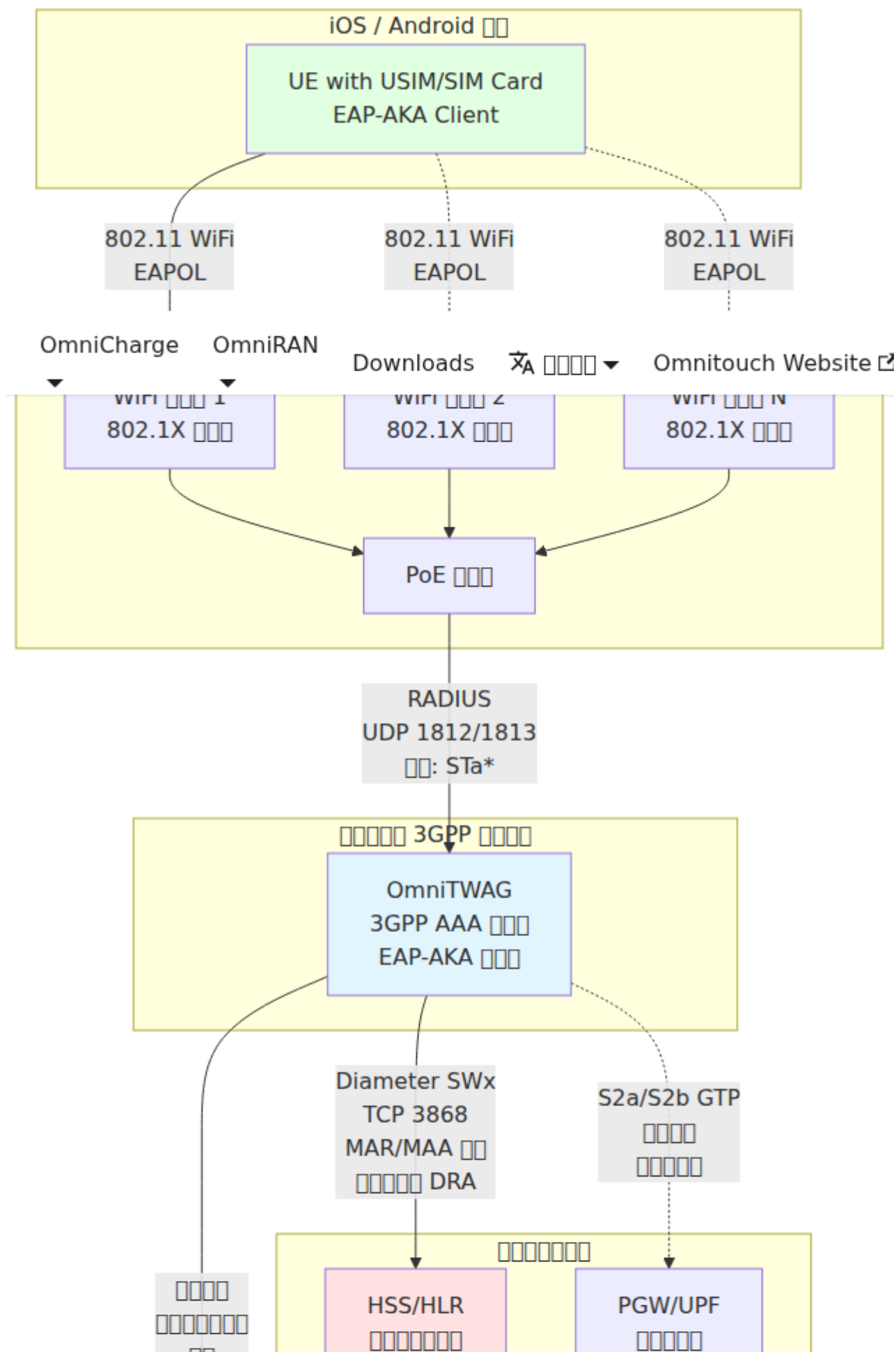
WiFi AP

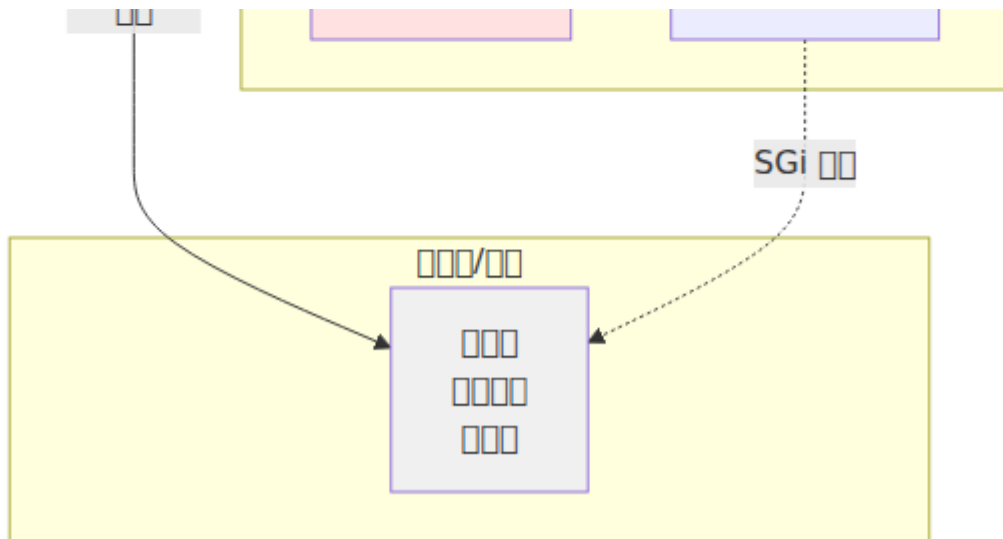
1. IMSI SIM 無線電波
2. WiFi AP 無線電波 RADIUS 無線電波 TWAG

3. TWAG 00000 HSS 0000000000
 4. 000 TWAG 0000 EAP-AKA 00-0000
 5. 00000000000 WiFi 0000
 6. 00000000000000000000000000
-

□□□□

□□□□



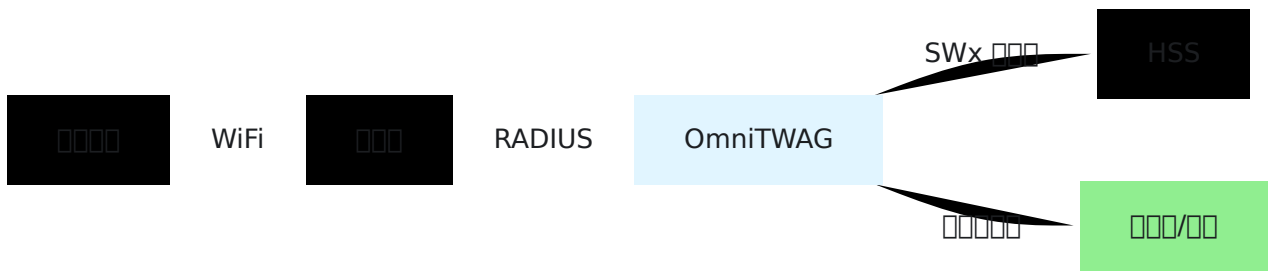


네트워크

- **STa***: WiFi AP ↔ TWAG ↔ RADIUS/Diameter ↔ 3GPP ↔ AAA
- **SWx**: TWAG ↔ 3GPP AAA ↔ HSS ↔ Diameter
- **S2a/S2b**: MME ↔ GTP ↔ S-GW
- **SGi**: S-GW ↔ Internet
- **802.11**: WiFi 네트워크
- **EAPOL**: EAP over LAN ↔ 802.1X

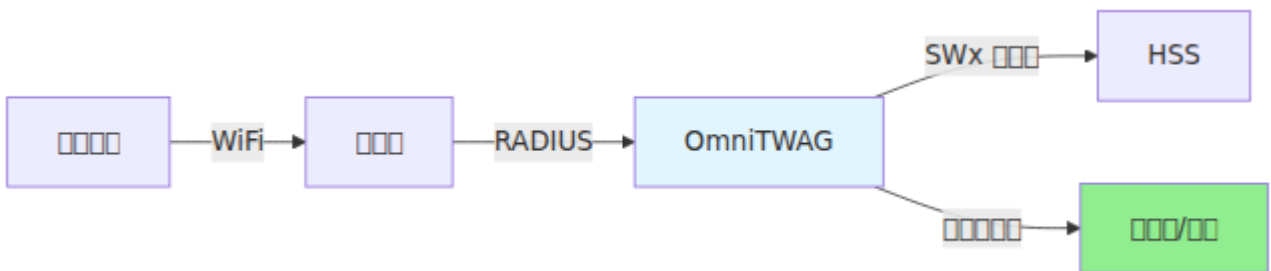
네트워크

1. 네트워크 구성



-
-
-
-

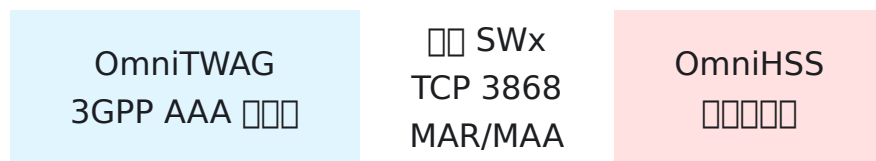
2



-
-
- VPN/
- WiFi

SWx

1
 HSS



HSS

OmniTWAG 3GPP AAA Client TWAG Client

OmniHSS

OmniHSS Client

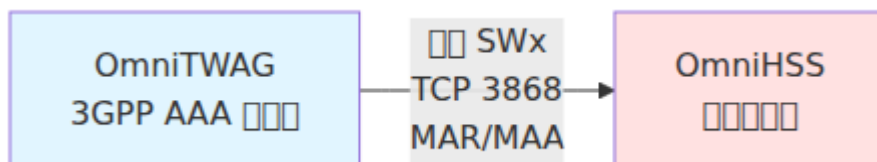
1. Client

OmniHSS Client

Client

- Omni/WiFi Client
- Omni/Client
- Omni/Client RADIUS Client

Client



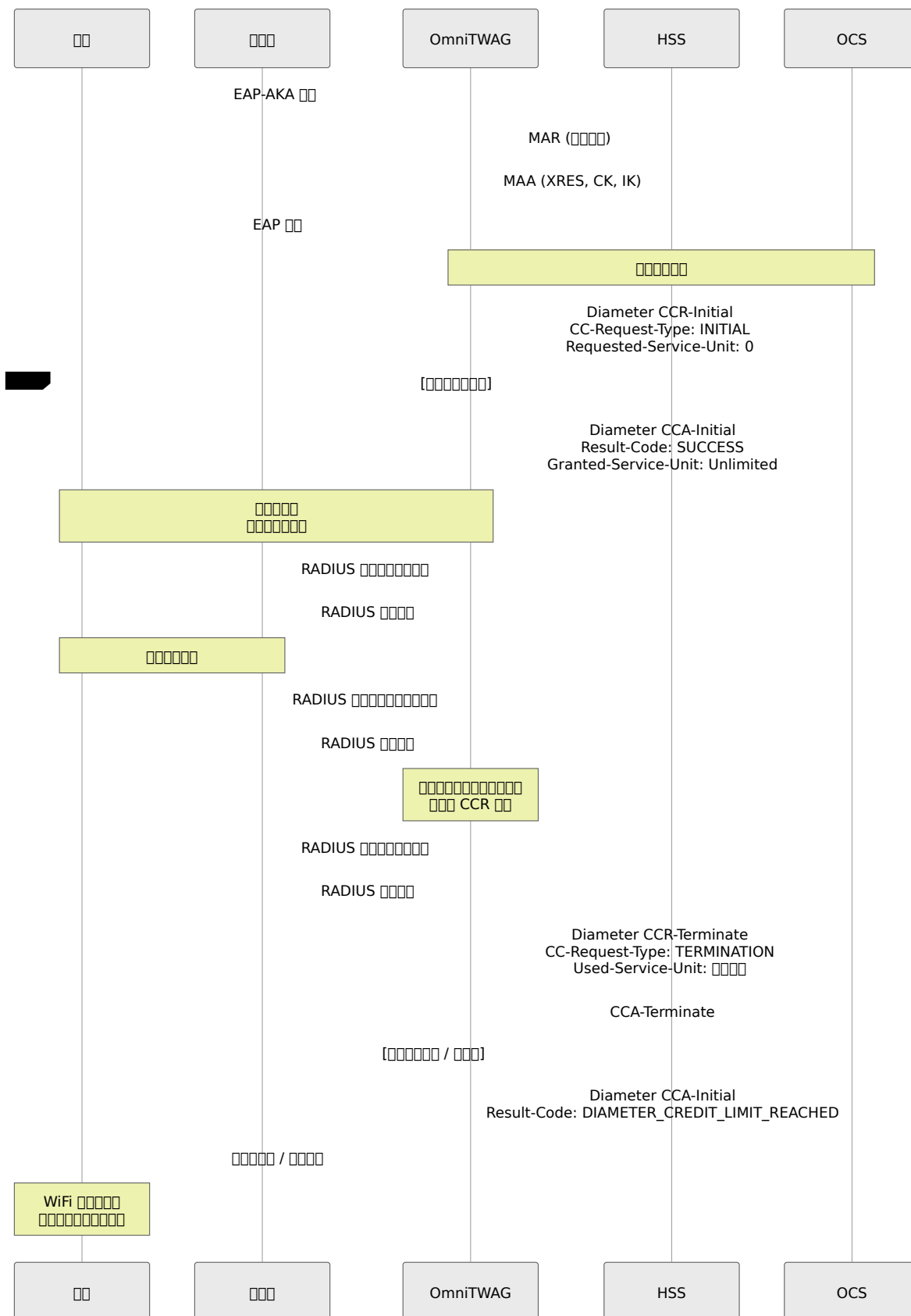
2. Client

Omni WiFi Client OCS Client CCR-Initial Client

Client

- 000000000000/00
- 00000000 WiFi 00
- 00000000000000
- 00 WiFi 0000000000/000000

000



...

- Diameter CCR-Initial message to OCS
- Diameter CCR-Terminate message

- 3GPP 3GPP
- 3GPP 3GPP

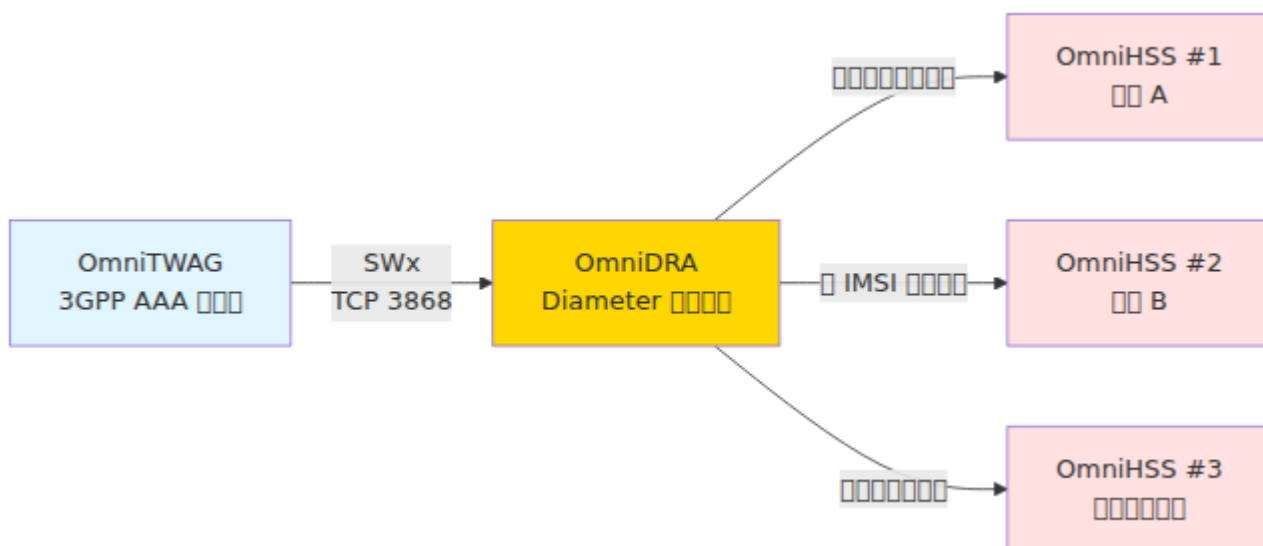
3. 3GPP Gy 3GPP

3GPP 3GPP WiFi 3GPP 3GPP 3GPP OCS 3GPP 3GPP

3GPP

- 3GPP
- 3GPP WiFi
- 3GPP 10GB 3GPP
- 3GPP

3GPP

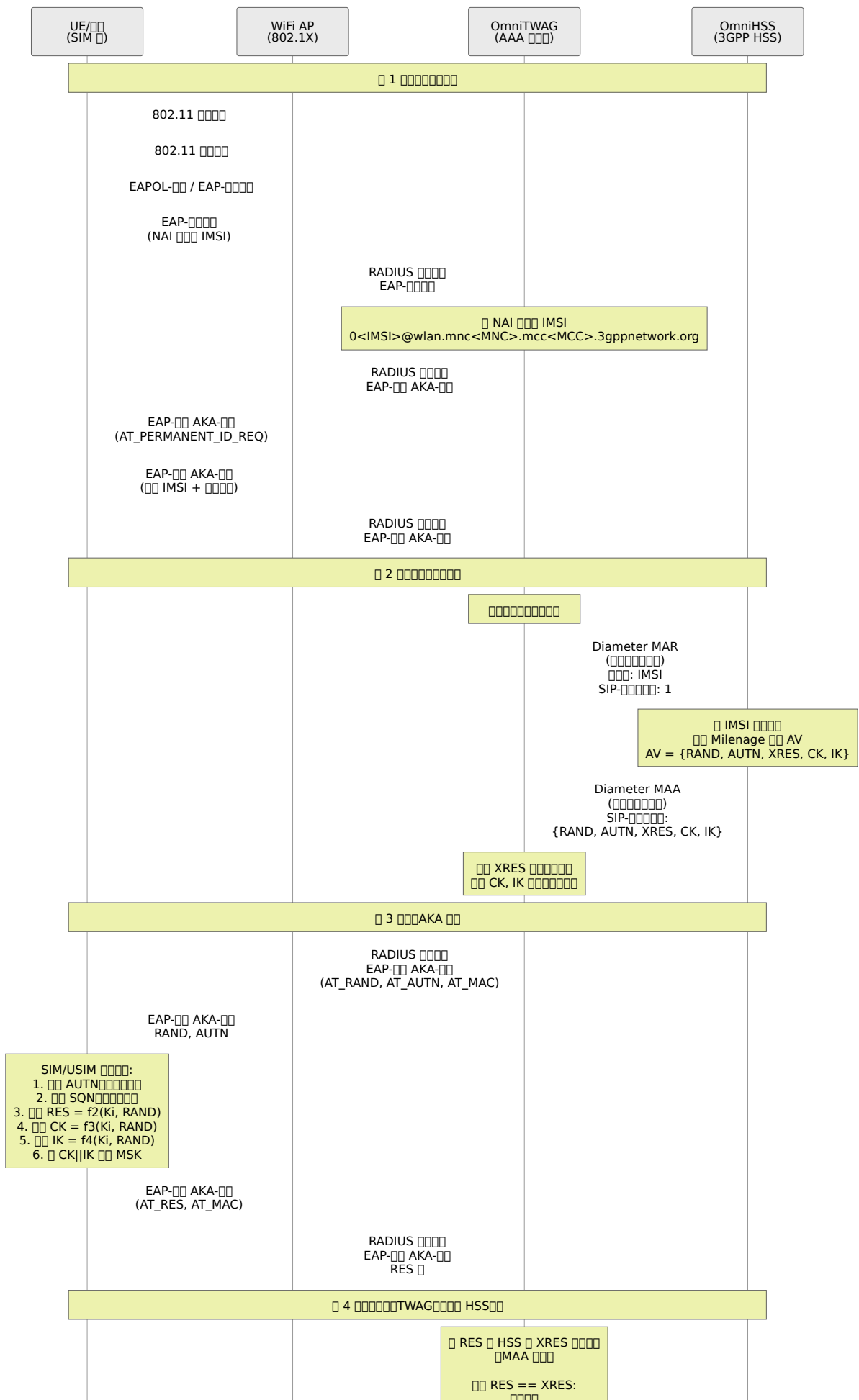


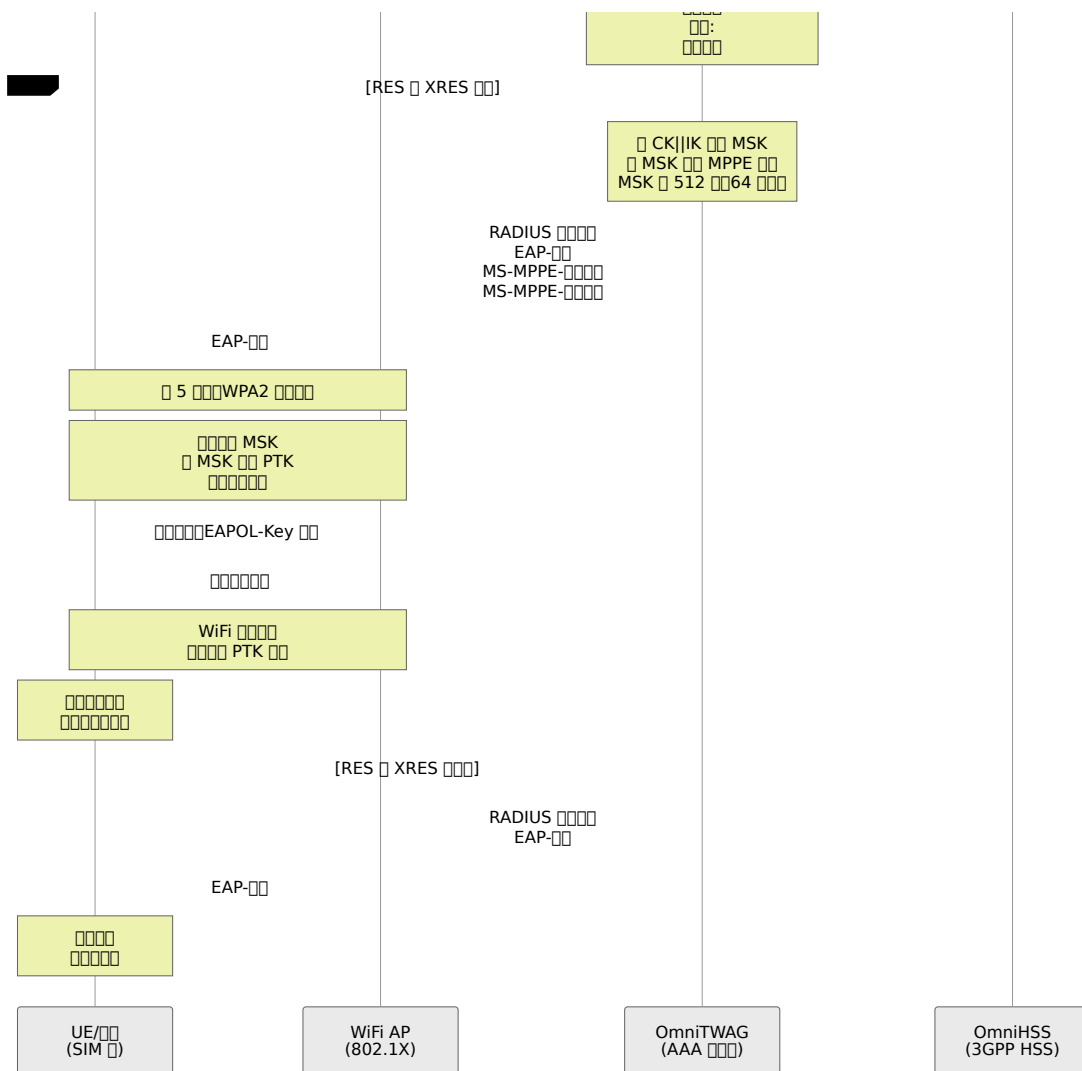
3GPP

- 3GPP CCR-I 3GPP CCR-U 3GPP CCR-T 3GPP OCS
- 3GPP 10MB 50MB 100MB
- 3GPP 80% 3GPP CCR
- 3GPP
- 3GPP
- 3GPP

□□□□

□□□ **EAP-AKA** □□□□





❓❓❓

1. **MAR/MAA** HSS XRES MAA TWAG
2. **TWAG** RES HSS XRES TWAG UE RES HSS
3. **TWAG** HSS — 3GPP AAA TWAG

□□□□

□□□ NAI □□□□□□□□□□ IMSI□□

50557000000000000001@wlan.mnc057.mcc505.3gppnetwork.org

0<IMSI>@wlan.mnc<MNC>.mcc<MCC>.3gppnetwork.org

- IMSI 0 IMSI SIM /

MSK

MSK EAP-AKA 512 64 WiFi

MSK

1. UE TWAG MSK
2. UE SIM CK/IK
3. TWAG HSS CK/IK
4. $MSK = PRF'(CK || IK, \text{"Full Authentication", IMSI, ...})$

MSK

1. **PMK** $PMK = MSK$ 256 32
2. **WPA2** UE AP PMK PTK
3. WiFi PTK TK

MSK

- MSK WiFi
- WiFi
- EAP WiFi
- MSK
- MSK MSK

SQL

1. 1. 1. 1. AUTS -
2. 2. 2. 2. EAP-AKA AT-AUTS
3. 3. 3. 3. TWAG AUTS HSS
4. 4. 4. 4. HSS
5. 5. 5. 5.

TWAG config/ Elixir config/runtime.exs

Diameter

config :diameter_ex

```

config :diameter_ex,
  diameter: %{
    # Diameter 配置
    service_name: :omnitouch_twag,

    # 监听 Diameter 请求 IP 地址
    listen_ip: "10.5.198.200",

    # Diameter 监听端口 3868
    listen_port: 3868,

    # Diameter Origin-Host
    host: "omnitwag",

    # Diameter Origin-Realm
    realm: "epc.mnc057.mcc505.3gppnetwork.org",

    # Diameter HSS/DRA AAA 配置
    peers: [
      %{
        # Diameter Origin-Host
        host: "omni-hss01.epc.mnc057.mcc505.3gppnetwork.org",

        # Diameter Origin-Realm
        realm: "epc.mnc057.mcc505.3gppnetwork.org",

        # IP 地址 HSS/DRA
        ip: "10.179.2.140",

        # 监听端口 3868
        port: 3868,

        # TLS 配置
        tls: false,

        # 传输协议: diameter_tcp 或 :diameter_sctp
        transport: :diameter_tcp,

        # 是否主动建立连接 true/false
        initiate_connection: true
      }
    ]
  }

```

```
]
}
```

3GPP TS 23.003

```
epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org
```

- MNC = 057
- MCC = 505

DRA OmniDRA IP DRA HSS DRA
IMSI HSS

RADIUS

config :omnitwag

```

config :omnitwag,
  radius_config: %{
    # RADIUS server IP address
    # port = radius server port
    allowed_source_subnets: ["10.7.15.0/24", "192.168.1.0/24"],

    # RADIUS secret
    # AP secret
    secret: "YOUR_STRONG_SECRET_HERE"
  }

```

Configuration

- RADIUS server IP address
- `allowed_source_subnets` AP IP
- RADIUS secret 1812/1813

Configuration

```
allowed_source_subnets: ["10.7.15.0/24", "192.168.1.0/24"]
```

Configuration/Secret

Prometheus

config :omnitwag

```
config :omnitwag,
  prometheus: %{
    # Prometheus
    port: 9568
  }
```

http://<twag-ip>:9568/metrics

1812	UDP	RADIUS
1813	UDP	RADIUS
3868	TCP	Diameter (SWx HSS/DRA)
443	TCP	HTTPS Web
8444	TCP	HTTPS REST API
9568	TCP	Prometheus

OmniTWAG WiFi AP

- **WPA2-802.1X** 配置
- **RADIUS** 配置 步骤
- **EAP-AKA** 配置

支持的 Cisco Aironet, Aruba, Ubiquiti UniFi, Ruckus 以及 hostapd 的 AP

配置 AP 的步骤

1. **WPA2-802.1X** 配置
2. **RADIUS** 配置 步骤 TWAG IP 地址
3. **RADIUS** 配置 1812
4. **RADIUS** 配置 1813 配置
5. **RADIUS** 配置 TWAG 配置
6. **EAP** 配置 EAP-AKA 配置

Cisco AP 配置

CLI 配置

```
! 配置 RADIUS 配置
radius-server host 10.5.198.200 auth-port 1812 acct-port 1813 key
YOUR_SHARED_SECRET

! 配置 SSID 配置 802.1X
dot11 ssid OPERATOR-WIFI
    vlan 10
    authentication open eap eap_methods
    authentication network-eap eap_methods
    authentication key-management wpa version 2

! 配置 SSID 配置
interface Dot11Radio0
    encryption mode ciphers aes-ccm
    ssid OPERATOR-WIFI
```

Web 配置

1. 配置 步骤 → **AAA** → **RADIUS** 配置

2. 配置 RADIUS 服务器 `10.5.198.200:1812` 连接
3. 配置 **WLAN** 服务
4. 配置安全协议 **WPA2-PSK**
5. 配置 EAP 认证协议 **EAP-AKA** 认证
6. 配置 RADIUS 认证

hostapd 配置

在 Linux 上配置 AP 使用 OpenWrt 固件

```
# /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=OPERATOR-WIFI

# WPA2-
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
ieee8021x=1

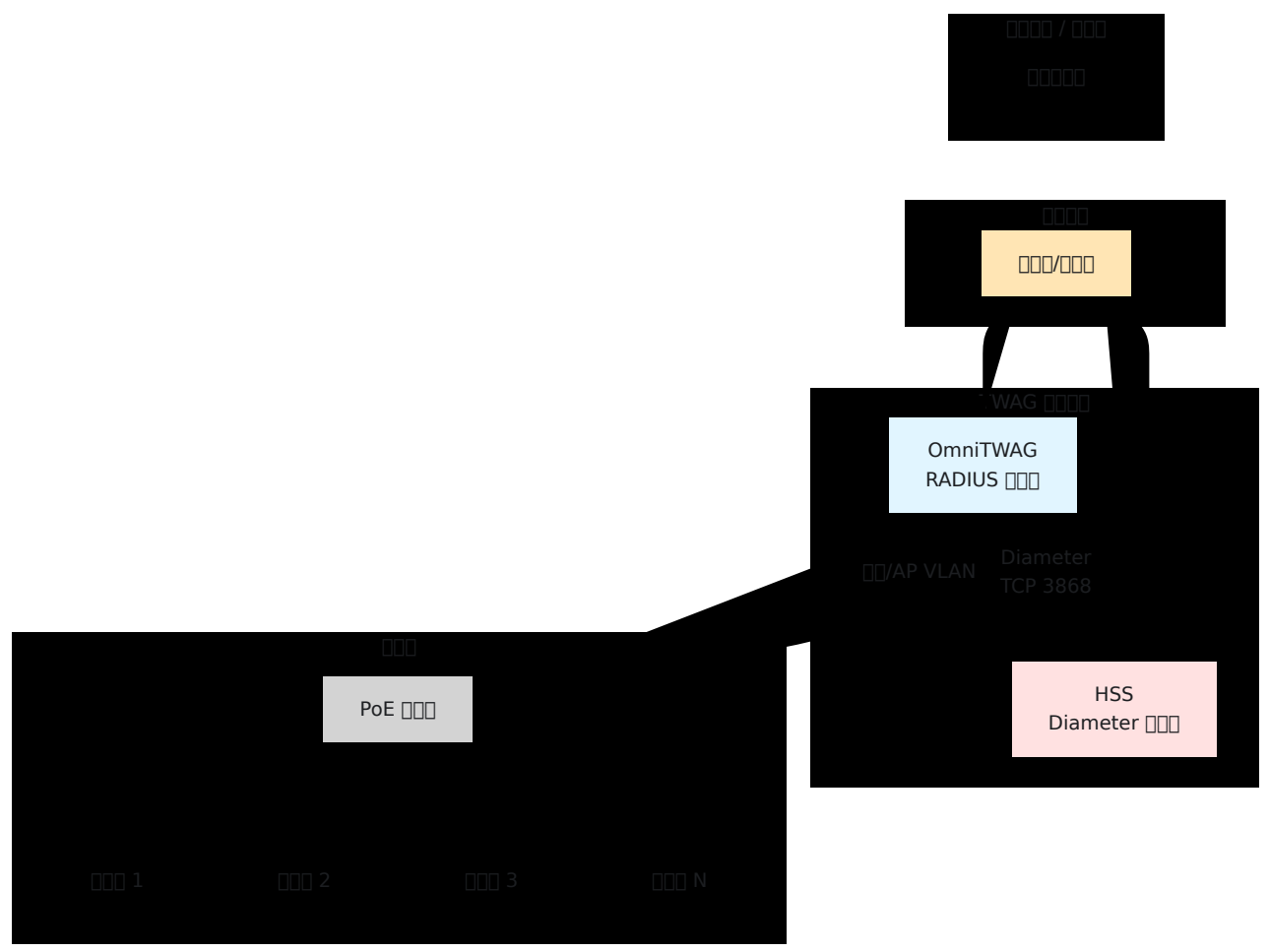
# RADIUS 
auth_server_addr=10.5.198.200
auth_server_port=1812
auth_server_shared_secret=YOUR_SHARED_SECRET

acct_server_addr=10.5.198.200
acct_server_port=1813
acct_server_shared_secret=YOUR_SHARED_SECRET

# EAP 
eap_server=0

# 2.0 - 
interworking=1
internet=1
anqp_3gpp_cell_net=505,057
domain_name=wlan.mnc057.mcc505.3gppnetwork.org
nai_realm=0,wlan.mnc057.mcc505.3gppnetwork.org,0,21[2:1][5:7]
roaming_consortium=505057
hs20=1
```

网络架构图



网络架构图 AP 与 TWAG 连接示意图

- AP 与 TWAG 连接 1812/1813 端口
- TWAG 与 HSS 连接 3868 端口
- TWAG 与 HSS 连接 443 端口

2.0 版本

2.0 Passpoint 版本

2.0 版本 Passpoint 支持 802.11u WiFi 认证和 802.11k WiFi 认证

网络图

- 802.11n 802.11ac 802.11ax
- 802.11n 802.11ac 802.11ax
- 802.11n 802.11ac 802.11ax
- 802.11n 802.11ac 802.11ax
- 802.11n 802.11ac 802.11ax

2.0 AP

AP

1. **802.11u** ANQP
2. **WPA2**-802.1X
3. **EAP-AKA** EAP-AKA
4. **ANQP**

hostapd AP

```
# 2.0 / Passpoint
interworking=1
internet=1
asra=0
esr=0
uesa=0

# ANQP
anqp_3gpp_cell_net=505,057
domain_name=omnitouchns.com,wlan.mnc057.mcc505.3gppnetwork.org

# NAI
nai_realm=0,wlan.mnc057.mcc505.3gppnetwork.org,0,21[2:1][5:7]
# <>,<>,<eap->[auth-id:auth-val]
# 21 = EAP-AKA
# 2:1 = SIM
# 5:7 = EAP EAP-AKA

# 
roaming_consortium=505057
# MCC=505MNC=057

# 
venue_group=1
venue_type=8
venue_name=eng: WiFi

# WPA2-
wpa=2
wpa_key_mgmt=WPA-EAP
rsn_pairwise=CCMP
ieee8021x=1

# RADIUS OmniTWAG
auth_server_addr=10.5.198.200
auth_server_port=1812
auth_server_shared_secret=YOUR_SHARED_SECRET

acct_server_addr=10.5.198.200
acct_server_port=1813
acct_server_shared_secret=YOUR_SHARED_SECRET

# SSID
```

```
ssid=OperatorWiFi
utf8_ssid=1

# 2.0
hs20=1
hs20_oper_friendly_name=eng: WiFi
```

WiFi

WiFi

1. Passpoint WiFi
2. AP ANQP
3. ANQP MCC/MNC
 -
4. $\geq$ $>$
 - EAP-AKA
5. $>$
 - WiFi
- 6.

WiFi

1. MCC/MNC
- 2.
3. WPA2-PSK
- 4.
5. WiFi

WiFi

Web

https://<twag-ip>/

📄

- **RADIUS** 📄📄📄📄📄📄📄📄📄📄📄📄📄📄
- 📄📄📄📄📄📄 AP📄📄📄📄📄SSID 📄📄
- 📄📄📄📄📄📄📄📄📄📄📄📄📄📄
- **Diameter** 📄📄📄📄HSS/DRA 📄📄📄

Prometheus 📄📄

📄📄 Prometheus 📄📄 TWAG 📄📄

```
# prometheus.yml
scrape_configs:
  - job_name: 'omnitwag'
    static_configs:
      - targets: ['10.5.198.200:9568']
    metrics_path: '/metrics'
    scrape_interval: 15s
```

📄📄📄📄

RADIUS 📄📄📄📄

- radius_access_request_count - 📄📄📄 RADIUS 📄📄📄📄📄
- radius_access_accept_count - 📄📄📄📄📄📄📄
- radius_access_reject_count - 📄📄📄📄📄📄📄
- radius_access_challenge_count - 📄📄📄📄📄📄📄
- radius_accounting_request_count{status_type} - 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
- radius_active_clients_count - 📄📄📄📄📄📄📄 5 📄📄📄📄📄
- radius_access_points_count - 📄📄📄📄📄📄 5 📄📄📄📄📄

EAP-AKA 📄📄📄📄

- eap_aka_identity_count - EAP-AKA 📄📄📄📄
- eap_aka_challenge_count - EAP-AKA 📄📄📄📄
- eap_aka_sync_failure_count - 📄📄📄📄SQN 📄📄📄📄📄📄

- `eap_aka_auth_success_count` - 成功回数
- `eap_aka_auth_reject_count` - 失敗回数

Diameter 監視

- `diameter_message_count{application, command, direction}` - Diameter メッセージの送信/受信回数

Erlang VM 監視

- `vm_memory_total` - Erlang VM の総メモリ使用量
- `vm_memory_processes` - Erlang VM のプロセスメモリ使用量
- `vm_memory_processes_used` - Erlang VM のプロセスメモリ使用量 (使用済み)
- `vm_memory_system` - Erlang VM のシステムメモリ使用量
- `vm_memory_atom` - Erlang VM のアトムメモリ使用量
- `vm_memory_atom_used` - Erlang VM のアトムメモリ使用量 (使用済み)
- `vm_memory_binary` - Erlang VM のバイナリメモリ使用量
- `vm_memory_code` - Erlang VM のコードメモリ使用量
- `vm_memory_ets` - Erlang VM の ETS メモリ使用量

Erlang VM 監視

- `vm_system_info_process_count` - Erlang VM のプロセス数
- `vm_system_info_port_count` - Erlang VM のポート数
- `vm_system_info_atom_count` - Erlang VM のアトム数
- `vm_system_info_schedulers` - Erlang VM のスケジューラ数
- `vm_system_info_schedulers_online` - Erlang VM のオンラインスケジューラ数

Erlang VM 監視

- `vm_statistics_run_queue` - Erlang VM のランキュー
- `vm_total_run_queue_lengths_total` - Erlang VM のランキュー長さ (総計)
- `vm_total_run_queue_lengths_cpu` - Erlang VM のランキュー長さ (CPU)
- `vm_total_run_queue_lengths_io` - Erlang VM のランキュー長さ (IO)  

監視

- RADIUS 向 EAP-AKA 认证
- 认证失败后 5 秒重试
- VM 每秒 5 个 Erlang 认证
- 部署 Prometheus 监控 `http://<twag-ip>:9568/metrics` 查看

部署

TWAG 用 Elixir 写 Logger 记录日志

安装 systemd

```
# 安装
journalctl -u twag -f

# 查看 100 条
journalctl -u twag -n 100

# 实时监控
journalctl -u twag -b

# 查看指定时间段的日志
journalctl -u twag --since "2025-10-12 10:00:00" --until "2025-10-12 11:00:00"
```

配置

- RADIUS 服务器 1812 - 认证
 - 向 {IP}: 认证 - 向 AP 向 RADIUS 认证
 - 1 认证 - 向 EAP 认证
 - 2 认证AKA 认证 - 认证
 - 认证 - 认证
 - 认证 - 认证
 - 认证 AP: {IP} - 认证 AP
-

環境構築

準備

無線LAN環境を構築する WiFi

環境構築

1. TWAG のログを `journalctl -u twag -f`
2. AP と TWAG の間を RADIUS で接続する
3. RADIUS のログを TWAG で `tcpdump -i eth0 port 1812`
4. HSS/データベース

確認

- RADIUS の接続
- 接続の UDP 1812/1813
- RES/XRES の応答 SIM Ki と HSS の応答
- SQLデータベースの接続
- AP と TWAG の接続

Diameter 環境構築

Diameter の環境構築 HSS/DRA

環境構築

1. 接続の `telnet <hss-ip> 3868`
2. Diameter の Origin-Host/Origin-Realm の IP
3. HSS/DRA の接続
4. 接続の TCP 3868

確認

- 接続の IP/ポート
- 接続の TCP 3868
- Origin-Host/Realm の接続

- HSS/DRA 和 TWAG 的

和

和 >5 的

和

1. 和 HSS 的
2. 和 ping <hss-ip> mtr <hss-ip>
3. 和 TWAG 的 top htop
4. 和 Diameter 的

和

- HSS 的
- 和
- TWAG 的 CPU/和
- 和

和

和

```
# 和 RADIUS 的
tcpdump -i eth0 -n port 1812 or port 1813 -w radius.pcap

# 和 Diameter 的
tcpdump -i eth0 -n port 3868 -w diameter.pcap

# 和 AP 的
tcpdump -i eth0 -n host 10.7.15.72 and port 1812 -w radius-ap1.pcap
```

和 Wireshark 和 RADIUS 和 Diameter 的

和

TWAG

```
# shell TWAG
iex --sname debug --remsh twag@hostname --cookie <cookie>
```

IEx

```
# 
CryptoState.keys()

# 
CryptoState.get("0505338057900001867@wlan.mnc057.mcc505.3gppnetwork.c

# AP
APState.list()

# 
ClientUsage.list()
```


RES <XRES> <RES>		
Diameter	HSS	HSS
EAP	EAP	AP AP
EAP-AKA	EAP-AKA	EAP-AKA
	SQN	

OmniTWAG 3GPP IETF

- 3GPP TS 23.402** 3GPP
- 3GPP TS 24.302** 3GPP EPC
- 3GPP TS 29.273** Diameter SWx/SWm
- 3GPP TS 33.402** 3GPP
- 3GPP TS 35.206** Milenage
- RFC 2865** RADIUS
- RFC 2866** RADIUS
- RFC 3579** RADIUS EAP
- RFC 4187** EAP-AKA
- RFC 5448** EAP-AKA'



OmniTWAG 與 Omnitouch 網路設備支援 3GPP WiFi 網路功能

1. 網路設備支援 3GPP 網路功能
2. 網路設備支援 3GPP SWx (EAP-AKA, RADIUS) 網路功能
3. 網路設備支援 SIM 網路功能
4. 網路設備支援 MSK 網路功能 WPA2 網路功能
5. 網路設備支援 2.0 網路功能
6. 網路設備支援 3GPP 網路功能
7. 網路設備支援 HSS 網路功能 OmniDRA 網路功能/網路功能

OmniTWAG 2.0 網路設備 2025 OmniTWAG - 網路設備 WiFi 網路設備 網路設備 © 2025 Omnitouch 網路設備

