

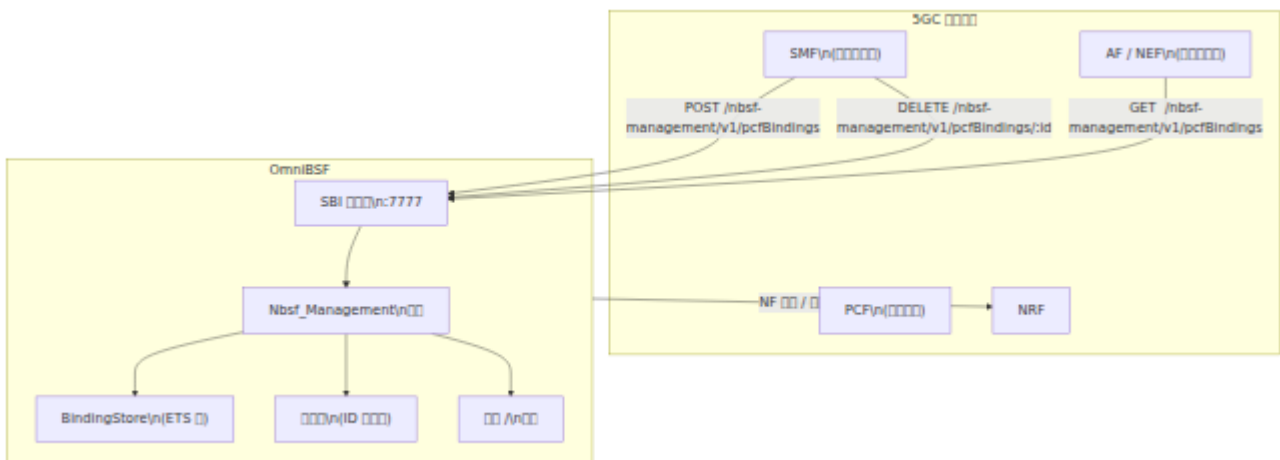
# OmniBSF 架构

## 目录

1. 概述
2. 3GPP 标准
3. SBI 接口
4. 部署
5. 配置
6. 测试
7. 安全
8. 附录

## 简介

OmniBSF 实现 3GPP TS 29.521 定义的 BSF (BSF) 功能。BSF 是 PCF 的扩展，用于支持 UE IP 地址 IPv4 和 IPv6 的分配。PDU 会话 PCF 通过 NF 接口与 SMF、AF、NEF 交互。此外，NRF 也支持 PCF。



## ETSI

ETSI ETS 3GPP TS 29.500

ETSI ID	Cardinality	Field Name	Field Type
bsf_bindings	:set	binding_id (string)	string
bsf_ipv4_index	:set	IPv4 address (string)	binding_id
bsf_ipv6_index	:set	IPv6 address (string)	binding_id
bsf_supri_index	:set	SUPI (string)	binding_id

## 3GPP

Field Name	Reference
BSF NF ID	3GPP TS 23.501 § 6.2.15
Nbsf_Management ID	3GPP TS 29.521
PCF ID	3GPP TS 29.521 § 4.2.2
PCF ID	3GPP TS 29.521 § 4.2.3
PCF ID	3GPP TS 29.521 § 4.2.4
SBI ID	3GPP TS 29.500
NRF ID / NF ID	3GPP TS 29.510
PcfBinding ID	3GPP TS 29.521 § 6.1


PCF BSF PDU PCF

## SBI

/nbsf-management/v1

Method	Path	Description	Status	Response
POST	/pcfBindings	PCF UE IP ID Location	201	400 snssai dnn PCF
GET	/pcfBindings	PCF PcfBinding 204	200 OK 204	400
DELETE	/pcfBindings/{bindingId}	ID PCF	204	404

# GET 消息

項目	形式	長さ	説明
ipv4Addr	文字列	1文字列	UE IPv4 文字列
ipv6Prefix	文字列	2	UE IPv6 文字列
dnn	文字列	3文字列	文字列 snssai 文字列
snssai	JSON 文字列	3文字列	S-NSSAI文字列 sst 文字列 sd 

TS 29.521 の open5gs bsf-sm.c には ipv4Addr 文字列  
ipv6Prefix 文字列 DNN+S-NSSAI 文字列

# POST 消息 — 消息

参数	类型	描述
snsai	字符串	S-NSSAI由sst 和 sd 组成
dnn	字符串	
pcfFqdn	字符串	PCF FQDN由 pcfIpEndpoints 组成
pcfIpEndpoints	字符串	PCF IP 由 pcfFqdn 组成
ipv4Addr	字符串	UE IPv4 地址
ipv6Prefix	字符串	UE IPv6 地址
supi	字符串	
gpsi	字符串	
ipv4FrameRouteList	字符串	IPv4 帧路由列表
ipv6FrameRouteList	字符串	IPv6 帧路由列表
suppFeat	字符串	

## 消息

消息体包含 config/runtime.exs

```
config :omnibsf,  
  sbi_scheme: "http",  
  sbi_addr:   "127.0.0.15",  
  sbi_port:   7777,  
  nrf_uri:    "http://127.0.0.10:7777",  
  mcc:        "999",  
  mnc:        "70",  
  heartbeat_interval: 10_000
```

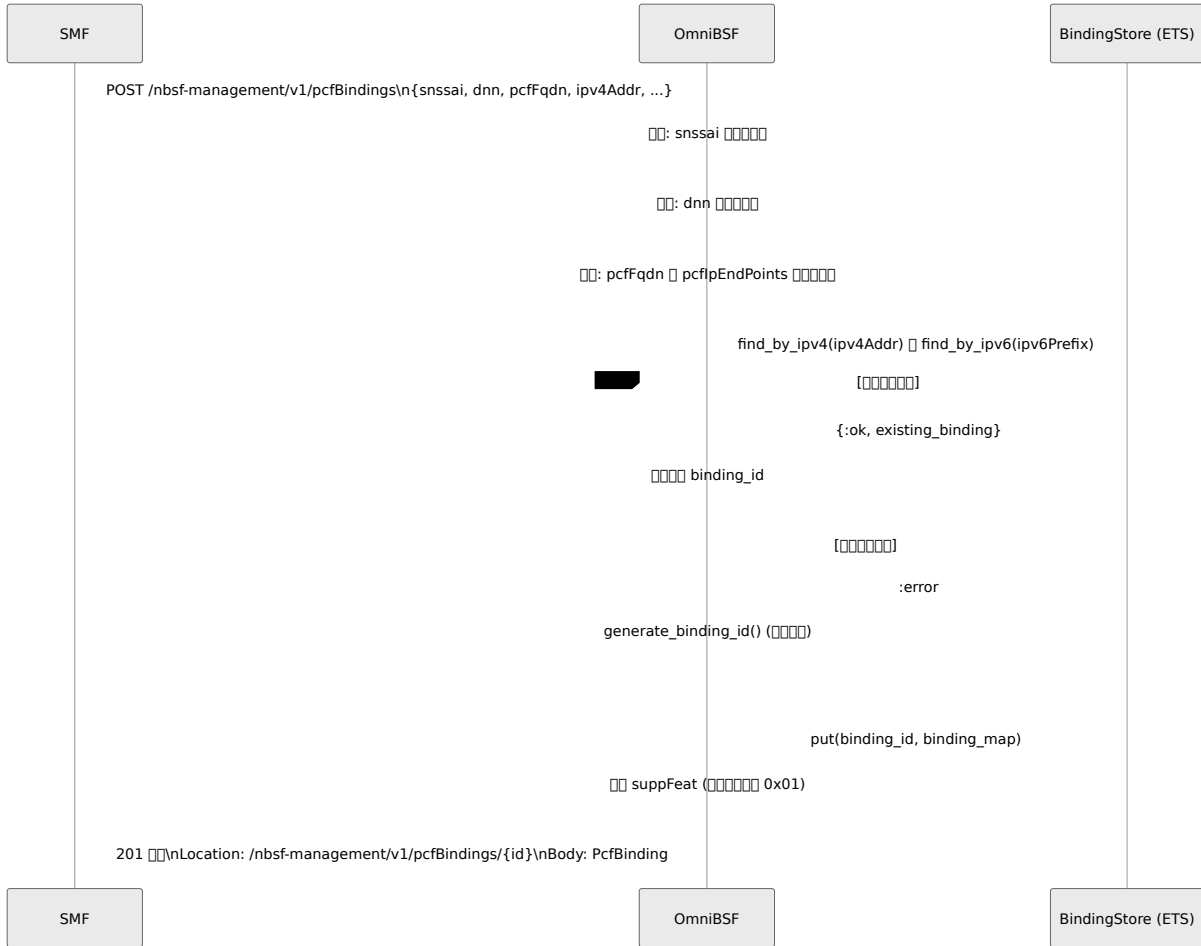
□□□

□□	□□□	□□	□□
sbi_scheme	"http"	□□ □	SBI □□□□□□□□ □□ TLS □□□ OmniBSF □□□□□ □□□□ "https" □
sbi_addr	"127.0.0.15"	□□ □	SBI HTTP □□□□□ □ IP □□□
sbi_port	7777	□□	SBI HTTP □□□□□ □ TCP □□□
nrf_uri	"http://127.0.0.10:7777"	□□ □	□□ NF □□□□□□ NRF □□ URI□
mcc	"999"	□□ □	□□□□□□□□□□□□ □□ NRF □ NF □□ □□□□□□
mnc	"70"	□□ □	□□□□□□□□□□□□ □□ NRF □ NF □□ □□□□□□
heartbeat_interval	10_000	□□ □□ □□	NRF □□ □NFUpdate□□□ □□□□□□□□ 10_000 □10 □□□ □□□□□□□□ NRF TTL □□□□□□□□

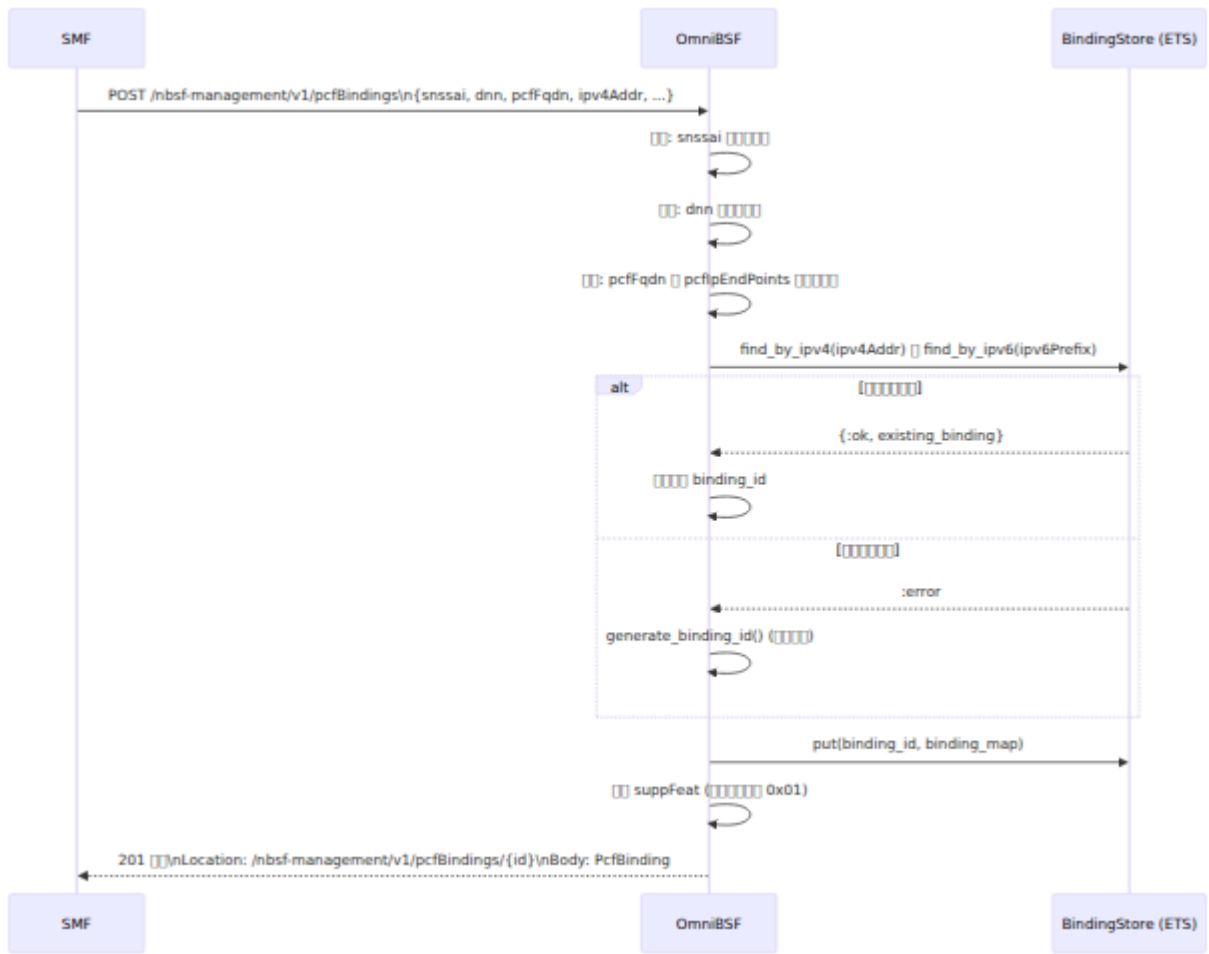
---



## PCF 注册 (POST)







# PCF 创建 (GET)


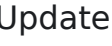





# PCF (DELETE)



## NRF

OmniBSF  Omni5gEx  BSF NF  NRF 

heartbeat\_interval  NFUpdate  NRF  NRF 

OmniBSF 

---

□□□□

□□□□

□□	□□	□□	□□
<code>[ :omnibsf, :pcf_binding, :register]</code>	count	dnn	□□□□□□□□ □□
<code>[ :omnibsf, :pcf_binding, :discover]</code>	count	result (found/not_found)	□□□□□□□□
<code>[ :omnibsf, :pcf_binding, :deregister]</code>	count	binding_id	□□□□□□□□
<code>[ :omnibsf, :binding, :register]</code>	count	—	□□□□□□
<code>[ :omnibsf, :binding, :discover]</code>	count	result	□□□□□□
<code>[ :omnibsf, :binding, :deregister]</code>	count	—	□□□□□□
<code>[ :omnibsf, :bindings, :active]</code>	count	—	□□□□□□□□ □□
<code>[ :omni5g, :nrf, :registration]</code>	status	nf_type	NRF □□□□ □1=□□□□0= □□□□

# Prometheus

## BSF

Label	Type	Label	Description
<code>omni_bsf.pcf_binding.register.count</code>	counter	<code>dnn</code>	PCF Count
<code>omni_bsf.pcf_binding.discover.count</code>	counter	<code>result</code>	PCF Count
<code>omni_bsf.pcf_binding.deregister.count</code>	counter	<code>binding_id</code>	PCF Count
<code>omni_bsf.binding_registers.total</code>	counter	--	Count Count
<code>omni_bsf.binding_discovers.total</code>	counter	<code>result</code>	Count Count
<code>omni_bsf.binding_deregisters.total</code>	counter	--	Count Count
<code>omni_bsf.active_bindings.count</code>	gauge	--	Count Count
<code>omni_bsf.nrf.registration.status</code>	gauge	<code>nf_type</code>	NRF Count 1= Count 0= Count

## BEAM VM

名前	タイプ	説明
<code>beam.memory.total</code>	gauge	BEAM の総メモリ
<code>beam.memory.processes</code>	gauge	Erlang のプロセス数
<code>beam.memory.processes_used</code>	gauge	プロセスに使用されているメモリ
<code>beam.memory.system</code>	gauge	システムメモリ
<code>beam.memory.atom</code>	gauge	アトム数
<code>beam.memory.atom_used</code>	gauge	アトムに使用されているメモリ
<code>beam.memory.binary</code>	gauge	バイナリ
<code>beam.memory.code</code>	gauge	コード
<code>beam.memory.ets</code>	gauge	ETS のメモリ
<code>beam.processes.count</code>	gauge	Erlang のプロセス数
<code>beam.ports.count</code>	gauge	Erlang のポート数
<code>beam.atom.count</code>	gauge	アトム数
<code>beam.vm.uptime</code>	gauge	VM の稼働時間

□□□□

□□	□□	□□
info	PCF binding registered: id=&lt;N> ipv4= &lt;addr> ...	□□□□
info	PCF binding deregistered: id=&lt;N>	□□□□
debug	PCF binding discovered: id=&lt;N>	□□□□
debug	PCF binding not found for query: ...	□□□□□
warning	PCF binding not found for deregister: id=&lt;N>	□□□ ID □ DELETE

---





□□□□□□

OmniBSF □□□□□□□□ 0x01 □BindingUpdate□□□□ supFeat □□□□□□□□□□□□□□□□  
□□□□□□□□ TS 29.521 □ 6.1.6 □□□□□□