

# 5G SMF Overview


- Feature Summary and Revision History, on page 1
- Product Description, on page 2
- Converged core architecture and ULB integration, on page 3
- Converged Core Overview, on page 11
- Use Cases and Features, on page 13
- Deployment Architecture and Interfaces, on page 21
- Life Cycle of Data Packet, on page 25
- Session Affinity, on page 30
- License Information, on page 31
- Standards Compliance, on page 31
- ULB-based deployment model, on page 31


# Feature Summary and Revision History

## Summary Data

***Table 1: Summary Data***

| | |
|---|---|
| Applicable Product(s) or Functional Area | SMF |
| Applicable Platform(s) | SMI |
| Feature Default Setting | Not Applicable |
| Related Changes in this Release | Not Applicable |
| Related Documentation | Not Applicable |

# Revision History

*Table 2: Revision History*

| Revision Details | Release |
|---|---|
| The converged core support for combined SMF + cnSGWc is added in this release. | 2021.01.0 |
| First introduced. | Pre-2020.02.0 |

# Product Description

The Cisco Session Management Function (SMF) is one of the Control Plane Network Functions (NF) of the 5G core network (5GC). The SMF is responsible for the session management with the supported individual functions on a per-session basis.

A single instance of SMF can support some or all the functionality of the SMF. As specified in *3GPP TS 23.501*, the SMF supports the following functionality:

- Handles session management. For example, session establishment, modification and release, including the tunnel between the User Plane Function (UPF) and the access network (AN).
- Handles user element (UE) IP address allocation and management, which includes an optional authorization.
- Performs Dynamic Host Configuration Protocol for IPv4 (DHCPv4) and DHCPv6 functions, both as server and client.
- Performs Address Resolution Protocol (ARP) proxying and IPv6 Neighbor Solicitation Proxying functionality for the Ethernet PDUs. The SMF responds to the ARP and the IPv6 Neighbor Solicitation Request by providing the MAC address. This address corresponds to the IP address that exists in the request.
- Selects and controls the UPF for the Ethernet PDU sessions. The UP function includes controlling the UPF to proxy ARP or IPv6 Neighbor Discovery, and forwarding all ARP or IPv6 Neighbor Solicitation traffic to the SMF.
- Configures Traffic Steering at the UPF to route traffic to the corresponding Data Network (DN).
- Terminates interfaces toward the Policy Control Function (PCF).
- Handles the Lawful Intercept (LI) for Session Manager (SM) events and interface to the LI system.
- Controls and synchronizes the charging data collection at the UPF.
- Terminates the SM parts of Non-Access-Stratum (NAS) messages.
- Routes packets and ensures the delivery of information through the Downlink Data Notification (DDN).
- Initiates the AN-specific SM information that is sent through the Access and Mobility Management Function (AMF) to AN over the N2 interface.
- Determines the session and service continuity (SSC) mode of a session.

- Provides the following roaming functionality:
  - Manages the local enforcement to apply Quality of Service (QoS) SLAs (VPLMN).
  - Collects charging data and supports the charging interfaces.
  - Supports communication with the external DN. The communication is for the transport of signaling for the PDU session authorization or authentication by an external DN.

# Converged core architecture and ULB integration

The Converged Core architecture integrates the Unified Load Balancer (ULB) with the SMF Service Pods to support direct routing of GTPC and PFCP signaling traffic. By eliminating the need for dedicated GTPC and PFCP endpoint pods, the architecture reduces signaling latency, improves scalability, and enables efficient distribution of signaling traffic across Converged Core deployments.

The ULB distributes ingress signaling messages to the appropriate SMF Service Pod based on the configured load-balancing policy and routes egress signaling messages to the corresponding external network peer. The ULB maintains session affinity to ensure that all signaling messages associated with an active subscriber session are consistently processed by the same SMF Service Pod.

The ULB provides a common routing framework for GTPC and PFCP signaling, enabling direct communication between the ULB and the SMF Service Pods. This architecture supports efficient session processing, consistent message routing, and protocol-specific peer management for supported signaling procedures.

The Converged Core ULB integration provides the following capabilities:

- GTPC ingress and egress signaling message routing
- PFCP ingress and egress signaling message routing
- Session affinity for subscriber sessions
- GTP and PFCP peer management

## ULB integration in converged core

***Table 3: Feature history***

| Feature name | Release | Feature description |
|---|---|---|
| ULB integration for PFCP traffic | 2026.03.0 | This release introduces ULB-based ingress and egress routing for PFCP traffic, enabling scalable traffic distribution. The enhancementoptimizes PFCP control-plane traffic handling and supportsefficient scaling in cloud-native deployments. |

| Feature name | Release | Feature description |
|---|---|---|
| ULB integration for GTPC traffic | 2026.03.0 | This release introduces ULB-based ingress and egress routing for GTPC traffic, enabling scalable traffic distribution and session-aware message steering. The enhancement optimizes control-plane traffic handling and supports efficient scaling in cloud-native deployments. |

### ULB integration design

The ULB integration uses the existing Unified Load Balancer infrastructure to distribute protocol signaling traffic across the available SMF service instances. The ULB selects an SMF service instance for new session establishment and maintains routing information for subsequent protocol messages associated with the session.

For new sessions, the ULB distributes the initial signaling request across the available Service Pods. The selected Service Pod establishes ownership of the session and maintains the session information for subsequent signaling procedures.

For existing sessions, the ULB identifies the Service Pod associated with the session and forwards all subsequent signaling messages to the same Service Pod, ensuring session affinity throughout the session lifecycle.

After processing the signaling request, the Service Pod generates the appropriate response and sends it directly through the ULB to the peer network element.

This routing mechanism is used for both GTPC and PFCP signaling traffic.

### Session routing and affinity

The ULB maintains session affinity to ensure that all protocol messages belonging to the same subscriber session are forwarded to the same SMF service instance.

For a new subscriber session, the ULB selects an SMF service instance based on the configured load-balancing algorithm. After the session is established, the routing information is maintained so that subsequent protocol messages associated with the session continue to be processed by the selected service instance.

Session affinity applies to both GTPC and PFCP session-level signaling and remains valid throughout the lifetime of the subscriber session.

### Call flow summary

The ULB integration supports the complete lifecycle of GTPC and PFCP signaling procedures, including session establishment, session modification, session deletion, and session release.

# ULB integration of GTPC ingress/egress traffic

The ULB routes GTPC ingress and egress signaling messages to the appropriate SMF Service Pod while maintaining session affinity. GTPC signaling is processed directly by the SMF Service Pod without requiring dedicated GTPC endpoint pods. The feature supports both 4G and 5G signaling procedures.

### GTPC message routing

The ULB receives GTPC ingress signaling messages from external network elements and forwards them to an SMF Service Pod based on the configured load-balancing policy.

For a new session, the ULB distributes the Create Session Request to an SMF Service Pod. The selected Service Pod processes the request and establishes the routing information required for subsequent GTPC messages.

After the session is established, the ULB maintains session affinity and forwards all subsequent session-level messages to the same SMF Service Pod. Egress GTPC messages generated by the SMF Service Pod are routed through the ULB to the appropriate external peer.

The ULB supports routing of both ingress and egress GTPC signaling while maintaining consistent session processing.

### GTP peer management

The Unified Load Balancer (ULB) supports GTP peer management in a Converged Core deployment by forwarding peer management messages to the GTPC pod. The GTPC pod, deployed in non-host networking mode with an Active-Active deployment, processes peer management procedures using the existing GTP peer management framework.

The ULB forwards GTP peer management messages, such as Echo Request and Echo Response messages, to the appropriate GTPC pod for processing. Session-level GTPC signaling is routed independently based on session affinity to the appropriate SMF Service Pod, while peer management procedures continue to be processed by the GTPC pod.

## ULB integration of PFCP ingress/egress traffic

The ULB routes PFCP ingress and egress signaling messages to the appropriate SMF Service Pod while maintaining session affinity. PFCP session-level signaling is processed directly by the Service Pod, while node-level peer management continues to use the existing PFCP protocol pod.

### PFCP message routing

The ULB receives PFCP ingress signaling messages from external network elements and forwards them to an SMF Service Pod based on the configured load-balancing policy.

For a new PFCP session or association, the ULB distributes the initial PFCP request to an SMF Service Pod. The selected Service Pod processes the request and establishes the routing information required for subsequent PFCP messages.

After the session or association is established, the ULB maintains session affinity and forwards all subsequent session-level messages to the same SMF Service Pod. Egress PFCP messages generated by the SMF Service Pod are routed through the ULB to the appropriate external peer.

The ULB supports routing of both ingress and egress PFCP signaling while maintaining consistent session processing.

### PFCP peer management

In a Converged Core deployment, the Unified Load Balancer (ULB) differentiates between PFCP node-level and session-level signaling and routes messages to the appropriate processing component.

The ULB eBPF forwards **PFCP association (node-level) messages**, such as Association Setup, Association Update, Association Release, and Heartbeat Request/Response messages, to the **PFCP protocol pod**. The protocol pod performs message encoding and decoding, while the existing node management logic processes these procedures.

The ULB forwards **PFCP session-level messages** to the **SMF Service Pod**. The Service Pod processes PFCP session establishment, modification, and deletion procedures and maintains session affinity so that all messages belonging to an active PFCP session are consistently processed by the same Service Pod.

This routing mechanism enables separate handling of node-level and session-level PFCP signaling while maintaining consistent message processing and session affinity across the Converged Core deployment.

# ULB integration for REST/SBI traffic

| Feature | Release | Description |
|---|---|---|
| ULB integration for REST/SBI traffic | 2026.04.0 | ULB-based routing of REST/SBI traffic uses NodeIndex information for appropriate service-instance selection and supports session-aware routing across supported interfaces. |

The ULB supports REST/SBI traffic routing in addition to GTPC and PFCP signaling traffic. The ULB uses NodeIndex information to distribute REST/SBI requests to the appropriate service instance.

The NodeIndex table is provided to App-Infra for NodeIndex-based routing. The table is updated when a node goes down, comes up, or is added to the cluster.

The application uses the NodeIndex information to determine the appropriate service instance for processing REST/SBI requests. If the local lbs-agent is unavailable, the application can obtain the required NodeIndex information from another available lbs-agent.

### REST-based node table integration

ULB provides the NodeIndex table to App-Infra through a REST-based notification mechanism. The NodeIndex table is updated whenever a node goes down, comes up, or is added to the cluster.

The lbs-agent running on each node sends the latest NodeIndex table snapshot to the co-located SMF service pods whenever the table changes.

App-Infra receives the update through its internal REST command interface, parses the payload, and stores the latest NodeIndex table in local memory for each registered LB service.

Each lbs-agent updates only the subscribed application pods running on the same node. The application uses the locally stored NodeIndex information for node-aware routing decisions.

If the co-located lbs-agent is unavailable, the application receives NodeIndex table updates from other available lbs-agents running on different nodes.

### Affinity and compatibility

The Common ID is carried in SBI and internal messages to maintain session affinity across supported interfaces.

Following a pod or node failure, ULB consistently selects the same serving pod for all interfaces associated with the same session, including GTPC, PFCP, and GTP-U.

The existing REST message-handling behavior remains unchanged when ULB is disabled. Backward compatibility is also maintained during rolling upgrades.

### Call flow summary

The ULB integration supports the following signaling and REST/SBI message flows:

- Incoming GTPC Create Session Request (CSR) with TEID=0.
- Incoming GTPC message with TEID (MBR).
- Incoming SBI SM Context Create request with IMSI.
- Incoming SBI Charging Notify message with Common ID key.

## Resiliency Support

With ULB enabled, the system monitors the health of SMF service pods, GTPC pods, protocol pods, and nodes. When a service pod or node becomes unavailable, ULB updates the forwarding information and redistributes new session traffic to the available healthy service pods. When a GTPC or protocol pod becomes unavailable, the remaining healthy pods continue handling node-level message traffic.

When a failed service pod or node recovers, ULB updates the forwarding information to account for the recovered resources. Temporary traffic imbalance is acceptable during recovery while the system progressively rebalances traffic.

### Health Monitoring

An eBPF watchdog or health-monitoring mechanism monitors the health of SMF service pods, GTPC pods, and protocol pods. The health status is reported at one-second intervals.

With ULB enabled, traffic diversion from a failed pod or node is expected to complete within four seconds. This target is subject to validation against existing baseline data.

### Service Pod Failure and Recovery

When a service pod becomes unavailable, ULB detects the pod failure and updates its forwarding information to redirect new session traffic to the remaining healthy service pods.

The following behaviour applies during service pod failure and recovery:

- **SMF service pod failure:** New session traffic is redistributed evenly among the remaining healthy service pods on the same node.
- **GTPC pod failure:** The remaining healthy GTPC pods continue handling node-level message traffic.
- **Protocol pod failure:** The remaining healthy protocol pods continue handling node-level message traffic.
- **SMF service pod recovery:** New sessions may initially be directed to the recovered service pod, resulting in temporary traffic imbalance. Traffic progressively rebalances across the available service pods.
- **GTPC/protocol pod recovery:** The recovered pod resumes handling node-level message traffic.

### Node Failure and Recovery

When a node becomes unavailable, ULB redistributes traffic across the remaining healthy nodes. When the failed node recovers, temporary service traffic imbalance is acceptable while the system converges.

The following behaviour applies:

- Traffic is redistributed across the remaining healthy nodes when a node fails.
- GTPC and protocol pods resume node-level message handling after node recovery.
- Other pod behaviour remains unchanged.
- When a node resumes or a new node is added, SMF service pods are expected to be distributed evenly across the available nodes.
- If the required service is unavailable on the selected node, ULB excludes that node from the selection.

GTPC and protocol pods are recommended to run in Active/Active mode for improved traffic management.

# Service Optimization

Flex-Fused mode optimizes GTP-C and PFCP session-message processing by moving service-owned message handling into the main SMF service processing context. This reduces internal message handoffs and repeated processing.

Only service-owned session traffic is processed in Flex-Fused mode. Heartbeat, association, and other node-management traffic continue to use their existing processing components.

### Flex-Fused mode overview

In the existing Flex mode, network message reception and SMF session processing are handled by separate contexts. GTP-C traffic enters the GTPC-EP context, and PFCP traffic enters the Protocol context. Each context forwards the request to the SMF service through an internal IPC bridge.

Flex-Fused mode moves service-owned GTP-C and PFCP session-message handling into the main service processing context. The SMF service pod directly owns its configured session endpoints, performs one full message decode, processes the received message under a single Message Processor, applies the existing session rules, and sends network responses without crossing an internal bridge.

Heartbeat, association, and other node-management traffic retain their existing ownership. Non-Flex deployments continue to use the existing external service path, where GTP-C and PFCP messages are handled by the GTPC-EP and Protocol pods, respectively.

Peer network functions continue to use the same GTP-C and PFCP interfaces, message types, identifiers, and timers. Existing session establishment, modification, release, report, and failure outcomes remain unchanged. Flex-Fused is an internal service optimization and does not change the customer-facing protocol behavior.

### Request Processing in Flex-Fused Mode

Flex-Fused uses a direct UDP path, while the existing Flex mode continues to use the IPC path. Both paths converge at the same SMF service Message Processor.

### Request Ingress and Convergence

In Flex-Fused mode, the request arrives directly at the SMF service UDP endpoint. The service reads the message header to identify the message and create a service transaction.

In the existing Flex mode, requests arrive through IPC and either create a new transaction or resume a waiting transaction.

Both processing paths converge at the same SMF service Message Processor.

### Common Service Processing

After the message reaches the Message Processor, the service decodes the message and identifies or creates the associated session.

The existing session procedure performs the required service and user-plane interactions, updates the session, and builds the final response.

### Response Delivery

For Flex-Fused requests, the response is encoded as a GTP-C or PFCP message, saved for duplicate-request handling, and sent directly through the service-owned UDP endpoint.

For existing Flex requests, responses continue through the IPC stream. The response follows the same transport path as the corresponding request.

### GTP-C and PFCP Peer-State Handling

SMF service pods observe peer information from session messages. The Node Manager (NodeMgr) remains the central authority for accepting and storing shared GTP-C and PFCP peer-state changes.

### Peer-State Observation

After decoding a session message, the service extracts relevant peer information, including peer identity, Recovery value, last-contact activity, and load or overload indications.

This information is obtained during the existing message decoding process and does not create an additional service transaction.

### Peer-State Publication

Observed peer-state changes are sent asynchronously to the NodeMgr responsible for the peer.

Repeated or unchanged observations are combined locally to reduce unnecessary updates when multiple service pods observe the same peer.

### State Validation and Distribution

NodeMgr validates the peer owner, revision, and update order before accepting a change.

Accepted peer-state changes are stored centrally and distributed to the required consumers. Duplicate or outdated updates are ignored.

### GTP-C Peer Handling

The first valid GTP-C session request can create a missing peer record using an atomic create-if-absent operation.

Recovery, activity, and load changes are applied only once, even when multiple service pods report the same change.

### PFCP Peer Handling

PFCP session messages can update activity, load, and overload information for an existing UPF peer record.

PFCP association, heartbeat, and peer creation procedures remain owned by the Protocol component. PFCP session traffic does not create a missing PFCP peer record.

### RAB Optimization (SGW Specific)

RAB optimization allows SGW-service to send the S11 response before the related PFCP operations complete. This allows the MME to continue without waiting for all user-plane updates.

### Early Response

SGW-service validates the request, updates the local bearer state, and sends one cached S11 RAB response.

The response is not modified or sent again after the related PFCP processing completes.

### Parallel PFCP Processing

SGW-service creates an independent PFCP operation for each affected PDN and starts the operations in parallel.

Each PFCP result is matched to its corresponding PDN.

### Completion and Failure Handling

The RAB transaction resumes after all PFCP operations reach a final state.

Failed operations are handled independently for each PDN, followed by the required cleanup. No additional S11 response is sent.

### SC-MBR Optimization (SGW Specific)

SC-MBR optimization improves response handling for Modify Bearer Requests (MBRs) that do not require immediate network-side processing.

SGW-service uses the existing subscriber, PDN, and bearer state instead of maintaining separate MBR grouping or timer caches.

### Request Classification

Each MBR is classified into one of the following categories:

- **Invalid:** The request is classified as invalid.
- **Replay:** The request is handled through duplicate-request processing.
- **No Change:** The request does not require a state change and can be accepted immediately.
- **Deferred:** The request contains changes that can be processed separately.
- **Critical:** The request requires the final downstream result before the response can be sent.

### Deferred and Critical Processing

Safe metadata-only changes are assigned to a response-free internal transaction before the Accepted response is sent.

Changes involving tunnels, bearers, peers, or N4/S5 dependencies remain in the original transaction until the final result is available.

### Response and Failure Handling

The network transaction owns one S11 response. The response is cached for duplicate replay without repeating the internal processing.

If deferred work cannot be admitted, SGW-service uses the Critical processing path or returns the defined failure.

A failure in deferred processing does not generate a second S11 response.

# Converged Core Overview

The converged core solution provides an advanced, cloud-native, converged control plane with the capability to support 4G and 5G devices, and use cases.



**Important** This release supports only the cloud-native integrated S-GW and SMF instance with S5C and cnSGW-C functionalities.

The converged core solution removes the operational complexity by providing a unified core network to handle all types of subscribers and use cases.

The operator has the following benefits:

- Improves the overall network efficiency by reducing signaling between cnSGW-C and SMF while handling a 4G subscriber or handoff from 5G to 4G coverage area.
- Reduces latency introduced due to the extra hop SGW-U for a subscriber in 4G coverage area, by collapsing the data path in the Converged UPF, thus improving the overall user experience.
- Provides ability to use a unified subscriber policy and billing infrastructure using SBA interfaces for 4G and 5G devices.

The solution supports the following converged control plane and user plane functions:

- Converged Control Plane Functions
  - Integrates S-GW and SMF network functions as a single deployment, under a single Kubernetes namespace, to support 4G and 5G devices from E-UTRAN/NR (converged core gateway)
  - Supports logical network functions (data)
- Converged User Plane Functions
  - Integrates UPF and SGW-U functionalities as a single network function
  - Provides simultaneous support for N4 and Sxa interfaces

- Terminates multiple control planes in a single deployment

# Interservice Pod Communication

## Feature Description

When the cnSGW service and SMF service selected for a subscriber are on the same cluster and same rack, the messages exchanged between the two services flow through the gtpc-ep pod.

If a collocated session is identified and **enable-gtpc-bypass** CLI command is configured under GTP endpoint, then the SMF and cnSGW-C directly communicate with each other without exchanging the messages through the gtpc-ep pod. This approach reduces the latency and the processing load on the gtpc-ep. For details on the command, see the Feature Configuration, on page 13 section.

The SMF service directly communicates with cnSGW service for processing the following requests:

- Create Bearer Request
- Update Bearer Request (UBR) (expect Modify Bearer Command (MBC) triggered UBR)
- Delete Bearer Request (DBR) (expect Delete Bearer Command (DBC) triggered DBR)

The cnSGW service directly communicates with SMF service for processing the following requests:

- Create Session Request
- Modify Bearer Request
- Delete Session Request

If the subscriber session is not collocated, the inbound and outbound messages from SMF or cnSGW-C continue to be exchanged through the gtpc-ep pod.

For this feature support on cnSGW, see the *UCC 5G cnSGWc Configuration and Administration Guide*.

## How it Works

This section describes how this feature works.

Perform the following steps to implement this feature.

1. Identify the deployment type of SMF and cnSGW-C.

   To identify the deployment type, the SMF or cnSGW-C compares the target GTPC peer IP address of the message with the locally configured IP address of S5e or S5 interface for the concerned GR instance. The SMF or cnSGW-C marks the subscriber session as collocated service based on the comparison result.

2. Identify the target service pod

   SMF uses session affinity in cnSGW namespace based on TEID, which is derived from Common ID to appropriately route the message towards cnSGW service pod instance.

3. Route the messages to the appropriate peers based on the identified deployment type and target service pod.

   Interservice pod communication uses the existing framework along with protocol buffer to carry the signaling message content.

The interservice communication between SMF and cnSGW-C happens with the following exceptions:

- GTPC messages cannot be captured using the packet sniffer tool and **monitor subscriber** command.
- Path management is not performed for collocated GTPC peers.
- GTPC message level metrics (at GTPC endpoint) will not be pegged for interservice GTPC messages as GTPC endpoint is bypassed for such messages.
- Existing interservice metrics will be pegged for interservice messages.
- UBR and DBR initiated on Command Messages follow the existing message flow path. That is, the SMF sends the command messages to cnSGW service through gtpc-ep pod.

## Feature Configuration

To enable GTPC bypass between cnSGW and SMF service, use the following sample configuration:

```
config
  instance instance-id gr_instance_id
    endpoint gtp
      enable-gtpc-bypass { false | true }
      end
```

**NOTES**:

- **endpoint gtp**: Enter the GTP endpoint configuration.
- **enable-gtpc-bypass { false | true }**: Specify the option to enable or disable the GTPC bypass between cnSGW and SMF service.

  When set to true, the GTPC bypass is enabled between SMF and cnSGW. That is, SMF and cnSGW directly communicate without involving the gtpc-ep pod. By default, it is false.

## OAM Support

This section describes operations, administration, and maintenance support for this feature.

### Bulk Statistics Support

The following statistics is updated to support this feature.

- smf_service_stats: This statistics includes gtpc_bypass label to track the GTPC bypass messages.

For more information on bulk statistics support, see the *UCC 5G SMF Metrics Reference*.

# Use Cases and Features

This section describes the use cases that SMF supports.

# Base SMF Configuration

The SMF base configuration provides a detailed view of the configurations that are required for making the SMF operational. This includes setting up the infrastructure to deploy the SMF, deploying the SMF through SMI, and configuring the Ops Center for exploiting the SMF capabilities over time.

For more information on SMI, see the *Ultra Cloud Core SMI Cluster Deployer Operations Guide*.

The following feature is related to this use case:

- Deploying and Configuring SMF through Ops Center

# 4G Session Support

***Table 4: Feature History***

| Feature Name | Release Information | Description |
|---|---|---|
| FWA Support | 2025.01.0 | SMF validates and supports 4G Fixed Wireless Access (FWA) for separate subscriber groups: those using 4G interfaces (RADIUS and Gz) and those using 5G SBA interfaces (N7 and N40). |

For UEs, the SMF supports both 5G and 4G NAS to connect to both 4G and 5G core networks. The SMF includes the EPS interworking support and acts as a PGW-C+SMF. The interfaces, such as the Gx, Gy, or Gz, which are used for a 4G session creation are replaced with the corresponding 5G core SBI interfaces, such as the Npcf and Nchf.

The SMF supports interworking with EPS by using the N26 interface (which is an inter-CN interface between the MME and the 5GS AMF) to enable interworking between the Evolved Packet Core (EPC) and the NG core networks. Support of the N26 interface in the network is optional for interworking. The N26 interface supports a subset of the functionalities over S10 interface to enable interworking. The UE uses the EPC NAS or 5GC NAS procedures that are based on the core network. The SMF supports QoS flow failures for access and mobility procedures.

Following are the two scenarios that are validated to ensure the support from SMF to handle the FWA subscribers:

1. Subscribers using RADIUS Accounting and Gz Accounting.
2. Subscribers using N7 for policy and N40 for online charging.

Following table lists the details of the supported and unsupported interfaces for FWA subscribers in the precding scenarios.

| | RADIUS Auth | RADIUS Acct | N7 | N40 | N10 | Gz (CDR) | Gy | Gx |
|---|---|---|---|---|---|---|---|---|
| Scenario 1 | No | Yes | No | No | No | Yes | No | No |
| Scenario 2 | No | No | Yes | Yes (only Online) | No | No | No | No |

**Note** SMF supports both types of subscribers using 4G and 5G SBI interfaces on the converged core across different DNNs.

SMF allows all the subscribers for:

- QoS Class Identifier (QCI) upgrade and downgrade for default QoS Flow Identifier (QFI)
- APN-AMBR upgrade and downgrade: Capability to adjust the APN Aggregate Maximum Bit Rate.

The following features are related to this use case:

- 4G to 5G Data Session Handover
- EPS Interworking
- Flow Failure Handling for Access and Mobility Procedures
- Managing Multi-Generation Device Support with Unified SMF
- Session Timers

# 5G Session Support

The Session and Service Continuity (SSC) support in 5G system architecture addresses the continuous requirements of different applications and services for a User Equipment (UE). The 5G system supports the SSC modes such that the network maintains the connectivity service to the UE. The SMF manages the UE IP address and ID allocation for establishing sessions. The SMF also maintains session connectivity on interfaces, such as N40, N4, N7, and N10, to facilitate charging.

The SMF uses the Xn interface to handover a UE from a source NG-RAN to the target NG-RAN when the AMF is unchanged, and without relocating the UPF. The SMF includes the N3 tunnel profile configuration to enable the notifications on the Control Plane (CP) and enable buffering on the UPF. The SMF supports activation and deactivation of the User Plane (UP) connection of a PDU session. The SMF also includes the DNS proxy feature to configure proxy servers for resolving the host names and their IP addresses.

The following features are related to this use case:

- Inter gNodeB Handover
- IP Pool Allocation per DNN
- UP Session Activation and Deactivation Service Request Procedures
- Session and Service Continuity Mode
- Static IP Support
- TAI Selection from AMF

# Access and Mobility Support

The SMF supports the access and mobility through session management procedures for PDU session establishment, modification, and release. The SMF supports N2-based handovers for intra-SMF or inter-AMF

when a UE moves from one NG-RAN to another NG-RAN for Data Forwarding Tunnel (DFT) and Indirect Data Forwarding Tunnel (IDFT) cases. With the multi-DNN support, SMF has multiple PDN connections for providing various services including Internet and Voice over New Radio (VoNR) services. The SMF supports network-initiated messages when a UE is either in the CM-Idle state or in the CM-Connected state.

Access and mobility support includes the intra-5G handover use case, which has the following handover support:

- Xn Handover
- Intra-AMF N2 Handover
- Inter-AMF N2 Handover

The following features are related to this use case:

- 5GSM Cause Code Handling
- AN-initiated Session Modification and Release
- CHF and PCF Integration for Access and Mobility Procedures
- Inter gNodeB Handover
- MTU Support in PCO
- DNN Support
- Network-initiated Session Modification Procedures
- Policy and User Plane Management
- UDM Integration
- Voice over New Radio

# Charging Integration

The SMF supports converged charging and uses the Nchf or N40 interface to generate charging events. The SMF supports offline failover for charging when a charging (CHF) server fails. Based on the charging data information that SMF receives, it provides reporting level support for online and offline charging.

The following feature is related to this use case:

- Subscriber Charging

# Cloud Native Infrastructure

The SMF services includes the configuration to process PDU Session Management API calls. The IP Address Management (IPAM) technique is integrated with the SMF in the Application Services layer for tracking and managing the IP address space of a network. The SMF uses the Operations Center interface, which is a system-level infrastructure, to initiate the deployment of micro-services, to push application specific configuration to one or more micro-services, and to run application-specific commands to invoke APIs in application-specific pods.

The following feature is related to this use case:

- Load and Overload Management

# Converged Core Network

The SMF supports standalone deployment or an integrated deployment with cnSGWc for serving 4G and 5G subscribers. Converged Control Plane function comprises a combination of 4G and 5G control plane instances, that is, SMF and cnSGWc.

With converged core deployment, for the same PDN session, the S-GW and SMF select the same UPF instance so that the data path is optimized. The converged core architecture reduces the operational cost and the complexity of maintaining multiple different networks, leverages new interfaces and business avenues.

The converged core deployment involves changing some basic configurations of SMF, pod layout, and optimizing performance with call processing.

The following features are related to this use case:

- Alerts
- Deploying and Configuring SMF through Ops Center
- Dynamic Routing
- EPS Interworking
  - GTP Path Failure Handling, Restoration, and Recovery
  - Support for UE Initial Attach
- Monitor Subscriber and Monitor Protocol
- Pods and Services Reference
- Policy and User Plane Management
  - Support for UPF Node Reports and Proprietary Session Reports
  - Static PCC Rules Support
- Metrics
- UPF Path Management and Restoration
- Wireless Priority Services

# IMS Support

The IP Multimedia Subsystem (IMS) connects to the LTE network and 5G core (through UPF node) for delivering voice services such as Voice over LTE (VoLTE) and Voice over New Radio (VoNR).

The following features are related to this use case:

- Voice Over LTE Support
- Voice over New Radio
- NPLI Support for VoLTE and VoNR

# IPAM Support

IP Address Management (IPAM) is a technique for tracking and managing IP addresses of a network. IPAM is one of the core components of the subscriber management system. The IPAM provides all the functionalities necessary for working with the cloud-native subscriber management system. Also, the IPAM acts as a generic IP address management system for the different network functions such as the Session Management Function (SMF), Policy Control Function (PCF), and so on.

The following feature is related to this use case:

- IP Address Management

# Lawful Intercept

The Lawful Intercept (LI) feature enables law enforcement agencies (LEAs) to intercept subscriber communications. The LI functionality provides the network operator the capability to intercept and control data messages of targeted mobile users. The SMF that handles the Control Plane actions for the PDU sessions includes an IRI-POI that has the LI capability to generate the related xIRI.

For more details, contact your Cisco account representative.

# NF Management

Based on the 3GPP-defined architecture model for 5G systems for data connectivity, SMF discovers the set of NF instances and their associate NF service instances. These instances, which are based on the NF profiles, are registered in the Network Repository Function (NRF) and meet the various input query parameters.

The following features are related to this use case:

- NF Discovery and Management
- Failure Handling Support

# OAM Support

This use case covers all the Operation, Administration, and Maintenance (OAM) functions of the SMF.

The following features are related to this use case:

- Alerts
- Bulk Statistics and Key Performance Indicators
- Deploying and Configuring SMF through Ops Center
- Logs
- Metrics
- Monitor Subscriber and Monitor Protocol
- Pods and Services Reference
- SMF Rolling Software Update

# Policy Integration

The SMF communicates with the Unified Data Management (UDM) and Policy Control Function (PCF) to perform the following:

- Procure the subscribed and authorized QoS parameters for the Guaranteed Bit Rate (GBR) and non-GBR flows
- Pass the relevant information to the UE (NAS), gNB (NGAP), and UPF (PFCP)

This ensures that all nodes on the network provide the desired QoS to the PDU session.

The SMF uses the service-based N7 interface with the PCF to retrieve the session management policy information corresponding to the PDU session of the UE. The SMF selects the PCF during the PDU Session Establishment procedure. It also acts as a consumer of the PCF-provided session management policy service.

The following features are related to this use case:

- Policy and User Plane Management
- Wireless Priority Services

# RADIUS Support

In the 5G architecture, the serving network authenticates the Subscription Permanent Identifier (SUPI) during authentication and the key agreement between the UE and the network. In addition, the serving network can perform a secondary authentication for data networks outside the mobile operator domain.

The following feature is related to this use case:

- RADIUS Authentication and Accounting

# Redundancy Support

The SMF deployment in K8 cluster plays a vital role to support High Availability (HA) and Geographic Redundancy (GR). The redundancy support ensures stateful session continuity among the clusters during the rack or cluster failures.

The SMF achieves HA through redundant set-up of each cluster component such that any single point of failure is avoided.

The GR provides rack-level redundancy to replicate data between two separate K8 clusters across racks so that, on rack or cluster failure, traffic can switch to a remote rack to process the traffic. Rack or cluster failure can be due to power failure, multi-compute failures, network failure, multi-pod failure, BFD link failure, and so on.

The following features are related to this use case:

- High Availability Support
- Inter-Rack Redundancy Support
- Mesh Connectivity to All UPFs

# Roaming Support

Mobile network operators make roaming partnerships to provide services to the subscribers seamlessly in geographies beyond their network reach. PLMNs define the operator network boundaries. HPLMN is the Subscriber's home network and VPLMN is the visited network from where the service is rendered.

The following features are related to this use case:

- Roaming Support
- Multiple PLMN Support

# SMF Inline Services

The SMF uses the Inline Services feature such as the Enhanced Charging Service (ECS) that enables operators to reduce billing-related costs and gives the ability to offer tiered, detailed, and itemized billing to their subscribers. Using shallow and deep packet inspection (DPI), the ECS [also known as Active Charging Service (ACS)] allows operators to charge subscribers based on the actual usage, number of bytes, premium services, location, and so on. The ECS also generates charging records for postpaid and prepaid billing systems.

The following features are related to this use case:

- Event Detail Records
- Policy and User Plane Management

# SMF Specification Compliance

The SMF supports different 3GPP specification versions for the SMF interfaces. It processes the messages from the interfaces as per the compliance profile configured for the corresponding services.

The following feature is related to this use case:

- Interfaces Support

# Subscription Management

The SMF handles the user subscription management over the N10 interface.

The following feature is related to this use case:

- UDM Integration

# UPF Integration

The SMF uses the available StarOS-based UPF node to meet the non-standard requirements on the UPF node to interwork with this UPF. To comply with the IPv6 Stateless Auto-configuration, the SMF supports ICMPv6 Router Solicit and Advertisement.

The following features are related to this use case:

- Policy and User Plane Management

- IPv6 PDU Sessions
- UPF Path Management and Restoration

## Wi-Fi Support

The SMF supports Voice over Wi-Fi (VoWiFi). The VoWiFi technology provides the telephony services using Voice over IP (VoIP) from the mobile devices that are connected across a Wi-Fi network.

The following features are related to this use case:

- VoWi-Fi Support
- Wi-Fi Handover

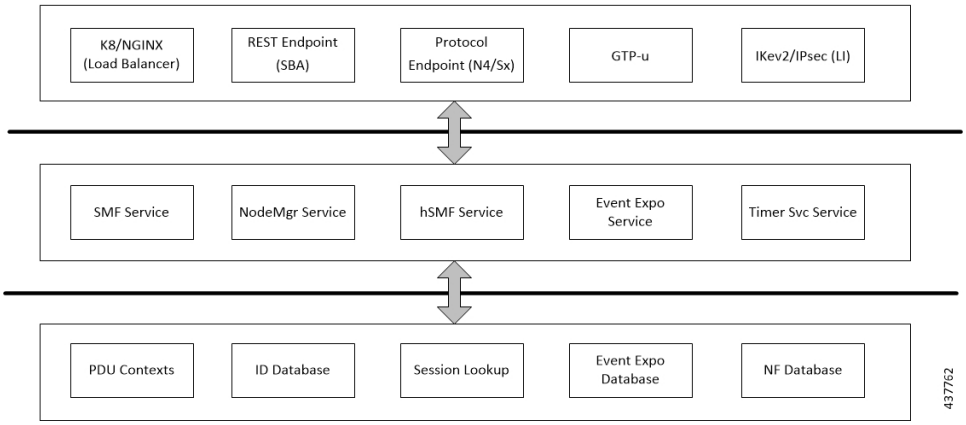
# Deployment Architecture and Interfaces

The Cisco SMF is a part of the 5G core network functions portfolio with a common mobile core platform architecture. The core network functions include Access and Mobility Management Function (AMF), Network Repository Function (NRF), Policy Control Function (PCF), Network Slice Selection Function (NSSF), and User Plane Function (UPF).

## SMF Architecture

The SMF network function consists of loosely coupled microservices together. The microservice decomposition is based on a three-layered architecture as illustrated in the following figure.

***Figure 1: SMF 3-Layered Micro Services Architecture***




Following are the three layers of the SMF architecture:

- Layer 1—Protocol and Load Balancer services (Stateless)
- Layer 2—Application services (Stateless)
- Layer 3—Database services (Stateful)

# SMF Deployment

The 5G Mobility NFs deployment supports the following modes:

- Standalone mode: In this mode, each NF together with the required microservices is deployed in a separate name space in Kubernetes.
- Converged mode: In this mode, several NFs are deployed together in a single name space and micro-service common to NFs render the service to all the deployed NFs.

# Converged Core Architecture

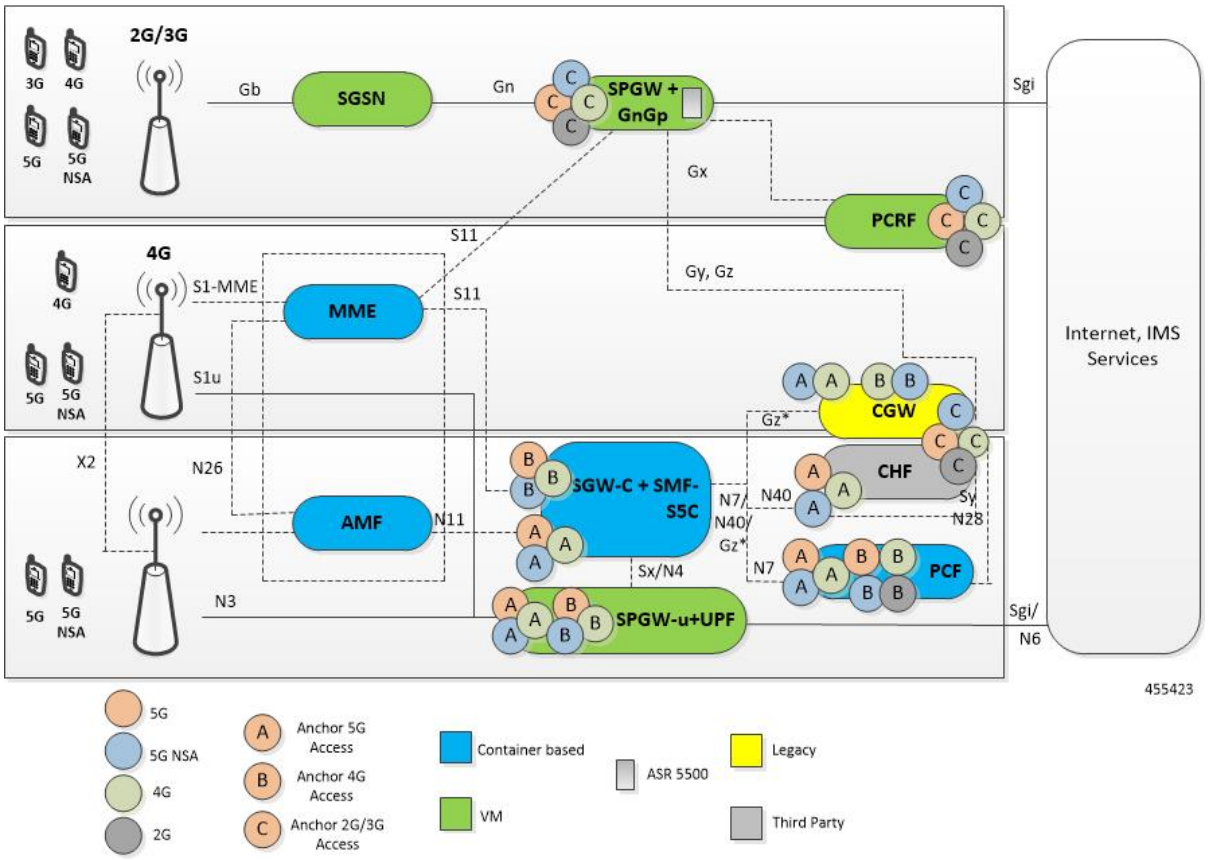
The converged core solution provides a single unified platform which is based on SMI architecture. The supporting architecture integrates the cloud-native S-GW and SMF deployment with 5GC and cnSGW-C functionalities. The solution uses 3GPP-defined SBA interfaces for policy and charging functions.

In the converged core architecture, the 4G and 5G capable UEs are anchored on the same control plane instance. The control plane instance provides the SMF, 5GC, and cnSGW-C functionalities.

The handoffs between 4G and 5G access types are seamless for 5G capable devices. The handoffs from LTE to UTRAN (bi-directional communication between 4G/5G and 3G/2G) are not seamless for 4G capable devices.

The following figure illustrates the supported network architecture.

*Figure 2: Converged Core Architecture*




The UPF deployed as a part of this solution is a VPC-SI VM. The UPF deployment is VM-based, and supports:

- SGW-U, PGW-U, and UPF functionalities in the same instance, and exposes the Sxa, Sxb, Sxab, or N4 interface towards the control plane.
- Multiple CP instances (up to 4) simultaneously.

# Converged Core Deployment

The converged core deployment is based on the converged control plane and unified user plane infrastructure for all use cases.

In the converged core deployment, all 4G and 5G-capable UEs are anchored on the 5G core (SMF) with SBA interfaces towards PCF.

The converged core deployment has a converged Ops Center that allows the configuration of cnSGW-C and SMF services along with other microservices. A single product helm chart is used to install components.

The following figure illustrates the Kubernetes deployment for the converged S-GW and SMF network function.

*Figure 3: Kubernetes Deployment*

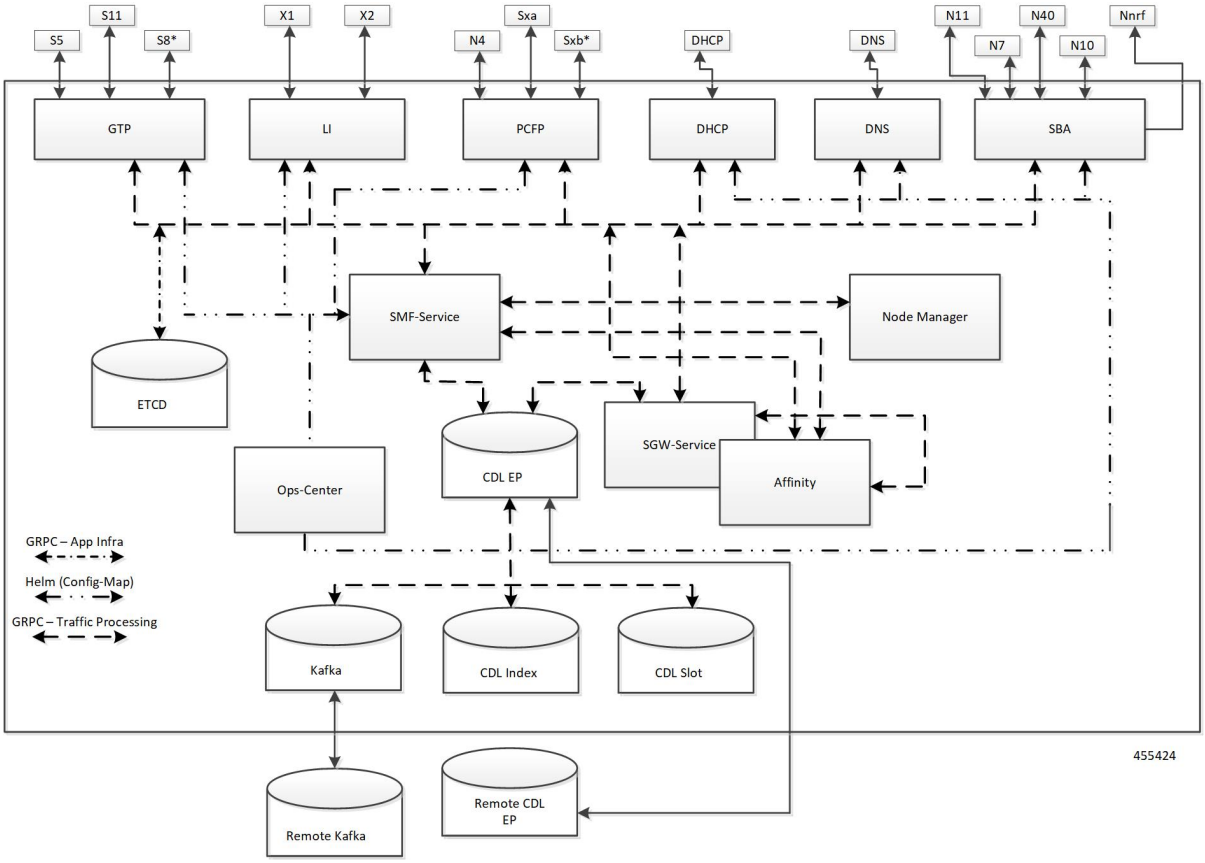



*Figure 4: Kubernetes Deployment*

The protocol layer services are shared across SMF and S-GW. The GTP endpoint terminates the S11 interface and S5/S8 interface. Similarly, the PFCP (protocol) endpoint terminates the N4 and Sxa interfaces.

The SMF and S-GW services are deployed as distinct pods and the session processing is segregated. Both the service pods use CDL for storing subscriber sessions.

# Supported Interfaces

This section describes the interfaces supported between the SMF and other network functions in the 5GC.

- Diameter—Interface that provides framework for services that require Access, Authorization, and Accounting (AAA) or Policy support across IP-based networks.
- GTP—Uses the N9 interface as the reference point between two core UPFs.
- Gx—Interface between SMF and PCRF.
- Gy—Interface between SMF CTF and OCS Charging Data Function (CDF).
- N1/NAS—Reference point between the UE and AMF.
- N2/NGAP—Reference point between the RAN and AMF.

- N4—Reference point between the SMF and UPF.
- N7—Reference point between the SMF and PCF.
- N10—Reference point between the UDM and SMF.
- N11—Reference point between the AMF and SMF.
- N40—Reference point between the SMF and CHF.
- Nnrf—Interface displayed by NRF on 3GPP 5G system architecture.
- RADIUS—Interface that manages network access.
- S2b—Interface between the PGW-C and ePDG.
- S5—Interface between the PGW-C and S-GW.
- SBA—Interface for NFs to communicate with each other.

For details on the supported interfaces, see the Interfaces Support chapter.

# Life Cycle of Data Packet

The following call flow depicts the life cycle of a data packet traversing through various pods of the SMF for a successful PDU session establishment.

The SMF application includes the following pods:

- REST-EP
- Cache
- Service
- Nodemgr
- Protocol
- UDP-Proxy
- CDL

*Figure 5: 4G Session Procedure - Complete ByPass(PFCP and GTP)*

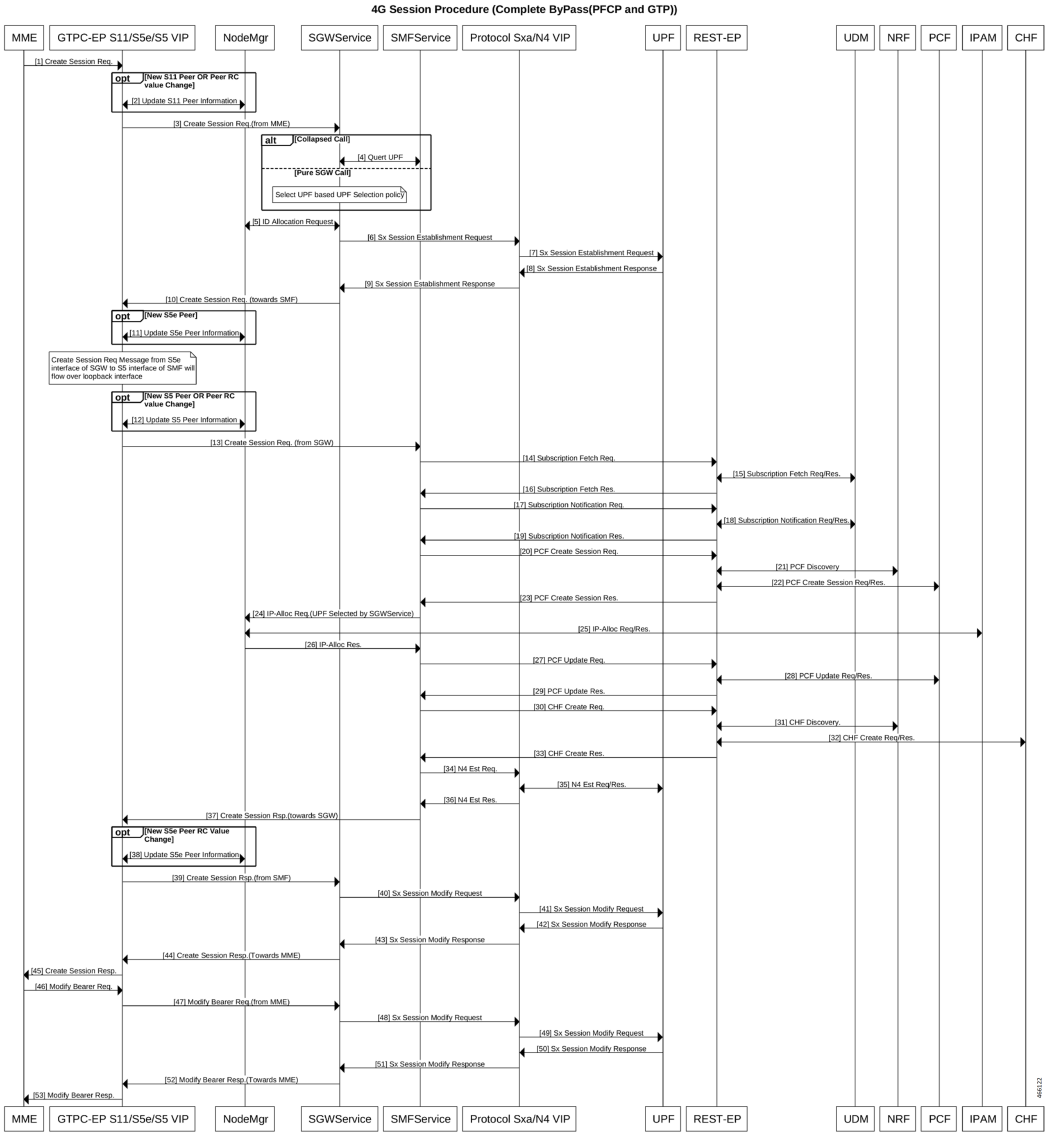


*Figure 6: 4G Session Procedure with UDP Proxy for PFCP and GTP*

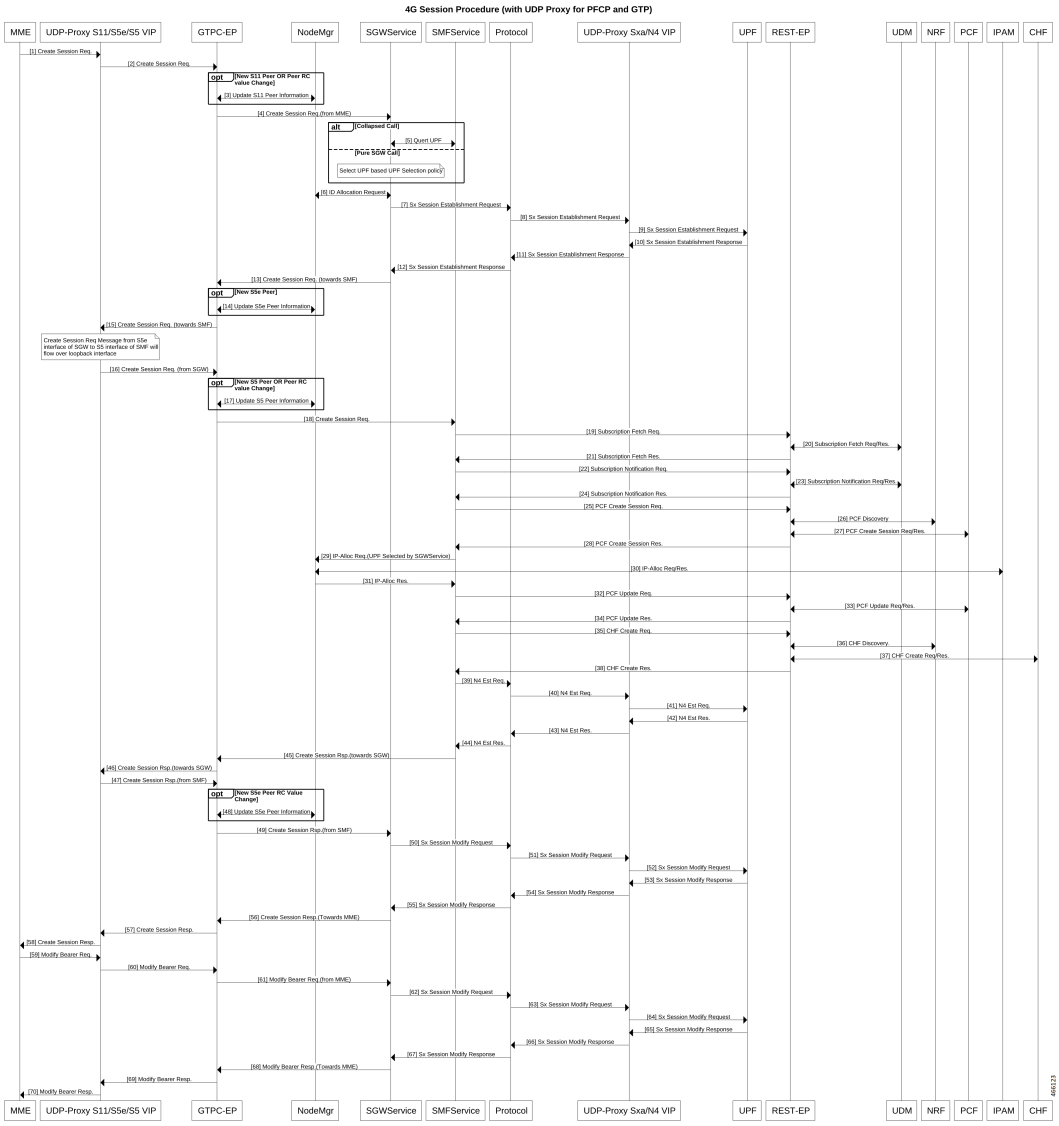



*Table 5: End-to-End PDU Session Establishment Call Flow Description*

| Step | Description |
|---|---|
| 1 | The AMF sends N11:SMContextCreateRequest to the SMF, which terminates on the VIP-IP/external IP of REST-EP pod. |
| 2 | The REST-EP pod performs look-up for session affinity with cache pod. The SMF does not have the entry for the user session. The cache output does not result in any SMF-service affinity for the user session.<br><br>Kubernetes service/ISTIO load balancer selects one SMF-service pod from multiple SMF-service pods that are configured. |

| Step | Description |
|---|---|
| 3 | The REST-EP starts the timer associated with transaction-1. The PDU session establishment procedure involves using three transactions which are started at different stages of the call flow.<br>The default transaction timer on SMF is 10 seconds. The transaction timers are configurable through Service Level Agreement (SLA) feature. |
| 4 | The REST-EP forwards the N11:SMContextCreateRequest to the selected SMF-service. |
| 5 | The SMF-service starts procedure timer (guard timer/SLA timer). The SLA timers are configurable. |
| 6 | The SMF-service adds affinity entry with cache pod for the session. The SMF continues to use the same selected SMF-service in the subsequent stages of the call flow until the cache is expired. |
| 7 | The SMF-service instructs the REST-EP pod to trigger N10: Registration Request. |
| 8 | The REST-EP decides whether to perform NF discovery or static NF selection of UDM based on the configuration. |
| 9 | The REST-EP encodes and sends UDM discovery request to the NRF and receives a successful response with the list of UDMs. |
| 10 | The REST-EP encodes and sends N10:RegistrationRequest to the selected UDM. |
| 11 | The REST-EP starts messageResponseTimer. The default value of the configurable messageResponseTimeout is 2 seconds. The messageResponseTimer is applicable for all outbound HTTP2 messages initiated by SMF. They are not explicitly called out in the subsequent stages of the call flow. |
| 12 | The REST-EP receives successful N10:RegistrationResponse from the UDM. |
| 13 | The REST-EP stops messageResponseTimer. |
| 14 | The REST-EP forwards the N10:RegistrationResponse to the SMF-service. |
| 15 | The SMF-service instructs the REST-EP pod to trigger N10:SubscriptionFetchRequest. |
| 16 | The REST-EP encodes and sends N10: SubscriptionFetchRequest to the UDM. The REST-EP receives a response from the UDM. |
| 17 | The REST-EP forwards the N10:SubscriptionFetchResponse to the SMF-service. |
| 18 | The SMF-service instructs the REST-EP pod to trigger N10:SubscribeNotificationRequest. |
| 19 | The REST-EP encodes and sends N10:SubscribeNotificationRequest to UDM. The REST-EP receives a response from the UDM. |
| 20 | The REST-EP forwards the N10:SubscribeNotificationRequest to the SMF-service. |
| 21 | The SMF-service sends N11:SMContextCreateResponse to the REST-EP. |
| 22 | The REST-EP forwards the N11:SMContextCreateResponse to the AMF. |
| 23 | The REST-EP stops the transaction-1 timer started in step 3. |
| 24 | The SMF-service adds the session entry information in the CDL. |
| 25 | The SMF-service starts an internal transaction by sending NIntSelfTxnPduSetup message. |
| 26 | The SMF-service starts the timer associated with transaction-2. |

| Step | Description |
|---|---|
| 27 | The SMF-service instructs the REST-EP pod to trigger N7:SMPolicyCreateReq. |
| 28 | The REST-EP decides whether to perform NF discovery or static NF selection of PCF based on the configuration. |
| 29 | The REST-EP encodes and sends the PCF discovery request to the NRF and receives a successful response with the list of PCFs. |
| 30 | The REST-EP encodes and sends N7:SMPolicyCreateReq to the selected PCF. The REST-EP receives a response from the PCF. |
| 31 | The REST-EP forwards N7:SmPolicyCreateSuccess to the SMF-service. |
| 32 | The SMF-service performs the UPF selection. |
| 33 | The SMF-service sends ResourceMgmtReq to IPAM module of Nodemgr to request the IP address for the UE. |
| 34 | The SMF-service receives ResourceMgmtResp from the IPAM module of the Nodemgr with the IP address to the UE. |
| 35 | The SMF-service instructs the REST-EP pod to trigger N7:SMPolicyUpdateReq with trigger "UE_IP_CH". |
| 36 | The REST-EP encodes and sends N7:SMPolicyUpdateReq with UE_IP_CH trigger to the selected PCF. The REST-EP receives a response from the PCF. |
| 37 | The REST-EP sends N7:SMPolicyUpdateSuccess to the SMF-service. |
| 38 | The SMF-service instructs the REST-EP pod to trigger N40:ChargingDataRequest. |
| 39 | The REST-EP decides whether to perform the NF discovery or static NF selection of CHF based on the configuration. |
| 40 | The REST-EP encodes and sends the CHF discovery request to the NRF. The REST-EP receives a successful response with the list of CHFs. |
| 41 | The REST-EP encodes and sends N40:ChargingDataRequest to the selected CHF. The REST-EP receives a response from the CHF. |
| 42 | The REST-EP forwards N40:ChargingDataResponse to the SMF-service. |
| 43 | The SMF-service instructs the SMF-Protocol pod to trigger N4:SessionEstablishmentRequest. |
| 44 | The SMF-Protocol encodes and sends the N4:SessionEstablishmentRequest to the UDP-Proxy pod. |
| 45 | The UDP-Proxy pod sends the N4:SessionEstablishmentRequest to the UPF. The UDP-Proxy receives a response from the UPF. |
| 46 | The UDP-Proxy forwards the N4:SessionEstablishmentResponse to the SMF-Protocol pod. |
| 47 | The SMF-protocol forwards the N4:SessionEstablishmentResponse to the SMF-service. |
| 48 | The SMF-service instructs the REST-EP to trigger N11:N1N2MessageTransferReq. |
| 49 | The REST-EP decides whether to perform NF discovery or static NF selection of AMF based on the configuration. |
| 50 | The REST-EP encodes and sends the AMF discovery request to the NRF. The REST-EP receives a successful response with the list of AMFs. |

| Step | Description |
|---|---|
| 51 | The REST-EP encodes and sends N11:N1N2MessageTransferReq to the selected AMF. The REST-EP receives a successful response from the AMF. |
| 52 | The REST-EP forwards the N11:N1N2MessageTransferSuccess to the SMF-service. |
| 53 | The REST-EP stops the transaction-2 timer started in step 26. |
| 54 | The SMF-service updates the session entry in the CDL. |
| 55 | The REST-EP receives N11:SMContextUpdate from the AMF. |
| 56 | The REST-EP looks-up for session affinity in the cache pod and identifies the SMF-service handling the session. |
| 57 | The REST-EP starts the timer associated with transaction-3. |
| 58 | The REST-EP forwards the N11:SMContextUpdate to the SMF-service pod learnt in step 56. |
| 59 | The SMF-service instructs the SMF-Protocol pod to trigger N4:SessionModificationRequest. |
| 60 | The SMF-Protocol encodes and sends the N4:SessionModificationRequest to the UDP-Proxy pod. |
| 61 | The UDP-Proxy pod sends the N4:SessionModificationRequest to the UPF. The UDP-Proxy receives a response from the UPF. |
| 62 | The UDP-Proxy forwards the N4:SessionModificationResponse to the SMF-Protocol pod. |
| 63 | The SMF-protocol forwards the N4:SessionModificationResponse to the SMF-service. |
| 64 | The SMF-service forwards the N11:SMContextUpdateSuccess to the REST-EP. |
| 65 | The REST-EP forwards the N11:SMContextUpdateSuccess to the AMF. |
| 66 | The REST-EP stops the transaction-3 timer started in step 57. |
| 67 | The SMF-service updates the session entry in the CDL. |
| 68 | The SMF-service stops the procedure timer (guard timer/SLA timer). |

# Session Affinity

The SMF supports session affinity to facilitate stateless architecture.

When a session management procedure is ongoing for a subscriber session in some SMF service instance and another event from the network comes for the same subscriber in the meantime. Then, the SMF protocol layer micro-services, such as "smf-rest-ep" and "smf-protocol" direct these events towards the concerned SMF service instance. This ensures that all network events pertaining to an ongoing procedure of a subscriber session are handled by the same SMF service instance until the completion of the procedure.

Upon completion of the procedure, the subscriber session information is updated in the database and the session affinity towards the SMF service instance is removed. Subsequent network events can be handled by any of the available SMF service instances, by fetching the relevant subscriber session information from the database.

# License Information

The SMF supports Cisco Smart Licensing, a cloud-based licensing model that simplifies the purchase, deployment, and management of Cisco software assets. For more information, refer to the Smart Licensing collection page.

# Standards Compliance

Cisco SMF complies with the following 3GPP standards as per Release 15 June 2019:

- *3GPP TS 23.510, version 15.4.0*
- *3GPP TS 29.274, version 15.8.0*
- *3GPP TS 23.007, version 15.4.0*
- *3GPP TS 23.501, version 15.6.0*
- *3GPP TS 29.244, version 15.6.0*
- *3GPP TS 33.515, version 0.4.0*
- *3GPP TS 29.510, version 15.3.0*
- *3GPP TS 32.255, version 15.3.0*
- *3GPP TS 32.291, version 15.3.0*
- *3GPP TS 32.290, version 15.4.0*
- *3GPP TS 29.501, version 15.4.0*
- *3GPP TS 23.503, version 15.6.0*
- *3GPP TS 24.501, version 15.4.0*
- *3GPP TS 24.502, version 15.4.0*
- *3GPP TS 24.503, version 15.4.0*
- *3GPP TS 29.518, version 15.4.0*
- *3GPP TS 23.402, version 15.3.0*
- *3GPP TS 38.413, version 15.4.0*
- *3GPP TS 23.401, version 15.8.0*
- *3GPP TS 29.500, version 15.8.0*

# ULB-based deployment model

The SMF supports the large-ulb deployment model, which integrates ULB support into the converged core. This model facilitates direct forwarding of GTP session ingress messages from the ULB eBPF program to

the smf-service or sgw-service pods, bypassing the GTP proxy pod. This optimization significantly reduces communication latency and improves overall system throughput.