Service based Interface Development in 5G CN

Luhan Wang
Beijing Univ. of Posts & Telec.
Beijing, 6. 23, 2018
1.1 Service based Architecture Overview

3GPP TS 23.501 defines the 5G System Architecture as a Service Based Architecture, i.e. a system architecture in which the system functionality is achieved by a set of NFs providing services to other authorized NFs to access their services.

The following Control plane interfaces within 5G Core are defined as service based interface:

- Namf, Nsmf, Nudm, Nnrf, Nnssf, Nausf, Nnet, Nsmsf, Nudr, Npcf, N5g-air
1.2 Benefits of Service-based Architecture

**Modularity & Reusability**
- The network is composed of modularized services, also known as Microservices.
- Services can be reused among different network functions.

**Cloud-Native**
- Continuous delivery, shrinking testing and integration timescales (moving towards continuous integration) which reduces the time to market for installing bug fixes, and rolling out new features.
- Containerization, allowing individual services updated/extended with minimal impact to other services.

**Extensibility**
- Service based interfaces can be easily extended without introducing new reference points.
- Traffic can be easily balanced or offloaded by deployment new NF service instance.

**Openness**
- Together with some control functions (i.e. authentication), service based interface can be easily exposed to external users, such as 3rd-party application providers.
2.1 Protocols over Service Based Interfaces

- The service based interfaces use HTTP/2 protocol, HTTP/2 provide low latency, security, high efficiency, bidirectional communication among NF services.

- Use JSON as the application layer serialization protocol.

- For the security protection at the transport layer, all 3GPP NFs shall support TLS and TLS shall be used within a PLMN.

- Interface Definition Language: OpenAPI Specification shall be used as Interface Definition Language (IDL) of SBI. [https://github.com/OAI/OpenAPI-Specification]
2.2 URI Structure

• A URI uniquely identifies a resource. In the 5GC SBI APIs the resource URI structure shall be specified as follows:

  {apiRoot}/{apiName}/{apiVersion}/{apiSpecificResourceUriPart}

  - “apiRoot” is a concatenation of the following parts:
    - Scheme: http://, or https://
  - “apiName” defines the name of the API
  - “apiVersion” defines the version of the API
2.3 Use of HTTP Methods

• Creating a Resource
  - Procedures that allow an NF service consumer to create a new resource at the NF service producer shall be specified to either use the HTTP POST method, or the HTTP PUT method with procedures.

Creating a Resource using POST

1. POST .../parent_resource (ResourceRepresentation)
2. 201 Created (ResourceRepresentation)

Creating a Resource using HTTP PUT

1. PUT .../resource (ResourceRepresentation)
2. 201 Created (Resource Representation)
• Reading a Single Resource
  - Procedures that allow a service consumer NF (client) to read information from the server shall be specified to use the HTTP GET method to obtain the current representation of a resource.

Reading a resource using GET
2.5 Use of HTTP Methods

• Updating a Resource
  - Procedures that allow a service consumer NF (client) to update information stored at the server by means of a complete replacement shall be specified to use the HTTP PUT method to replace the current representation of a resource with a new representation.

![Diagram of PUT method]

Updating a resource using PUT

1. PUT .../resource (ResourceRepresentation)
2. 204 No Content ()
or 200 OK
2.6 Use of HTTP Methods

• Deleting a Resource
  - Procedures that allow a service consumer NF (client) to delete a resource from the server shall be specified to use the HTTP DELETE method

Deleting a resource using DELETE

1. DELETE .../resource ()
2. 204 No Content ()
The Service Based Architecture shall support the NF Service Framework that enable the use of NF services.

The NF Service Framework includes the following mechanisms:

- NF service registration and de-registration: to make the NRF aware of the available NF instances and supported services;
- NF service discovery: to enable a NF Service Consumer to discover NF Service Producer instance(s) which provide the expected NF service(s);
- NF service authorization: to ensure the NF Service Consumer is authorized to access the NF service provided by the NF Service Producer.
3.2 General Functionalities in Service Based Architecture

• **Routing Mechanisms**
  - Route HTTP messages between Network Functions, routing within a PLMN, routing across PLMN, Message forwarding.

• **Server-Initiated Communication**
  - Subscribe-Notify service operations require bidirectional communication between the NFs when the server needs to initiate communication with the client.
    - NF service consumer acts as an HTTP client and NF service producer acts as an HTTP server when NF service consumer subscribes to NF service producer’s notifications;
    - NF service producer acts as an HTTP client and NF service consumer acts as an HTTP server when NF service producer delivers notifications to NF service consumer.
3.3 General Functionalities in Service Based Architecture

• Load Control
  - NRF may provide the NF instance and/or the NF service instance information with the NF capacity information advertised during NF Service Registration and/or NF Service Update

• Overload Control
  - Congestion control between two endpoints is provided by TCP layer and HTTP2 layer.
  - An NF Service Producer may mitigate a potential overload status by sending the NF Service Consumer the following HTTP status codes as a response to requests received during, or close to reaching, an overload situation:
    - 503 Service Unavailable;
    - 429 Too Many Requests;
    - 307 Temporary Redirect

- Support of Stateless NFs
  - A NF may become stateless by decoupling the "compute" resource and "storage" resource e.g. AMF may become stateless by storing the UE related information in the UDSF
4.1 Network Function Repository Services

- The Network Function (NF) Repository Function (NRF) is the network entity, which enables service-based architecture in the 5G Core Network (5GC). It supports the following functionality:
  - Maintains the NF profile of available NF instances and their supported services;
  - Allows other NF instances to subscribe to, and get notified about, the registration in NRF of new NF instances of a given type;
  - Supports service discovery function. It receives NF Discovery Requests from NF instances, and provides the information of the available NF instances fulfilling certain criteria (e.g., supporting a given service).

Reference architecture for the 5GC, with focus on the NRF
## 4.2 Network Function Repository Services

<table>
<thead>
<tr>
<th>Network Function Repository Services</th>
<th>Nnrf_NFManagement</th>
<th>Nnrf_NFDiscovery</th>
<th>OAuth2 Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Allows a Network Function Instance in the serving PLMN to register, update or deregister its profile in the local NRF.</td>
<td>• Allows a Network Function Instance to discover services offered by other Network Function Instances, by querying the local NRF.</td>
<td>• Offers an OAuth2 authorization, following the &quot;Client Credentials&quot; authorization grant. It exposes a &quot;Token Endpoint&quot; where the Access Token Request service can be requested by NF Service Consumers.</td>
</tr>
<tr>
<td></td>
<td>• Allows an NF to subscribe to be notified of newly registered NF Instances along with their NF services.</td>
<td>• Allows an NRF in a PLMN to re-issue a discovery request towards an NRF in another PLMN.</td>
<td></td>
</tr>
</tbody>
</table>
4.3 Network Function Repository Services

- Nnrf_NFManagement service:
  - NF Register
  - NFUpdate
  - NFDeregister
  - NFStatusUnsubscribe
  - NFStatusNotify
  - NFStatusSubscribe

**NFRegister Operation:**

1. PUT 
   
   .../nf-instances/{nfInstanceID} (NFProfile)

2. 201 Created (NFProfile)

The NF Service Consumer shall send a PUT request to the resource URI representing the NF Instance. The URI is determined by the NF Instance. The variable \{nfInstanceID\} represents an identifier, provided by the NF Service Consumer, that shall be globally unique inside the PLMN of the NRF where the NF is being registered. The format of the NF Instance ID shall be a Universally Unique Identifier (UUID)
4.4 Network Function Repository Services

**NFDiscover:**

It provides to the NF service consumer the IP address(es) or FQDN of the NF Instance(s) or NF Service(s) matching certain input criteria.

✓ **NFDiscover Operation:**

1. The NF Service Consumer shall send an HTTP GET request to the resource URI "nf-instances" collection resource.
2. On success, "200 OK" shall be returned, contain a validity period, during which the search result can be cached by the NF Service Consumer, and an array of NF profile objects, that satisfy the search filter criteria (e.g., all NF Instances offering a certain NF Service name). If the NF Service Consumer is not allowed to discover the NF services for the requested NF type provided in the query parameters, the NRF shall return "403 Forbidden" response.
4.5 Network Function Repository Services

**OAuth2 Authorization:**
The NRF offers an OAuth2 authorization service, following the "Client Credentials" authorization grant. It exposes a "Token Endpoint" where the Access Token Request service can be requested by NF Service Consumers.

**Access Token Request:**

1. The NF Service Consumer shall send a POST request to the "Token Endpoint", The "Token Endpoint" URI shall be:
   
   `{nrfApiRoot}/oauth2/token`

2. On success, "200 OK" shall be returned. If the access token request fails at the NRF, the NRF shall return "400 Bad Request" status code, including a JSON object in the response payload, that includes details about the specific error that occurred.
5.1 Implementation of HTTP/2 in 5GCore

• The traditional way to use HTTP

  ![Diagram of HTTP server](image)

  HTTP server
  Apache/Tomcat/Ngnix

  Client

  Request/Response

• The traditional way won’t work in 5G core
  - Each NF can be both server and client
  - oai-cn code doesn’t work in Apache/Tomcat/Ngnix
5.2 Implementation of HTTP/2 in 5GCore

- Intergrate ngHttp2 into oai-cn codes.
  - This is an implementation of the HTTP2 in C language.
  - Server, Client, Proxy are implemented in ngHttp2 lib.
5.3 Implementation of HTTP/2 in 5G Core

※ Develop 5G Core by restructuring openair-cn.

※ Steps

Step 1. Split of MME into AMF and SMF

Step 2. Split of SPGW into UPF and SMF

Step 3. Split of HSS to UDM and AUSF
MME Splitting Principles

- Change synchronous calls into asynchronous messages

- Split Coupled contexts

- Change multiple threads into multiple progresses.
Current Status

■ Develop from Scratch
  ✓ 28th June, 2nd Meeting
  ✓ SBA skeleton Design
  ✓ MicroServices approach Design

■ Current Develop Team
  ✓ Eurecom- Lionel
  ✓ B-com - Frédéric LER, Nadir BOUYAHMED, Cao-Thanh PHAN
  ✓ BUPT- Luhan Wang, Zeyu Pu, Xinli Zhou, Keliang Du
  ✓ Blackned - Dincer Beken, Andreas Eberlein
  ✓ Ng4t - Jens Irrgang
  ✓ Orange - CHONG CHAUVOT Linda IMT

■ Trello Task Manager
  ✓ https://trello.com/b/FkusxAie/oai-5g-cn
Thank you!

wluhan@bupt.edu.cn