Service Based Architecture

Owned by David Cherney DISH
Last updated: Jun 21, 2023 • Add Workflow

Contents

Service based architecture is a term that 3GPP coined to describe a way that the 5G core would be significantly different than 4G. Software development has made progressively more and more use of RESTfull (representational state transfer) API calls with protocol HTTP/2 since their invention in the year 2000. Foreseeing that future developers will want to create applications that interact with the 5G core, 3GPP decided to make all control plane communication be through such API calls. Every 5G NF has an interface called a service based interface (SBI, another 3GPP specific term) to which API calls can be made to access the services provided by that NF.

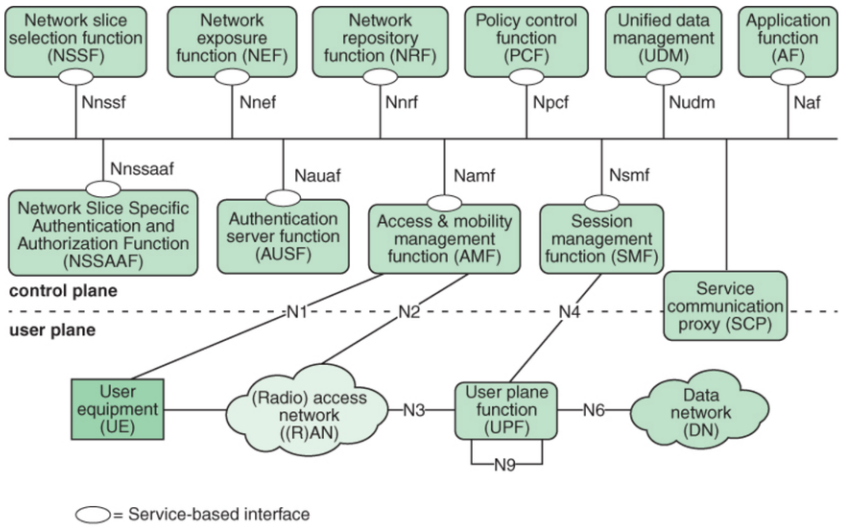


FIGURE 9.4 Non-Roaming 5G System Architecture

For example, the AMF has a service based interface called the Namf through which a service operation called `UEContextTransfer` can be accessed. (That service operation allows information about a UE to be passed from one AMF to another.)

(The service communication proxy (SCP) is a NF that can act as an intermediary and translator in SBI communication. If two network functions are from different software vendors, the differences in the vendors' software might introduce complications in SBI communication between the NFs. Then the SCP is where the network operator can manage those complications without relying on the software vendors to update their software.)

HTTP Lingo

HTTP is a request-response based protocol (as opposed to a session based protocol like GTP, or a one way protocol like UDP). Every communication has a server, a client, and a resource.

Definition: A server is an object that can deliver a service upon request.

Definition: A client is an object that makes a request to a server for a service.

Definition: A resource is an object or procedure provided by a server upon request from a client.

The terms 'service' and 'resource' have a very subtle difference in meaning; a service is access to a resource.

Methods

The client specifies the resource it wants and also an action to perform on it. The possible actions are called "HTTP methods". Here is an incomplete list of HTTP methods.

- `GET` is for retrieving information
- `PUT` is for requesting that info be stored in the server under a URI provided by the client, thereby either creating or overwriting a document.
- `PATCH` is for modifying a resource
- `POST` is like put but appends to the end of a document
- `DELETE` is for deleting a resource
- `OPTIONS` is for returning the HTTP methods supported by the server
- `CONNECT` is for establishing a TCP session

The 4 Parts of and API Call

1. API Endpoint: This is a type of Uniform Resource Identifier (URI) in that it uniquely identifies a resource and operation that can be provided on it. (All API endpoints are URIs, but not all URIs are API endpoints.)
   - e.g. of a SBI API endpoint is `http://172.20.137.254:7777/nnrf-nfm/v1/nf-instances`
     - it identifies the service operation `nf-instances` on the network function management (`nfm`) service exposed on the Nnrf SBI.
     - The IP address of the NRF here is `172.20.137.254`
     - The port on which the service is exposed is `7777`
     - The version of the API is `v1`
   - the scheme can be `http` or `https`
   - the host and port can be specified with address or with a domain name
   - the rest describes a resource.
2. Method: (described above)
3. Headers:
   a. These often specify that the data sent or received is in JSON format.
4. body:
   - this is the payload delivered in a put or post, and is usually in JSON format.

5G Lingo For APIs in the 5G Core

The pair of terms client-server is replaced in 5GC lingo with consumer-producer.

The subtle difference between a service and the resource available through the resource is explicitly recognized in 5G lingo; the syntax of a SBA API call includes both a name for the service and a name for the resource. That name for the resource is called the service operation.

5G URIs

In 5GC, an absolute URI for a SBI is required to have the following structure. (Examples are provided in the following subsections.)

```
<apiRoot>/<apiName>/<apiVersion>/<apiSpecificResourceUriPart>
```

That comes in two parts:

1. the base URI
   - `<apiRoot>` is the concatenation of
     - scheme (`http` or `https`)
     - `://`
     - host and, optionally, port
     - optional API prefix that starts with `/`
   - `apiName` is the name of the API, and it includes

    - the SBI interface (e.g. `Nnef`)
    - the nickname of the service (e.g. `nfm` for network function management.)
  - `apiVersion` is the version of the API
2. resource part
  - `apiSpecificResourceUriPart` specifies a resource, and is referred to as a service operation.

As an example, the following linux `curl` command is an SBI API call for a resource/service operation called `nf-instances`.

```
curl --http2-prior-knowledge GET http://172.20.137.254:7777/nnrf-nfm/v1/nf-instances
```

The `curl` command includes a flag forcing use of HTTP/2. The resource `nf-instances` is the list of network function instances registered in the NRF. The base URI is a NRF instance's address and port (7777). The API name has both the name of the SBI reference point `Nnrf` and the nickname `nfm` (for network function management, or NFManagement). The service operation that lists the network function instances currently registered with the NRF.

The following example uses some different options.

```
curl --http2-prior-knowledge GET https://nrf7.slice-v2x.opx.3gpp/nrf-nfm/v1/nf-instances/smf5-slicev2x
```

Here the server is a NRF addressed by a name `nrf7.slice-v2x.opx.3gpp`. The method is again `GET`, and the service operation `nf-instances/smf5-slicev2x` specifies that information about the SMF instance with instance ID `smf5-slicev2x` shall be retrieved.

That was just two of the API calls that can be made to 5G core NFs. To demonstrate the ways that the control plane uses SBI communication between network functions to make things work in the core, some examples of larger scale operations are presented below.

e.g.1 Network Function Registration

When a new instance of a network function (NF) is created it is required (by standards) to register with the network repository function (NRF). This is done by the new NF making a `PUT` API call to the NRF. Thus, in this API call the new NF is the consumer and the NRF is the producer. The API call has body that contains a description of the new NF. That information includes the kind of NF (e.g. AMF, SMF,…) and the services that the new NF can provide other NFs. The NRF is meant to add this document, along with a unique identifier for the new NF (NFID), to its database of NFs. If the NRF does add this new document then it sends a response message to the new NF.

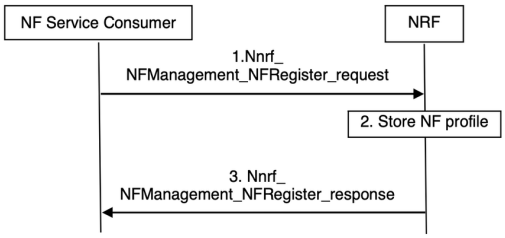


Figure 4.17.1-1: NF Service Registration procedure

That was a simple example of a operation in the core with just one API call (and response). The second example is more complex.

e.g.2 PDU Session Establishment

The following diagram outlines how a PDU session is *established* with time running downward. Since PDU sessions are central to 5G core function, this example is presented in some detail.

Note: This diagram and it's verbal description below provide an example of the complexity of specifications of the 5G network and its processes; the 3GPP specification of this initiation process alone takes up 21 pages; the diagram skips many steps, and the verbal description is minimal.

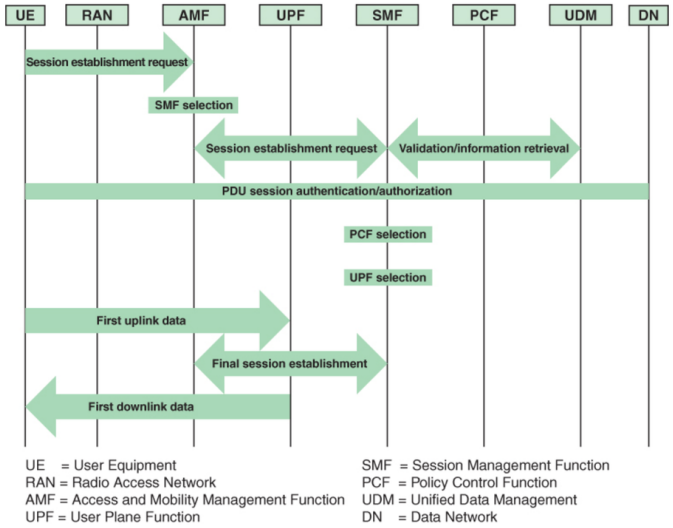


FIGURE 3.10 UE-Requested PDU Session Establishment

Verbally,

- A UE sends a Session Establishment Request to AMF (via N2 tunnel). Part of this request is a session ID so the session can be kept separate from other sessions.
- (Not shown:) The AMF performs Service Discovery; it makes a `GET` API call to the NRF to obtain a list of SMF instances that are available to manage the requested PDU session; the AMF receives a list of appropriate SMF instances via API response.
- The AMF tells one of those SMF instances to create of a PDU session by a `GET` API call. The SMF sends a confirming response message to the AMF and initiates the PDU session as follows.
- The SMF makes a `GET` call to the unified data management (UDM) to see if the UE's policies allow the UE to have the requested PDU session, and gets a response message.
- The SMF sends tunneling information, including tunneling endpoint IDs (TEIDs) to the UE, but this can not be done directly since there is no reference point between the UE and SMF; The SMF sends makes an API call to the AMF who then forward the message to the UE over N2 tunnel, and the AMF sends an confirmation of message sent response to the SMF.
- The SMF also needs to send tunneling information to a UPF (after choosing one, after making a call to the NFR to see which UPFs are available.). This is done via API call to a UPF.
- The SMF must also register the PDU session with the UDM via another `put` API call to the UDM; the UDM maintains a list of current PDU sessions.
- Both the UE and core are now prepared for SDFs to be created and placed in the PDU session.

The point here is that there are quire a few API calls in the process. In fact, quite a few are left out of this description and there are quite a few more that are optional. The following diagram of the same process shows more of the details and has the names of some of the SBA APIs shown; they start with e.g. `Nsmf`.

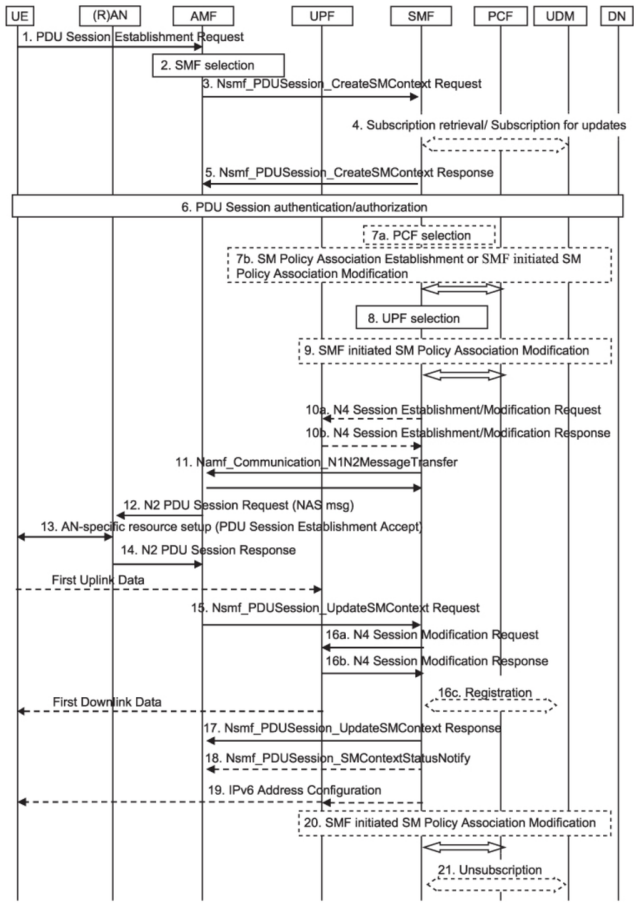
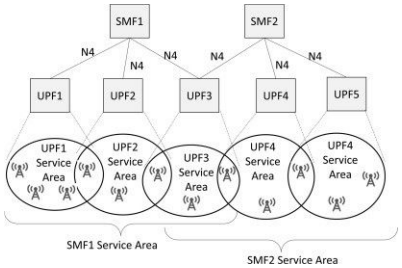


FIGURE 9.6 UE-Requested PDU Session Establishment

SMF and UPF Service Area

Some complexity in PDU session establishment was glossed over above; the SMFs and UPFs called to control and anchor PDU sessions, respectively, depend on the location of the UE. The UPFs are given a list of gNBs they serve, and the SMFs given a list of UPFs they serve. These lists are called UPF service area and SMF service area respectively. When the AMF serving the UE receives a PDU session establishment request, the serving AMF is aware of the location of the UE, and queries the NRF for the list of relevant SMFs. As you can in the diagram below, sometimes two SMFs are relevant. After the AMF chooses an SMF, the SMF chooses a UPF in a similar fashion, using the NRF to query for relevant UPFs.

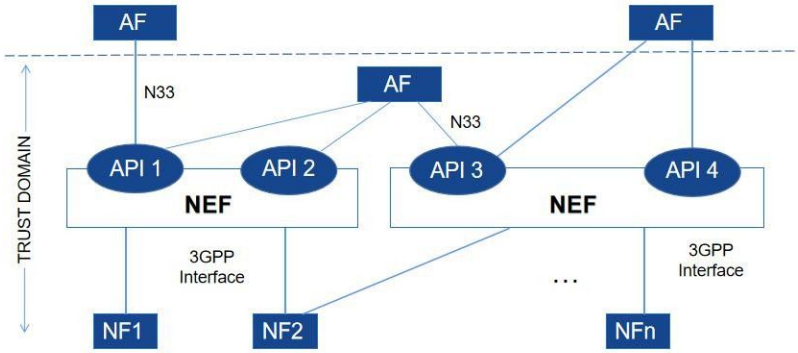


Note that UPF1 has physical connections to the gNBs in its UPF service area, but does not have a physical connection the gNBs in disjoint UPF service areas.

NEF and AF

API calls are practically without competition in exposing functionality of web services over the internet. 3GPP designed the 5G core to be able to interact with that ecosystem. In particular, the network exposure function (NEF) is meant to shrink the gap between mobile telecommunications cores and software development.

However software applications are not intended to interact with the network functions directly. Rather, the NEF is intended to be the sole point of contact. This exposure to application functions is called the north bound interface, a term borrowed from software defined networking (SDN).



An obvious question is why NFs other than the NEF have REST API endpoints if it would suffice to have only the north bound interface use REST APIs to meet the goal of connecting application development with mobile core development. The answer given by 3GPP is that this “service-based change design” was intended to be forward looking in having the internal workings of the core obey the same functional principles as the functional exposure. This way, 3GPP has more freedom in the future to modify core specifications without worrying about a boundary between exposed functionality and internal workings.

Recap

The 5G core uses HTTP/2 API calls for communication between core network functions. The reason for this change (relative to 4G) is to enable developers of applications to use the API calls for interactions between their apps and the 5G core.