Technical Disclosure Commons

Defensive Publications Series

September 2020

CONSUMER NF AUTHENTICATION AND SERVICE AUTHORIZATION USING AN SCP FOR AN ESBA BASED 5G CORE NETWORK

Ravi Shekhar

Dishant Parikh

Suyog Belsare

CONSUMER NF AUTHENTICATION AND SERVICE AUTHORIZATION USING AN SCP FOR AN ESBA BASED 5G CORE NETWORK

AUTHORS:
Ravi Shekhar
Dishant Parikh
Suyog Belsare

ABSTRACT

For a Third Generation Partnership Project (3GPP) Fifth Generation (5G) core (5GC) enhanced Service Based Architecture (eSBA) network, 3GPP Specifications identify a number of communication models. For example, the 3GPP specification 23.501 defines four communication models, encompassing direct and indirect modes, between a Consumer Network Function (NF) and a Producer NF. Within the indirect modes, a Service Communication Proxy (SCP) performs a range of important activities including among other things load balancing and load distribution of signaling traffic between for example different NF instances of a NF Set or different NF Services of a NF Service set. As a network scales and as traffic complexity grows, a range of challenges may arise encompassing excess or redundant signaling traffic, token invalidation, etc. Techniques are presented herein that address these challenges through the incorporation of an Open Authorization (OAuth) service into a SCP, yielding a number of improvements including, for example, a reduction in the overall number of signaling messages; significant optimization of the signaling between Consumer, SCP and NRF; simplification of the Auth call flow in scenarios where an initial producer instance is changed or multiple SCPs are deployed in a system; among others.

DETAILED DESCRIPTION

For a 3GPP 5GC eSBA network, 3GPP Technical Specification (TS) 23.501 defines four communication models, as shown in the Figure 1 (below), which can be used for a direct mode (Model A and Model B) or an indirect mode (Model C and Model D) of communication between a Consumer NF and a Producer NF. Model C and Model D are used for an indirect mode of communication where a NRF is used, by a Consumer directly

in Model C and by a SCP in Model D, for NF Discovery, Consumer Authentication and Service Authorization.

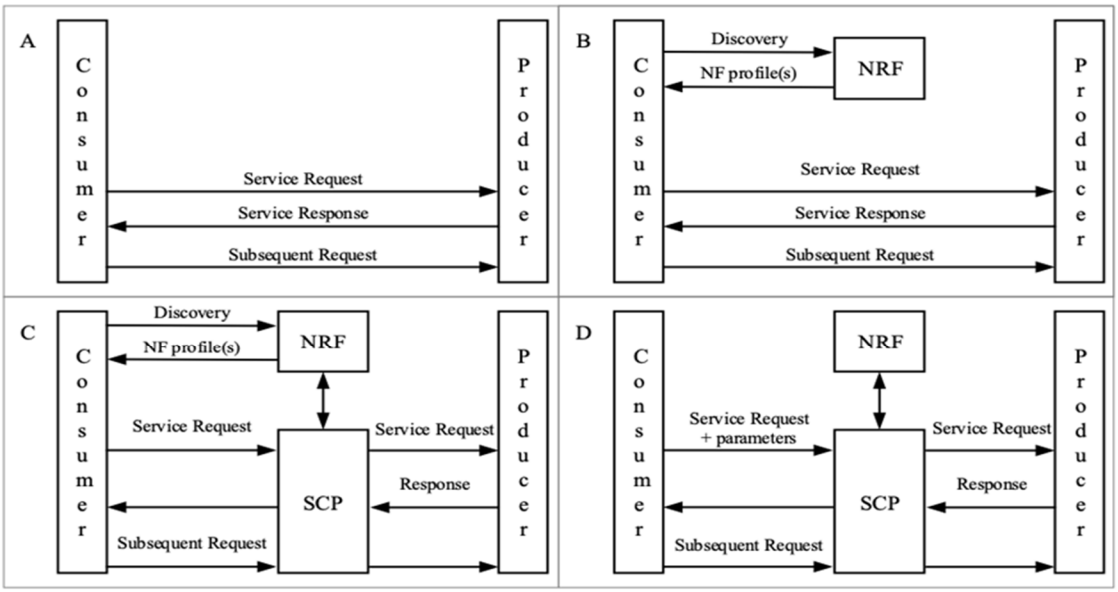

Figure 1: Communication Models for NF/NF Services Interactions

As detailed in various 3GPP Specifications, for an eSBA deployment the indirect modes of communication offer a range of benefits such as, for example, topology hiding, load-balancing between NFs of a NF set or between NF services within NF service set, etc. Considering all of the benefits that a SCP offers, an operator which plans to deploy an eSBA will likely choose between Model C and Model D or a mix of both.

In both Model C and Model D, the NRF is performing Discovery followed by an Auth function (as per the 3GPP standards, it plays the role of OAuth Server). In the current Architecture a NRF offers an `Nnrf_AccessToken` 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.

However, the current architecture has a number of gaps including, for example:

- For Model C, each Consumer interacts with a NRF multiple times (first for Discovery and then for Auth) before sending a service request message to a SCP. A SCP acts as a routing agent and it can decide to override the selected

Producer (for reasons of for example load balancing, overload control, operator policy, etc.) and in that case the SCP needs to fetch another Auth token from a NRF for a newly selected Producer instance. This will make the initial NF discovery and Auth, through the NRF, redundant.

- In large deployments, there could be multiple SCP instances running in a network - to cover different Public Land Mobile Network (PLMN) areas or slices. A multiple SCP deployment is already agreed to by the 3GPP body with a recommendation to select the next hop SCP through either a NRF or an operator policy. In such cases, the initial Auth token taken by Consumer will no longer be valid as a service request goes through multiple SCPs, which interact with different NRFs.

- If a SCP is enabled with features such as, for example, load-balancing, overload control support, etc. then there is a good chance that the SCP may override the a NRF-selected NF Producer. In such cases, the NRF-supplied Auth token for the consumer will not be valid.

A SCP plays an important role in load balancing of signaling traffic between different NF instances of a NF Set or different NF service instances of a NF Service Set. There are many cases where a SCP can override the NRF-discovered NF Instance in a NF set or NF Service in a Service set. For all such cases in which a SCP overrides a NRF decision, the initial token taken by NF consumer for the service request will become invalid. The techniques described herein solve this problem by incorporating an OAuth service inside a SCP.

Figure 2, below, provides a high level depiction of an OAuth service operating within a SCP. The SCP routing service will interact with the OAuth service internally to generate the authentication token before making any routing decision.

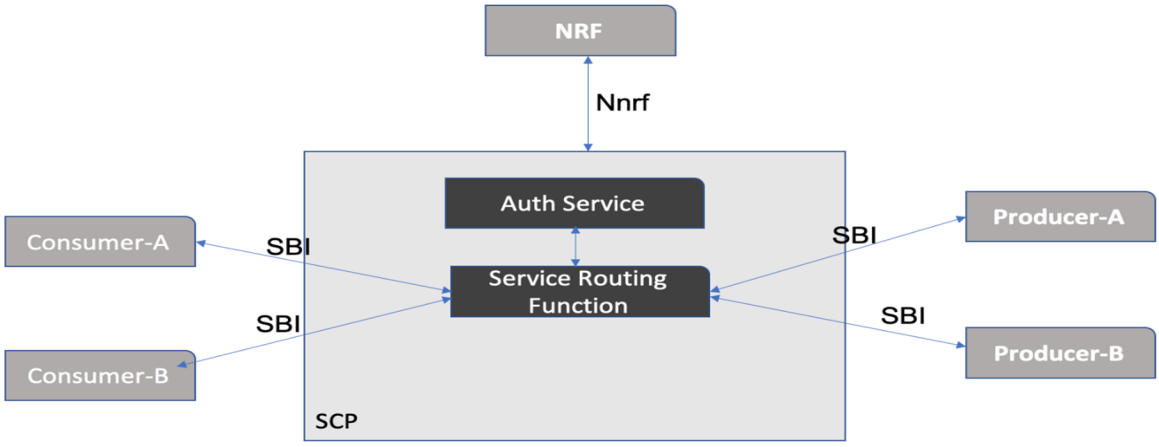

Figure 2: OAuth service within SCP

As shown in Figure 2, the `Nnrf` interface is only used in the case of Model D in which a SCP uses a NRF service for NF discovery and management services.

Figure 3, below, depicts the overall call flow where a SCP acts as an OAuth Server and provides an access token to a consumer before the consumer initiates a service request.

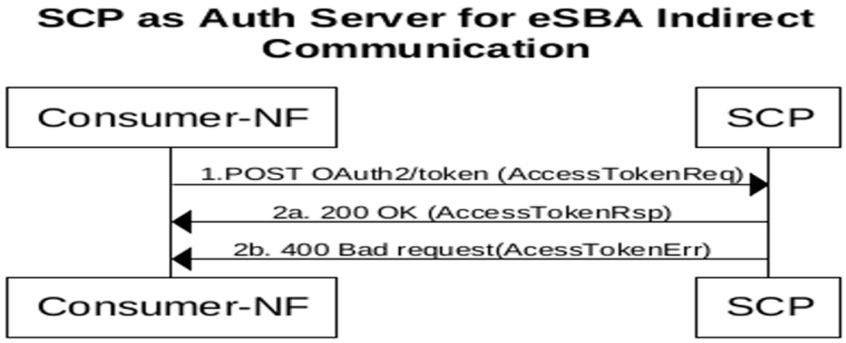

Figure 3: SCP as an OAuth Server for an eSBA Indirect Communication Model

The service operation as illustrated in Figure 3 is used by a NF Service Consumer to request an OAuth2 access token from the authorization server (i.e., a SCP). For example, at Step 1 as shown in Figure 3, the NF Service Consumer sends a Hypertext Transfer Protocol (HTTP) POST request to the "Token Endpoint" (i.e., a SCP). The OAuth 2.0 Access Token Request includes in the body of the HTTP POST request:

- An OAuth2 grant type is set to "client_credentials," having a scope set to the name of NF services that the Consumer is trying to access, such as the NF

Instance Id of the consumer, the NF type of the consumer and the producer, or the NF Instance Id of the producer for cases where an access token request is for a specific NF Service Producer (e.g., Model C).

- The Home and Serving PLMN IDs, if this is an access token request for use in a roaming scenario.

- Optionally, the NF Set ID of expected service producer instances may be included for cases in which an NF Set is configured in the network.

Step 2a illustrates a success case. For success cases a SCP returns "200 OK" with the payload body of the POST response containing the requested access token.

Step 2b illustrates a failure case. If the access token request fails at the SCP then it returns a "400 Bad Request" status code, including in the response payload a JavaScript Object Notation (JSON) object that provides details about the specific error that occurred.

It should be noted that if a NF consumer instance provides the NF Set information, then a SCP generates the tokens for the complete NF Set – instead of the selected producer instance. This will avoid generating another access token in case a SCP decides to later change the selected NF producer instance (due to for example overload, node failure, etc.).

For authentication in a Model C deployment, the consumer uses a NRF service for discovering producer instances and a NF Set to which the producer instances belong. Once the selection of a producer instance (and the NF Set that it belongs to) is made, it can generate a token for using the Auth service of an SCP, instead of using a NRF OAuth service. The SCP generates the access tokens for all of the NF Set to which the selected producer. In the future, if a SCP decides to switch to a different NF producer instance within the same NF Set then the consumer need not generate another token for communication.

For authentication in a Model D deployment, a SCP uses the discovery services of a NRF to select the producer instances and the NRF sends a list of available NF producer instances to the SCP. The SCP performs the final selection and, accordingly, generates the token for the whole NF Set to which the selected producer instance belongs.

For authentication in roaming scenarios or in environments in which multiple SCPs are deployed, the current architecture and call flow can be complicated. The consumer NRF would send the Auth request to a visited PLMN NRF, which would then forward the request to a Home PLMN NRF. After all of the message exchanges, the destination SCP may decide to change the producer instance, which would trigger re-allocation of access tokens at the consumer.

Through the techniques described herein, the destination SCP would generate the token so the case of regeneration of a token never arises. For a Model D deployment, the destination SCP can interact with a local NRF (in the same PLMN) to discover the Producer instance (or instance set) before generating the access token.

A SCP as an OAuth Server, as explicated through the techniques presented herein, provides a number of advantages including, for example:

- For indirect communication in which a SCP is making routing decisions, with assistance from a NRF for Model D, there is a good chance that a SCP would change the initial NF producer instance selected by NF consumer. For such cases, the access token taken by a NF consumer from an NRF will no longer be valid and extra messaging with a NRF will be required to get another token. This can be avoided if a SCP is the OAuth server.

- For multiple SCP deployments, the next hop could be a SCP instead of a NF producer in the message routing path. For such cases, the SCP needs to discover another NRF (in the realm of a destination SCP) and request another access token before forwarding the request. This can all be avoided if the destination SCP itself serves as an OAuth server and provides the source SCP with an access token.

- A reduction in the signaling load from a NRF, which would only be needed for NF Discovery and NF Management.

- Simplification of the security architecture and call flows for the cases in which multiple SCPs and NRFs are deployed.

In summary, techniques herein provide for an SCP to serve as an Auth server, which optimizes the signaling between a Consumer, the SCP, and an NRF. Techniques herein may further simplify the call flow in scenarios in which an initial producer instance is changed or multiple SCPs are deployed in a system.