NF Discovery Services

The Nnrf_NFDiscovery service queries the local NRF to enable an NF instance to discover services offered by other NF instances.

NF Discovery Service Operation

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-NRF
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**
First introduced.	2026.01

Feature Description

The NFDiscovery service operation enables an NF instance to discover other NF instances based on their IP address(es) or FQDN. It also enables an NF instance to discover NF services that matches a certain input criteria.

The NFDiscovery feature provides the following functionality:

- NRF performs authorization of the NF client requesting discovery of target NFs.
- Service discovery in the same PLMN.
- Discovery based on the following query parameters:
  - service-names
  - target-nf-type
  - snssais
  - dnn
  - tai
  - target-plmn-list
  - preferred-locality
  - supi
  - group-id-list
  - routing indicator
  - gpsi
  - external-group-identity
  - data-set
  - target-nf-instance-id
  - target-nf-fqdn
  - requester-nf-instance-fqdn
  - amf-region-id
  - amf-set-id
  - guami
  - ue-ipv4-address
  - ip-domain
  - ue-ipv6-prefix
  - requester-snssais
  - requester-plmn-list

    - access-type
    - supported-features
    - required-features
    - smf-serving-area
    - pgw-ind
    - pgw
    - dnai-list
    - upf-iwk-eps-ind
    - plmn-specific-snssai-list
    - nsi-list
    - limit
    - max-payload-size
    - pdu-session-types

Note The syntax of IEs and NRF response to the requests is based on 3GPP TS 29.510, Table 6.2.3.2.3.1-1.

- Discovery based on weight capacity configured for a profile.
- Discovery based on attributes of the following logical entities:
    - UDRInfo
    - UDMInfo
    - AUSFInfo of NFProfile
    - FQDN (Fully Qualified Domain Name)
- The requester-nf-instance-fqdn refers to user FQDN. To match the discovery request, this parameter must be present in allowedNfDomains field of services inside NFProfile.

How it Works

The discovery request contains multiple filter parameters based on which the NF Profiles, that matches the filter, are selected. The DB is queried with all the target profile filter parameters with an AND condition. If an attribute has multiple values (that is, list) in the discovery request, then the profile is selected that has at least one value from the multiple values in the list.

After querying the DB, the profiles are further filtered out to check if the service can be discovered in the PLMN or if it can be discovered by the sourceNfType, and so on.

NRF REST Endpoint

1. The NRF Rest endpoint receives the NFDiscovery request over HTTP2/JSON. The HTTP method is GET and the URL is {apiRoot}/nnrf-disc/v1/nf-instances?<query-parameters> with no body.

   Note: The default value, /root, is a configurable parameter.

2. On receiving the request, it is validated to check if the parameters and their value and type are allowed for discovery. If the validation fails, the response is sent with status 400 (BAD REQUEST).
3. The NFDiscovery request is transformed into protobuf format and sent toward the service engine for processing.
4. The service engine sends a response toward the endpoint after processing the request. The response is then transformed from protobuf-based format to JSON-based format.
5. The response is then checked for the response code; if it’s a success response and the response contains the list of profiles, then the profiles are sent back in the response body with the status code as 200 (OK). Else, if it’s an error code, the error code is sent back to the client along with problem details.

NRF Service Engine

1. The NRF Service Engine receives the protobuf-based request from the REST endpoint through the IPC system. The discovery request is then sent to the request-processor thread pool for further processing.
2. The NRF Engine gets all the query-parameters and prepares the query to the Datastore service.
3. If the request contains the Unique key, the engine queries the DB with the key.
4. If the request does not contain the Unique key, the engine queries the DB with the combination of non-Unique keys.
5. The engine then creates a new DBRecordFilter request containing all the non-Unique keys and triggers a Datastore query.
6. The Datastore service returns the list of NFProfiles matching the filter criteria.
7. The NF Services are then filtered out based on whether it can be discovered in the source PLMN, discovered by the source NF.
8. The resulting NFProfiles are sent back to the client with success code of 200.
9. If the source NF is not allowed to discover the NF service, response is sent back with code 403.
10. In case of errors, response is sent back with error code 500 and problem details.

Discovery Request

The processing of Discovery request follows the following sequence of steps:

1. The profile DB is first queried based on the Unique or set of non-Unique keys from the request. If the Unique key is present, it’s used to query the DB, else the combination of non-Unique keys is used.
2. After the NFProfiles are retrieved based on the query, the NFProfiles and NFServices are filtered out from the records.
3. The NFProfile retains only those services which are requested by NFClient.

4. Records are filtered based on allowed PLMN, NF Type, and NSSAI for requested service.

Discovery Key

The following lists the details of Discovery API key:

- The query should contain either the Unique key or at least one non-Unique key.
- In this release, Unique key is not supported.
- The supported non-Unique key is: nfType

  If more than one non-Unique key is present in the query, those NF profiles are retrieved for whom all the parameters are matched.

Discovery of Profile based on Weight Capacity

This section describes how NRF performs Discovery of profiles based on weight capacity.

- NRF performs the weight capacity, as defined in IETF RFC 2782, on NF profile and NF services if the preferred-locality is provided as part of query.
- NRF does not filter the NF profile or NF service based on preferred-locality.
- NRF modifies the priority of the NF profile and NF services to a higher value (higher the value, lower the priority) for which the locality of the NF profile does not match the preferred-locality.
- NRF sorts NF profiles and NF services based on respective load, capacity, and priority, to ease NRF consumer to select the appropriate NF profile and NF service.

Call Flows

NFDiscovery Success Call Flow

This section describes the successful NF Discovery call flow.

Figure 1: NFDiscovery Success Call Flow

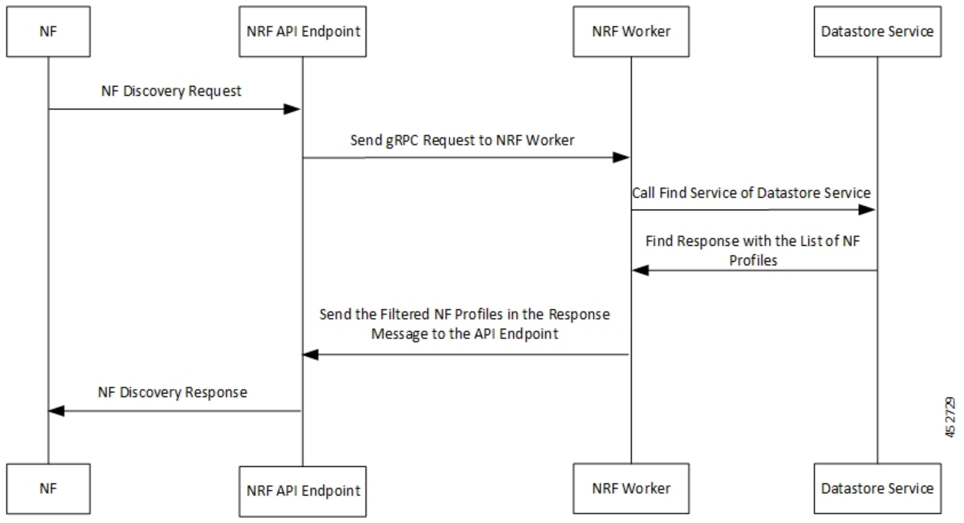
NF
NRF API Endpoint
NRF Worker
Datastore Service
NF Discovery Request
Send gRPC Request to NRF Worker
Call Find Service of Datastore Service
Find Response with the List of NF Profiles
Send the Filtered NF Profiles in the Response Message to the API Endpoint
NF Discovery Response
452729

Table 3: NFDiscovery Success Call Flow

Step	Description
1	The NF sends NF Discovery Request to the NRF API endpoint.
2	The NRF API endpoint transforms the REST request to gRPC. The NRF API endpoint sends gRPC request to the NRF worker.
3	The NRF worker decodes gRPC request and prepares DBRecordFilter message to be sent to the Datastore service. The NRF worker sends call find service to Datastore service.
4	The Datastore service responds to NRF worker with the list of NF Profiles.
5	The NRF worker filters the NF Profiles based on allowedPlmns/allowedNssais/allowedNfTypes. Then, the NRF worker sends the filtered NF Profiles in the response message to the NRF API endpoint.
6	The NRF API endpoint transforms the gRPC NFResponse message to REST-based response message. Then, the NRF API endpoint sends NF Discovery Response to the NF.