# Commonalities WG

08.12.2021
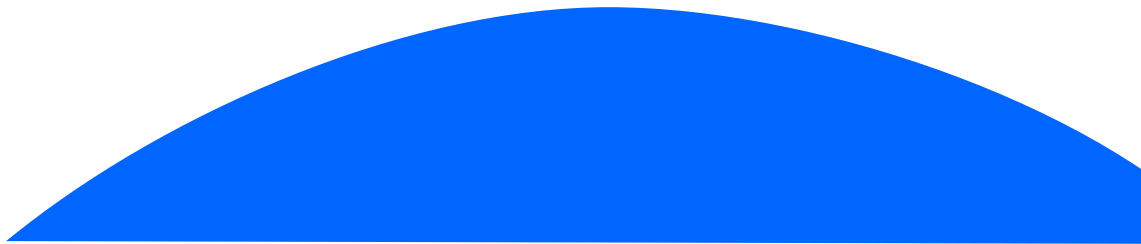
# According to APs from last call (10.11.2021)...

| Contributions | | |
|---|---|---|
| | Telefonica | Template for documenting Userstories |
| | Microsoft | Proposal for Standard resource description doc |
| | DT | Proposal for API documentation template<br>Authentication & Authorization concept doc |

# RECONNECT

Telefónica

# Actions for the 'commonalities' workstream (1/2)

1. **Document the principles/criteria that the in-scope service APIs shall fulfill -> OPAG collaboration**
   - The input from customers (e.g. hyperscalers) is a must. Customer shall declare their expectations on the usability of APIs.
   - Used as a basis to decide whether a SDO API is eligilible for direct usability or not.
2. **Draft a template to describe user stories.**
   - Business-driven use cases is what really should drive the API design + implemnetation work in the TGA Alliance.
3. **Decide on the service API spec format**
   - There is a tendency in the industry in going for OSA3
   - However, industry-wide validators and benchmarks (e.g. TMForum) keep using Swagger 2.0.
4. **Decide on the authentication and authorization solution**
   - Authentication mechanisms: OAuth2, OpenID, SAML, TLS, JWT Authentication,…
   - Authentication workflow: for some service APIs, the operator may be required to call an identity provider (IDP) service.

# Actions for the 'commonalities' workstream (2/2)

5. **Decide on the testing principles and tools for validation in telco network(s)**
   - Document this is a must, so as to ensure consistency.
6. **Decide on the criteria to go from step 3 (implementation) to step 4 (validation)**
   - This is important to manage expectations, especially on API developer side.
   - We do not need to have the API fully implemented – fleeing from traditional waterfall approach.
   - Example: At least [TBD] tests defined in the ATP passed.
7. **Decide on the criteria for API documentation publication**
   - Documentation should include: 1) complete API specification, with all methods and attributes; 2) API code and version; 3) succeed tests cases and related documentation
   - In order to manage expectations from customers and external fora/SDO, it is recommedable to publish 'sucess test cases' vs 'ATP' (i.e. all test cases planned for the service API to be 100% functional). The goal is to get this ratio increased for every new API release/version.

# Template for user story

● ● ● ● ●

# User stories (1/2)

User stories to be developed by product owner.

Start by developing user stories as a **user persona**(*), focusing on expectations from the point of view of an end-user. The best practice format for user persona-based stories is: "**As a** (Persona), **I want** (Need), **so that** (Goal)".

- Consider user stories for admin users vs non-privileged users.

Next, using the target and source systems you're integrating, **clarify the 'data journey'**

- Think about triggers for workflows: how and when does data need to be moved between the application and the service?
- Think about dependencies of data objects: does the data in underlying objects need to be regularly kept in sync with another system?
- Think about any parameters the user might need to configure or change. This is particularly important when building self-serve integrations for non-technical end users.
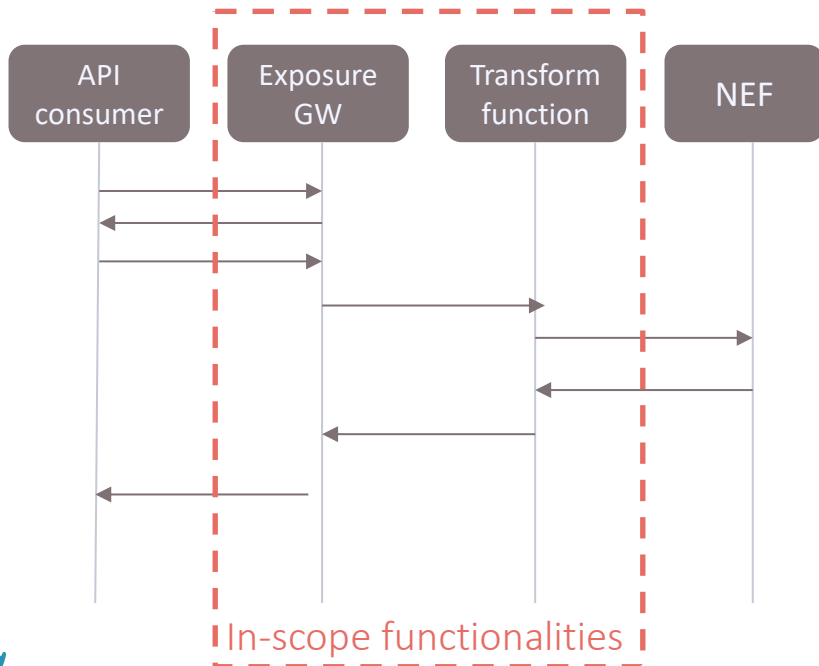
## Template for user story

| Item | Description |
|---|---|
| User story | "**As a** (Persona), **I want** (Need), **so that** (Goal)". |
| Actors and Roles | |
| Telecom resources | |
| Pre-conditions | |
| Begins when | |
| Step 1 (M/O/CM) | |
| Step 2 (M/O/CM) | |
| … | |
| Step N (M/O/CM) | |
| Ends when | |
| Exceptions | One of the mandatory steps fails. |
| Post-conditions | |

*NOTE(*): This is contrary to a user story based on system persona, which is designed to represent background system functions that do not require interaction from the end-user (i.e., elaborate on the behind-of-the-scenes integration tasks that are not user-centric).*
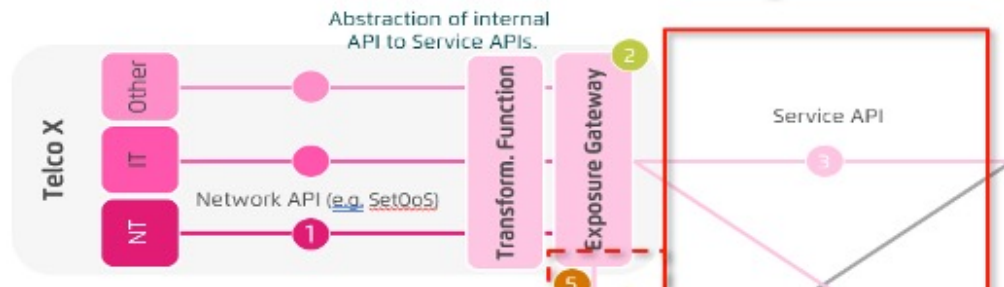
# User stories (2/2)

We can translate these steps into workflows, to see how underlying components (SDO-defined NF, transformation function, exposure GW) are involved.



In-scope functionalities

| Item | Description |
|---|---|
| User story | "**As a** (Persona), **I want** (Need), **so that** (Goal)". |
| Actors and Roles | |
| Telecom resources | |
| Pre-conditions | |
| Begins when | |
| Step 1 (M/O/CM) | |
| Step 2 (M/O/CM) | |
| ... | |
| Step N (M/O/CM) | |
| Ends when | |
| Exceptions | One of the mandatory steps fails. |
| Post-conditions | |



Abstraction of internal API to Service APIs.

Telco X — Other / IT / NT — Network API (e.g. SetQoS) — Transform. Function — Exposure Gateway — Service API

# Exposure Gateway – On the use of CAPIF

● ● ● ● ●

# CAPIF (Common API Framework)



**3GPP Architecture for Northbound APIs**

3rd Party — AF/AS(s)

N33

Service APIs

5GS — NEF

5GC interface

NF 1 • • NF x — NF y

NF specific interface

*API Set1: Capabilities of 5G network functions*
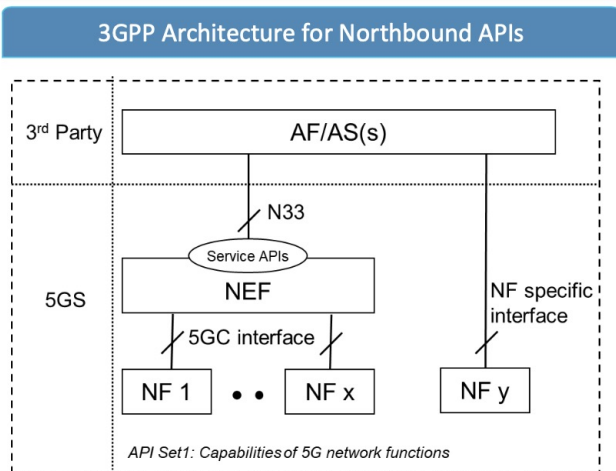
*NEF – Network Exposure Function*
*Service APIs = NB APIs*
*Framework APIs = CAPIF APIs*

**3GPP Architecture for Northbound APIs (with CAPIF)**

3rd Party — AF/AS(s)

CAPIF-2   CAPIF-1   CAPIF-2

Service Framework — Service APIs / Framework APIs / Service APIs / CAPIF

CAPIF-3/4/5

N33

5GS — NEF

5GC interface

NF 1 • • NF x — NF y

NF specific interface
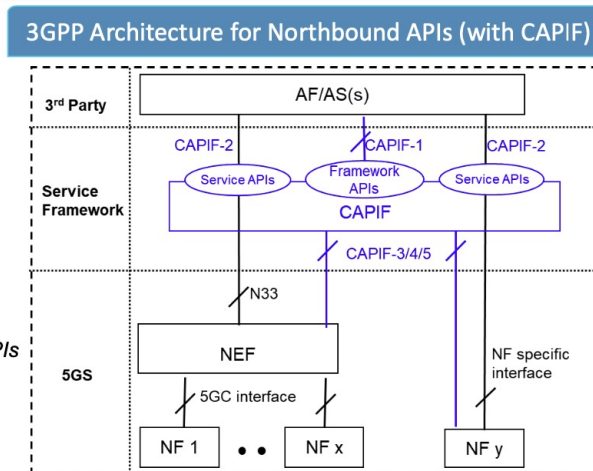
- 3GPP Core Network Capabilities are exposed as Northbound APIs, for consumption by the 3rd Parties outside MNO domain

- Northbound APIs are **configured offline** i.e. API discovery not supported, **dissimilar** security mechanisms, etc., **leading to fragmentation**

- Common API Framework (CAPIF) defines a framework to establish a **single and harmonized platform** (both EPS and 5GS) for all 3GPP Northbound APIs
  - On-boarding of API Invokers online is addressed
  - CAPIF provides common API registry on which unified discovery is supported
  - Common authentication and authorization mechanisms – so no fragmentation
  - Policy enforcement, auditing, accounting, support 3rd party domain, interconnection

# CAPIF - Architecture and Adoption
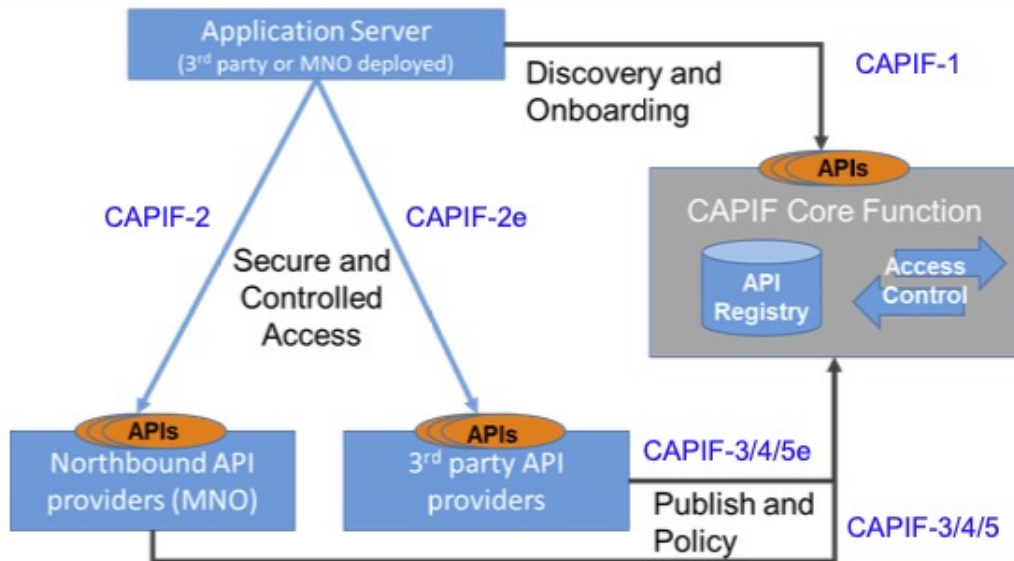


- CAPIF provides a unified exposure layer
  - Focused on API registration, discovery and access control
- Though defined by 3GPP (SA6 WG), the usability of CAPIF is applicable to any API.
  - CAPIF does not deal with API semantics
  - This means that CAPIF can be used for 3GPP APIs (e.g. SCEF/NEF APIs) but also non-3GPP APIs (including CAMARA APIs)

🌐 Standard API Framework
- Service Oriented Access
- IT/Web style API integration
- Controlled access to APIs

🌐 Business acceleration
- Unified API Gateway
- Supports API aggregation or API cloud

# On the use of CAPIF for Exposure Gateway

1. There is no yet a solution proposed for Exposure GW in CAPIC
   - So CAPIF can be a potential solution
   - Reference standard framework
2. TEF might bring a first release of CAPIF in 2022Q1
   - Outcome from EVOLVED-5G project (https://evolved-5g.eu) (https://www.youtube.com/watch?v=S7BhVV6PgpE)



EVOLVED-5G Overall concept
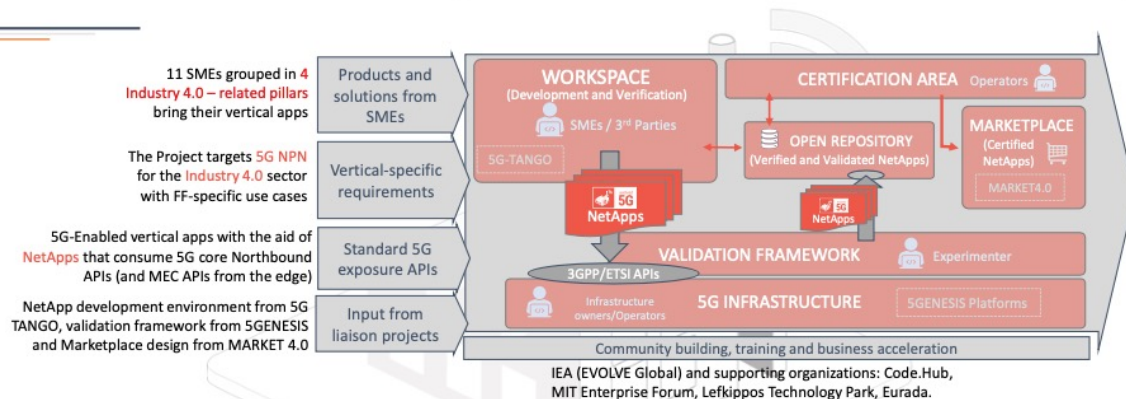
A NetApp:
- consumes network APIs (3GPP/ETSI APIs)
- Exposes service APIs (CAMARA APIs)

#RECONNECT