**SNAAPP use cases (draft)**

* Typical use case scenarios:
  + **Gaming**  
    An end user is playing a time-sensitive game communicating with a game provider's server. The end user wants to have a high-quality and low-latency communication for better service experience, so the end user asks the game server (AF or API invoker) to invoke the QoS API to change the end user's QoS. The end user chooses "Yes" when the UE shows a message saying whether the game server may change the user's QoS. After this authorization, the game server sends the service API invocation request to the API exposing function.
  + **Supporting 3rd party API invoker**  
    A small API provider exposes some service APIs to MNO subscribers. The API provider does not want to develop the authentication/authorization functionalities by itself and wants to farm out those functionalities to the MNO. The API provider may rely on the authentication/authorization mechanisms provided by the MNO and can concentrate on providing the service APIs.
  + **Supporting UEs as API invokers**  
    An end user wants to allow another end user to invoke service API on its behalf but wants to be able to do it contextually and does not want to provide the authorization equally to the server or to other users.

For example, the UE X end user chooses "every day during work hours only” to message requesting consent for UE Y to track its location. This consent however is not given to an application server, so if UE Y goes to sleep, an application server cannot continue tracking UE X on its behalf. In addition, such consent can be provided independent of application, i.e. UE Y cannot track UE X during this time via another app. (This description does not exclude the ability to provide authorization per-application and to application servers, but those aspects are assumed already provided for). This consent configuration applies for a specific duration and can be renewed, i.e. does not need to be provided at each API invocation.

In the same example, UE X end user can choose to provide consent for an entire G1 group of UEs (e.g. involved in a cooperative game), only if the invoker group member is located within a certain geographic area. When a group member makes the request from outside the area, the UE X end user wants to be triggered to provide consent on a case-by-case basis. This consent configuration applies for a specific duration and can be renewed.

* SNAAPP allows the end users of a particular UE to give authorization to the API invoker before API invocation relating to that UE. The UE is considered to have a user interface that can display the authorization dialogue. The process should differentiate between authorizing an application server and another UE (see SA1 requirements below)
* “Resource owner” is the end user, and the UE shows certain user interfaces to let the end user authorize or deny the API invocation.
* Unlike the user consent in UDM/UDR, SNAAPP aims to realize at least near real-time (Solution #4) authorization; (real-time authorization is also a possibility (Solution #3)). At the same time, SNAAP maintains the ability to provide persistent “authorization configurations” which may be renewed periodically. “Authorization configurations” can include settings such as “always allow/deny”, “conditionally allow/deny”, “ask for consent at each API invocation”, etc.
* The solution should not require developers to learn new mechanisms in addition to Oauth 2.0 (which is already used for CAPIF, anyways). Addition of further flows is considered acceptable, though.
* Two different API invocation scenarios (corresponding to SA1 requirements):
  + UE-originated API invocation (Key Issue #1, TR 23.700-95)
    - [SA1 requirement: *The 5G system shall be able to provide a UE with secure access to APIs (e.g. triggered by an application that is not visible to the 5G system), by authenticating and authorizing the UE.*]
  + AF-originated API invocation (Key Issue #2, TR 23.700-95)
    - [SA1 requirement: *The 5G system shall be able to provide a third-party with secure access to APIs (e.g. triggered by an application that is visible to the 5G system), by authenticating and authorizing both the third-party and the UE using the third-party's service.*]
* End user wants to permit a certain application to change its QoS, obtain user location, etc., but does not permit others to do so.