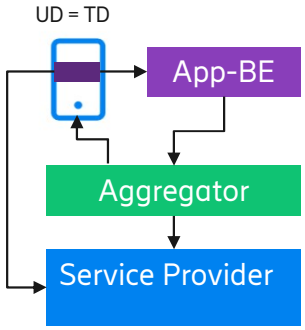
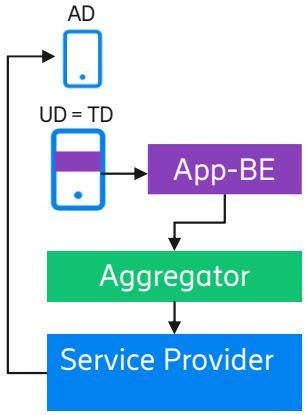
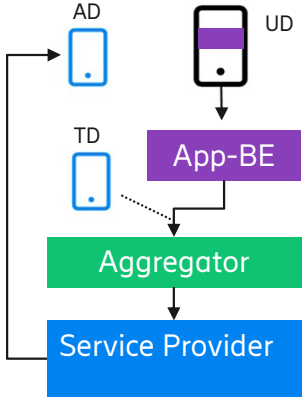
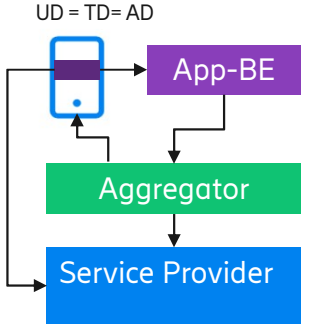
CAMARA ISSUE 224

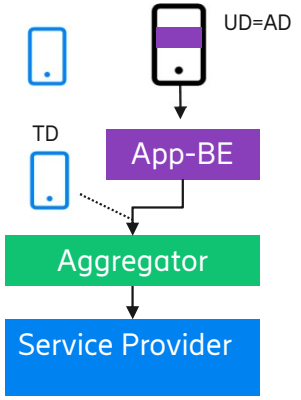
How and when to bring up Consent Capture dialog











Option 1: Frontend flow/OIDC Authorization code flow

- App requests access code for IP, scope, purpose (potentially other login hint)
- IP is translated to MSISDN
- Consent check happens potentially resulting in lack of consent
- If prompt = none, AuthZ returns error to API Invoker m(see RFC 6749, 4.1.2.1)
- If prompt=consent, AuthZ server should prompt the end user for consent (**INDEPENDENT** of consent is there or not)

Option 2: CIBA flow

- App requests access code for MSISDN, scope, purpose
  - MSISDN must be target device
- Consent check happens potentially resulting in lack of consent
- Asynchronous notification shall be sent to authorizing party
- Authorizing party potentially configures consent by bringing up consent capture app of CSP
- API client polls for access token and might receive timeout

Option 3: Out of bounds consent capture

- App checks for consent status for [MSISDN of target device, purpose, scope] with CSP
- CSP returns consent status & consent capture URL
- App decides to bring up consent capture dialog

Legend:
UD = User Device
TD = Target Device of API call
AD = Authorization Device

Technical Considerations – Option 3

- Consent Info API must be called on exposure platform of home CSP of target device
- Proposal to use CIBA flow for calling Consent Info API to allow for use cases UD = TD and UD ≠ TD

- Consent Info API is treated exactly as any other NW API
  - Application must sign up for usage
  - Service Provider must configure privacy treatment, e.g. legal basis to be applied
  - App must acquire access token to call the API
  - Access token generation at service provider must consider consent configuration of TD
    - Sub claim in access token refers to TD
  - Access token must be used on API call

Find my Friend: Mary wants to locate Jane