SMART ON FHIR OG FØRERRETT

norskhelsenett

FØRERRETT

- Førerrett benytter felleskompontene:
  - Skjemautfyller, benyttes både av Helsenorge og SMART-applikasjonen
  - Skjemakatalog, benyttes både av Helsenorge og SMART-applikasjonen
  - Helsenorges Dokumentarkiv for å kunne dele egenerklæringen med fastlege
  - HelseID for pålogging av Helsepersonell
  - SMART-rammeverk (egentlig kun en spesifikajson)

- Førerrett består av:
  - Egenerklæringsskjema som innbygger kan fylle ut før oppmøte hos fastlege
  - SMART-applikasjon: Helseattest som fastlege benytter ved undersøkelse av innbygger
  - SMART-støtte og FHIR-API på EP-siden
  - SVV-API for mottak av konklusjon fra helseattesten

Arkitektur

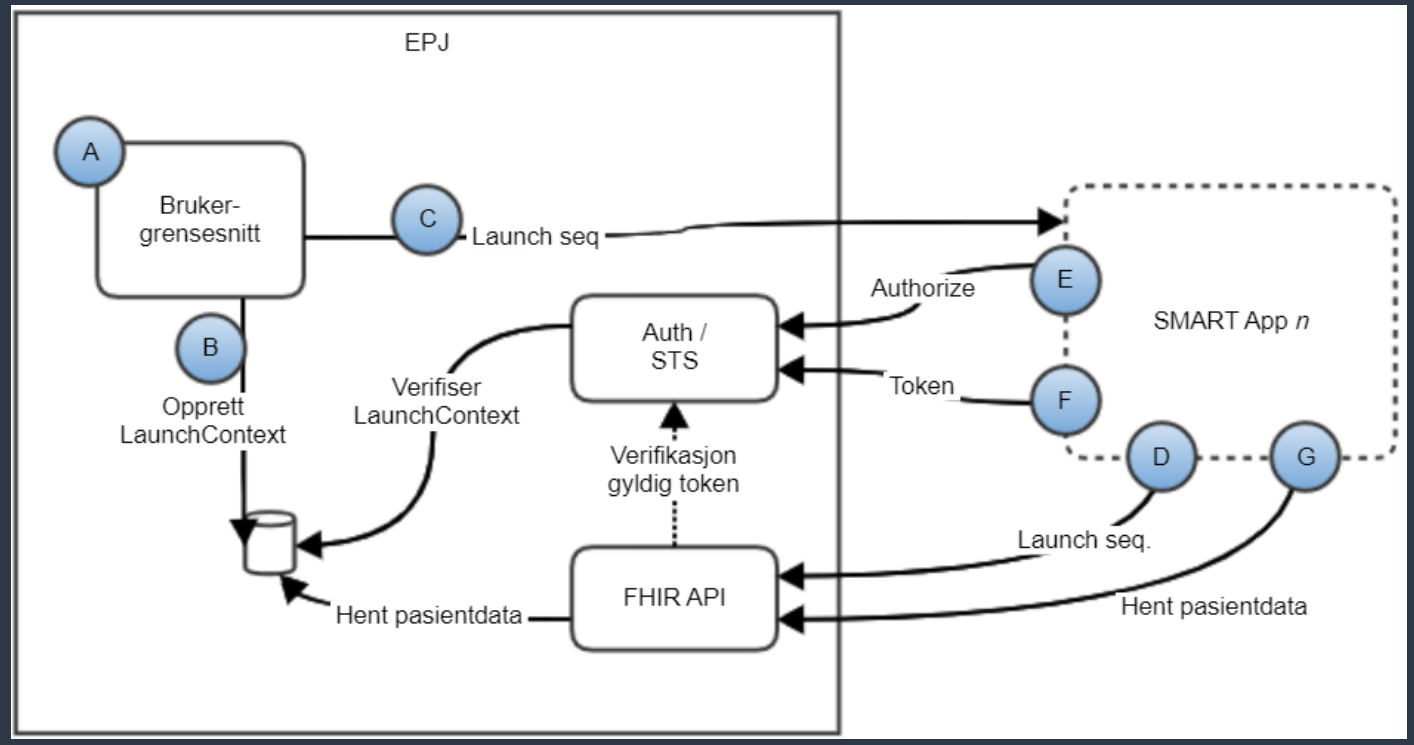
SMART APP LAUNCH FRAMEWORK

– SMART gir tredjepartsapplikasjoner autorisert tilgang til data i elektroniske pasientjournaler via en pålitelig og sikker autorisasjonsprotokoll. Applikasjonene kan starte som en del av eller utenfor brukergrensesnittet til et EPJ-system. Rammeverket støtter per i dag fire bruksscenarioer beskrevet i Argonaut prosjektets fase 1. Argonaut prosjektet tar for seg følgende bruksscenarioer:

1. Applikasjoner for pasienter som kan starte frittstående
2. Applikasjoner for pasienter som kan starte fra en portal
3. Applikasjoner for klinikere som kan starte frittstående
4. **Applikasjoner for klinikere som kan starte i en EPJ eller portal**

Arkitektur

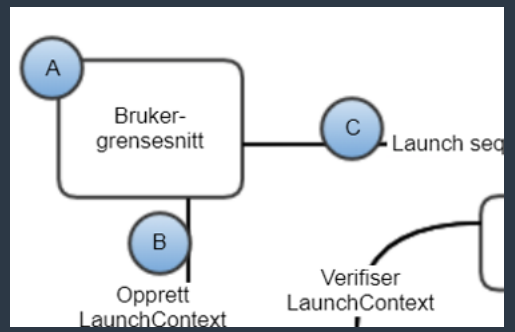
SMART APP LAUNCH FRAMEWORK - OPPSTARTSFLYT



Arkitektur

A. STARTER SMART-ENABLED WEBAPPLIKASJON

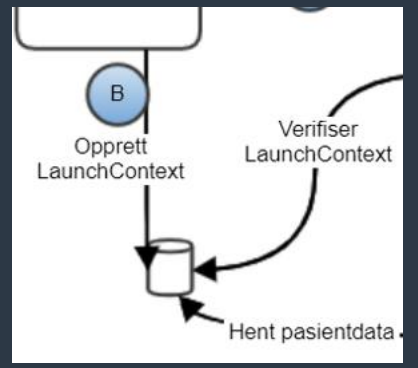
- Klassisk desktop EPJ benytter integrert nettleser for å starte en webapplikasjon med SMART App Launch Framework-støtte
- Web EPJ kan starte applikasjonen i samme vindu eller åpne ny nettleserfane. (utfordringer ved kommunikasjon med HelseID dersom man benytter Iframe)
  - X-Frame-Options
    - **HelseID støtter kun X-Frame-Options: SAMEORIGIN**
    - **Ikke avklart p.t.**



Arkitektur

B. LAUNCHCONTEXT OPPRETTES

- EPJ oppretter en LaunchContext som tildeles en unik identifikator og assosieres med client_id for SMART-applikasjonen. Konteksten består av:

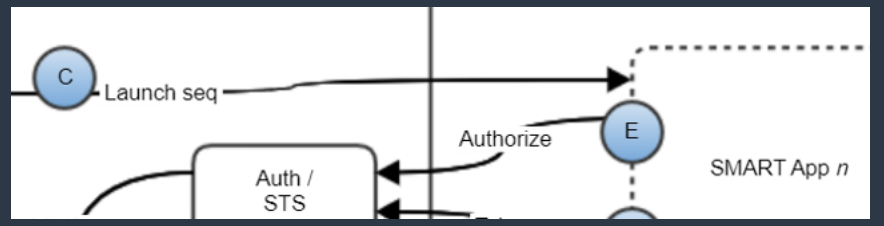


Parametere		
patient	Påkrevet	Den logiske logiske ressursidentifikatoren for pasienten
practitioner	Valgfri	Den logiske ressursidentifikatoren for helsepersonellet som benytter applikasjonen
encounter	Valgfri	Den logiske ressursidentifikatoren for konsultasjonen

Arkitektur

C. LAUNCH-SEKVENSEN INITIERES

- Launch-sekvensen initieres ved å kalle web-applikasjonen GET https://app/launch?iss=https%3A%2F%2Fehr%2Ffhir&launch=xyz1



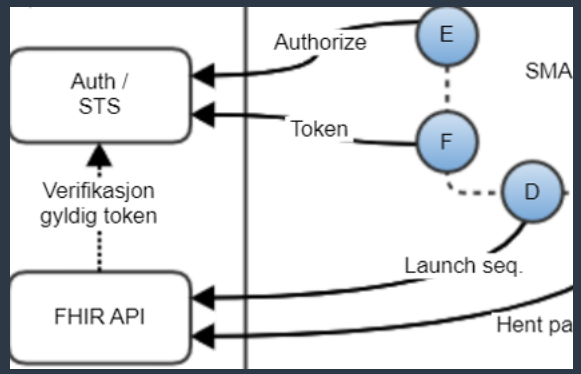
Parametere		
iss	Påkrevet	Url til EPJens FHIR endepunkt. Web-applikasjonen benytter dette endepunktet for å skaffe ytterligere detaljer EPJen, inkludert URLen til autorisasjonserveren
launch	Påkrevet	Ikke-transparent identifikator for denne oppstartsekvensen. Dette parameteret kommuniseres tilbake til EPJen på autorisasjonstidspunktet.

Arkitektur

D. APPLIKASJON MOTTAR LAUNCH-NOTIFIKASJON

- Ved mottak av launch-notifikasjonen forespør applikasjonen utstederen (iss) sitt /metadata/ endepunkt eller .well-known/smart-configuration.json endepunkt. Disse inneholder URLene til EPJen sitt authorize og token endepunkt.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "authorization_endpoint": "https://ehr.example.com/auth/authorize",
  "token_endpoint": "https://ehr.example.com/auth/token",
  "token_endpoint_auth_methods_supported": ["client_secret_basic"],
  "registration_endpoint": "https://ehr.example.com/auth/register",
  "scopes_supported": ["openid", "profile", "launch", "launch/patient", "patient/*.*", "user/*.*",
"offline_access"],
  "response_types_supported": ["code", "code id_token", "id_token", "refresh_token"],
  "management_endpoint": https://ehr.example.com/user/manage
  "introspection_endpoint": https://ehr.example.com/user/introspect
  "revocation_endpoint": "https://ehr.example.com/user/revoke",
  "capabilities": ["launch-ehr", "client-public", "client-confidential-symmetric", "context-ehr-
patient", "sso-openid-connect"]
}
```

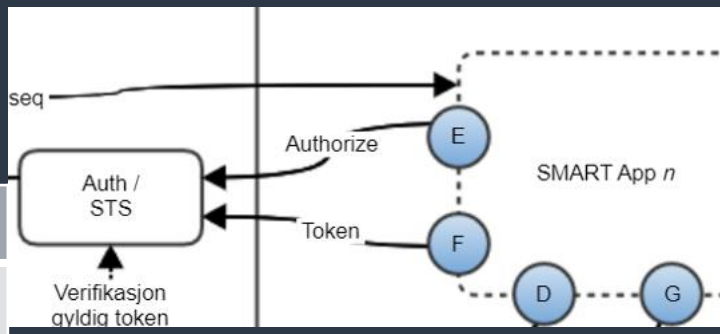


Arkitektur

E. APPLIKASJONEN UTFØRER EN FORESPØRSEL MOT AUTHORIZE ENDEPUNKTET

- Applikasjonen gjør en forespørsel mot authorize endepunktet på EPJens autorisasjonsserver med følgende parametere:

Parametere		
response_type		Fiksert verdi ‘code’
launch	Påkrevet	Samsvarer med den mottatte launch-parameteren fra EPJ.
scope	Påkrevet	Angir hvilken aksess den trenger til helsedata. Inkluderer scope som: • patient/*.read • openid • fhirUser • launch



Arkitektur

E. URL-EKSEMPEL AUTHORIZE OG REDIRECT

Kall til authorize-endepunkt:

```
GET https://ehr/authorize?
  response_type=code&
  client_id=app-client-id&
  redirect_uri=https://app/after-auth&
  launch=xyz123&
  scope=launch+patient/Observation.read+patient/Patient.read+openid+fhirUser&
  state=98wrghuwuogerg97&
  aud=https://ehr/fhir
```

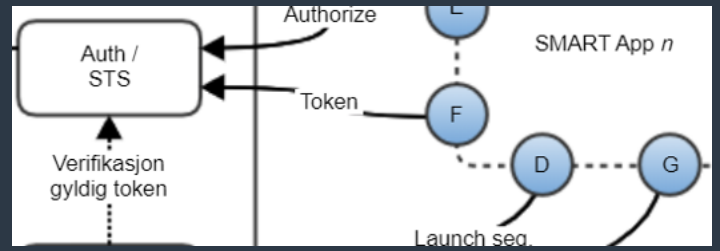
Redirect tilbake til SMART-applikasjon etter suksessfull autorisasjon

```
GET https://app/after-auth?
  code=123abc&
  state=98wrghuwuogerg97
```

Arkitektur

F. APPLIKASJONEN VEKSLER INN AUTORISASJONSKODEN I ET TILGANGSTOKEN

- Etter at applikasjonen har mottatt en autorisasjonskode veksles denne inn i et tilgangstoken via et HTTP POST kall til EPJens autorisasjonsserver token-endepunkt



Parametere (Request)		
grant_type	Påkrevet	Fiksert verdi ‘authorization_code’
code	Påkrevet	Autorisasjonskoden mottatt fra autorisasjonsserveren sitt authorization-endepunkt

Parametere (Response)		
access_token	Påkrevet	Tilgangstoken utstedt av autorisasjonsserveren
token_type	Påkrevet	Fiksert verdi ‘bearer’

F. EKSEMPEL PÅ RESPONS FRA TOKEN-ENDEPUNKT

```
{
  "access_token": "i8hweunweunweofiwweoijewiwe",
  "token_type": "bearer",
  "expires_in": 3600,
  "scope": "patient/Observation.read patient/Patient.read",
  "intent": "client-ui-name",
  "patient":  "123",
  "encounter": "456“
}
```

G. APPLIKASJONEN HAR NÅ TILGANG TIL HELSEDATA

- Applikasjonen har nå tilgang til helsedata via det mottatte tilgangstoken.
- Endepunktet applikasjonen mottok i iss-parameteret ved oppstart er url til FHIR-server
- access_token mottatt fra token-endepunktet benyttes for autorisasjon

G. EKSEMPEL PÅ FORESPØRSEL

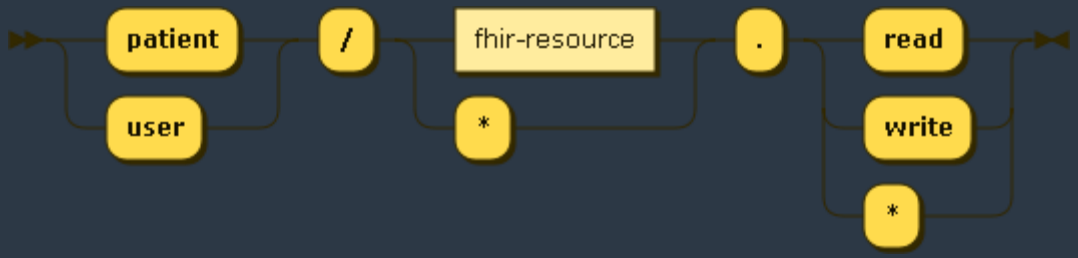
```
Forepørsel:
GET https://ehr/fhir/Patient/123
Authorization: Bearer i8hweunweunweofiwweoijewiwe
----

Respons:
{
  "id": "123",
  "resourceType": "Patient",
  "identifier": [
  {
    "system": "urn:oid:2.16.578.1.12.4.1.4.1",
    "value": "210377xxxxx«
  }],
  "name": [
    {
      "use": "official",
      "family": [
        "Myhra«
      ],
      "given": "Kenneth"
    }
  ],
  (...)
}
```

Arkitektur

SCOPES OG LAUNCH-KONTEKST

- SMART App Launch Framework benytter OAuth scopes for å kommunisere og forhandle krav til tilgang. I tillegg til hvilke scope som er satt i tilgangstokenet er tilgang i tillegg begrenset til de privilegiene eller autorisasjonen brukeren har tilgang til. Generelt benyttes scopes for tre typer data:
  - Kliniske data
    - patient/*.read
    - patient/observation.read
  - Kontekstuelle data
    - launch
  - Identitetsdata
    - openid
    - fhirUser



Arkitektur og infrastruktur

TILGANGSSTYRING OG KOMMUNIKASJONSFLYT

- SMART App Launch Framework benyttes for oppstartsekvens og tildeling av tilgang
- FHIR REST API for å hente ut informasjon
  - HelseAPI
    - Patient
    - Practitioner
- HelseID
  - pålogging av helsepersonell
  - klient-autentisering ved kommunikasjon med Skjemakatalog
- Konklusjon etter utfylt helseattest
  - REST-API Statens vegvesen (SVV)
  - Tilgang autoriseres ved helsepersonellets HelseID-token
- Helsenorge Dokumentarkiv
  - Førerrett App defineres som en intern applikasjon av Helsenorge
  - Tilgang autoriseres ved system til internt API token