Føderert API management

2021-02-10

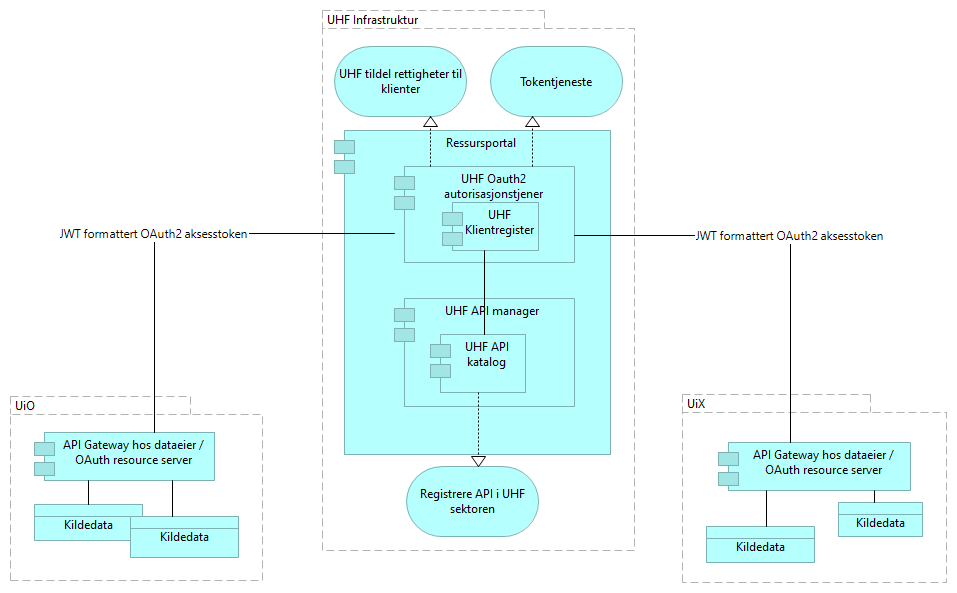
Føderert API Management innebærer at et domene (føderasjon) kan gi brukere i andre domener man har et tillitsforhold til (ofte kalt fødereringspartnere) tilgang til sine egne data gjennom APIer. Dette er et prinsipp for API Management som har etablert seg de siste årene. Det er flere grunner for dette, blant annet at store virksomheter i dag gjerne har virksomhetsdata lagret distribuert og gjerne fordelt mellom systemer som kjører på lokale servere (on-premise) og i skybaserte-løsninger som kjører på egen infrastruktur.

Det sterke politiske fokuset på deling av data gjennom orden i eget hus og nasjonale og sektorvise digitaliseringsstrategier understøtter også behovet for et API Management regime som understøtter deling av data til 3. part for innovasjon, og fokus på sammenhengende brukerreiser (eksemplifisert ved de syv livshendelser og tjenestekjeder) som forutsetter at data kan deles mellom virksomheter som betjener slike brukerreiser.

Virksomhetene i UH/F har mange fellestrekk og noen fellestjenester på tvers, men deres autonomi gjør at de også er ulike på enkelte områder, f.eks. når det gjelder teknisk infrastruktur. Dette understreker behovet for en føderert API Management løsning som er mer fleksibel mtp. tekniske løsninger og produktvalg lokalt.

Designen avbildet er laget for å støtte følgende karakteristikk:

* Intstitusjonene er sikret råderett over egne data gjennom tilgangstjeneste i resssursportalen og API gateway hos institusjonen som kan styre og holde oversikt over datatilgang
* Institusjonene har valgfrihet over API Gateway såfremt funsjoner og grensesnitt er ivaretatt
* Deling av data på tvers av institusjoner er støttet gjennom felles ressursportal



Føderert API management

Elementer i view for Føderert API management

|  |  |
| --- | --- |
| Element | Beskrivelse |
| UHF Infrastruktur <grouping> |  |
| Tokentjeneste <application-service> | Tjeneste som utsteder sikkerhetsbilletter. Sikkerhetsbillett utstedes basert på tildelte rettigheter og eventuelle representasjonsforhold. |
| Ressursportal <application-component> |  |
| UHF Oauth2 autorisasjonstjener <application-component> | Autorisasjonstjeneter realiserer tilgangskontroll gjennom å utstede OpenID Connect eller OAuth 2 grants og tokens. En autorisasjonstjener er også brukt til å realisere tilgangs policy.  Scope (rettighet) og Audience (identifikator eller ressurs-server f. eks. FS)  Per i dag er Scopes er grovkornet Rich authorization requsts (ny spesifikasjon) for å utvide scope begrep i IETF underenhet og behandlingssted, behandlingsgrunnlag og roller jobbes det med. |
| UHF Klientregister <application-component> | Oversikt over alle klienter som er blitt tildelt tilgang til ressurser. |
| UHF API manager <application-component> | Administrativ komponent som kontrollerer policy og livssyklus for definisjon og forvaltning av API. Tilganger gjennom et API defineres i API manager og tildeles av autorisasjonstjener. Ansvaret til API manager inkluderer:  \* Sentralisert API administrasjon og forvaltning av API-katalogen \* Håndtering av registrerings- og introduksjonsprosesser for API utviklere \* Håndtere livssyklusen til et API  Ivareta registrering, forvaltning og og utvikling av klientrettigheter  Hva tilbyr API av datamodell Hvordan ønsker API å sikre seg selv En klient må forholde seg til |
| UHF API katalog <application-component> | Oversikt over alle API som er registrert |
| Registrere API i UHF sektoren <application-service> |  |
| UHF tildel rettigheter til klienter <application-service> |  |
| UiX <grouping> |  |
| Kildedata <data-object> |  |
| API Gateway hos dataeier / OAuth resource server <application-component> | API gateway beskytter ressurser og monitorerer tilgang. Den håndhever tilgangsstyring ved å validere aksesstokens og utfører eventuell fin-granulert tilgangskontroll. Den håndhever også volumbegrenseninger og samler data om bruken av APIer (logging).  Komponent som beskytter ressursene og monitorerer tilgang inklusivt:  \* Validere tilgangstoken \* Beskytte mot inntrenging og andre trusler • Håndtere volumbegrensninger og andre abonnementsordninger • Håndheve tilgangsstyring • Samle inn data om bruken av API-er • [orkestrering mellom interntjeneser??]  Prinsipp fra Steiner: Oppgaver både sikkerhetsperspektiv og forvaltningsperspektivet Dersom gatway skal stå for sikkerheten, MÅ VÆRE SIKKER PÅ at API ikke kan nås utenom gateway Noen må passe VELDIG godt på gatewayen og holder den oppdatert.  Det må være et tydeligvalg om hvem (gateway eller API) har ansvar for sikkerhetsmessig. |
| Kildedata <data-object> |  |
| UiO <grouping> |  |
| Kildedata <data-object> |  |
| API Gateway hos dataeier / OAuth resource server <application-component> | API gateway beskytter ressurser og monitorerer tilgang. Den håndhever tilgangsstyring ved å validere aksesstokens og utfører eventuell fin-granulert tilgangskontroll. Den håndhever også volumbegrenseninger og samler data om bruken av APIer (logging).  Komponent som beskytter ressursene og monitorerer tilgang inklusivt:  \* Validere tilgangstoken \* Beskytte mot inntrenging og andre trusler • Håndtere volumbegrensninger og andre abonnementsordninger • Håndheve tilgangsstyring • Samle inn data om bruken av API-er • [orkestrering mellom interntjeneser??]  Prinsipp fra Steiner: Oppgaver både sikkerhetsperspektiv og forvaltningsperspektivet Dersom gatway skal stå for sikkerheten, MÅ VÆRE SIKKER PÅ at API ikke kan nås utenom gateway Noen må passe VELDIG godt på gatewayen og holder den oppdatert.  Det må være et tydeligvalg om hvem (gateway eller API) har ansvar for sikkerhetsmessig. |
| Kildedata <data-object> |  |