Anpassa tjänster till ny samverkans-arkitektur

Rapport

inera
Ett företag inom SKR

Innehållsförteckning

Revisionshistorik

VERSION	DATUM	FÖRFATTARE	KOMMENTAR
0.1	2023-02-17	Björn Beskow	Initial version
0.2	2023-03-06	Björn Beskow	Avstämning med uppdragsägare
0.3	2023-03-23	Björn Beskow	Preliminär version för granskning
0.4	2023-04-04	Björn Beskow	Mindre uppdateringar efter granskningskommentarer
1.0	2023-04-14	Björn Beskow	Version för presentation för ledningsgrupp
1.0.1	2023-04-28	Björn Beskow	Slutversion

1 Sammanfattning

1.1 Bakgrund

Ineras programråd har under 2022 prioriterat ett utvecklingsinitiativ inom utvecklingsramen för anpassning av fler tjänster till ny samverkansarkitektur (REST/ FHIR).

1.2 Uppdragsbeskrivning

Genom det här uppdraget så tas en sammanställning och beskrivning fram av behov som påverkar Ineras tjänster samt en portföljanalys med samband och beroende om vad som måste göras. Uppdraget kommer också göra en prioriteringslista om i vilken ordning som utvecklingen bör ske och där det är möjligt så kommer också ett tids- eller kostnadsestimat göras.

1.3 Sammanfattning av resultat

Nya stödtjänster behöver tas fram för att stödja samverkansfederationer enligt T2:

- Federationskatalog
- Tjänstekatalog

Nya förmågor krävs även för att stödja Inera som producent av APIer:

- Biljettutfärdande enligt OpenID Connect och OAuth 2.0
- API Gateway för realisering av gemensamma förmågor:
    - Åtkomstkontroll
    - Tjänstidentifiering och lastbalansering
    - Loggning, monitorering och statistik
    - Trafikbegränsning
    - Mellanlagring
- Process- och verktygsstöd för skalbar API hantering

1.4 Rekommendationer

1. Utveckla en minimal Tjänstekatalog för Skriv till VIS behov
2. Ta fram konkreta realiseringsanvisningar för IAM referensarkitektur
3. Kravställ och realisera initial central OAuth 2.0 tjänst
4. Utvärdera, välj och realisera taktisk lösning för API gateway
5. Initiera en förstudie av lösning för syndikerad Federationskatalog
6. Initiera en förstudie av lösning för syndikerad Tjänstekatalog

7. Realisera fullvärdig central OAuth 2.0 tjänst
8. Ta fram konkreta realiseringsanvisningar för REST-baserade APIer
9. Initiera en förstudie av fullvärdigt verktygsstöd för API management

2 Introduktion

2.1 Om utredningen

Denna utredning har bedrivits av Björn Beskow under perioden januari till mars 2023, med David Ulfstrand som administrativ uppdragsledare. Kartläggning av behov har gjorts genom intervjuer med utpekade projekt och initiativ, samt genom löpande dialog med nyckelpersoner inom sektionerna för Arkitektur och Samverkansarkitektur.

Följande personer har bidragit med input till utredningen:

Deltagare	Initiativ/projekt
Anders Malmros	Samverkansarkitektur
Henrik Emilsson	1177 eTjänster, Skriv till VIS
Martin Svensson	Arkitektur, Skriv till VIS
Björn Hedman	Samverkansarkitektur
Andreas Tell	SDK
Anders Larsson	Arkitektur, 1177 eTjänster
Johan Zetterström	1177 Rådgivningsstöd
Stig Carlsson	IAM
Peter Hernfalk	Terminologitjänsten
Niklas Frantzel	Terminologitjänsten
Tomas Fransson	IAM
Emanuel Bergsten	Sömlös vård
Christina Kling Hassler	Sömlös vård
Peter Merikan	Formulärhantering
Jenny Åshammar	Formulärhantering
Oskar Thunman	Standarder för informationsutbyte

2.2 Syfte och omfattning för utredningen

Syftet med utredningen är att sammanställa de eventuella behov av nya och förändrade stödtjänster som kommer med den nya samverkansarkitekturen T2 liksom med nya och ökade krav på exponering av interna såväl som interna APIer, utifrån de utvecklingsprojekt och initiativ som Arkitektur och Digital Infrastruktur bedriver eller planerar att bedriva under 2023 - 2024. Resultatet skall vara en prioriterad lista av åtgärder som bör genomföras för att säkerställa de identifierade behoven.

2.3 Förutsättningar och antaganden

Jag har i detta arbete gjort följande antaganden, som varit vägledande i att formulera kravbild och rekommendera åtgärder:

Bevara styrkorna i RIV-TA baserade interaktioner: RIV-TA och den nationella tjänsteplattformens viktigaste framgångsfaktorer är och har varit dess fokus på *överenskomna kontrakt*, samt den lösa koppling som ges mellan sändare och mottagare genom *logisk adressering*. Det är av central betydelse att bevara dessa egenskaper, även om både arkitektur, teknik och informatik moderniseras och ändras.

Stegvisa förändringar med fokus på verksamhetsnytta: Både den nya samverkans-arkitekturen och Ineras ökade fokus på exponering av APIer ställer krav på sofistikerade stödtjänster, verktyg och plattformar. Det är dock klokt att skilja på vad som krävs på kort sikt för att ge nytta i de närmast förestående initiativen/projekten, jämfört med vad som krävs i förlängningen för en fullt utbyggd, skalbar och kostnadseffektiv lösning. Det möjliggör att realisera stödtjänster och verktyg iterativt och inkrementellt, med fokus på tidigt nytta. Genom att börja med att realisera en ”minsta livskraftig produkt” som ändå kan leverera verksamhetsnytta, åstadkommer man flera saker: Förutom den uppenbara fördelen med att leverera verksamhetsnytta tidigt, så får man dessutom värdefull kunskap om både kraven och lösningen. Förutsättningarna ökar därmed att realisera resterande lösning både bättre och effektivare.

2.4 Viktiga begrepp och akronymer

Begrepp	Förklaring
APIM	En API Management plattform är en samling verktyg som möjliggör skalbar och effektiv livscykelhantering av APIer.
Apifiering	APIfiering är att bryta ned lösningar i enskilda autonoma tjänster som följer varandras API-specar för att underlätta informationsutbyte.
FHIR	Fast Health Interoperability Resources, standard för utbyte av elektronisk hälsodata definierad av HL7.
Interoperabilitets-specifikation	Ett samlingsbegrepp för överenskommelser som beskriver förutsättningar och krav för digitala tjänster, och inbegriper både legala, organisatoriska, semantiska och tekniska aspekter.
Portföljanalys	Analys över vilka tjänster i portföljen som behöver förändras baserad på den nya samverkansarkitekturen.
OAuth 2.0	Open Authorisation, standard för biljett-baserad, säker delegerad auktorisation.
OIDC	OpenID Connect, standard för autentisering byggd ovanpå OAuth 2.0.

REST	Representational State Transfer eller RESTful webbtjänst är en arkitekturellt mönster för tjänster för maskin-till-maskin-kommunikation som tillhandahållas via webbteknologi.
SAML	Security Assertion Markup Langauage, XML-baserad standard för att utbyta data för autentisering och auktorisering mellan olika parter.
Samverkansarkitektur	Gemensam digital infrastruktur och arkitektur.
VIS	Regionernas vårdinformationssystem

2.5 Referenser

[1] Regelverk för interoperabilitet inom vård och omsorg: https://rivta.se/

[2] Referensarkitektur för vård och omsorg - T-boken: https://inera.atlassian.net/wiki/spaces/RTA/pages/3632866/Referensarkitektur+f+r+v+rd+och+omsorg+-+T-boken+REV+D

[3] CDA Release 2: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7

[4] Representational State Transfer (REST): https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[5] Open API Specification (OAS): https://spec.openapis.org/oas/v3.1.0

[6] Fast Health Interoperability Resources (FHIR): https://www.hl7.org/FHIR/

[7] SMART-on-FHIR: http://hl7.org/fhir/smart-app-launch/

[8] Security Assertion Markup Language 2.0 (SAML): https://www.oasis-open.org/committees/download.php/56776/sstc-saml-core-errata-2.0-wd-07.pdf

[9] OAuth 2.0: https://oauth.net/2/

[10] Open ID Connect: https://openid.net/connect/

[11] Open ID Connect Federation 1.0 draft: https://openid.net/specs/openid-connect-federation-1_0.html

[12] DIGG: Svenskt ramverk för digital samverkan: https://www.digg.se/kunskap-och-stod/svenskt-ramverk-for-digital-samverkan

[13] European Interoperability Framework (EIF): https://joinup.ec.europa.eu/collection/nifo-national-interoperability-framework-observatory/european-interoperability-framework

[14] Hypertext As the Engine of Application State (HATEOAS): https://en.wikipedia.org/wiki/HATEOAS

[15] DIGG: REST API-profil: https://dev.dataportal.se/rest-api-profil

[16] Attribute Based Access Control (ABAC): https://en.wikipedia.org/wiki/Attribute-based_access_control

[17] Referensarkitektur för Identitet och Åtkomst, Rev A: https://rivta.se/documents/ARK_0046/Referensarkitektur-Identitetochatkomst-RevA.pdf

[18] Övergripande design av tjänsteregistrering och tjänstesökning inom vård och omsorg: https://inera.atlassian.net/wiki/spaces/T2/pages/2907046422/UTREDNING+vergripande+design+av+tj+nsteregistrering+och+tj+nstes+kning+inom+v+rd+och+omsorg

[19] Målarkitektur för stödtjänster och samverkan enligt T2: https://inera.atlassian.net/wiki/spaces/T2/pages/3138322690/M+larkitektur+f+r+st+dtj+nster+och+samverkan+enligt+T2

[20] Exempel på tillämpning av T2:
https://inera.atlassian.net/wiki/spaces/T2/pages/2798715797/Exempel+p+till+mpning+av+T2

[21] Referensarkitektur för grunddata och katalog, version 1.0:
https://rivta.se/documents/ARK_0059/Referensarkitektur_for_grunddata_och_katalog.pdf

[22] Ineras stöd för sömlöst vårdflöde - presentation för LG, Christina Kling Hassler *et.al.*

[23] Strategi för Ineras integrationsarkitektur, version 0.3, Martin Svensson.

[24] Intressenter för API Management lösning:
https://ineraab.sharepoint.com/:w:/s/TeamAPIManagementPlattform/ERWcYH8k5k1FjapEYxqTVqoBOAGdF-0gfoSHpf1LOqeabw

[25] Analys av nytt utvecklingsbehov: Gemensamt regelverk för enskilds direktåtkomst i invånarapplikationer, version 1.0 2022-12-30, Maria Ekendahl.

2.6 Dokumentöversikt

Kapitel 3 ger en kort beskrivning av nuläge i form av referensarkitekturer och pågående utvecklingsinitiativ. Kapitel 4 går igenom gemensamma identifierade behov. Kapitel 5 diskuterar lösningsalternativ och åtgärder för att möta de identifierade behoven. Kapitel 6 beskriver de enskilda projektens och initiativens behov av nya eller förändrade stödtjänster. Kapitel 7 avslutningsvis ger tidsuppskattningar för de rekommenderade åtgärderna.

3 Bakgrund och nuläge

3.1 Samverkansarkitektur

Samverkansmönster mellan system inom vård och omsorg har sedan 2010 beskrivits och styrts av RIV-TA [1] och den nationella referensarkitekturen i T-boken [2]. Två bärande principer ger förutsättningar för interoperabilitet: centralt *överenskomna tjänstekontrakt* och lös koppling mellan samverkande parter via *logisk adressering*. Arkitekturen realiseras baserat på synkrona RPC-anrop över SOAP, med en central meddelande-växel som ansvarar för logisk adressering och åtkomst-regelverk (”tjänste-plattformen”). Adressering och åtkomstregler tillförs genom en Tjänsteadresseringskatalog (TAK). Tjänstekontrakten beskrivs med XML Schema (som en del av WSDL-beskrivningen av SOAP-ändpunkterna) tillsammans med ytterligare textuell beskrivning i dokumentform (Tjänstekontraktsbeskrivning, TKB). HL7’s CDA (Clinical Document Architecture [3]) eller dess förenklade form (”Green CDA”) har i förekommande fall använts som hälsoinformatisk standard.

Principerna om standardiserade tjänstekontrakt och logisk adressering har gjort stor nytta och bidragit till digitalisering inom vård och omsorg. Men både teknologier och informatik-standard har med tiden blivit omsprungna av mer moderna teknologier och standards, främst REST [4] och FHIR [5]. På grund av RIV-TAs hårda koppling till SOAP, XML och en central meddelande-växel, har det visat sig svårt att anpassa den till de nya krav som dessa tekniker ställer.

3.1.1 Samverkan enligt T2

En vidareutveckling av T-bokens referensarkitektur har därför gjorts, som bättre svarar mot dagens krav. Samtidigt har nomenklaturen i arkitekturen anpassats till Svenskt ramverk för digital samverkan [7] och EIF [13], och referensarkitekturen har skiktats i två skikt: en mer generell referensarkitektur för interoperabilitet inom svensk välfärd, och en specialisering för vård och omsorg. Referensarkitekturerna går under namnet T2, och är tänkt att komplettera snarare än ersätta RIV-TA och T-boken.

T2 är uttryckt med högre abstraktionsgrad är tidigare, och en viktig styrande princip är att kontrakt bör utformas teknikoberoende. Därmed kan olika kommunikationstekniker och protokoll såväl som olika hälso-informatiska standarder stödjas.

Den viktigaste nyheten ligger i ett explicit och (potentiellt) mer fingranulärt federations-begrepp. En *informationsfederation* tjänar ett specifikt syfte, och olika federationer kan ha olika förutsättningar, lagrum, regler och specifikationer. En *federationsoperatör* hanterar och samordnar federationens verksamhet. En sådan operatör kan vara Inera men kan även vara annan part.

Den viktigaste ändringen ligger i att T2 öppnar upp för direkt punkt-till-punkt-interaktion mellan samverkande parter (utan att trafiken passerar en central meddelande-växel). Detta nya samverkansmönster motiveras dels av förändrade anropsmönster (framför allt drivet av REST-API:er såsom t.ex. FHIR som följer HATEOAS [14] designprinciper med mer fingranulära informationsresurser länkade till andra resurser via hypertext-länkar), dels av en strävan efter förenkling i de fall där en central tjänst/växel inte tillför verksamhetsnytta i proportion till dess komplexitet. Det är ett väl beprövat mönster för samverkan mellan oberoende parter.

En arkitekturell konsekvens av en decentraliserad punkt-till-punkt-kommunikation mellan samverkande parter, är att API-klient och -producent får överta ansvaret för logisk adressering respektive tillit/åtkomstkontroll. För att stödja dessa kapabiliteter definierar T2 ett antal stödtjänster som en federationsoperatör skall erbjuda till klienter och producenter inom federationen: En *tjänstekatalog* ger stöd för en klient att hitta den tekniska anropsadressen till en API-producent, givet en interoperabilitets-specifikation (motsvarigheten till ett Tjänstekontrakt enligt T-boken) och en logisk adress (typiskt ett verksamhetsid). En *federationskatalog* ger klient och producent möjlighet att verifiera verksamheters tillhörighet till den aktuella federationen, som basis för tillit och åtkomstbeslut.

3.2 API:er

Utöver APIer inom formella samverkansfederationer utvecklar och förvaltar Inera även andra APIer, både interna och externa. Interna REST-APIer används bland annat inom Formulärhantering för kommunikation mellan invånar- respektive medarbetar-portaler och

bakomliggande backend. 1177 e-Tjänster har påbörjat en liknande skiktning i sidor/vyer/e-tjänster mot bakomliggande APIer. Terminologitjänsten skall tillgängliggöra terminologier och kodverk via externa FHIR-baserade APIer.

3.3 Identitet och åtkomst

Identitet, autentisering och åtkomstkontroll är en nödvändig förutsättning för samverkan. RIV-TA bygger på tillit mellan organisationer, där de samverkande systemen autentiserar sig mot tjänsteplattformen med SITHS funktionscertifikat via mutual TLS och åtkomstregler realiseras genom registreringar i tjänsteadresseringskatalogen (TAK). Stöd för delegerad åtkomst saknas – i den mån operationer utförs för en inloggad användares räkning, får slutanvändarens identitet skickas som en del av nyttolasten (utan tekniska garantier om äkthet).

Ineras referensarkitektur för Identitet och åtkomst [17] pekar på en mer modern och kapabel federerad modell, baserad på s.k. biljettbaserad teknik. Inloggning och autentisering är frikopplad från biljett/intygsutfärdande, där ett identitetsintyg och/eller åtkomstintyg på ett säkert sätt förmedlar identitetsdata och grundläggande behörighetsgrundande information (och i fallet med åtkomstintyg även information om en eller flera rättigheter för specifika resurser med ett visst omfång).

Två olika tekniska standarder för intyg pekas ut: SAML 2.0 [8] och OAuth 2.0 [9]. SAML rekommenderas för äldre, XML-baserade lösningar utan krav på delegerad åtkomst (vilket inte stöds av SAML), medan OAuth 2.0 med OpenID Connect (OIDC, [10]) rekommenderas för REST/JSON-baserade lösningar eller där krav finns på delegerad åtkomst. FHIR-standarens lösning för autentisering och auktorisation (som är en del av SMART-on-FHIR [7]) bygger på OAuth2.

Revision A av referensarkitekturen saknar explicit stöd för server-till-server-autentisering och åtkomstkontroll. I kommande revision B läggs detta till, i form av OAuth 2.0 *Client Credentials* Flow med autentisering baserat på asymmetrisk autentisering (private_key_jwt) eller mutual TLS. Denna mekanism harmonierar väl med SMART-on-FHIR (som dock inte explicit stödjer autentisering via mutual TLS). Information om det anropande systemets identitet (subject_dn alternativt publika nyckel) måste registreras hos intygsutfärdaren, varför SMART-on-FHIR rekommenderar att OAuth 2.0-utökningen RFC7591 *Dynamic Client Registration Protocol* stöds för att möjliggöra programmatisk anslutning av klienter.

För delegerad åtkomst via ett system rekommenderar revision B av referensarkitekturen OAuth 2.0-utökningen RFC8693 *Token Exchange*, där ett system As åtkomstbiljett tillsammans med en biljett för den part B (som kan vara en inloggad användare eller annat system) som åtkomsten skall göras för kan bytas mot en biljett som indikerar att system A har rätt att agera för Bs räkning. Denna mekanism kan även användas för ett federativt förlitande mellan åtkomstintygstjänster.

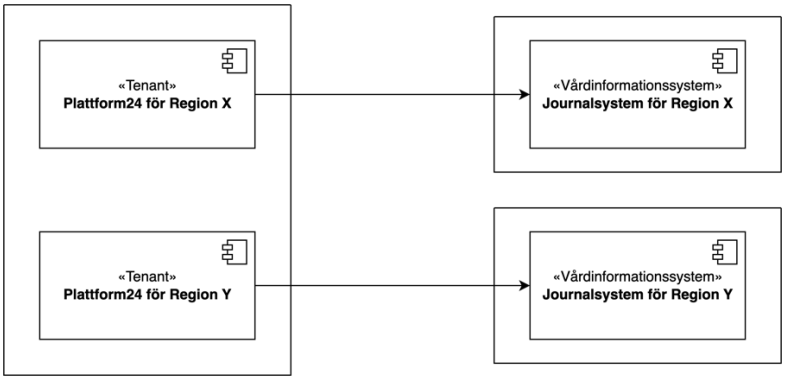
3.4 Pågående och planerade utvecklingsinitiativ

3.4.1 Skriv till VIS

”Skriv till VIS” är en identifierad förmåga inom ramen för Symptombedömning och hänvisning (SBH, även kallat 1177 direkt). Resultatet av processen för ”Automatisk symptombedömning och hänvisning” utgör en journalhandling, och skall därmed journalföras i respektive vårdgivares vårdinformationssystem. Förmågan lyftes ur SBH-upphandlingen för att realiseras i ett separat projekt. Liknande behov ses även inom andra 1177 e-tjänster (t.ex. nya Rådgivningsstödet, Stöd och behandling, 1177 ärendehantering och Formulärhantering).

Produkten Plattform24 har upphandlats för realisering av SBH. Produkten förutsätts kunna anpassas till en överenskommen interoperabilitetsspecifikation, troligtvis baserad på FHIR över REST och Smart-on-FHIR. I ett första skede skall ett begränsat antal regioner ingå i federationen. En förstudie har påbörjats för Skriv till VIS, som pågår t.o.m. maj 2023. Ambitionen är sedan att starta ett genomförandeprojekt under 2024.

Aktörsrelationerna i Skriv till VIS är initialt att betrakta som en till en: En instans (logisk eller fysisk) av Plattform24 för en specifik region skriver journalanteckningar till samma regions Vårdinformations-system. Tjänstekonsument och tjänsteproducent är här alltså samma organisation, men där samverkan uppstår av tekniska skäl eftersom Platform24 upphandlas och erbjuds som en tjänst av Inera.



Det finns dock en ambition att förbereda för en mer generell lösning, där även andra e-tjänster inom 1177 kan nyttja samma mekanism. Aktörsmönstret blir då en till många eller många till många.

3.4.2 1177 e-tjänster

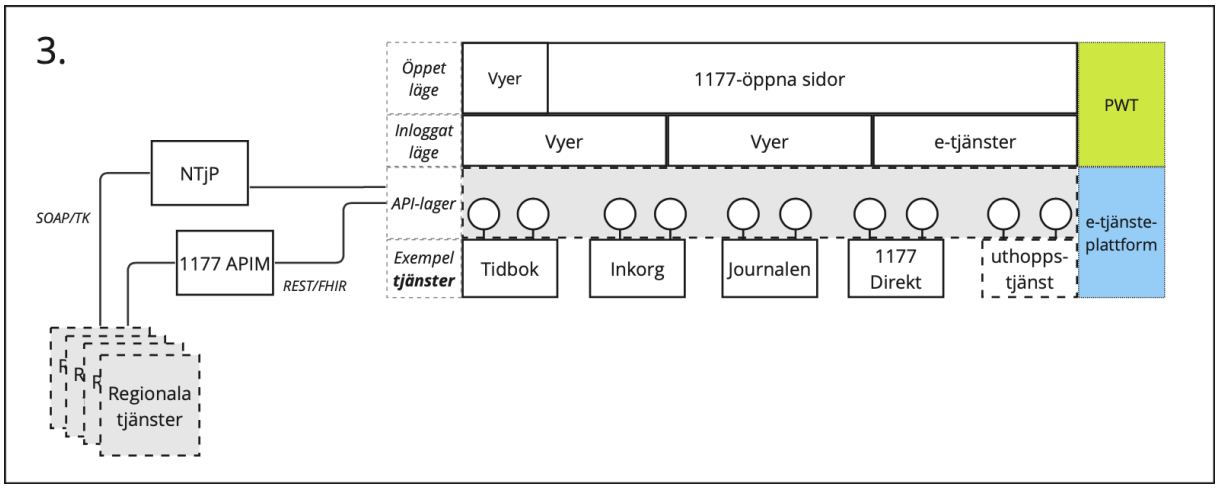
1177 e-tjänster har utvecklats under lång tid, och bestod initialt av ett antal mer eller mindre isolerade e-tjänster utan tydlig skiktning mellan användargränssnitt och verksamhetslogik. Det har försvårat möjligheten att dela gemensam information eller verksamhetslogik mellan e-

tjänster, liksom att skapa aggregerade användargränssnitt/vyer som spänner över flera informations-mängder eller tjänster.

1177 Målarkitektur pekar ut separation av nuvarande e-tjänster i separata underliggande tjänster exponerade via APIer och e-tjänster/användargränssnitt/vyer som en viktig förutsättning för att kunna realisera 1177 målbild. Gemensamma förmågor för API-tjänster såsom tjänsteuppslag (service discovery), accesskontroll, mellanlagring, loggning, monitorering och statistik skall externaliseras från API-tjänsterna och realiseras i ett separat API-lager. Anpassning av befintliga tjänster till denna arkitektur har påbörjats, men behöver konsolideras och accelereras.

3.4.2.1 Plattform Webtjänster (PWT)

Parallellt med den tekniska plattform som e-tjänsterna är implementerade på, inför 1177 en ny plattform (PWT) för främst redaktionellt innehåll. Det finns en uttalad ambition om en tydlig ansvarsfördelning mellan de båda plattformarna, där PWT hanterar redaktionellt innehåll, ”sidor”, ”vyer” och aggregering/orkestrering av information för e-tjänster, medan den ursprungliga tekniska plattformen hanterar underliggande tjänster och deras APIer.



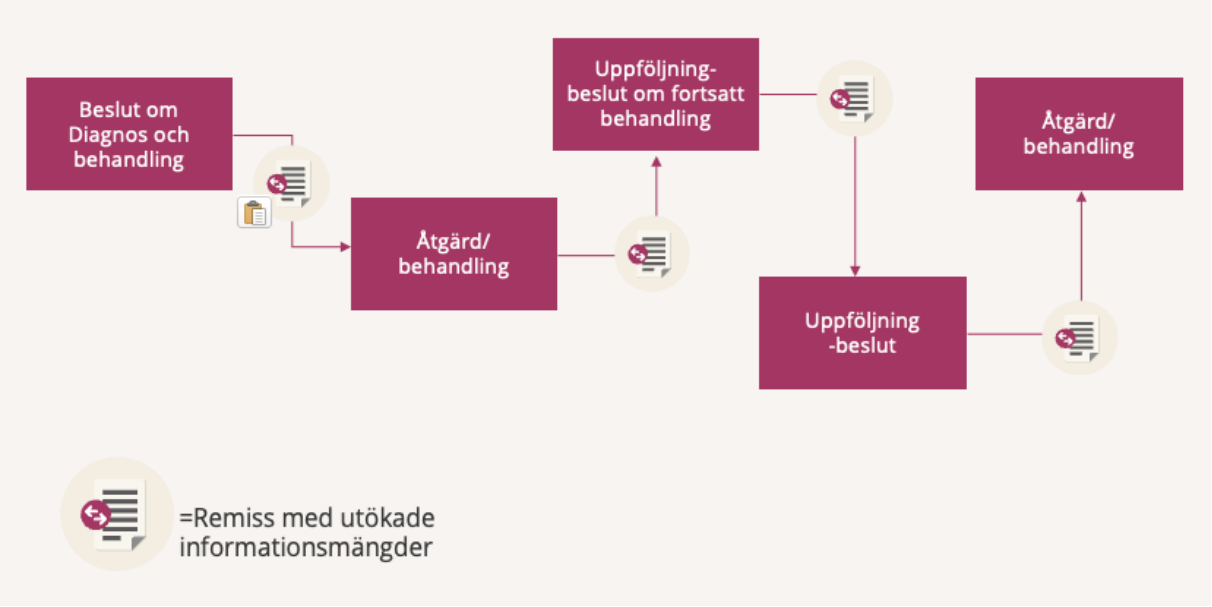
3.4.2.2 Sammanhållen planering

Sammanhållen planering är ett långsiktigt, samlande initiativ som syftar till att ge invånaren ett enkel och sammanhållet perspektiv på sin vårdrelaterade information, och därmed ges förutsättning att vara en aktiv medskapare. 1177 kommer vara den viktigaste kanalen, vilket ställer krav på nya sätt att aggregera, kombinera och presentera information från underliggande tjänster/APIer utifrån invånarens behov. 1177 målarkitektur är därför en viktig förutsättning för att realisera sammanhållen planering.

3.4.3 Sömlöst vård flöde

Sömlöst vårdflöde är ett initiativ för att bättre stödja samarbete mellan vårdgivare. En central pusselbit här är möjlighet till delning av information av olika slag, från enkel text till filmer, foton, röntgen, digital patologi etc. Flödet drivs via remisser, som skulle behöva kunna bära med sig multipla bifogade bilagor. RIV-TA kontraktet för e-remiss är inte helt ändamålsenligt, då det finns en begränsning i maximal meddelande-storlek som omöjliggör meddelanden större än 5mb. Inbäddade bilagor (där en kopia av originalhandlingen bifogas, och mottagaren därmed får ansvara för att hantera och eventuellt spara bilagan) är heller inte optimalt.

Möjlighet för meddelanden med bilagor via referens skulle behövas för att möta verksamheternas behov. Denna förmåga saknas i dagens lösningar: Att kunna skapa referenser till ursprungsartefakterna, och göra dem tillgängliga från källsystemen på ett säkert sätt. Optimalt skulle dessa referenser dessutom vara teknikoberoende och beständiga över tid, för att inte behöva dubbellagra artefakterna om mottagaren av remissen har krav på långsiktig beständighet (vilket beroende på tillämpning kan vara reglerat i lag och så långt som 10 år).



En initial utredning ”Ineras stöd för sömlöst vårdflöde” har gjorts [22], som identifierar 3 möjliga lösningsförslag på kort, mellan respektive lång sikt.

Förslag A innebär en anpassning av befintligt tjänstekontrakt för eRemiss, för att möjliggöra inbäddning av mindre informationsmängder (exempelvis journalutdrag eller små bilder). Denna lösning kräver inga ytterligare förmågor av Inera, och kan realiseras med relativt små medel. Den ger dock bara partiell nytta för verksamheten.

Förslag B innebär i stället att eRemiss anpassas att kunna referera till en eller flera bilagor, som finns lagrade i den remiss-utfärdande partens källsystem. Nationell Patientöversikt NPÖ föreslås utökas för att kunna användas för att hämta och visa de refererade bilagorna. Det skulle även

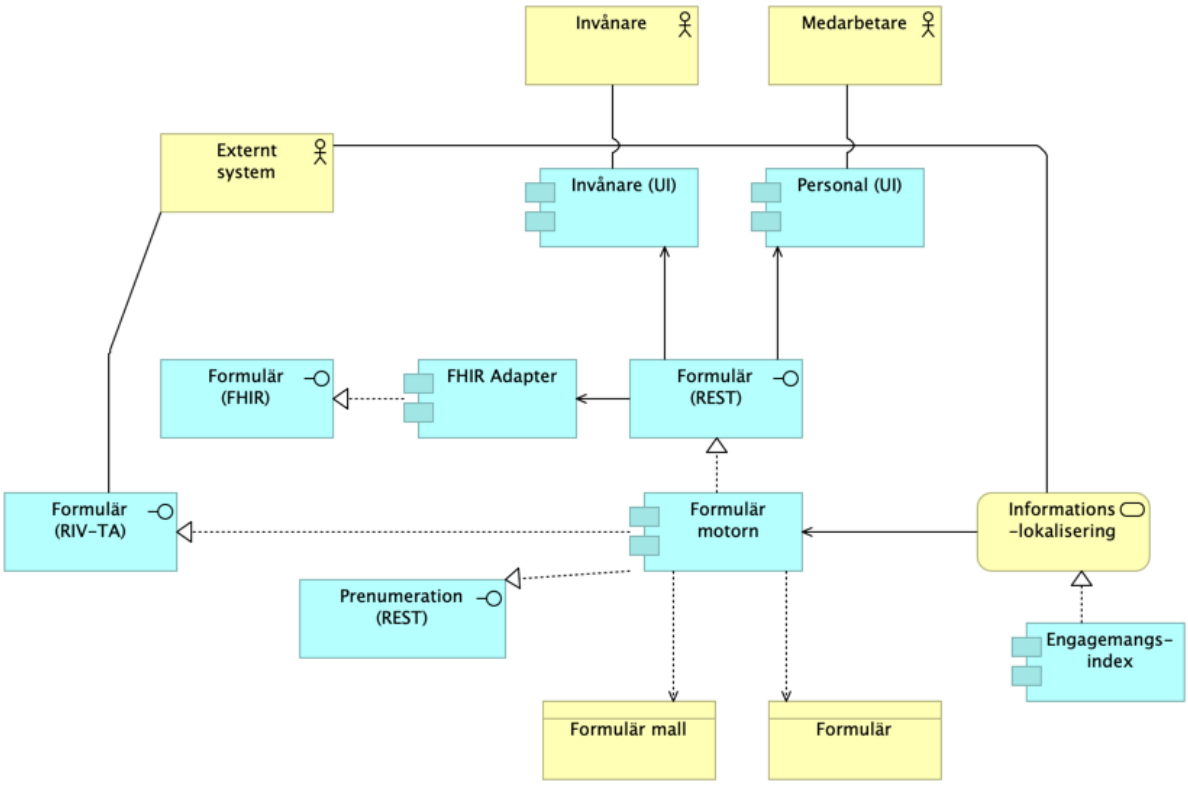
kräva nya APIer hos remiss-utfärdande partens källsystem för att möjliggöra hämtning av bilagorna.

Hantering/läsning av stora artefakter lämpar sig inte att göra via nationella tjänsteplattformen, på grund av storleken. För hantering av riktigt stora medier krävs i stället användning av s.k. strömmande API:er. Ett REST-baserat API inom en samverkan enligt T2 skulle vara mer ändamålsenligt. Tekniskt finns det inget som hindrar att ett REST-baserat API för strömmande medier kombineras med RIV-TA baserad eRemiss, men naturligare kanske är att även exponera eRemiss över ett REST-baserat API enligt T2-mönster. I T2s exempel på tillämpningar (se [20]) har ett initialt arbete gjorts att beskriva (delar av) en sådan lösning.

3.4.4 Formulärhantering

Formulärhantering är en tjänst som möjliggör inhämtning av strukturerad information från invånare via e-tjänst på 1177. Formulärmallar innehåller strukturerade frågor med svarsalternativ, och medger ett enkelt arbetsflöde med utskick, ifyllande och hämtning av besvarat formulär. Funktionerna kan hanteras via formulärmotorns RIV-TA API eller via ett web-baserat personal- respektive invånargränssnitt.

Formulärhantering består i dagsläget av en uppsättning backend-tjänster (”formulärmotorn”) som exponerar RIV-TA tjänster mot andra system, och användargränssnitt för vårdpersonal respektive invånare som servas av backend-tjänsten via ett internt REST-API. Informations-elementen i RIV-TA kontrakten och REST-api:et är i stort sett identiska.

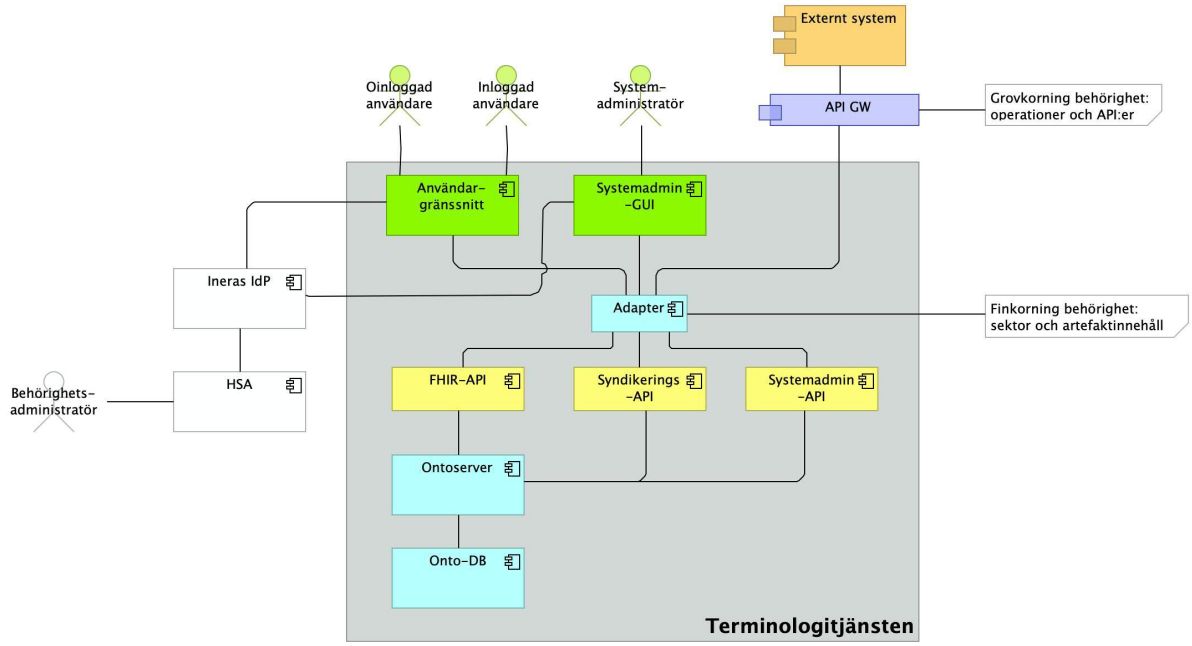


En Proof-of-Concept-implementation har gjorts för att exponera formulärmotorn med ett FHIR-API över REST. En del informatikarbete återstår för att mappa resterande attribut, liksom utvecklingsarbete för att göra implementationen produktionsfärdig. Då ingen gemensam lösning för bl.a. åtkomstkontroll finns i dagläget, har driftsättning av FHIR-APIet pausats.

Formulärhantering har även ett behov av en mekanism för hantering av prenumerationer på status-förändringar av formulär-instanser. Nuvarande, initiala realisering av denna förmåga använder Engagemangsindex för att propagera statusförändringar, vilket strider mot syftet med Engagemangsindex och driver onödig trafik. En annan, mer ändamålsenlig realisering håller på att designas. FHIRs *Resource Subscription* kan troligtvis användas som utgångspunkt.

3.4.5 Terminologitjänst

Terminologitjänsten skall erbjuda en tjänst för hantering av och sökning i terminologier och kodverk. Terminologitjänsten har upphandlat produkten Ontoserver för hantering och lagring av kodverk och terminologier. Terminologitjänsten utvecklar egna administrativa gränssnitt mot Ontoserver för multi-region stöd (s.k. *multi tenancy*, där data hålls isolerad per region eller vårdgivare trots en delad, central installation). Ontoserver implementerar FHIRs terminologi APIer som standard. Dessa FHIR-apier används av Terminologitjänsten administrativa gränssnitt, men skall också exponeras som externt API både i en samverkansfederation (för administration av kodverk) och som externt, öppet API.



Ontoserver erbjuder en säkerhetsmodell byggd på OAuth2 med SMART-on-FHIR [7]. Leverantören erbjuder även en färdig SMART-on-FHIR-kapabel OAuth2 server i form av Ontocloak (en paketering av Keycloak). Terminologi-tjänstens administrativa gränssnitt använder Ineras IdP med OIDC-baserat identitetsintyg för autentisering och åtkomstkontroll. För extern API

access skall krävas autentisering av anslutande system, och tillhörighet till relevant terminologi-federation kontrolleras. All extern API access skall initialt begränsas till läsande operationer (dvs http GET anrop över REST-API:et). På sikt finns även behov av ”sektor”-baserad accesskontroll, så att vissa informationselement bara får läsas via externt API av den eller de organisationer/regioner som äger informationen.

En första version 1.0 är planerad till slutet Q2 2023, som fokuserar på de administrativa gränssnitten. Stöd för extern access via FHIR-API är planerad strax därefter, under Q3 2023.

3.4.6 Regelverk för enskilds direktåtkomst i invånarapplikationer

Ett behov har identifierats för att möjliggöra att samma regelverk som tillämpas för invånares åtkomst till sin vårdinformation via 1177 Journalen skall kunna användas för åtkomstkontroll vid användning av andra invånarapplikationer. En del av regeluppsättningen (ombudsinformation) finns idag tillgänglig över ett RIV-TA kontrakt genom Ombudstjänsten. Andra regeluppsättningar kring försegling är i dagsläget bara implementerat i 1177 Journalen.

Ett initiativ [25] har därför lagts för beredning som syftar till att konsolidera regelverket för enskilds direktåtkomst, eventuellt genom att exponera regelverket via ett eller flera nya API.

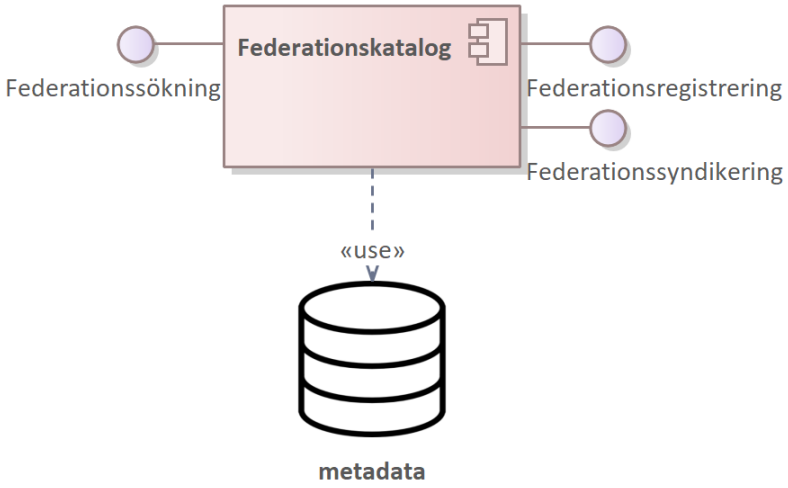
4 Behovsanalys

4.1 Samverkan enligt T2: Stödtjänster för Federationsoperatör

En federationsoperatör behöver stödja klienter och producenter med ett antal gemensamma förmågor för att underlätta organisatorisk och teknisk samverkan. Det kan finnas stora operationella fördelar med att dessa tjänster erbjuds centralt av Inera, även i de fall Inera inte agerar federationsoperatör.

4.1.1 Federationskatalog

En *federationskatalog* ger API-klienter och -producenter möjlighet att verifiera verksamheters tillhörighet till den aktuella federationen, som basis för tillit och åtkomstbeslut. Den kan även tjäna som en mekanism för federationsmedlemmar att administrera metadata om sin egen organisation, som andra intressenter därmed får tillgång till. Federationskatalogen utgör därmed metadata-katalog för organisatorisk och teknisk metadata om anslutna parter som är nödvändig för att etablera federativt förlitande mellan system över organisationsgränser.



4.1.1.1 Federationssökning

Vid federationssökning kan en API-klient eller -producent verifiera att en organisation är ansluten till en specifik federationen, och således har förbundit sig att följa federationens legala, organisatoriska och tekniska regler och riktlinjer. Federationssökning kan även användas för att hämta metadata om en organisation, inklusive tekniska metadata som t.ex. vilket publikt certifikat som organisationen använder för att autentisera sig eller vilka ip-adresser som organisationen gör anrop ifrån.

4.1.1.2 Federationsregistrering

Vid federationsregistrering registreras en organisation och dess enheter som medlem i federationen, tillsammans med metadata om organisationen och dess enheter. Federationsregistrering följer på ett anslutningsförfarande, där följsamhet mot federationens regler utvärderas innan en organisation registreras i federationskatalogen. Vid efterföljande förändringar bör federationsmedlemmar automatiskt kunna uppdatera sitt eget metadata.

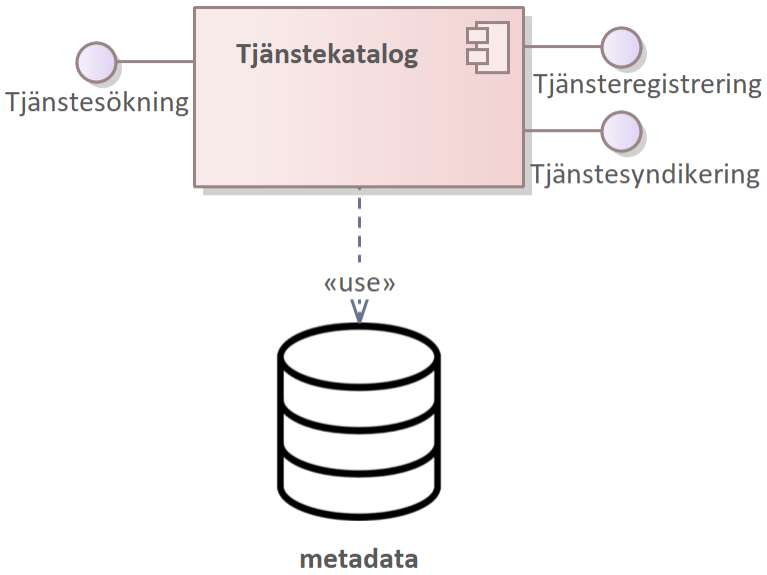
4.1.1.3 Federationssyndikering

Det finns även behov av ett prenumerationsförfarande eller synkroniseringsmekanism för federationsinformation, så att befintliga medlemmar i federationen blir notifierade när en medlem tillkommer eller försvinner, eller när metadata för en medlem ändras eller läggs till. Denna information kan ligga till grund för t.ex. automatisk konfiguration av brandväggar eller federativt förlitande mellan system.

4.1.2 Tjänstekatalog

Kopplingen mellan erbjudna digitala tjänster och de system som realiserar dem är i ständig förändring, vilket gör att kartan över samverkande system i en federation ständigt ritas om.

En *tjänstekatalog* ger stöd för en klient att hitta den tekniska anropsadressen till en API-producent, och för en producent att registrera och underhålla motsvarande information.



4.1.2.1 Tjänstesökning

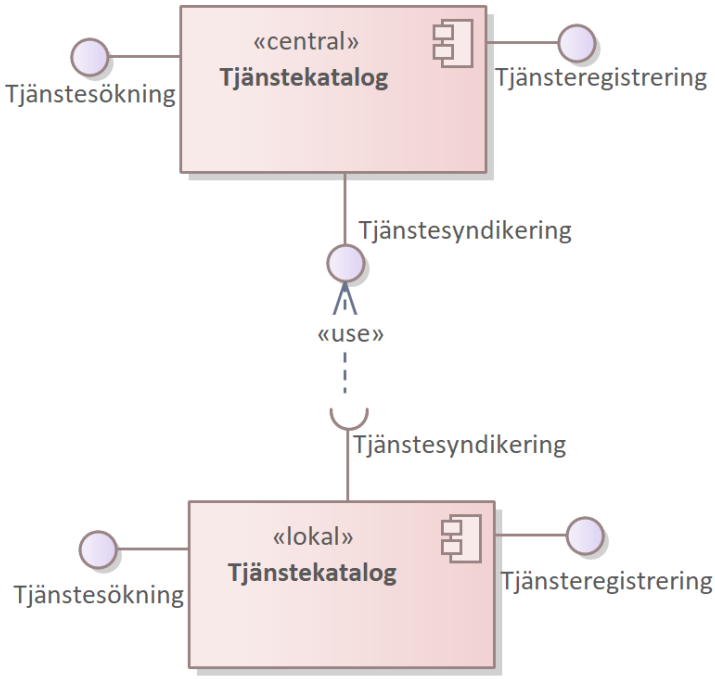
Vid tjänstesökning hittar en klient den tekniska anropsadressen till en API-producent och eventuell ytterligare information om utfärdare av åtkomstintyg, givet en interoperabilitets-specifikation för det efterfrågade APIet (motsvarigheten till ett Tjänstekontrakt enligt T-boken) och en logisk adress till producenten (typiskt en organisationsidentitet).

4.1.2.2 Tjänsteregistrering

Vid tjänsteregistrering registrerar en tjänsteproducent en digital förmåga beskriven av en interoperabilitets-specifikation. Förmågan registreras under en logisk adress (typiskt en organisationsidentitet), och knyts till dess tekniska anropsadress och eventuell ytterligare information om utfärdare av åtkomstintyg.

4.1.2.3 Lokal federering med central konsolidering

T2 rekommenderar att tjänstekatalogen realiseras som en federativ lösning med möjlighet till lokala tjänstekataloger med central konsolidering. Det möjliggör att tjänster kan ingå i både lokala och nationella federationer, med en distribuerad administration. Ett explicit gränssnitt för syndikering av tjänsteinformation krävs för automatisk syndikering.



4.1.3 Informationsindex

Aggregerande tjänster sammanställer information från flera bakomliggande API-producenter, som t.ex. Nationell Patientöversikt eller Journalen. I situationer där antalet potentiella bakomliggande producenter för ett specifikt sökobjekt är stort men där information om sökobjektet bara finns hos en del av dessa producenter, kan användning av ett informationsindex vara nödvändigt för att minska antalet onödiga anrop.

Ett index som relaterar en invånare till de API-producenter som har information om invånaren, kan dock innebära integritetsproblem. Utformning och användning av informationsindex måste därför balansera mellan integritetsaspekter och optimering av anrop. Nuvarande tjänst Engagemangsindex kan därför behöva anpassas. Det kan även vara rimligt att exponera Engagemangsindex över ett FHIR eller REST-baserat API.

4.2 Samverkan enligt T2: Centralt definierade tjänster

Några ytterligare tjänster måste finnas centralt definierade, men där realiseringen kan vara decentraliserad och federerad.

4.2.1 Identitetshantering

I dagsläget används uteslutande SITHS som identitetsutfärdare för medarbetare. För invånare skall alla godkända identitetsutfärdare enligt svensk e-legitimation stödjas.

4.2.2 Autentisering och intyg för identitet och åtkomsthantering

En autentiseringstjänst ansvarar för att med viss autentiseringsmetod autentisera en aktör, dvs. säkerställa aktörens identitet. En identitetsintygsutfärdare utfärdar säkra identitetsintyg, som intygar aktörens identitet och tillhörande egenskaper. En åtkomstintygsutfärdare utfärdar säkra åtkomstintyg, som intygar att aktören har en rättighet att nå en viss resurs/utföra en viss operation. Intygen utgör en grund på vilken en API-producent grundar sitt åtkomstbeslut.

En identitets- och behörighetsfederation bygger på tillit mellan de federerade organisationerna, vilket gör att man kan acceptera utfärdade identitetsintyg från annan part, och därigenom ge åtkomst till skyddade resurser utan att själv behöva administrera den andra partens aktörer och deras e-identiteter och egenskaper. Förutsättningar för identitets- och behörighetshantering och federering beskrivs i detalj i Referensarkitektur för Identitet och Åtkomst [17].

En Identitets och behörighetsfederation innefattar dock normalt inte system-identiteter, och revision A av referensarkitekturen saknar explicit stöd för server-till-server autentisering och åtkomstkontroll. Detta är tillagt i revision B, liksom skrivningar som antyder en federativ modell för tillit mellan åtkomstintygstjänster. Eventuellt kan en federationsmodell baserad på OIDC Federation [11] även användas för federering av system-identiteter eller klienter. Det är dock oklart exakt hur det skall realiseras tekniskt, och hur kopplingen till federationskatalogens metadata ser ut. Det saknas idag konkreta realiseringsanvisningar för de mönster som beskrivs, vilket är kritiskt då varje region såväl som Inera behöver kunna realisera dessa förmågor.

4.3 Inera som API producent

Tjänster som exponerar APIer (externt eller internt) har ofta ett antal likartade behov av generella förmågor såsom autentisering, åtkomstkontroll, teknisk adressering, lastbalansering, mellanlagring, loggning, statistik etc (även om detaljerna sannolikt skiljer sig åt för externa och interna APIer). En API-producent som exponerar sitt API externt inom ramen för samverkan enligt T2, måste t.ex. själv ansvara för att åtkomst bara beviljas enligt de regler som federationen kräver.

Ineras portfölj av exponerade APIer är stadigt växande (en initial inventering över APIer har gjorts, se [24]). APIerna har likartade gemensamma behov, oavsett om de används internt, externt eller inom ramen för en samverkan.

Förmågor av övergripande, generell karaktär kan med fördel externaliseras, dvs brytas ut från underliggande APIer för att realiseras separat. Därigenom minskar behovet av duplicerad funktionalitet och ger en uniform, anpassningsbar realisering av förmågorna som kan kvalitetssäkras separat. En mekanism för att realisera gemensamma förmågor kan förvaltas och livscykel-hanteras separat, även om vissa av policy-reglerna kan behöva ägas och förvaltas av de bakomliggande APIerna.

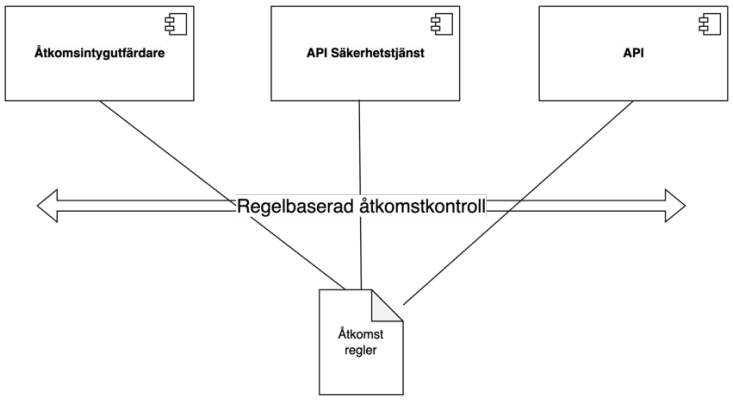
4.3.1 Autentisering och åtkomstkontroll

Alla exponerade APIer, såväl interna som externa, måste skydda sig mot obehörig åtkomst. För system-till-system-interaktion enligt T-boken används mutual TLS där klientens funktionscertifikat används för att identifiera anropande system och regler i Tjänsteaddresseringskatalogen (TAK) ger åtkomsträttigheter. I kommande revision B av IAM-strategin rekommenderas istället OAuth 2.0 åtkomstintyg, via *Authorization Code* flödet för autentisering av slutanvändare, *Client Credentials* flödet för system-till-system-interaktion och med *Token Exchange* för delegerad access (dvs när ett system utför anrop för en annan intressent, t.ex. en inloggad användares eller ett annat systems räkning).

Regelverket för åtkomstkontroll bör vara attributbaserat [16]. De behörighetsgrundande attributen kan komma från olika källor, såsom HSA-katalogen eller federationskataloger. Utfärdaren av identitets- och åtkomstintyg kan inkludera en del behörighetsgrundande attribut i de utfärdade intygen, medan andra attribut kan behöva hämtas separat.

Åtkomstbeslut görs med olika granularitet, baserat bland annat på vilka informationskällor beslutet baseras på.

Ett mer grovkornigt åtkomstbeslut kan fattas vid utfärdande av åtkomstintyg, baserat framför allt på attribut/egenskaper hos anroparen (så som t.ex. befattning för en användare eller federationstillhörighet för ett system), medan mer finkornig åtkomstkontroll eller filtrering (ibland benämnt kontroll/reglering av ”sökdjup”) kan behöva göras närmare API producenten baserat t.ex. på attribut/egenskaper hos den efterfrågade resursen. API-producenten kan även välja att delegera sitt åtkomstbeslut till en externaliserad s.k. API-Säkerhetstjänst.



4.3.1.1 Utfärdande av Identitets- och åtkomstintyg

För att kunna exponera REST-baserade APIer i enlighet med referensarkitekturen för IAM, måste åtkomstintyg-utfärdande enligt OAuth 2.0 realiseras. I de fall även inloggning skall kunna verifieras (i fallet med t.ex. Web-applikationer), bör OAuth 2.0 servern även realisera OpenID Connect (OIDC).

För åtkomst baserat på delegerade privilegier för en slutanvändare, används *Authorization Code*. För åtkomst baserat på privilegier för det anropande systemet används *Client Credentials*. Om åtkomst skall baseras på privilegier för det anropande systemet att agera för en slutanvändares eller annat systems räkning, krävs stöd för utökningen RFC8693 *Token Exchange*.

I APIer för samverkan inom ramen för T2 kommer sannolikt åtkomst baserat på privilegier för det anropande systemet att vara vanligast. Värt att notera är att information om ursprunglig slutanvändare fortfarande kan vara en del av interaktionen, men då inte i form av en kryptografiskt validerad biljett. Tilliten är då till det anropande systemet, inklusive dess förmåga att validera slutanvändare och skicka med korrekt information om denne.

4.3.2 Tjänstidentifiering och lastbalansering

Fysiska åtkomstadresser kan behöva ändras dynamiskt över tid, av olika tekniska skäl. Det kan handla om ändrade implementationer, nya versioner, nya driftsmiljöer, nya runtime-instanser etc. Genom en lokal tjänstidentifierings-mekanism backad av en dynamisk tjänstekatalog kan en tjänsts externa anrops-URI hållas konstant trots dynamiska förändringar av de underliggande tjänsternas anropsadresser. Ett sådant logiskt abstraktionslager mellan extern anrops-URI och realiserande runtime-instans(er) ger möjlighet till lastbalansering, behovsstyrd dynamisk skalning av kapacitet liksom stöd för avancerade driftsättningsmönster såsom s.k. *rolling upgrades* där nya versioner kan driftsättas utan nertid.

4.3.3 Monitorering, loggning och statistik

Monitorering, loggning av och statistik över anrop är viktiga redskap för att förstå, styra och förvalta APIer, liksom för att försäkra följsamhet mot regler och överenskommelser (*SLA*). Man behöver oftast kunna agera både reaktivt på uppkomna situationer och proaktivt för taktiska eller strategiska beslut baserad på trender över tid. Informationen som ligger till grund för dessa förmågor är oftast uniform mellan APIer, och de lämpar sig därför väl för att externaliseras och realiseras på ett enhetligt sätt.

4.3.4 Trafikbegränsning, ”throttling”

Regelverk för hur ett API får användas skiljer sig troligtvis åt i olika sammanhang eller samverkansfederationer, och kan ofta samspela med uppfyllnad av SLAer. För att säkerställa korrekt och avtalsenlig hantering av klienter, kan begränsningar för enskilda klienter eller grupper av klienter behöva införas och kontrolleras (t.ex. maximalt antal anrop per tidsenhet och klient, eller maximal storlek på anrop eller svar).

4.3.5 Mellanlagring

Läs-operationer av information som är dyr eller tidskrävande att beräkna, efterfrågas ofta och ändras mer sällan, kan behöva mellanlagras för snabbare och effektivare åtkomst. Det kan ibland finnas tekniska skäl att realisera sådan mellanlagring separerat från den underliggande tjänsten.

4.3.6 Protokoll- och meddelande adaption/transformation

Adaption av protokoll/transportpaketering och/eller meddelandeformat kan möjliggöra att ett API exponeras över fler än ett transportformat (t.ex. både REST och SOAP) eller meddelandeformat (t.ex. både XML och JSON). Det kan även finnas behov av meddelande-transformationer för att stödja flera olika major-versioner, eller för att göra anpassningar mot bakomliggande legacy-APIer.

4.3.7 Livscykelhantering av APIer

När Ineras samlade antal publicerade APIer ökar, kommer nya krav att ställas på effektiv och ändamålsenlig hantering av APIerna från initial design, utveckling, kvalitetssäkring, dokumentation, driftsättning och konfigurationsstyrning.

4.3.7.1 Design, utveckling, versionering

Gemensamma, tydliga och lättillgängliga anvisningar och rekommendationer för design, utveckling och förvaltning av APIer är en viktig förutsättning för enhetlighet och förvaltningsbarhet. En gemensam, explicit strategi för versionshantering av APIernas informationskontrakt är likaså kritisk. Verktygsstöd för automatisk kontroll av följsamhet mot regler och riktlinjer kan vara värdefullt.

4.3.7.2 Test och kvalitetssäkring

Möjlighet att effektivt och kostnadseffektivt kunna kvalitetssäkra både API-klient och -producent är kritisk för lyckad API-användning. Ett bra stöd för automatiserade API-tester, både kontraktsbaserade och för hela flöden end-to-end, är då en viktig förutsättning. Gemensamma, tydliga och lättillgängliga anvisningar och rekommendationer för test och kvalitetssäkring av APIer är en viktig förutsättning för enhetlighet och förvaltningsbarhet. Verktygsstöd för generering av stub-implementationer av klienter och producenter från formella API-specifikationer kan även vara värdefullt.

4.3.7.3 Policy-hantering

Realisering av gemensamma, generella förmågor i en API-gateway ger ett flertal fördelar, men innebär i sig också en viss ökat komplexitet. Ett effektivt och uniformt sätt att uttrycka, förvalta, visualisera och kvalitetssäkra de regler som styr de gemensamma förmågorna i en API-gateway blir nödvändig då antalet API-er växer. Begreppet *Policy* och tillhörande mekanismer för *policy-hantering* adresserar detta behov, och ger ett sammanhållande ramverk för att regel-styra olika aspekter av APIers livscykel.

4.3.7.4 Dokumentation och exempel

När antalet publicerade APIer ökar, kommer nya krav att ställas på möjligheten att tillgängliggöra adekvat, korrekt och aktuell dokumentation över APIerna och hur de skall användas. Dokumentationen måste vara enkel för befintliga och potentiella nya klienter till APIet att hitta och använda. Genom att samla dokumentation och exempel för alla Ineras APIer i gemensamma portaler, kan dokumentationen göras enhetlig och enkelt åtkomlig.

4.4 Inera som API klient

En API klient i ett samverkansmönster enligt T2 får också ett utökat ansvar jämfört med T-boken. Klienten måste göra tjänsteuppslag mot federationens tjänstekatalog, för att dynamiskt få tag i producentens tekniska anropsadress. Av prestandaskäl eller för att garanterna hög tillgänglighet, kan resultatet av tjänsteuppslaget behöva mellanlagras. Klienten kan även behöva anpassa sig till den eller de mekanismer för åtkomstkontroll som federationen föreskriver, till exempel genom att autentisera sig och hämta ut ett åtkomstintyg från en utpekad åtkomstintygstjänst.

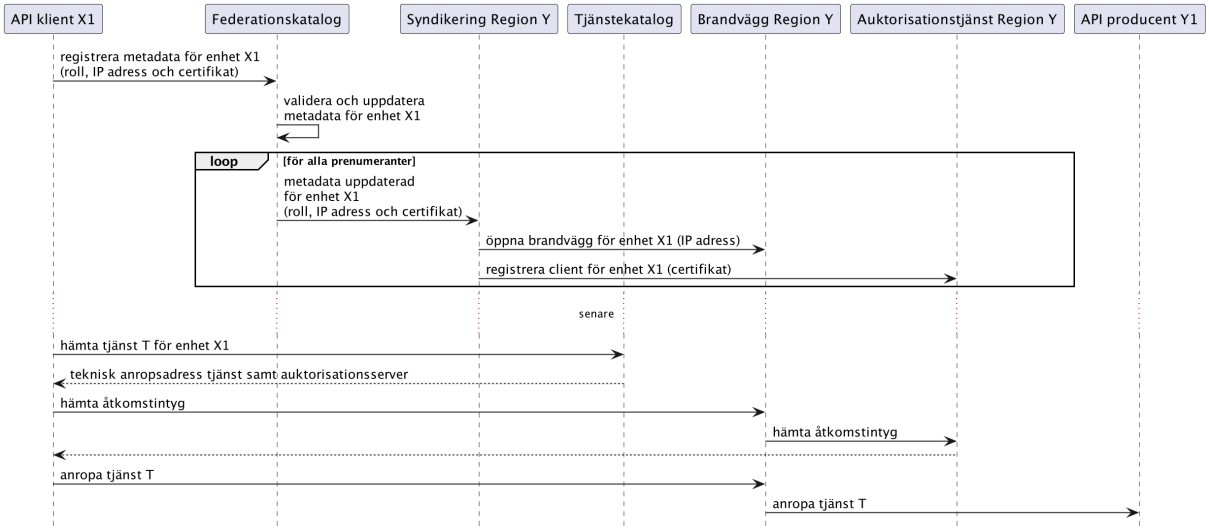
Om flera interna komponenter eller tjänster har likartade behov av att konsumera samma API(er), kan det finnas skäl att externalisera dessa kapabiliteter till en gemensam, förädlande ”proxy”-tjänst. På så vis kan man samla tekniska detaljer kring uppfyllnad av T2s samverkansmönster på ett ställe, och på motsvarande sätt förenkla de anropande klienterna. Det får dock vägas mot det beroende till proxy-tjänsten som då introduceras.

5 Lösningsalternativ

5.1 Stödtjänster för federationsoperatör

5.1.1 Federationskatalog

Federationskatalogens syfte på en konceptuell nivå är relativt tydligt – att administrera metadata om en federation och dess medlemmar ur både avtalsmässigt, organisatoriskt, semantiskt och tekniskt perspektiv. Ett inledande arbete som skissar på design-alternativ för federationskatalog har gjorts i exemplifieringen av T2 (se [20]). Federationskatalogens tekniska användningsområden behöver dock konkretiseras: Hur skall metadata användas av federationsmedlemmarna, och hur behöver katalogen distribueras/synkroniseras till medlemmarna? Arbetshypotesen är en händelse-driven mekanism för prenumeration på metadata-förändringar som kan användas av ingående parter för t.ex. brandväggsöppning och konfiguration av åtkomstkontroll.

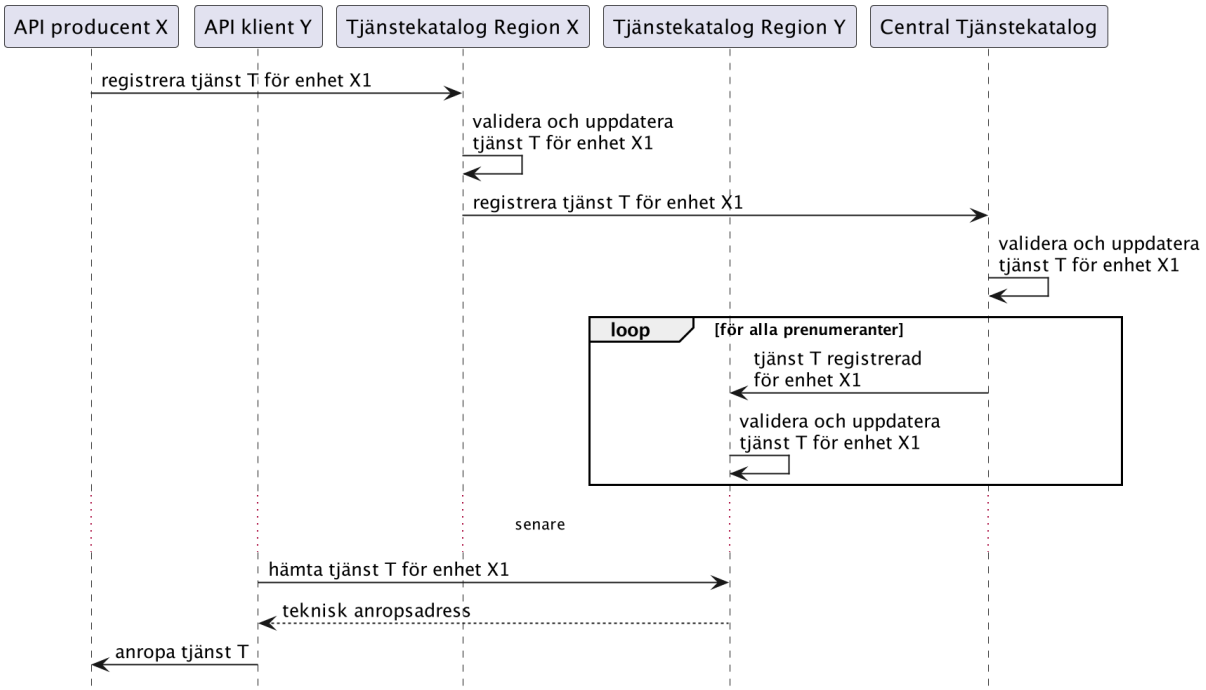


Katalogfunktionalitet verkar under alla händelser vara central, så val av och eventuell anpassning av någon befintlig katalog-produkt är nog en trolig lösning för långsiktig realisering. Lösningen bör sannolikt även linjeras med Referensarkitektur för grunddata och katalog [21].

5.1.2 Tjänstekatalog

En utredning som skissar på olika design-alternativ för tjänstekatalog har gjorts (se [18]). Av den framgår att APIerna för tjänstesökning och -registrering är relativt enkla. Om en syndikerad lösning bestående av lokala tjänstekataloger med central aggregering skall stödjas, så kan syndikeringen göras på olika sätt. Arbetshypotesen är en händelse-driven mekanism för

prenumeration på förändringar som i kombination med tjänsteregistrering och utökad tjänstesökning ger möjlighet till lokal-central syndikering utan påverkan på den centrala katalogen.



Precis som för federationskatalogen verkar Katalogfunktionalitet vara central, så val av och eventuell anpassning av någon befintlig katalog-produkt är nog en trolig lösning för långsiktig realisering. Alternativt skulle NTjPs befintliga Tjänsteadresseringskatalog eventuellt kunna utgöra grund för denna utökade funktionalitet.

5.1.3 Autentisering och åtkomsthantering

Referensarkitekturen för IAM revision B pekar ut OAuth2 som föredragen standard för server-till-server-integration mot REST/JSON-baserade APIer. När revision B godkänts, återstår ett arbete att utreda och ta fram realiserings-anvisningar för referensarkitekturens mönster för T2-baserad samverkan såväl som för interna APIer. En specifik frågeställning att utreda och detaljera är en skalbar lösning för federativ tillit till system-identiteter. Beroende på vilken lösning man väljer, så utgör federationskatalogen troligtvis en ingrediens i lösningen (se t.ex. skiss ovan).

Konkreta och detaljerade realiserings-anvisningar är viktiga både för regionernas planering av sina IAM-förmågor liksom för Inera i rollen som API-producent.

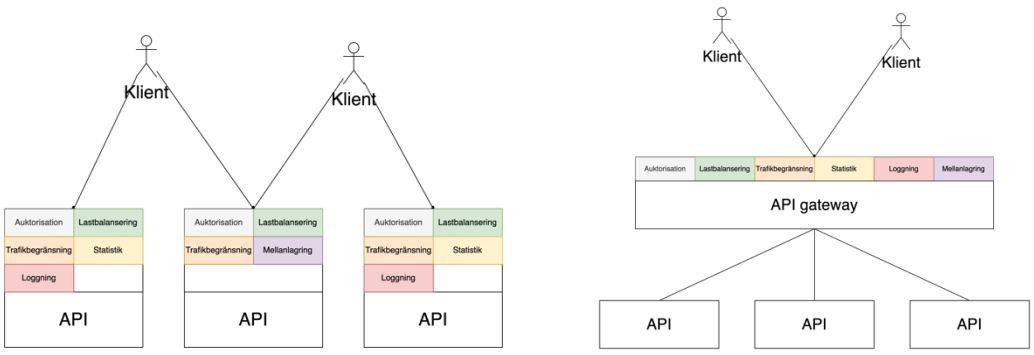
5.2 Stödtjänster för API producent

5.2.1 Gemensam OAuth 2.0 förmåga

Det kan finns skalfördelar med en Inera-gemensam OAuth 2.0 tjänst. En gemensam tjänst ställer dock krav på tjänstens möjligheter till konfiguration och anpassning till de olika APIernas behov, t.ex. m.a.p. vilka ”scopes” som skall erbjudas och vilka regler som gäller för deras utfärdande. Beroende på i vilka sammanhang APIerna skall används, kan även stöd behövas för federativ tillit mellan system för att nå en administrativt hanterbar lösning.

5.2.2 Realisering av gemensamma förmågor genom API gateway

Likartade behov av generella förmågor såsom autentisering, åtkomstkontroll, teknisk adressering, lastbalansering, mellanlagring, loggning, statistik etc för APIer realiseras ofta med hjälp av en *API gateway*. En API gateway är ett arkitekturellt mönster som introducerar en transparent mellanhand mellan API:er och deras klienter. Dess huvudsakliga roll är att agera som en gemensam kontaktpunkt och standardiserade process för interaktion mellan en organisations APIer och dess klienter (både externa och interna). All trafik passerar genom denna gateway, som då kan realisera gemensamma förmågor baserat på gemensamma och/eller API-specifika regelverk.



Fördelarna som därmed kan uppnås är flera: De underliggande APIerna avlastas, samtidig som de gemensamma förmågorna kan realiseras och förvaltas effektivt och uniformt, och därmed också med möjlighet till bättre kvalitet. En API gateway kan även agera fasad för klienters räkning, och därmed dölja underliggande APIers komplexitet eller förändringar.

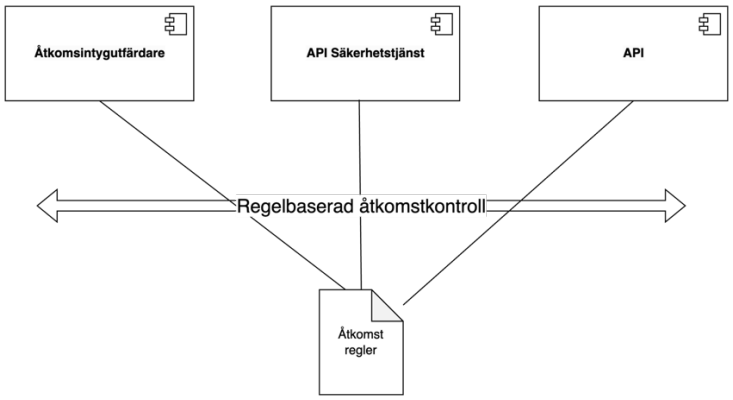
En API gateway utgör därmed en central beståndsdel i en modern API strategi för en organisation. En API gateway ingår nästan alltid som en central komponent i mer heltäckande plattformar för API hantering (s.k. API Management Platform, APIM).

5.2.2.1 Autentisering och åtkomstkontroll

Grovkorniga åtkomstbeslut (såsom rättighet att över huvud taget få göra ett anrop) kan ofta realiseras genom en mekanism som är gemensam för flera APIer. På så vis kan APIerna avlastas från lågnivå-detaljer kring autentisering och åtkomstkontroll, och en uppsättning gemensamma, generella policies för autentisering och åtkomstkontroll definieras.

Referensarkitektur för Identitet och Åtkomst benämner denna typ av generellt säkerhetslager API Säkerhetstjänst. Det kan med fördel realiseras av en API gateway.

Hur stor del av åtkomstkontroll-regler som kan och bör externaliseras till en API gateway beror på hur komplext regelverket för åtkomstkontroll är, och hur sofistikerade och kraftfulla policy-regler som kan uttryckas i en specifik API gateway.



5.2.2.2 Tjänstidentifiering och lastbalansering

Tjänstidentifiering (Service Discovery) är en implicit förmåga hos en API gateway, där externa URIer knyts till bakomliggande, interna URIer via dynamiska regler. Det ger en naturlig lös koppling mellan externa URIer (som kan hållas stabila över tid) och de bakomliggande, interna URIer som kan ändras över tid. Det ger också möjlighet till lastbalansering mot bakomliggande APIer. I moderna s.k. container-baserade applikationsplattformar som t.ex. Kubernetes integreras oftast en API gateway med plattformens s.k. Ingress service, vilket öppnar upp för plattformens fulla dynamiska förmågor såsom automatisk skalning baserat på resursbehov liksom sofistikerade driftsättnings- och uppdateringsmekanismer.

5.2.2.3 Monitorering, loggning och statistik

Monitorering, loggning och statistik är också exempel på förmågor som en API gateway brukar användas till.

5.2.2.4 Trafikbegränsning och -styrning

Regelstyrda begränsningar för enskilda klienter eller grupper av klienter (t.ex. maximalt antal anrop per tidsenhet och klient, eller maximal storlek på anrop eller svar) är andra exempel på förmågor som en API gateway kan realisera.

5.2.2.5 Mellanlagring

Mellanlagring av statisk information som efterfrågas ofta kan realiseras av en API-gateway.

5.2.2.6 Protokoll- och meddelande adaption/transformation

Adaption av protokoll/transportpaketering och/eller meddelandeformat kan möjliggöra att ett API exponeras över fler än ett transportformat (t.ex. både REST och SOAP) eller meddelandeformat (t.ex. både XML och JSON). Det kan även finnas behov av meddelande-transformationer för att t.ex. stödja flera olika major-versioner, eller för att göra anpassningar mot bakomliggande legacy-APIer.

5.2.3 Realisering av stöd för livscykel hantering av APIer

En explicit strategi för hantering av APIer är en förutsättning för en enhetlig, kostnadseffektiv, och skalbar förvaltning av en växande portfölj av APIer. Initialt är genomtänkta processer och väldokumenterade anvisningar viktigast, men för bättre effektivitet och skalbarhet kan verktygsstöd löna sig. En utvecklarportal kan ge en lättillgänglig och enhetlig vy för APIer (API-kontrakt, användningsdokumentation, stubbar för kontraktstester, statistik, etc). En mekanism för policy-hantering ger ett uniformt sätt att regelstyra centrala aspekter av APIers beteende.

Stöd för utvecklarportaler och policy-hantering (liksom API gateway funktioner) ingår normalt i en s.k. **plattform för API-hantering** (API Management Platform, APIM). Denna typ av verktyg ger stöd för ett tydligt, sammanhållet arbetsflöde och knyter ihop de ingående delarna till en koherent helhet via gemensamma administrationsgränssnitt och konfigurationsmekanismer.

6 Beroenden

Detta avsnitt beskriver projektens funktionella och tidsmässiga beroenden till de identifierade behovs och förmågorna.

6.1 Skriv till VIS

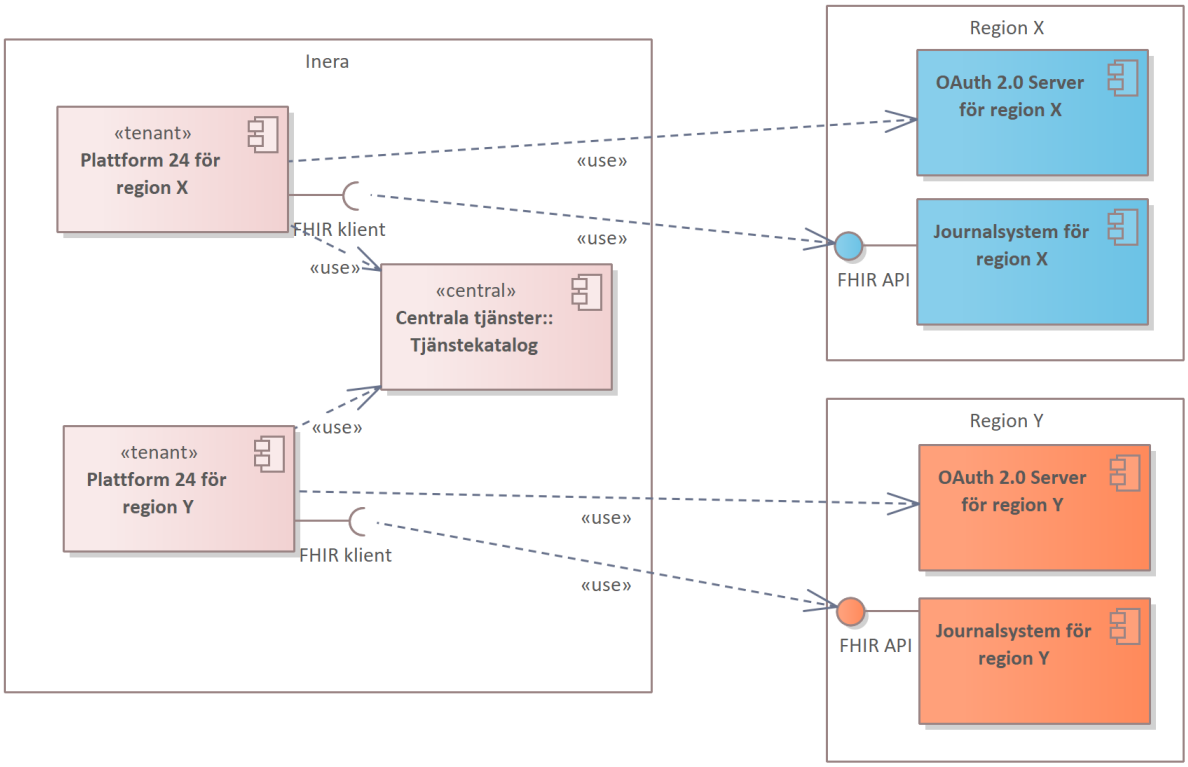
En förstudie har påbörjats för Skriv till VIS, som pågår t.o.m. maj 2023 med ambitionen att starta ett genomförandeprojekt i början på 2024.

6.1.1 Federationskatalog

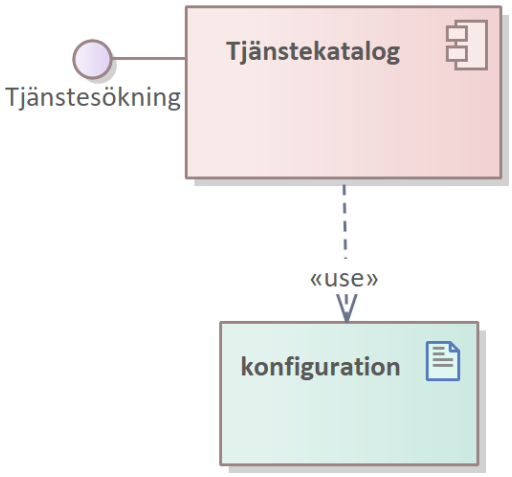
Användning av federationskatalog är sannolikt inte nödvändig för Skriv till VIS, då informationsmönstret initialt är en till en (alternativt en till många) och informationsflöden inte går över organisationsgränser. Teknisk konfiguration som krävs för samverkan (brandväggsregler, åtkomstkontroll m.m.) kan därmed sannolikt göras manuellt i ett första skede. Senare införande av en syndikerad federationskatalog skulle bara påverka dessa konfigurationer, inte de bakomliggande API-producenterna.

6.1.2 Tjänstekatalog

Logisk adressering av API-producenter bör ske genom sökning i central tjänstekatalog. Behov av lokal tjänstekatalog med central aggregering föreligger inte.



De initiala kraven på administration/registrering i tjänstekatalogen är modesta, då adressering kommer vara baserat på vårdgivare/region och antalet anslutna regioner initialt kommer att vara begränsat. Ett administrativt anslutningsförfarande med manuell registrering i tjänstekatalog via administrativt gränssnitt av något slag skulle troligtvis vara bra nog till en början. Tjänsteregistrering och syndikering behöver därmed inte realiseras ännu.



En minimal realisering av en tjänstekatalog skulle därmed kunna göras med en simplistisk implementation av tjänstesökning m.h.a en konfigurationsfil.

6.1.3 Identitets- och åtkomstkontroll för stödtjänster

Behov av identitets- och åtkomstkontroll för Inera-exponerade APIer i Skriv till VIS begränsas till stödtjänsten tjänstekatalogen. I förlängningen bör tjänstekatalogen använda OAuth 2.0 baserad åtkomstkontroll i enlighet med referensarkitekturen för IAM, vilket kräver en realiserad OAuth 2.0 förmåga. Om Skriv till VIS initialt är den enda klienten till tjänstekatalogen och en centralt realiserad OAuth 2.0 förmåga på Inera ännu inte finns på plats, kan det dock vara rimligt att initialt enbart använda mutual TLS för att autentisera och auktorisera anrop mot tjänstekatalogen.

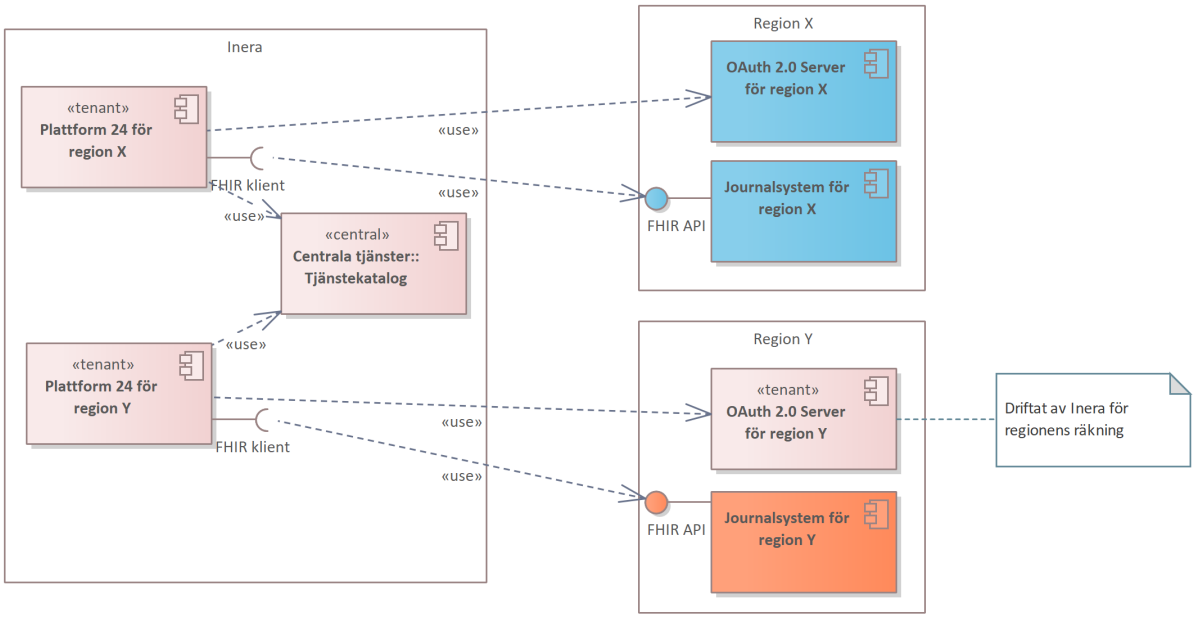
6.1.4 Identitets- och åtkomstkontroll för API producent

Smart-on-FHIR är huvudspåret för Identitets- och åtkomstkontroll för Skriv till VIS. Det finns två möjliga alternativ för Identitets- och åtkomstkontroll inom ramen för Smart-in-FHIR: system-auktorisation via SMART Backend Services (som bygger på OAuth 2.0 *Client Credentials*) eller slutanvändar-auktorisation via SMART App Launch (som bygger på OAuth 2.0 *Authorization Code*). Då slutanvändarna kommer ha behörighet till och arbeta i både Plattform24 och Journalsystemet, är slutanvändar-auktorisation via SMART App Launch ett troligt val. System-auktorisation via SMART Backend Services bygger på att Plattform24 kan använda en unik system-identitet för varje region, vilket i dagsläget är oklart.

6.1.4.1 Åtkomstkontroll för regioners APIer

Det har uttryckts oro att några regioner (t.ex. Region Skåne) som är aktuella för Skriv till VIS inte har möjlighet eller önskan att etablera egen OAuth 2.0 förmåga. Skriv till VIS signalerar därför ett

eventuellt behov/möjlighet för Inera att erbjuda en OAuth 2.0 förmåga som tjänst för dessa regioner.

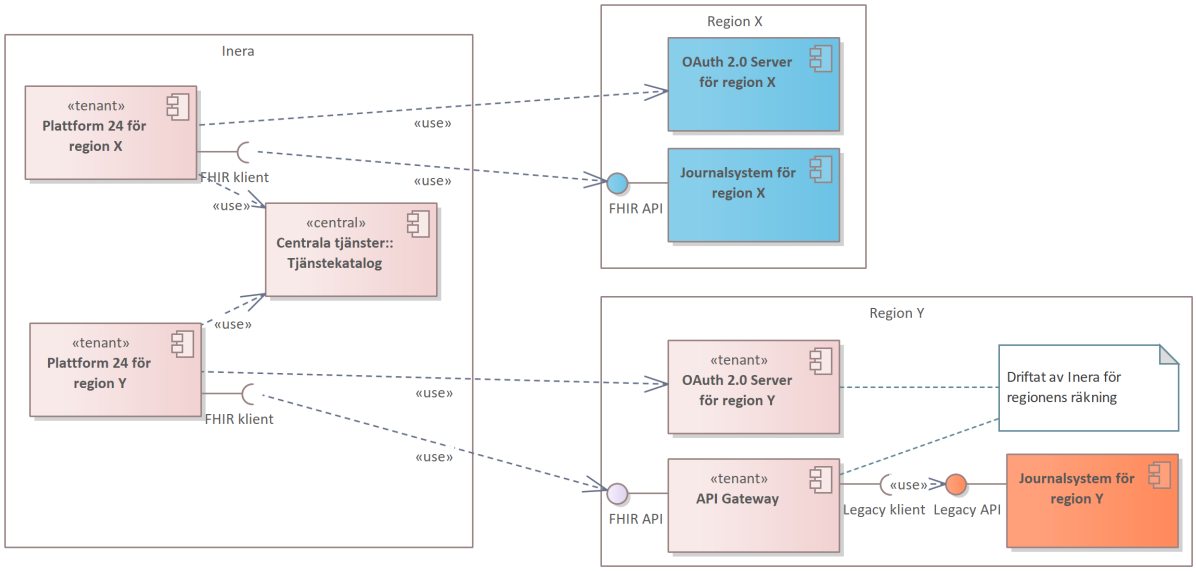


Inera blir då en service leverantör av en OAuth 2.0 tjänst till regionen, men regionen ansvarar själv för och förvaltar innehållet i tjänsten (klient-uppsättningar och åtkomstregler).

Huruvida det är något Inera skall och vill göra är ett separat strategiskt affärsbeslut. Det skulle under en övergångsperiod kunna vara en viktig möjliggörare för mindre aktörer att kunna samverka enligt villkoren för T2.

6.1.4.2 Protokoll- och meddelande adaption mot regioners legacy APIer

Det har även uttryckts oro att några regioner (t.ex. Region Skåne) som är aktuella för Skriv till VIS inte heller har förmåga att fullt ut uppfylla den överenskomna interoperabilitetsspecifikationen. Skriv till VIS signalerar därför ett eventuellt behov/möjlighet för Inera att erbjuda en anpassnings-förmåga som tjänst för dessa regioner, realiserad t.ex. genom en API gateway.



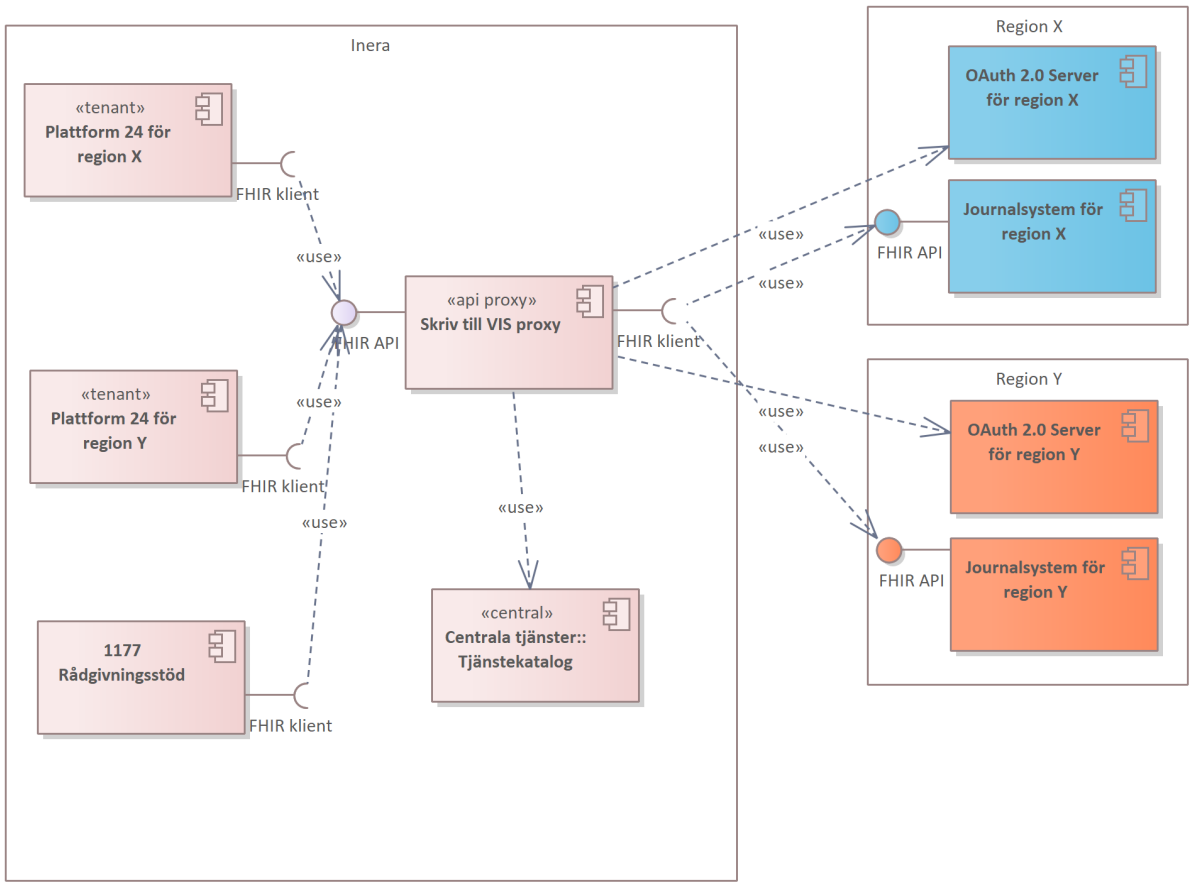
Inera blir då en serviceleverantör av en anpassningstjänst till regionen, men regionen äger och ansvarar själv för anpassningarna (inklusive eventuell CE-märkning).

Huruvida det är något Inera skall och vill göra är ett separat strategiskt affärsbeslut. De ansvarsmässiga implikationerna (t.ex. med avseende på CE-märkning) är en komplicerande faktor.

6.1.5 API proxy

Samverkan enligt T2 ställer krav på API-klient att realisera logisk addressering baserad på sökning i tjänstekatalog, liksom på autentisering mot OAuth 2.0 server.

Beroende på vilka klient-förmågor som Plattform24 har med avseende på OAuth 2.0, FHIR och Smart-on-FHIR klientförmåga, kan ytterligare anpassningar eventuellt behöva göras. Det kan då finnas skäl att externalisera dessa anpassningar, och realisera dem i en separat komponent (API proxy) som döljer anpassningarna och exponerar ett enklare API för Plattform24. Det är dock inte helt oproblematiskt, då det påverkar ansvarsfördelningen mellan Plattform24 och proxy-komponenten som gör anpassningarna (och som således kan behöva CE-märkas).



Om samma integrations- och anpassningsbehov finns inom andra delar av 1177 (t.ex. nya Rådgivningsstödet, Stöd och behandling, 1177 ärendehantering och eventuellt även Formulärhantering) eller från andra Inera-system skulle det även kunna finnas skäl att bryta ut denna API proxy till ett separat förvaltningsobjekt som kan användas av fler klienter. Nyttan med detta måste dock vägas mot de nackdelar som uppkommer med ett beroende till en gemensamt förvaltad komponent.

En API proxy kan realiseras med olika tekniker, bland annat med hjälp av en API gateway.

6.2 1177 e-tjänster

1177 e-tjänsters initiala behov under 2023 är att betrakta som interna APIer mellan mer fingranulära, modulariserade delar av 1177-leveransen. I nästa steg kan några av dessa APIer tänkas göras tillgängliga för andra tjänster inom Inera, och på ytterligare sikt även exponeras externt.

6.2.1 Federationskatalog

Användning av federationskatalog är initialt inte nödvändig för 1177 e-tjänster, då ingen federation krävs för intern användning. När 1177 APIer skall exponeras externt, kommer federations-syndikering eventuellt krävas för en skalbar lösning för åtkomstkontroll.

6.2.2 Tjänstekatalog

Användning av tjänstekatalog enligt T2 är inte nödvändig för intern exponering av 1177 e-Tjänster. Tjänstidentifiering via t.ex. API gateway torde vara en fullgod lösning för att åstadkomma lös koppling mellan stabil extern URI och teknisk anropsadress.

När 1177 APIer skall exponeras mot externa system kan en tjänstekatalog bli aktuell (1177 målarkitektur pekar ut en extern tjänstekatalog för detta ändamål).

6.2.3 Identitets- och åtkomstkontroll

En realiserad OAuth 2.0 förmåga krävs för följsamhet mot referensarkitekturen för IAM. 1177 APIer skall skyddas med krav på åtkomstbiljett enligt OAuth 2.0.

För 1177s egna gränssnitt för Medarbetare (i form av sidor och vyer) skulle detta kunna vara i form av en åtkomstbiljett för inloggad användare (med *Authorization Code* Flow enligt OIDC), men auktorisation av anropande system (dvs 1177 användargränssnitt) genom system-till-system-auktorisation (med *Client Credentials* Flow) är troligtvis en fullt adekvat lösning.

När 1177 APIer skall exponeras utanför 1177 eller mot externa system, krävs också system-till-system-autentisering. Federations-syndikering kommer eventuellt att krävas för en skalbar lösning för konfiguration av API-klienters anropsidentiteter liksom för åtkomstregler.

6.2.4 Realisering av gemensamma förmågor genom API gateway

1177 målarkitektur pekar ut ett ”API lager” för realisering av gemensamma förmågor:

- Åtkomstkontroll
- Tjänstidentifiering
- Mellanlagring
- Användningsstatistik
- Konvertering av transportpaketering

En API gateway är lämplig för realisering av dessa förmågor.

1177 målarkitektur pekar även ut en ”API proxy” vid anrop av externa APIer, för att kapsla in och dölja detaljer om det anropade systemet. En API gateway kan vara en lämplig realisering även av en API proxy.

6.3 Formulärhantering

Färdigställande och exponering av FHIR-API för formulärhantering har temporärt pausats i väntan på saknade gemensamma förmågor. Produktägarens önskan är att exponera FHIR-APIet för formulärhantering så snart som möjligt.

6.3.1 Federationskatalog

Användning av federationskatalog för Formulärhantering beror på om en formell federation är nödvändig. Ett anslutningsförfarande kan eventuellt initialt skötas manuellt för tjänsten, och accesskontroll skötas med manuell konfiguration av anslutna klienter i gemensam OAuth 2.0 server.

6.3.2 Tjänstekatalog

Användning av tjänstekatalog enligt T2 är inte nödvändig för Formulärhantering, då det är en många-till-en tjänst. Tjänstidentifiering via t.ex. API gateway torde vara en fullgod lösning för att åstadkomma lös koppling mellan stabil extern URI och teknisk anropsadress.

6.3.3 Identitets- och åtkomstkontroll

En realiserad OAuth 2.0 förmåga krävs för följsamhet mot referensarkitekturen för IAM. Formulärhanterings FHIR API skall skyddas med krav på åtkomstbiljett från anropande system enligt OAuth Client Credentials. Federations-syndikering kommer eventuellt att krävas för en skalbar lösning för konfiguration av API-klienters anropsidentiteter och certifikat. På kort sikt skulle dock en initial OAuth 2.0 förmåga med stöd för server-till-server-auktorisation vara tillräcklig för att exponera formulärhanterings FHIR-API till ett mindre antal konsumenter.

6.3.4 Realisering av gemensamma förmågor genom API gateway

Formulärhantering har behov av realisering av följande gemensamma förmågor:

- Åtkomstkontroll
- Tjänstidentifiering
- Loggning, monitorering, statistik

En API gateway är lämplig för realisering av dessa förmågor.

6.4 Terminologitjänst

Terminologitjänsten vill kunna exponera sitt FHIR-API externt under Q3 2023.

6.4.1 Federationskatalog

Användning av federationskatalog är initialt inte nödvändig för Terminologistjänster, då ingen formell federation är nödvändig. Ett anslutningsförfarande kan eventuellt initialt skötas manuellt för tjänsten, och accesskontroll skötas med manuell konfiguration av anslutna klienter i egen eller gemensam OAuth 2.0 server, alternativt i API gateway med egen accesskontrolllista (ACL).

6.4.2 Tjänstekatalog

Användning av tjänstekatalog är inte nödvändig för Terminologistjänster, då ingen formell federation är nödvändig och det är en många-till-en tjänst. Tjänstidentifiering via t.ex. API gateway torde vara en fullgod lösning för att åstadkomma lös koppling mellan stabil extern URI och teknisk anropsadress.

6.4.3 Identitets- och åtkomstkontroll

Terminologitjänstens initiala behov av identitets- och åtkomstkontroll är initialt ytterst modesta: Läs-access till tjänstens FHIR-API skall beredas de organisationer som anslutit till tjänsten. I ett senare skede ser man eventuellt behov av att även tillåta skriv-access till den eller de ”sektorer” som en organisation äger.

Om terminologitjänsten väljer att använda Smart-on-FHIR för identitets- och åtkomstkontroll, krävs en realiserad OAuth 2.0 förmåga. Denna kan realiseras med egen OAuth 2.0 server i form av Ontocloak. Det kan dock finnas skalfördelar med en centralt realiserad OAuth 2.0 förmåga på Inera.

Alternativt kan det vara rimligt att initialt använda mutual TLS för autentisering av API-klient och realisera åtkomstkontroll med egen ACL via en API gateway.

6.4.4 Realisering av gemensamma förmågor genom API gateway

Terminologitjänsten har behov av realisering av följande gemensamma förmågor:

- Åtkomstkontroll
- Tjänstidentifiering
- Loggning, monitorering, statistik

En API gateway är lämplig för realisering av dessa förmågor.

6.5 Sömlöst vård flöde

Förmåga att skicka bilagor via referens kommer sannolikt kräva en fullvärdig realisering av de stödtjänster som krävs av en federationsoperatör enligt T2, då aktörsmönstret för eRemiss är många till många.

6.5.1 Federationskatalog

En federationskatalog med API för federationsregistrering/hantering av klienters metadata och automatisk syndikering är sannolikt nödvändig för skalbar hantering av federationens trafik och åtkomstregler.

6.5.2 Tjänstekatalog

En tjänstekatalog med API för tjänstesökning och -registrering är nödvändig för skalbar hantering av logiska adresser och tillhörande metadata.

6.5.3 Identitets- och åtkomstkontroll för Federations- och Tjänstekatalog

Strikt identitets- och åtkomstkontroll krävs för federations- och tjänstekatalogernas registrerings-APIer. Minimalt krävs en realiserad OAuth 2.0 förmåga för grovkorniga åtkomstbeslut (anrop till APIerna endast för medlemmar i federationen). Mer finkorniga åtkomstbeslut kan realiseras genom ett API säkerhetslager (via en API gateway) och/eller i APIerna.

6.6 Regelverk enskilds direktåtkomst i invånarapplikationer

Den initiala analysen av Gemensamt regelverk för enskilds direktåtkomst [25] identifierar 2 principiella alternativ för konsolidering av regelverk: Lösningsförslag 1 tillgängliggör ”filtreringsregler” via ett eller flera APIer, som respektive applikation därefter tillämpar på den journalinformation som hanteras. Lösningsförslag 2 tillgängliggör istället en ny central tjänst för filtrerad journalinformation (i vilken journalinformation redan filtrerats baserat på regelverket).

Lösningen behöver utredas bättre för att förstå vilka krav på stödtjänster för samverkan de ställer. Under alla omständigheter finns troligtvis behov av realisering av följande gemensamma förmågor:

- Åtkomstkontroll baserat på anslutningsförfarande för betrodda tredjeparts-applikationer, eventuellt via federationskatalog
- Tjänstidentifiering via publik, välkänd URI
- Loggning, monitorering, statistik

7 Tids- och kostnadsuppskattningar

Detta avsnitt ger grova tids/kostnadsuppskattningar för de föreslagna åtgärderna. Uppskattningarna är uttryckta i idealiserad utrednings/utvecklingstid: den tid som en kompetent och erfaren utförare på fulltid skulle behöva för att lösa uppgiften. Uppskattningarna inkluderar således inte eventuell uppstartstid, tid för avstämning eller förankring med interna eller externa intressenter eller annan normal projektoverhead. Lämplig faktor bör därför appliceras på tidsuppskattningarna för att omvandla dem till realistiska projektestimat. Då flera av uppgifterna involverar avstämning och förankring med regionerna, krävs sannolikt betydligt längre kalendertid.

7.1 Federationskatalog

7.1.1 Förstudie, lösning för T2 Federationskatalog

Ett förberedande arbete för federationskatalog har gjorts i samband med exemplifiering av T2 [20]. Detta arbete behöver fördjupas:

- Utred tekniska behov och förutsättningar för syndikering
- Modellera informationsdomänen för Federationskatalog
- Designa APIer för Federationssökning, Federationsregistrering och Federations-syndikering
- Utred realiseringsalternativ, och utvärdera eventuellt möjliga tredjepartsprodukter
- Förbered eventuell intresseanmälan och avsiktsförklaring

En förstudie bör kunna genomföras på 3 manmånader. Lämpligt är att denna utredning utförs inom sektionen Samverkansarkitektur av en mindre grupp på 2-3 personer.

Förstudiens föreslagna lösning behöver därefter realiseras, och en förvaltningsorganisation etableras. Det är rimligt att denna hamnar under Plattformar, eventuellt som en utökning av det befintliga förvaltningsobjektet Tjänsteadresseringskatalog (TAK) inom NTjP.

7.2 Tjänstekatalog

7.2.1 Design och realisering, minimal Tjänstekatalog för Skriv till VIS

Beroende på hur tidplanen för ett implementationsprojekt för Skriv till VIS ser ut, kan en minimal realisering av tjänstesökning behöva tas fram för Skriv till VIS behov.

- Modellera informationsdomänen för Tjänstekatalog
- Designa APIer för Tjänstesökning och Tjänsteregistrering

- Implementera Tjänstesökning med enkel konfigurationsmekanism, men med produktionsfärdig paketering för driftsättning.
- Driftsätt lösning, med initial konfiguration för Skriv till VIS.

En möjlighet skulle kunna vara att realisera tjänstesöknings-APIet inom ramen för det befintliga förvaltningsobjektet Tjänsteadresseringskatalog (TAK) inom NTjP, antingen genom att utöka TAK-komponenten med ett nytt API eller som en ny teknisk komponent inom samma förvaltningsobjekt. Fördelarna med att utnyttja befintlig förvaltning behöver vägas mot nackdelarna (osäkerhet kring påverkan på befintlig TAK, behov av omtestning etc.).

Beroende på vilken förvaltningslösning som väljs, bör ett implementationsprojekt samt driftsättning kunna genomföras på 1-2 manmånader.

7.2.2 Förstudie, fullvärdig lösning för T2 Tjänstekatalog

Ett förberedande arbete kring tjänstekatalog har redan gjorts (se [14] och [16]). Detta arbete behöver fördjupas:

- Utred tekniska behov och förutsättningar för lokal-central syndikering.
- Förfina informationsdomänen för tjänstekatalog.
- Förfina APIer för tjänstesökning och tjänsteregistrering.
- Designa APIer för tjänstesyndikering.
- Utred realiseringsalternativ, och utvärdera eventuellt möjliga tredjepartsprodukter
- Förbered eventuell intresseanmälan och avsiktsförklaring

En förstudie bör kunna genomföras på 2 manmånader. Lämpligt är att denna utredning utförs inom sektionen Samverkansarkitektur av en mindre grupp på 2-3 personer.

Förstudiens föreslagna lösning behöver därefter realiseras, och en förvaltningsorganisation etableras. Det är rimligt att denna hamnar under Plattformar, eventuellt som en utökning av det befintliga förvaltningsobjektet Tjänsteadresseringskatalog (TAK) inom NTjP.

7.3 IAM förmågor

7.3.1 Realiseringsanvisningar för IAM referensarkitektur

Ta fram konkreta realiseringsanvisningar med exempel för referensarkitekturen för IAM. Anvisningar och exempel bör möjliggöra stegvis, inkrementell realisering av mönster i referensarkitekturen.

Anvisningar och exempel bör kunna tas fram på 1-2 manmånader. Lämpligt är att detta arbete utförs inom sektionen Samverkansarkitektur, i samarbete med Arkitektursektionen.

7.3.2 Realisering, initial central OAuth 2.0 tjänst

Tidsuppskattningar för en stegvis realisering av OAuth 2.0 tjänst är svårt att göra innan realiseringsanvisningar finns på plats som detaljerar stegen och ger ramar och omfattning. Implementationen kommer troligtvis göras av Ineras befintliga leverantör, som därmed måste involveras i planeringen.

En arbetshypotes är dock att en första leverans skulle kunna vara en OAuth 2.0 tjänst som enbart stödjer IAM-mönstren med tillit baserat på inloggad slutanvändare (Authorization Code) eller baserat på system-identitet utan delegerad användar-behörighet (Client Credentials med mutual TLS eller private jwt key). En försiktig gissning är att en sådan lösning skulle kunna realiseras under 2023.

7.3.3 Realisering, fullvärdig central OAuth 2.0 tjänst

Det är inte vettigt att spekulera i omfattning innan konkreta realiseringsanvisningar har tagits fram och diskuterats med Ineras befintliga leverantör av säkerhetstjänster.

7.4 Gemensamma förmågor för API klient och producent

7.4.1 Utvärdering, val och realisering av taktisk lösning för API gateway

Utvärdera API gateway produkter och välj produkt som uppfyller Terminologitjänstens omedelbara behov, och som i bästa fall kan utgöra grund för en långsiktig lösning för API management.

- Gör en kort inventering av möjliga produkter
- Värdera produkterna utifrån Terminologitjänstens omedelbara behov, men väg in möjlighet att bredda till en mer fullvärdig för API management
- Välj produkt och implementera minimal lösning i samarbete med Terminologitjänsten.

Utredning, val av produkt och minimal implementation bör kunna genomföras på 2 manmånader. Lämpligt är att detta arbete utförs inom sektionen Arkitektur, i nära samarbete med Terminologitjänsten. Syftet med arbetet är inte enbart att tillgodose Terminologitjänstens behov men även att få praktisk återkoppling och förståelse för behov, utmaningar och möjligheter med standardiserade API gateways.

7.4.2 Realiseringsanvisningar för REST-baserade APIer

Ta fram konkreta realiseringsanvisningar med exempel för REST-baserade APIer, med utgångspunkt i existerande anvisningar från t.ex. DIGG [15]. Anvisningar, riktlinjer och exempel

bör täcka in design av APIer och informationskontrakt, strategi för versionering, dokumentation liksom metod och ansats för test och kvalitetssäkring av APIer.

Anvisningar och exempel bör kunna tas fram på 1 manmånad. Lämpligt är att detta arbete utförs inom sektionen Arkitektur.

7.4.3 Förstudie, fullvärdigt verktygsstöd för API management

Ett gediget initialt arbete kring integrationsarkitektur har gjorts (se [23]), och ett utredande arbete kring behov och lösningar för API Management genomförs inom förstudien för Skriv till VIS. Detta arbete bör prioriteras, och kanske bedrivas som ett eget initiativ.

- Utforma Inera-gemensam strategi för API hantering och integration
- Utred omfattning och behov av process- och verktygsstöd för effektiv och skalbar API hantering i enlighet med strategin, förankra eventuellt med regionerna
- Förbered eventuell intresseanmälan och avsiktsförklaring
- Kravställ och utvärdera APIM produkter

En förstudie bör kunna genomföras på 2-3 manmånader. Lämpligt är att detta arbete utförs inom sektionen Arkitektur, i samarbete med Skriv till VIS och 1177 e-Tjänster.

Beroende på förstudiens resultat kan föreslaget systemstöd därefter behöva realiseras, och en förvaltningsorganisation eventuellt behöva etableras.