TOP-KT-007 - Koppeltaal Launch

Versiegeschiedenis...

Versie	Datum	Status	Wijzigingen
2.0.7	19 jan 2026	definitief	Tekstuele correcties doorgevoerd zonder wijziging aan de inhoud
2.0.6	6 jan 2026	concept	Verwijziging naar Multiple IdP support in TOP-KT023a toegvoegd aan de beschrijving op meerdere plekken
2.0.5	4 apr 2025	defitief	Verwijziging naar Topic 26 opgenomen (Related Person)
2.0.4	14 Mar 2025	concept	Een aantal koppen waar geen tekst, afbeeldingen of code onder stond zijn verwijderd.
2.0.3.	10 Feb 2025	concept	Onder “De koppeltaal launch in stappen” zijn een aantal zinnen aangepast om de tekst duidelijker te maken.
2.0.2	12 Dec 2023	definitief	Wijzigingen ter verduidelijking Relatie client_id en device reference
2.0.1	5 Dec 2023	definitief	Text aanpassingen betreffende `launch openid fhirUser`
2.0.0	15 Nov 2023	definitief	Wijzing mbt ActivityDefinition.extension[instantiates]
1.0.2	6 Nov 2023	definitief	Wijzigingen aan beschrijving SMART on FHIR
1.0.1	25 Sept 2023	definitief	aud in HTI aangepast naar Device Id (Device/123)
1.0.0	5 Jul 2023	definitief	Geen nieuwe wijzigingen sinds laatste concept. Geen wijzigingen in eisen sinds 0.2.0
0.2.2	26 Jun 2023	concept	Deadlink opgelost
0.2.1	14 Jun 2023	concept	aanpassing diagrammen
0.2.0	13 Mar 2023	concept	HTI versie 2.1 Scope is beter gespecificeerd Het vullen van de HTI token is gespecificeerd De parameters in de verschillende requests van de SMART on FHIR flow zijn verder gespecificeerd volgens spec.
0.1.1	30 Jan 2023	concept	De launch context velden zijn herzien.
0.1.0	12 Jan 2023	concept	

Beschrijving

De primaire use case van koppeltaal is het starten van modules vanuit een EPD of behandelportaal. In deze use case selecteert de gebruiker een taak om de module te starten en wordt de gebruiker door middel van een speciale link naar de juiste module gebracht. Op deze manier wordt het mogelijk in een zorgdomein om modules van verschillende leveranciers naadloos te integreren in het platform.

In koppeltaal wordt de launch gerealiseerd door de twee open standaarden, SMART on FHIR app launch en HTI, te combineren. Deze twee standaarden zijn complementair. HTI maakt het mogelijk de juiste informatie in de launch mee te geven, SMART on FHIR app launch zorgt ervoor dat de juiste validaties en checks voor het ontsluiten van gegevens wordt uitgevoerd.

Zowel HTI als SMART on FHIR app launch versie v2.0.0 zijn uitgebreid gedocumenteerd, details over de standaarden zijn daar te vinden.

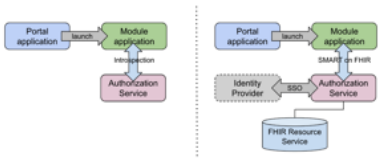
De launch in het kort

De koppeltaal launch bestaat uit een SMART on FHIR app launch met als launch waarde een HTI token. Afhankelijk van of de module persoonsgegeven en/of medische gegevens verwerkt kan de module kiezen voor het valideren van het HTI token bij de autorisatie service of het uitvoeren van een SMART on FHIR app launch. De SMART on FHIR app launch is vereist in het geval dat de module persoonsgegevens en/of medische gegevens verwerkt.

In het eerste geval krijgt de module applicatie de inhoud van het HTI bericht terug na een token introspection van de autorisatie service. Dit bericht bevat de gegevens om een sessie op de module te kunnen starten en de bijbehorende FHIR resources bij de FHIR resource service op te halen. Om met het token introspection endpoint te kunnen communiceren wordt gebruik gemaakt van JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication (rfc7523), zie ook TOP-KT-021 - Token Introspection. Het endpoint wat voor de token introspection van het HTI gebruikt wordt is het endpoint zoals beschreven in TOP-KT-021 - Token Introspection.

In het tweede geval komt de relevante informatie terug uit de laatste stap van de SMART on FHIR app launch flow. Deze informatie bevindt zich in de context van het antwoord.

Verder speelt de autorisatiestap in de SMART on FHIR app launch een belangrijke rol. Hier wordt gevalideerd of de gebruiker die in de launch omschreven staat ook daadwerkelijk de gebruiker is. Dit wordt typisch gedaan door een SSO koppeling.



Overwegingen

De koppeltaal launch maakt gebruik van twee complementaire technologieën. Deze keuze komt voort uit een aantal overwegingen. De overwegingen die hebben geleid tot de vorming van de koppeeltaalstandaard zijn de volgende.

Hoofdlijnen voor de gekozen Launch oplossing en authenticatie

De rechten van de gebruikers die hun applicaties met Koppeltaal verbinden, zijn in Koppeltaal gelaagd opgebouwd. Koppeltaal neemt het autorisatiemodel van de gebruiker in de applicatie als uitgangspunt. Het EPD, platform en module zijn zelf verantwoordelijk voor de autorisatie van de individuele applicatiegebruiker. De Koppeltaal standaard voegt daar een autorisatie systeem van applicatie-instanties aan toe. Gegevens tussen applicaties als EPD's, platform's en modules worden in Koppeltaal 2.0 namelijk beveiligd uitgewisseld op basis van SMART backend services. Hierbij is de applicatie-instantie van EPD, platform of module de identiteit waarop geautoriseerd wordt.

Een veelgebruikte functionaliteit in Koppeltaal is de launch. Bij een launch lanceert een gebruiker vanuit zijn applicatie een andere applicatie, zoals bijvoorbeeld een eHealth module vanuit zijn EPD. Binnen Koppeltaal is de Launch niet gebonden aan één lancerend platform. Meerdere applicatie-instanties kunnen elkaar binnen een domein lanceren. Hierbij baseert Koppeltaal zich op het SMART Application Launch Framework van de FHIR-standaard. Het SMART Application Launch Framework is een beproefd protocol voor het lanceren van modules vanuit een platform en daarmee een goede basis. Toch zijn er enkele belangrijke kanttekeningen voor het gebruik van het SMART Application Launch Framework in Koppeltaal:

1. Het SMART Application Launch Framework hanteert het uitgangspunt dat een gelanceerde module rechten krijgt tot resources die onder het beheer van het lancerende platform vallen. Dit komt binnen Koppeltaal 2.0 niet voor! Koppeltaal geeft geen rechten en op resources die onder het beheer van het lancerende platform vallen. Gegevens worden tussen platform en module uitgewisseld via een FHIR-service op basis van autorisatie met SMART backend services.

2. Het SMART Application Launch Framework hanteert het uitgangspunt dat op gebruikersniveau geautoriseerd wordt voor gegevensuitwisseling tussen applicatie-instanties. En dat is in Koppeltaal 2.0 niet het geval. Zoals aangegeven: in Koppeltaal 2.0 worden rechten gegeven op applicatie-instantie niveau tussen applicaties en binnen applicaties op medewerkers niveau. Dit is voldoende. Een extra autorisatie laag van het toekennen van rechten op gebruikersniveau tussen applicaties-instanties is overbodig en ook verwarrend.

Daarom gebruiken we het SMART on FHIR Application Launch Framework zonder het autorisatie gedeelte Echter, daarbij blijft het wel nodig om een invulling te geven aan de context van de launch. In het SMART on FHIR Application Launch Framework is de context van de Launch besloten het token request. De context geeft bijvoorbeeld aan welke taak moet worden gelanceerd. Het niet gebruiken van het autorisatiedeel noodzaakt het gebruik van een Koppeltaal specifieke extensie in het SMART Application Launch Framework om alsnog de context toe te kunnen voegen. In Koppeltaal is gezocht naar een veilige en toekomstbestendige manier om het overbrengen van intent mogelijk te maken. Daarbij is gekozen voor het HTI-token van Health Tools Interoperability (HTI) voor overdracht van de context. HTI is een technische standaard waarmee leveranciers van eHealth modules kunnen integreren met zorgplatformen. HTI heeft draagvlak in de Zorg als kandidaat-bouwsteen in het duurzaam informatiestelsel voor de Zorg van het informatieberaad Zorg. HTI is bovendien al bij diverse leden uit de community van leveranciers van Koppeltaal in gebruik.

Waarom is alleen het token van HTI gebruikt en niet het gehele Launch concept van HTI? Uit securityonderzoek bleek een risico voor “unsolicited authentication”. “Unsolicited authentication” betekent in dit kader dat onbevoegden een Launch kunnen "overnemen" als een authenticatiestap ontbreekt. In de HTI standaard is dit risico ook onderkent en is voor persoons/medische informatie een authenticatie stap vereist. Voor Koppeltaal maken we voor deze authenticatiestap gebruik van de authenticatiemogelijkheden van het SMART Application Launch Framework. Door deze authenticatie stap van het SMART Application Launch Framework te gebruiken, in combinatie met het token van HTI is aan de security voorwaarden voldaan. Daarbij ontstaat bovendien een SSO (Single Sign On).

Context probleem

SMART on FHIR app launch framework gaat primair uit van een één op meerdere relatie tussen het launching platform en het ontvangende platform, dit om aan het einde van de SMART on FHIR app launch flow de context van de launch mee te kunnen geven. In koppeltaal kunnen er meerdere lancerende platformen zijn. Bij koppeltaal vindt de validatie en het verlenen van toegang van de gebruiker tot de launch plaats in de autorisatie service.

Waar deze autorisatie service in een typische SMART on FHIR app launch gekoppeld lijkt aan het lancerend systeem, of in ieder geval in staat lijkt informatie uit te wisselen, is dat bij koppeltaal niet het geval. Als gevolg mist er informatie in de SMART on FHIR flow die nodig is om de juiste FHIR context aan de module mee te geven vanuit de autorisatieservice in de laatste stap van de SMART on FHIR app launch, samen met een access_token. Over het access_token meer in het onderdeel Het access_token. Het betreft de informatie rond de taak, de patient en de activity definition, deze worden in eigen parameters in het antwoord op het token request meegegeven, er wordt hier bewust door Koppeltaal geen gebruik gemaakt van de `fhirContext`, omdat niet alle referenties FHIR resources zijn. Dit probleem wordt opgelost door gebruik te maken van het HTI token als launch parameter in de SMART on FHIR app launch flow; deze bevat een ondertekend JWT token met daarin deze informatie.

Unsolicited authenticatie

Een aanvraag naar een module in de vorm van een launch, waar de module niet de aanvraag initieert is een zogenaamde ongevraagde (unsolicited) authenticatie. Een dergelijke authenticatie vormt een risico rond de beveiliging. De inhoud van de launch geeft toegang tot gegevens en kan opzettelijk of onopzettelijk onderschept of opgeslagen worden. Zo kan een link in een browsergeschiedenis opgeslagen worden, door cross site scripting onderschept worden, of in de logs van slecht geconfigureerde webservers worden opgeslagen.

HTI is daarom, zonder extra identificatie van de gebruiker, onvoldoende beveiligd om toegang te verlenen tot persoonlijke en/of medische gegevens. HTI erkent dit probleem en stelt daarom ook dat het de ontvangende partij vrij staat extra maatregelen te treffen alvorens toegang te geven tot het systeem en de gegevens. Dit is dan ook de rol van SMART on FHIR app launch framework. Hier wordt in de autorisatiestap de gebruiker door middel van SSO opnieuw geïdentificeerd, en deze gegevens worden gematched met de inhoud van het HTI launch bericht.

> Voor de duidelijkheid, de SMART on FHIR flow is enkel noodzakelijk bij het verwerken van medische en/of persoonsgegevens, indien de module geen medische en/of persoonsgegevens verwerkt, is HTI voldoende. Denk hierbij aan een applicatie die enkel media aanbiedt.

Afhankelijk van de inhoud

De vraag of voor de launch gebruik gemaakt kan worden van HTI of van SMART on FHIR, is afhankelijk van de vraag of de module persoonsgegevens en/of medische gegevens verwerkt. Verwerken de modules van een applicatie-instantie geen gegevens over de gebruiker (bijvoorbeeld door alleen media en tekst aan te bieden), dan verwerkt deze geen persoonsgegevens of medische gegevens. In dergelijk geval is er geen noodzaak voor een extra authenticatie van de user agent en dus van SMART on FHIR. Niet in alle gevallen is er hier noodzaak voor. De aanbieder van de module moet communiceren met de domeinbeheerder of de module persoonsgegevens en/of medische gegevens verwerkt. Koppeltaal doet wél de observatie dat er een noodzaak is binnen alle domeinen zoals deze op dit moment bekend zijn.

Autorisatie en SSO

Zoals genoemd in het onderdeel unsolicited authenticatie is de autorisatiestap binnen het SMART on FHIR app launch framework van wezenlijk belang voor het beschermen van de gegevens. In deze stap moet de autorisatieservice de identiteit van de gebruiker vaststellen die past bij de te ontsluiten gegevens. Hoe die vaststelling plaatsvindt, hangt af van het domein. In de praktijk gaan we er vanuit dat dit door middel van een Single Sign On (SSO) op een Identity Provider (IdP) wordt gedaan. Een IdP vervult een centrale rol bij het inloggen van gebruikers en wordt in veel scenario's ook door portalen gebruikt om de gebruikers te authenticeren. Veelvoorkomende SSO oplossingen zijn SAML en OpenID Connect (OIDC).

Binnen Koppeltaal is het mogelijk dat **meerdere IdP's naast elkaar worden gebruikt**. Om dit te ondersteunen bevat het **HTI-token** dat door de client wordt aangeleverd een veld `idp_hint`. Deze hint geeft aan **welke IdP gebruikt moet worden** voor de authenticatie van de gebruiker. De autorisatieservice controleert vervolgens of deze vertrouwd is en selecteert de bijbehorende sleutelset om het HTI-token cryptografisch te valideren. Hierdoor kunnen verschillende organisaties of domeinen hun eigen IdP blijven gebruiken, terwijl de autorisatieflow uniform en interoperabel blijft. In TOP-KT-023a-Multiple IdP Support voor Patient, Practitioner & RelatedPerson | Oplossingsbeschrijving wordt beschreven hoe Multiple IdP support in de Koppeltaal Standaard wordt ondersteund.

> Koppeltaal dicteert niet wat er in de autorisatie stap exact moet gebeuren, anders dan dat in deze stap de beveiliging passend moet zijn bij de gegevens die ontsloten

> worden. Met passend wordt bedoeld dat het middel van beveiliging past bij de gegevens die ontsloten worden. Worden er geen persoons en/of medische gegevens ontsloten, is een minder of geen middel van toepassing, worden er persoonlijke en/of medische gegevens ontsloten, zijn meer strikte middelen passend. Wat passend is wordt bepaald door verschillende normenkaders (TODO, refereren).

POST vs GET

Hoewel HTI en SMART on FHIR appl launch framework prima met elkaar samenwerken; de launch parameter van de SMART on FHIR app launch is een prima plek waar het HTI token zijn rol kan vervullen, is er tussen beide standaarden een tweetal discrepanties over hoe deze parameter wordt uitgewisseld.

HTI doet de uitwisseling van het HTI token in het `token` parameter in de application/x-www-form-urlencoded encoded POST van de launch, het SMART on FHIR app launch framework doet dit door middel van het `launch` parameter in het GET request van de launch. Koppeltaal maakt de keuze voor het `launch` parameter in een application/x-www-form-urlencoded encoded POST. De keuze hiervoor is tweeledig. Ten eerste: een application/x-www-form-urlencoded encoded POST methode kent geen beperking wat betreft de lengte van de parameters. Ten tweede: een application/x-www-form-urlencoded encoded POST methode is iets veiliger dan een GET methode. Dit omdat de parameters geen deel uitmaken van de URL en dus minder snel in de browser historie of server logs verschijnen.

Het access_token

In koppeltaal is de keuze gemaakt om de applicaties - en niet de individuele gebruikers - toegang te geven op de FHIR resource server. SMART on FHIR app launch framework gaat uit van individuele toegang op een FHIR resource server. Dit laatste manifesteert zich in een access_token dat aan het einde van de SMART on FHIR app launch flow in het token response wordt meegeven. Dit veld is verplicht en kan dus niet leeg gelaten worden. Echter, het heeft in de koppeltaal context geen betekenis, en is eerder een bron van verwarring dan behulpzaam. Daarom kiezen we ervoor de waarde van het access_token in alle gevallen naar NOOP te zetten, en de FHIR resource service in geen geval toegang te laten verlenen

op basis van dit access_token. Het id_token en de context parameters zijn wel van toepassing.

Validatie van tokens (HTI, access_token, id_token).

Applicaties in een koppeltaaldomein kennen elkaar niet, en zijn niet op de hoogte van de JWKS endpoints van elkaar in het domein. Dit is een bewuste keuze van Koppeltaal, het beheer van het domein wordt door domeinbeheer uitgevoerd, deze geeft aan wie er deelneemt en met welke beperkingen en voorwaarden. Het is dus domeinbeheer die de JWKS endpoints bijhoudt. De autorisatie service stelt een token introspection endpoint beschikbaar waar zowel het HTI token, als de access_token en id_token van de autorisatie service zelf gevalideerd kunnen worden, zie voor details TOP-KT-021 - Token Introspection. Het is dus als applicatie onmogelijk zelf een HTI token te valideren binnen het stelsel van Koppeltaal.

Beveiliging van het token introspection endpoint

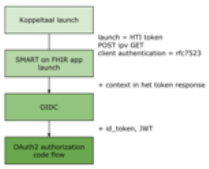
Het token introspection endpoint moet volgens de specificatie beveiligd worden om te beveiligen tegen *token scanning* aanvallen. In koppeltaal maken we de keuze de beveiliging te doen op basis van JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication (rfc7523) zoals in TOP-KT-021 - Token Introspection beschreven. Hiermee kiezen we voor een uniforme aanpak rond client authenticatie.

Toepassing, restricties en eisen

SMART on FHIR app launch framework toegelicht

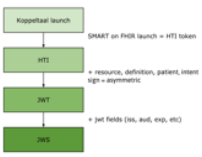
Hoewel SMART on FHIR app launch gedocumenteerd is, is het toch goed even iets toe te lichten over de standaard en over hoe deze wordt toegepast. Het diagram hieronder geeft aan hoe de verschillende standaarden gerelateerd zijn. De koppeltaal launch maakt gebruik van de SMART on FHIR app launch voor applicaties die persoons en/of medische gegevens ontsluiten. In deze launch is de launch parameter gevuld met een ondertekende HTI token. In de SMART on FHIR app launch documentatie wordt gebruik gemaakt van een GET request, koppeltaal doet de launch met een application/x-www-form-urlencoded encoded POST in een redirect form. Verder resulteert de app launch in een NOOP access_token (aangezien dit token in de context van één specifieke gebruiker

is). Toegang tot de FHIR resource service dient altijd op applicatie-niveau, middels rfc 7523, plaats te vinden. Hoe rfc 7523 wordt toegepast is beschreven in SMART on FHIR backend services. Op deze drie specifieke afspraken na, is koppeltaal een SMART on FHIR app launch. SMART on FHIR app launch, op zijn beurt is gebaseerd op Open ID Connect (OIDC), met als toevoeging dat naast het access_token en het id_token er in het token response ook contextgegevens meegegeven mogen worden. In koppeltaal is dat de inhoud van het HTI token. Deze kan wordt aangevraagd met de `launch` scope. OIDC is op zijn beurt een uitbreiding op de OAuth2 authorization code flow, met als toevoeging het id_token, een JWT token met daarin de gegevens van de gebruiker. De id_token kan door middel van de scope `openid` aangevraagd worden, net als de `fhirUser` checkt of de openid user ook echt in de FHIR resource service bestaat. Omdat in koppeltaal vooralsnog de gebruiker altijd onderdeel is van de launch, is de scope altijd `launch openid fhirUser`. Koppeltaal maakt geen gebruik van het refresh token bij de SMART on FHIR app launch.



Health Tool Interoperability (HTI)

Health Tool Interoperability (HTI) is een standaard geïnspireerd door, maar niet gebaseerd op, Learning Tool Interopability (LTI). De filosofie achter HTI is eenvoud en eenduidigheid. HTI voegt een aantal launch gerelateerde velden toe aan de reeds bestaande JWT standaard, die op zijn beurt aantal velden vastgesteld op de JSON web signature standard. Daarnaast beschrijft HTI hoe het token uit te wisselen. Koppeltaal maakt gebruik van HTI versie 2.0.



JSON Web Keys (JWKS)

JWKS is geen eigen standaard, maar komt voor in de OAuth 2.0 Authorization Server Metadata standaard als optionele URL. Het document bevat een JSON structuur met in de

“keys” waarde een lijst van JSON Web Keys (rfc7517). In koppeltaal maakt de autorisatie service gebruik van een JWKS URL om zijn keys bekend te maken. Voor de applicaties in het domein is dit optioneel. Dit wordt in detail besproken in het onderdeel TOP-KT-020 - Uitwisseling publieke sleutels.

Code challenge en verifier

Het SMART on FHIR App launch framework vereist dat er bij de authorisatiestap een `code_challenge` challenge en `code_challenge_method` wordt meegegeven, die bij het token request dient een `code_verifier` meegegeven te worden. De challenge bestaat uit een hash van de verifier die wordt samengesteld met de methode die in de `code_challenge_method` wordt aangegeven. Het SMART on FHIR App launch framework legt vast dat de waarde voor de `code_challenge_method` `"S256"` moet zijn, de "plain" waarde voor de methode is niet toegestaan.

De koppeltaal launch in stappen

Voordat de launch start gaan we er vanuit dat de volgende situatie van toepassing is:

- Alle betrokken applicaties zijn aangemeld in het domein en heeft zijn public key bekend gemaakt door een JWKS URL of door een directe registratie van de public key. De private key dient geheim gehouden te worden. Zie ook TOP-KT-008 - Beveiliging aspecten .
- In de FHIR resource service moet een Device zijn aangemaakt waarin de client_id overeenkomt met de logical identifier van het Device resource. Praktisch gezien is de Device resource dus altijd Device/<client_id>.
- De applicatie instantie moet een applicatierol zijn toegekend en heeft de benodigde rechten op de FHIR resource service.
- De module applicatie heeft één of meerdere redirect_uris geregistreerd bij het domein.
- De module applicatie heeft een FHIR ActivityDefinition gepubliceerd. De endpoint extension (ActivityDefinition.extension['endpoint']) in deze definition wijst naar het launch endpoint van de module.
- De gebruiker moet ingelogd zijn in de applicatie en een FHIR resource van deze gebruiker moet bestaan. Deze kan een Patient, Practitioner of RelatedPerson resource zijn..
- Bij een SMART on FHIR app launch scenario moet de business identifier van de resource van de gebruiker overeenkomen met de identiteit van de gebruiker bij de Identity Provider van het domein.
- Er is een FHIR Task object aangemaakt, en toegewezen aan de gebruiker. Wie het FHIR Task object aanmaakt staat niet vast, dit kan een andere applicatie in het domein zijn. De FHIR Task verwijst naar de ActivityDefinition door middel van het `Task.extension['instantiates']`
- De portal en module applicatie heeft toegang tot de FHIR resource service door middel van rfc 7523, beschreven in SMART on FHIR backend services.

De algemene launch flow

- De gebruiker selecteert een taak om uit te voeren.
- De portal applicatie genereert een HTI token op basis van het HTI protocol met de gegevens uit de taak en ondertekent deze met de bijbehorende private key. De velden worden als volgt gevuld:

    - `resource` : de Task reference
    - `definition` : de reference naar de ActivityDefinition van het veld `Task.extension[instantiates]`
    - `sub` : de referentie naar de lancerende gebruiker (Patient/Practitioner/RelatedPerson). Typisch is dit de huidig ingelogde gebruiker.
    - `patient` (optioneel): indien de sub niet de toegewezen gebruiker van de taak is, kan deze hier worden toegevoegd worden. Typisch wordt dit veld gevuld met de toegewezen gebruiker van de taak, als de huidige ingelogde gebruiker niet de toegewezen gebruiker van de taak is.
    - `intent` : intentie van de launch, dit veld is niet verplicht. Dit veld kan gebruikt worden om verschillende scenario's van gebruik te ondersteunen. Zie ook de Valueset Task Intent.
    - `iss` = De client_id van de portal applicatie
    - `aud` =De device reference van de applicatie (Device/123). Deze kan gevonden worden in het `ActivityDefinition.extension['resource-origin']`.
    - `idp_hint` = De unieke identifier waaronder de configuratie bekend is in de Koppeltaal Domein Configuratie. (Optioneel)
- De portal applicatie stuurt een redirect naar de launch URL van de ActivityDefintion door middel van een application/x-www-form-urlencoded encoded POST met als waarden het HTI token als `launch` parameter en de FHIR server URL als `iss` parameter. De parameters worden als application/x-www-form-urlencoded uitgewisseld. De launch URL staat in het ActivityDefintion.extension['endpoint.address'] veld.
- De module applicatie haalt het smart configuratie op bij de FHIR service (.well-known/smart-configuration). Dit statement bevat de volgende relevante endpoints:
    - jwks_uri
    - authorization_endpoint
    - token_endpoint
    - introspection_endpoint
- De module applicatie ontvangt de launch met de launch parameter en de iss. De iss moet de base URL van de FHIR resource service zijn.

De HTI flow voor applicaties die geen persoons en/of medische gegevens verwerken doorlopen de volgende vervolgstappen.

- De module applicatie stuurt de inhoud van de launch parameter, het HTI token, naar het introspection endpoint. De module maakt gebruik van JWT gebaseerde client credentials (rfc7523) om het endpoint te benaderen. Het token introspection endpoint is beschreven in TOP-KT-021 - Token Introspection.
- De authorization service valideert het token en, indien valide, stuurt de inhoud van het token in het response. De validatie betreft de issuer (`iss`), issued at (`iat`), expiry (`exp`), not before (`nbf`), JWT id (`jti`) en indien mogelijk de audience (`aud`) .
- De module applicatie kan op basis van de inhoud van het introspection antwoord de juiste resources bij de FHIR resource service ophalen. De applicatie moet zich authenticeren bij het token endpoint door middel van JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication (rfc7523), zoals beschreven in TOP-KT-005c - Applicatie toegang: SMART on FHIR backend service s.

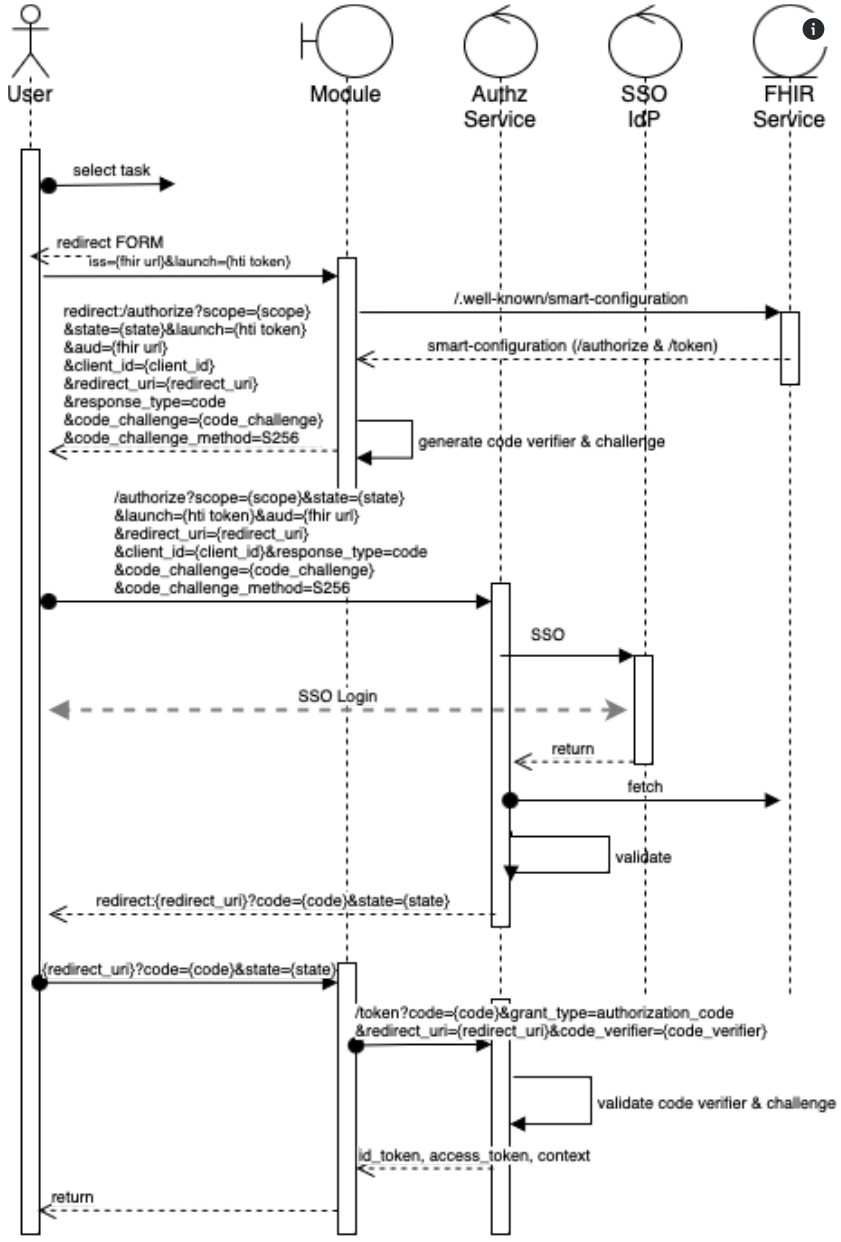
De SMART on FHIR app launch flow voor applicaties die persoons en/of medische gegevens verwerken doorlopen de volgende vervolgstappen.

- De module applicatie genereert een waarde voor de `code_verifier` aan maakt een `code_challenge` op basis van de `code_challenge_method` en de code_verifier zoals beschreven in rfc7636.
- De module applicatie redirect de gebruiker naar het authorize endpoint met de volgende verplichte parameters.
    - response_type, altijd “code”. Verplicht door SMART on FHIR app launch
    - client_id, de client id van de module applicatie. Verplicht door SMART on FHIR app launch

    - redirect_uri, de redirect locatie waar de browser na het autoriseren heen wordt gestuurd. Verplicht door SMART on FHIR app launch
    - launch: de inhoud van het launch parameter, het HTI token. Verplicht in koppeltaal.
    - scope: Verplicht door SMART on FHIR app launch, een van de vaste waarden:
        - launch openid fhirUser
    - state: een unieke identifier voor de correlatie en validatie van het autorisatieverzoek. Verplicht door SMART on FHIR app launch
    - aud: de FHIR resource service URL.
    - code_challenge: de code challenge uit de vorige stap.
    - code_challenge_method: de vaste waarde S256.
- De autorisatie service voert de validatie uit van client_id en redirect_uri met de gegevens zoals deze over de applicatie en module bekend zijn.
- De autorisatie service identificeert de gebruiker volgens de SSO implementatie van het domein.
- De autorisatie service valideert de inhoud van het HTI token met de identiteit van de gebruiker bij de vorige stap. Het HTI token heeft in het sub veld een referentie naar de gebruiker, dit kan een Patient, Practitioner of RelatedPerson zijn. In deze resource zijn één of meerdere identifiers verplicht. Iedere identifier heeft een waarde en een systeem. Afhankelijk van de inrichting van het domein, moet er een mapping bestaan tussen de identiteit die de gebruiker krijgt van de identity provider en één van de identifier velden van de FHIR resource.
- De autorisatie service redirect de gebruiker naar de module met de state en de code.
- De module applicatie ontvangt de code en de state, en valideert de state met de actieve sessie van de gebruiker.
- De module applicatie voert een authorization_code request uit naar het token endpoint van de authorization service. Dit endpoint vereist client authenticatie met rfc 7523, zoals bescheven in SMART on FHIR backend service. De module applicatie verzend in dit request als application/x-www-form-urlencoded encoded POST de volgende parameters:
    - `grant_type` : altijd ‘authorization_code’
    - `code` : de eerder ontvangen code.
    - `redirect_uri` : dezelfde waarde voor de redirect_uri als bij de autorisatie stap.
    - `code_verifier` : de waarde van de eerder gegenereerde code_verifier gerelateerd aan de code_challenge.
    - `client_assertion` : De client identificatie van de module applicatie door middel van rfc 7523
    - `client_assertion_type` : altijd de waarde `urn:ietf:params:oauth:client-assertion-type:jwt-bearer`
- De autorisatieservice valideert de ontvangen client credentials (rfc 7523) van de module en checkt de ontvangen code. Verder checkt hij of de client_id uit de iss (issuer) van de client credentials (rfc 7523) overeenkomen met de redirect_uri in het authorizatie request en de redirect_uri uit dit request.
- De autorisatieservice genereert een id_token op basis van de identiteit van de gebruiker.
- In het antwoord van de autorisatie service worden de standaard velden als volgt gevuld:
    - `id_token` : de JWT token met de claims over de gebruiker.
    - `access_token` : altijd de waarde 'NOOP'.
    - `scope` : de inhoud van de waarde van scope bij de authorize stap (vaste waarde '`launch openid fhirUser`').
    - `token_type` : altijd 'bearer'
    - `expires_in` : now() + 5min
- De SMART on FHIR app launch specificatie staat toe dat naast de standaard velden in de context van het token response ook extra velden meegegeven mogen worden. In Koppeltaal 2.0 moet de context gevuld worden met de volgende velden uit het HTI token (indien deze aanwezig zijn):
    - `resource` : taak referentie `Task/123`

- `definition` : `de reference naar de  ActivityDefinition van het veld Task.extension[instantiates]`
- `sub` : de referentie van de gebruiker die de launch uitvoert, dit kan de patient, behandelaar of derde zijn. `Patient/123`
- `patient` : optioneel de patient identifier, vooral wanneer de `sub` niet de patient zelf is. `Patient/321`
- `intent` : optioneel, om onderscheid te kunnen maken van het type launch.

- De module applicatie ontvangt het antwoord op het token request en kan de juiste resources bij de FHIR resource service ophalen.



Gebruikte standaarden

- HTI: https://github.com/GIDSOpenStandaarden/GIDS-HTI-Protocol/blob/master/HTI.md
- SMART on FHIR app launch (v2.1.0): https://hl7.org/fhir/smart-app-launch/STU2.1/app-launch.html
- JSON Web Key (rfc7517): https://www.rfc-editor.org/rfc/rfc7517
- OAuth 2.0 Token Introspection (rfc7662): https://www.rfc-editor.org/rfc/rfc7662
- JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants: https://www.rfc-editor.org/rfc/rfc7523.html
- OAuth 2.0 Authorization Server Metadata https://www.rfc-editor.org/rfc/rfc8414.html
- Proof Key for Code Exchange by OAuth Public Clients https://www.rfc-editor.org/rfc/rfc7636

De restricties en eisen

Deze restricties en eisen zijn restricties en eisen die additioneel gelden over de betrokken standaarden. Het zijn dus de eisen die gelden over hoe koppeltaal de standaarden toepast, aanpast of specifieker maakt.

Key disseminatie

- Een applicatie in het domein moet zijn public key(s) beschikbaar maken in het domein. Dit kan door
  - a) het configureren van de public key bij de domein registratie of
  - b) het registreren van een JWKS endpoint bij de domein registratie.
- In het JWKS endpoint wordt in een JSON bestand onder de “keys” een lijst van JSON Web Keys beschikbaar gesteld.

Applicatie registratie

- Bij de registratie van de applicatie bij domeinbeheer krijgt de applicatie een client_id toegewezen. Deze client_id wordt op verschillende plekken in de OAuth2 flows van SMART on FHIR app launch framework als SMART on FHIR backend services gebruikt.
- Applicaties die gebruik maken van de SMART on FHIR app launch framework moeten één of meerdere redirect_uris registreren bij domeinbeheer.

Publicatie van behandelingen

- De module applicatie moet een ActivityDefinition publiceren met een endpoint extension als het launch endpoint van de module.
- De gepubliceerde URL moet een https URL zijn.

FHIR resource server

- De SMART configuratie (zie ook TOP-KT-016 - SMART on FHIR Conformiteit) van de FHIR resource service moet de volgende URLs kenbaar maken:
-
  - token (token_endpoint)
  - JWKS (jwks_uri)
  - authorization (authorization_endpoint)
  - token introspection (introspection_endpoint)introspection_endpoint

SMART on FHIR app launch

- De launch maakt gebruik van een application/x-www-form-urlencoded encoded POST in plaats van een GET zoals beschreven in de SMART on FHIR app launch standaard.

- De launch parameter wordt gevuld met het HTI token en is verplicht, net als in SMART on FHIR app launch.
- De iss parameter is de base URL van de FHIR resource server.
- De URL van de launch is de URL van de module vastgelegd in de ActivityDefinition
- De scope parameter bij het authorize request is gevuld met vaste waarde '`launch openid fhirUser`'.
- Het token endpoint maakt gebruik van *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants* voor het authenticeren van het token request. De client_id zit hier in het iss veld van het JWT token.
- De autorisatiestap in de SMART on FHIR app launch moet gebruik maken van de SSO oplossing van het domein indien dat een vereiste is. Initieel wordt SAML en OIDC ondersteund.

HTI

- Koppeltaal maakt gebruik van versie 2.0 van HTI.
- Indien een applicatie gebruik maakt van het JWKS endpoint, MOET deze de kid header vullen in het HTI token.
- De issuer (iss) in het HTI token is de client_id van de portal applicatie.
- De audience (aud) van het token is de Device Id (Device/123) van de module applicatie.

Token introspection

- Het token introspectie endpoint maakt gebruik van *JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants (rfc7662)* om de aanvrager te autoriseren.

RelatedPerson

Wanneer voor de RelatedPerson (De Naaste van de Patient) een launch wordt uitgevoerd moet met een aantal specifieke restricties rekening worden gehouden die beschreven worden in TOP-KT-026 - Uitbreiding: Rol van de naaste | Toepassing en restricties

{}