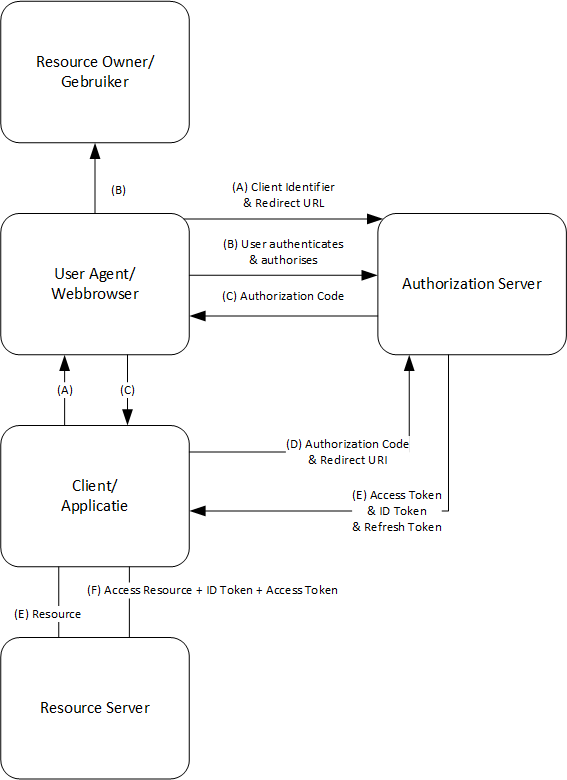
Inleiding

De tweede use case betreft, naast applicaties van het DSO, ook ingelogde eindgebruikers. Deze gebruikers kunnen ingelogd zijn met bijvoorbeeld DigiD, eHerkenning of een DSO inlogmiddel. Een voorbeeld van deze use case is een initiatiefnemer die ingelogd is in het loket, en een aanvraag in wil dienen via een gebruikerstoepassing. Het knooppunt moet de identiteit van de eindgebruiker doorgeven. In deze use-case werken meerdere componenten van het DSO samen en wordt gebruik gemaakt van de GDI componenten DigiD / EHerkenning. Deze use case is dus een uitbreiding/ specialisatie van de eerste usecase.

Beschrijving Use Case 2

De tweede use case (interne applicaties met ingelogde gebruikers) gaat over eindgebruikers die applicaties van het DSO gebruiken. Deze externe gebruikers zullen ingelogd zijn (of worden ingelogd) met eHerkenning, DigiD of een DSO ID. In elk geval zullen gebruikers identificeerbaar moeten zijn voor de resource server (de aanbieder van de API) voor bijvoorbeeld fijnmazige autorisatie. Dit is dus essentieel anders dan use case 1 waarbij alleen applicatie identiteiten een rol spelen. Tevens is de eigenaar van het token nu niet een specifieke applicatie, maar een eindgebruiker. Er zijn verschillende applicaties die de identiteit van de gebruiker nodig hebben voor hun services; zoals bijvoorbeeld het DMS. Omdat, anders dan bij use case 1, er nu 4 rollen een belang hebben (client of relying party, de authorization server of openID provider, resource server en in dit geval ook de eindgebruiker of resource owner met zijn of haar user-agent) en er naast autorisatie (OAuth) ook een belangrijke rol voor authenticatie van de eindgebruiker is weggelegd, is gekozen om OpenID Connect te gebruiken als uitbreiding op OAuth. OpenID Connect biedt ook de mogelijkheid om de identiteit van een gebruiker door (of terug) te geven in de vorm van een zogenaamd identity token. Dit token bevat informatie over de eindgebruiker in de vorm van een DSO identifier. In tegenstelling tot een bearer token, bevat een ID token dus wel informatie over de eindgebruiker. Deze informatie is echter zonder toegang tot de Identity Server niet terug te leiden tot persoonsgegevens. Om te kunnen valideren dat dit token door de juiste instantie is uitgegeven, is dit token is digitaal ondertekend met een JSON websignature [RFC7515]. Om deze signature te zetten is gebruik gemaakt van de sleutel (private key) van de identity server die de token heeft uitgereikt. Deze private key is een PKIO certificaat. Door de public key van deze sleutel te delen met de partijen die een Identity Token ontvangen kunnen die altijd valideren dat de token inderdaad van de Identity Server vandaan is gekomen. Tevens wordt dit token door de client, rechtstreeks opgehaald bij de Authorisation server. De inhoud van het token is een eenvoudige claim, de zogenaamde DSO ID. Dit DSO ID is een voor DSO unieke Identifier voor een eindgebruiker. Hierdoor kan wel herleid worden welke eindgebruiker het betreft, maar alleen door clients met toegang tot de persoonsgegevens service van de Indentity Server.

Het grant type dat gebruikt wordt voor de tweede gebruikersgroep is de authorization code flow. In dit grant type wordt gebruik gemaakt van redirects van de browser van de gebruiker om inloggen te ondersteunen. Daarnaast is er een backchannel tussen de Replying party en de OpenID Provider welke het token uitgeeft. Hierdoor worden het access token en het ID token niet over de browser verstuurd waardoor dit niet onderschept kan worden.



Figuur 1 OAuth Grant Type Authorization Code voor gebruikersgroep 2 – Interne applicaties met gebruikers. Voor deze gebruikersgroep wordt niet pure OAuth, maar openIDConnect gebruikt [OIDC Core]. De tokens komen na autorisatie via een backchannel bij de client terecht, om deze op te halen wordt de Authorization code gebruikt die terugkomt via de user-agent.

 Sequence Diagram Use Case 2

**Scenario's**

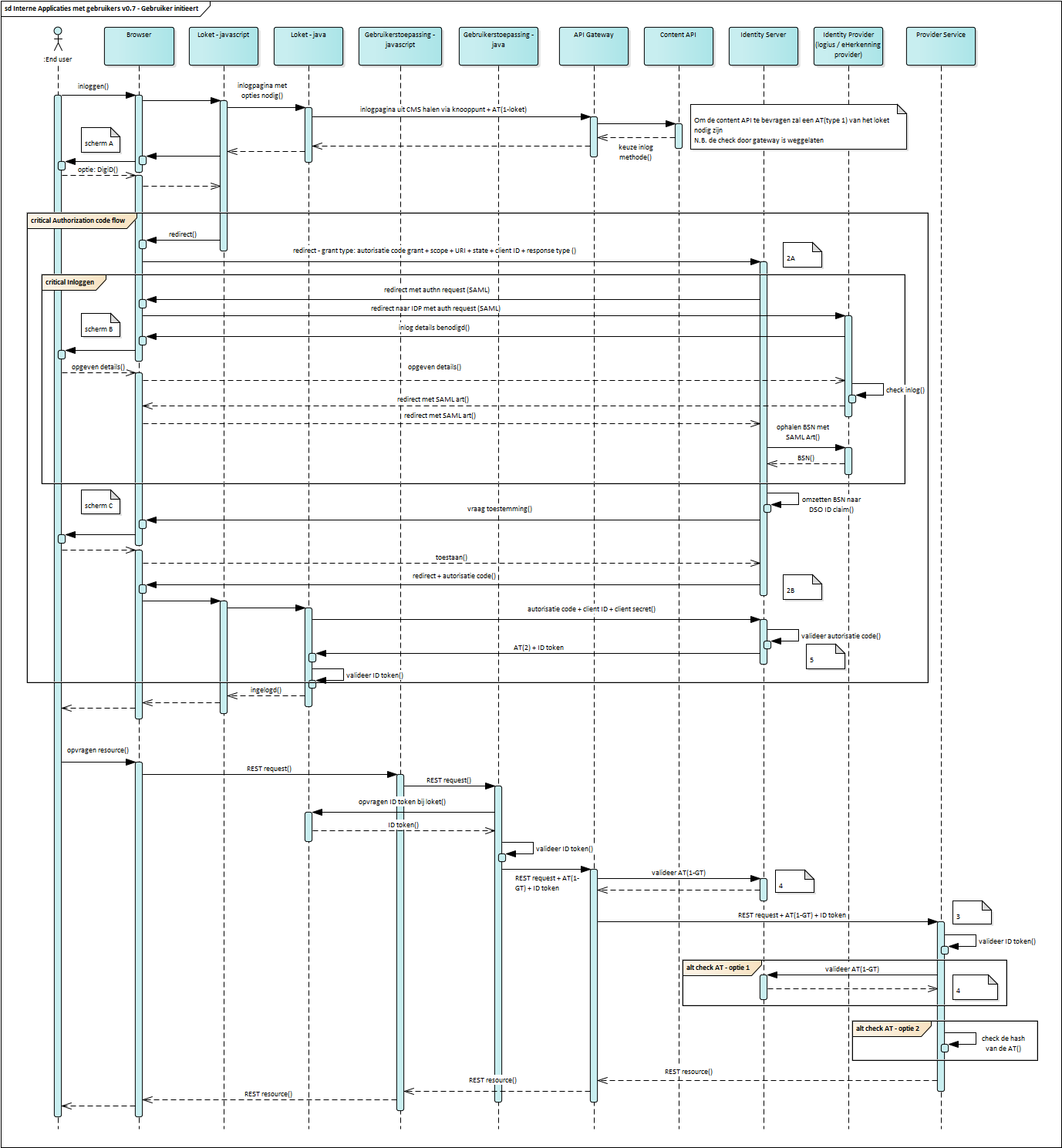
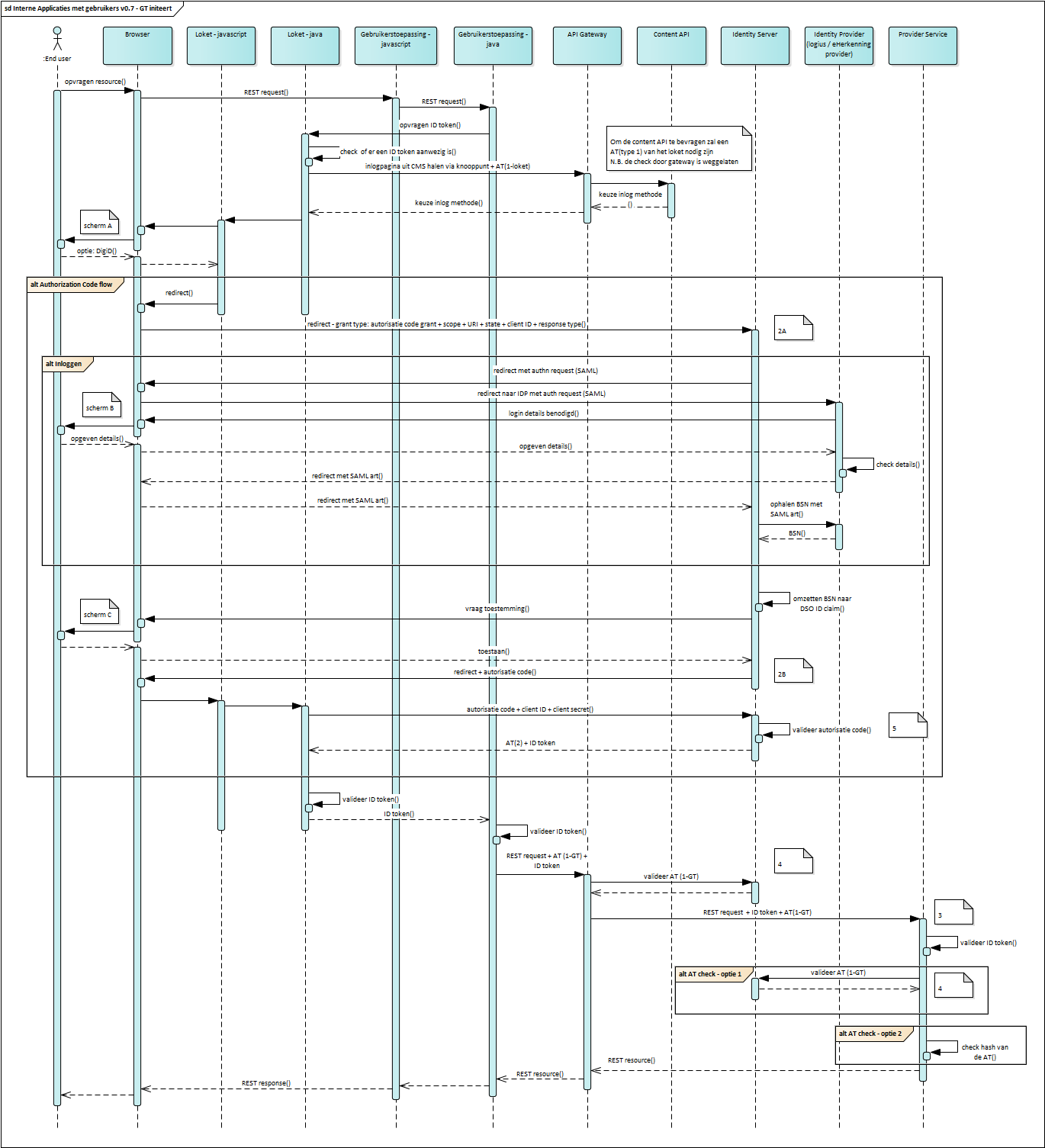
Het inloggen van een externe gebruiker (m.b.v. DigiD of Eherkenning of een DSO ID) kan op twee verschillende manieren geïnitieerd worden. De gebruiker kan zelf op een willekeurig moment besluiten om in te gaan loggen, door middels van de inloggen knop in het loket, of de gebruiker kan een onderdeel van een gebruikerstoepassing raadplegen waar een API achter schuil gaat waarvoor inloggen vereist is. Omdat dit twee wezenlijk andere processen zijn, worden deze in verschillende sequence diagrams uitgewerkt: Inloggen op interne applicaties – Gebruiker initieert en Inloggen op interne applicaties – Gebruikerstoepassing initieert.

Deze flows zijn anders omdat de trigger om in te loggen vanaf een andere kant komt. De Relying party (client) is wel dezelfde in beide gevallen: het loket vervult deze rol. Nadat het loket het inlogverzoek heeft gedaan richting de authorization server (de identity server) en de gerbuiker heeft ingelogd, bij een van de drie identity providers, zal een autorisatie code worden verkregen door het loket. De java component van het loket zal deze autorisatie code vervolgens inwisselen voor een nieuw access token (2) en een ID Token. Het loket deelt dit ID token met de server component van een gebruikerstoepassing die erom vraagt.

**Verschillende access tokens**

Om duidelijk te maken in het sequence diagram dat er verschillende access tokens zijn, wordt er verwezen naar AT (1); het token van de client credentials grant type en AT(2); wat verkegen is in combinatie met een ID Token en opgehaald is met de authorization code grant type. Deze beide tokens verschillen dus qua vorm en inhoud tevens is de eigenaar van beide tokens verschillend. De eigenaar van de eerste access token is de applicatie, terwijl de eigenaar van het tweede access token (en bijbehorend ID token) de eindgebruiker is. Wel is het zo dat beide tokens behoren tot 1 relying party/ client. Hierin is onderscheid gemaakt in de sequence diagrams door erbij te zetten wie de eigenaar is van het access token; AT(1-GT) is van een gebruikerstoepassing, en AT(1-loket) is van het loket.

De rollen in deze flow zijn alsvolgt verdeeld; de identity server is de OpenID provider (OIDP of de authorization server), het loket of de gebruikerstoepassing is de Relying party of de client, de eindgebruiker is degene die de user agent (in dit geval een browser) aanstuurt.

Figuur 2 Sequence diagram Applicaties koppelingen met ingelogde gebruikers use case 2. Links: het inloggen word gestart door een gebruiker. Rechts: Het inloggen wordt gestart door een Gebruikerstoepassing