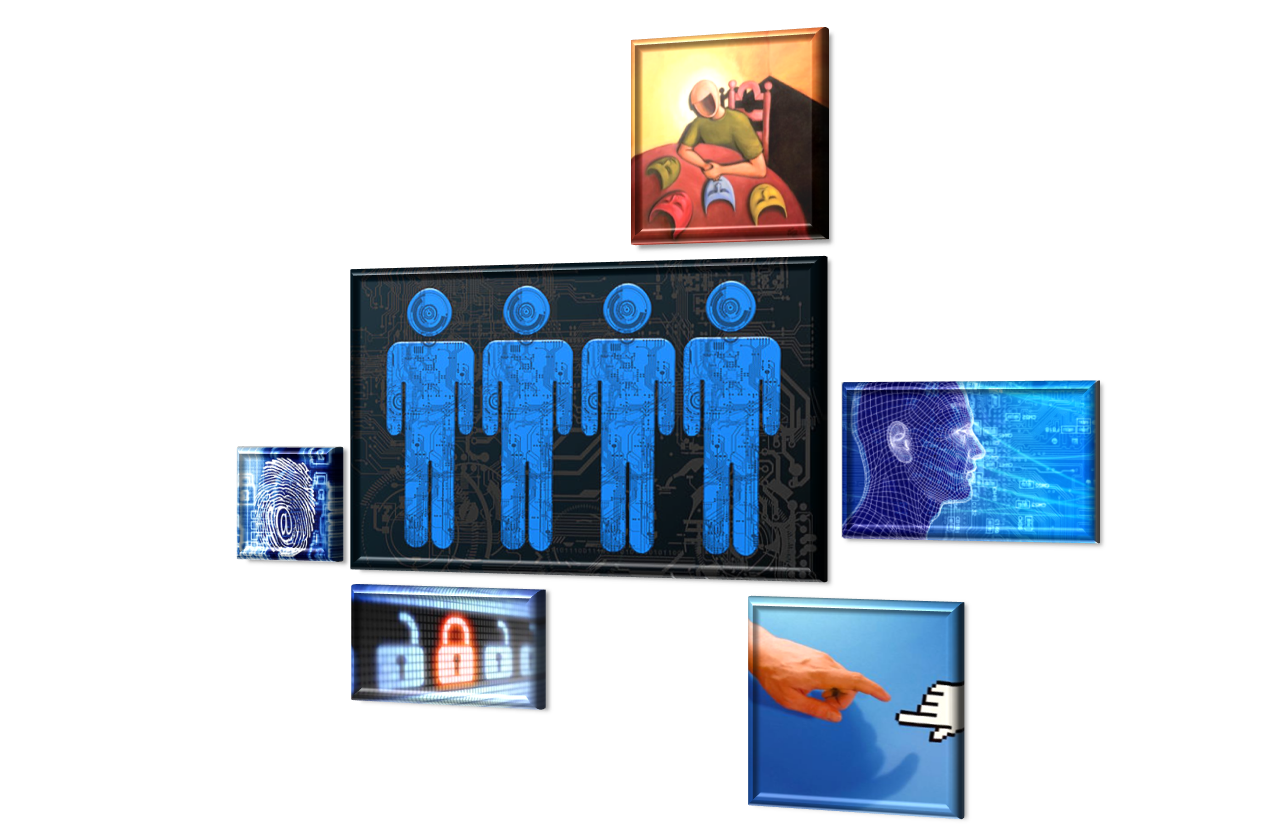
Eenmaal authentiseren, onbeperkt navigeren.

Maar eerst even de problemen oplossen die komen kijken bij het aan elkaar koppelen van digitale identiteiten en websites.



Auteur: Hugo van der Zandt  
Datum: 7-8-2012  
Locatie: Eindhoven

# *iStock_000011441899XSmall.jpg*Titelpagina

**Auteur:**

Naam: Hugo van der Zandt

Voorletters: H.V.N.

Studentnummer: 2130020

PCN: 208343

**Afstudeerbedrijf:**

Organisatie: E-ID

Adres: Lage Biezenweg 1

4131 LV Vianen

Telefoonnummer: 0347-344 345

Website: www.e-id.nl

**Mentoren**:

Stagedocent Fontys: Frank Haverkort

Bedrijfsmentor E-ID: Edvard Scheffers

# Voorwoord

Dit rapport is geschreven in het kader van mijn afstudeerstage voor de opleiding ICT aan Fontys Hogeschool ICT te Eindhoven.

De afstudeerstage is uitgevoerd bij E-ID internet strategies en heeft plaatsgevonden van maart tot en met juli 2012.

Bij deze wil ik graag een aantal mensen bedanken voor hun medewerking tijdens mijn afstudeerstage. Graag wil ik Patrick Schiks en Frouke Verburg bedanken voor hun medewerking bij het formuleren van de opdracht. Het onderwerp waar wij op uit zijn gekomen sprak mij erg aan doordat deze mij de gelegenheid gaf mijn expertise, in het domein van software security, verder te ontwikkelen.

Daarnaast wil ik mijn mentoren Edvard Sheffers en Frank Haverkort bedanken voor hun begeleiding tijdens het project. Zij stonden voor me klaar als dit nodig was.

Ook Erik Post en Wouter van Daalen wil ik bedanken voor hun betrokkenheid bij het project en mijn collega’s bij E-ID die van mijn stage periode een leuke leerzame periode hebben gemaakt.

07 augustus 2012

Hugo van der Zandt

# Samenvatting

Tegenwoordig zitten er steeds meer mensen op internet. Zij verwachten daarbij dat bepaalde handelingen automatisch gebeuren en ervaren het storend als dit niet het geval is. Authentiseren (inloggen) is een van deze handelingen. Organisaties willen de gebruikersvriendelijkheid vergroten door websites aan elkaar te koppelen; zodat er voor toegang tot alle websites van een organisatie, gebruikers maar één keer hoeven te authentiseren.

Om er voor te zorgen dat de websites goed blijven functioneren, is het belangrijk dat websites werken met de juiste digitale identiteit van een gebruiker. Dit kan mogelijk worden gemaakt door van een gebruiker, de persoonlijke gegevens van verschillende websites aan elkaar te koppelen. Zo wordt er effectief een Federated Identity gerealiseerd. Het voordeel van het gebruiken van een Federated Identity is dat een gebruiker altijd beschikt over de juiste digitale identiteit voor de website waar de gebruiker zich op dat moment bevindt. Zo kan worden bereikt dat een gebruiker zich maar één keer hoeft te authentiseren, om vervolgens gebruik te maken van alle websites die een organisatie aan elkaar heeft gekoppeld. Dit wordt Cross-Domain-SSO genoemd. Er is echter een probleem betreffende sessie management wat zich voordoet bij Cross-Domain-SSO, namelijk dat websites geen gebruik kunnen maken van eigen sessie cookies. Dit komt omdat websites geen toegang hebben tot de cookies van een andere website, waardoor het niet mogelijk is voor een website om te achterhalen of de gebruiker een geldige sessie heeft op een andere website.

Het eerder genoemde probleem kan worden opgelost met behulp van SAML (Security Assurtion Markup Language). SAML is een XML standaard welke is opgesteld door OASIS. De standaard definieert hoe securityinformatie op een veilige manier kan worden uitgewisseld tussen componenten. Dit maakt het mogelijk om websites op te splitsen in twee componenten namelijk een authentiserend component, Identity provider (IDP) genoemd, en een service aanbiedend component, service provider (SP) genoemd. Een website krijgt dan de rol van een SP en zal worden ontheven van de taak om gebruikers te authentiseren, deze taak neemt de IDP op zich. Het is mogelijk voor de IDP om zich op een ander domein te bevinden dan een website. Het is ook mogelijk dat meerdere SP’s vertrouwen op de authenticatie claims van dezelfde IDP. Zo wordt Cross-Domain-SSO mogelijk gemaakt.

Doordat SAML veilig en tegelijkertijd erg flexibel is, is het een populaire standaard waardoor er veel frameworks beschikbaar zijn die het ondersteunen. Voor deze opdracht is het Java Open Single Sign On (JOSSO) framework uitgekozen om te evalueren of het mogelijk is om een Federated Identity oplossing in de praktijk te realiseren. Dit is gedaan middels het realiseren van een Proof Of Concept (POC). Hieruit kwam naar voren dat de Model Driven Development (MDD) aanpak van het JOSSO framework ingezet kan worden om, op een veilige manier en binnen zeer korte tijd, Federated Identity oplossingen te realiseren. Doordat het framework gebaseerd is op SAML kunnen ook IDP’s en SP’s van andere leveranciers gebruikt worden in een oplossing, mits deze ook SAML compliant zijn. In een poging om specifieke eisen te vervullen, bleek dat het framework nog verschillende bugs heeft. Mocht er voor gekozen worden het framework in de praktijk in gebruik te gaan nemen, zal er contact opgenomen moeten worden met de organisatie die JOSSO heeft ontwikkeld om tot een oplossing te komen.

# Summary

An increasing amount of people use the internet on a daily basis. These people have grown accustomed to being relived from certain actions. Authenticating (logging in) is one of the actions. To increase usability, organisations want to connect their websites to one another; in such a way that users can get access to all the websites an organisation owns while having to authenticate only once.

In order for the websites to keep functioning accordingly, it is important that the websites work with the correct digital identity of a user. This can be facilitated through connecting the personal information of a user which he or she has over multiple websites. By doing this a Federated Identity is born into existence. The advantage of a Federated Identity is that a user is always in possession of the correct digital identity when he is using a particular website. This makes it possible for a user to authenticate on one website and navigate to all other websites an organisation connected to it. This is called Cross-domain-SSO. However, there is one problem regarding the realisation of Cross-Domain-SSO. Cookies that are used to maintain user sessions can’t be accessed by other websites. This means that it’s not possible for any website to check if a user has a valid session on another website. This is a problem that has to be overcome before Cross-Domain-SSO can be realised.

The previously mentioned problem can be tackled through use of the SAML. SAML is a XML standard which is ratified by OASIS. The standard defines a way security information can be exchanged between components. This makes it possible to split websites into two components. One component that offers a service, called SP and one component that will handle authorisation, called IDP. Websites will assume the role of SP and will be relieved from the task of authenticating users. An IDP will for fill this task and communicate assertions regarding a principle to a SP. It is possible for an IDP to be located on another domain and it is also possible for multiple SP’s to connect to one IDP. Because of this, it is possible to realise a Cross-Domain-SSO solution by use of the SAML standard.

A lot of SAML based frameworks exist because SAML is secure and flexible at the same time. For this graduation-assignment the JOSSO framework has been selected to demonstrate if it is possible to realise a Federated Identity in practice. This is done through realising a POC. During the realisation processes, it appeared that a secure Federated Identity solution could be rapidly realised through use of the MDD support of the framework. Because the framework is also SAML based it allows for use of IDP and SP components from different suppliers (as long as these are also SAML compliant). However, in an attempt to realise specific requirements it appeared that the framework still has some bugs. If the framework is going to be used in future projects, the organisation that build the framework should be contacted to find a solution for the bug related issues.

Inhoudsopgave

[1 Onderzoeksvoorstel 8](#_Toc331148395)

[1.1 De aanleiding 8](#_Toc331148396)

[1.2 Doelgroep 8](#_Toc331148397)

[1.3 De doelstelling van het onderzoek 8](#_Toc331148398)

[1.4 Methodologie 9](#_Toc331148399)

[1.5 de onderzoeksvraag 9](#_Toc331148400)

[1.5.1 De deelvragen 9](#_Toc331148401)

[1.6 Opbouw van het rapport 9](#_Toc331148402)

[2 Het bedrijf 10](#_Toc331148403)

[3 De opdracht 11](#_Toc331148404)

[3.1 Aanpak en planning 11](#_Toc331148405)

[3.2 Technische uitdaging/probleem 11](#_Toc331148406)

[3.3 Het onderzoek 12](#_Toc331148407)

[3.3.1 De requirements 12](#_Toc331148408)

[3.3.2 Technologieën 13](#_Toc331148409)

[3.3.3 Frameworks 18](#_Toc331148410)

[3.3.4 Framework keuzen 19](#_Toc331148411)

[3.3.5 Uitvoering POC 21](#_Toc331148412)

[4 Het JOSSO framework 22](#_Toc331148413)

[4.1 De Gateway (Identity provider) 22](#_Toc331148414)

[4.2 De Agent (Service provider) 23](#_Toc331148415)

[4.3 Beveiliging 24](#_Toc331148416)

[4.3.1 Broken authentication and session management 24](#_Toc331148417)

[4.3.2 CSRF 24](#_Toc331148418)

[4.4 Framework versies 25](#_Toc331148419)

[4.4.1 JOSSO1 25](#_Toc331148420)

[4.4.2 JOSSO2 25](#_Toc331148421)

[5 Proof of concept 28](#_Toc331148422)

[5.1 Onderzoeksaanpak vaststellen 28](#_Toc331148423)

[5.1.1 Resultaat 29](#_Toc331148424)

[5.2 De basis functionaliteit testen 29](#_Toc331148425)

[5.2.1 Resultaat 30](#_Toc331148426)

[5.3 Custom application 30](#_Toc331148427)

[5.3.1 Resultaat 30](#_Toc331148428)

[5.4 Behoud van de login pagina 31](#_Toc331148429)

[5.4.1 Resultaat 31](#_Toc331148430)

[5.4.2 Branding van de login pagina als alternatief. 32](#_Toc331148431)

[5.5 Werken met een Federated Identity 32](#_Toc331148432)

[5.5.1 Problemen 32](#_Toc331148433)

[5.5.2 Resultaat 33](#_Toc331148434)

[5.6 Behoud van oude account 34](#_Toc331148435)

[5.6.1 Problemen 34](#_Toc331148436)

[5.6.2 Resultaat 34](#_Toc331148437)

[5.6.3 Een alternatieve oplossing 35](#_Toc331148438)

[5.7 Ondersteuning voor spring security 36](#_Toc331148439)

[5.7.1 Aanpak 36](#_Toc331148440)

[5.7.2 Resultaat 36](#_Toc331148441)

[6 Conclusie en aanbevelingen 37](#_Toc331148442)

[7 Evaluatie 38](#_Toc331148443)

[8 Literatuurlijst 39](#_Toc331148444)

[A. Verklarende woordenlijst 1](#_Toc331148445)

[B. Project Initiation Document 7](#_Toc331148446)

[C. Onderzoeksdocument 28](#_Toc331148498)

[D. Requirements 49](#_Toc331148532)

[E. Protocollen 55](#_Toc331148542)

[F. Frameworks 57](#_Toc331148543)

[G. Loginpagina code 67](#_Toc331148546)

[H. Scenario 68](#_Toc331148547)

# Onderzoeksvoorstel

Voor u ligt de scriptie betreffende de afstudeerstage uitgevoerd bij E-ID te Vianen. E-ID heeft deze stage mogelijk gemaakt door middel van het geven van een opdracht. Het onderwerp van de afstudeerstage is als volgt: het vinden van een manier om een SSO verbinding tussen websites te realiseren, waarbij gebruikersaccounts van de verschillende websites aan elkaar worden gekoppeld. In dit hoofdstuk worden de aanleiding, de doelstelling, de onderzoeksvraag, de deelvragen en de randvoorwaarden beschreven. Als laatste volgt de opbouw van het rapport.

## De aanleiding

Er zijn veel sites die eisen dat gebruikers zich authentiseren voordat ze toegang krijgen tot bepaalde functionaliteiten of content. Dit hoeft op de meeste sites maar een keer te gebeuren omdat er tegenwoordig bijna altijd gebruik wordt gemaakt van SSO. Dit wordt gedaan om aan de verwachtingen van de gebruikers tegemoet te komen. Gebruikers verwachten namelijk als ze een keer een website hebben verteld wie ze zijn, dat de website dit onthoudt.

Tegenwoordig komt het veel voor dat een grote organisatie in het bezit is van meerdere websites. Als gebruikers in het bezit zijn van een account op deze websites hebben zij de volgende verwachting: “*Ik ben bekend bij website A en website B. Website A en B zijn van dezelfde organisatie, dan zou ik niet opnieuw moeten inloggen als ik van site A naar B ga*”. Om aan deze verwachting van gebruikers tegemoet te komen, dient er een oplossing voor deze organisaties te worden bedacht die het mogelijk maakt user-accounts van verschillende identiteitsmanagement-systemen aan elkaar te koppelen. De organisatie wil aan haar gebruikers de mogelijkheid geven om van een naadloze SSO ervaring te genieten als ze navigeren tussen de domeinen die de organisatie bezit.

## Doelgroep

Dit document presenteert de resultaten van de afstudeeropdracht van de auteur. De opdracht is uitgevoerd in opdracht van E-ID internet strategies en is bedoeld als vooronderzoek voor projecten die wellicht in de toekomst uitgevoerd gaan worden. Dit document is bedoeld voor:

* mensen die interesse hebben in technologieën die nauw in verband staan met authenticatie, SSO en Federated Identity.
* mensen/medewerkers die SSO voor bedrijven gaan realiseren in toekomstige projecten.

## De doelstelling van het onderzoek

Het doel van het onderzoek is het vinden van een manier om SSO tussen domeinen (Cross-domain-SSO) mogelijk te maken voor gebruikers. Dit moet worden ondersteund door het koppelen van de achterliggende identiteitsmanagement-systemen van de domeinen. Door het koppelen van de identiteitsmanagement-systemen worden de digitale-identiteiten van de gebruiker als het ware samengesmolten tot één digitale-identiteit. Een dergelijke digitale-identiteit wordt een Federated Identity genoemd.

## Methodologie

De volgende methodieken worden toegepast tijdens de uitvoering van de afstudeeropdracht:

* Er wordt gebruik gemaakt van de fasering methode van Roel Grit [1]. In de planning wordt ook aangegeven in welke Grit fase milestones gehaald moeten worden. De fases zijn: de definitiefase, ontwerpfase en realisatiefase. De complete fasering is te vinden in het project initietion document (PID) en deze is te vinden in bijlage B.
* Het onderzoek is gestructureerd volgens het tienstappenplan (TSP) van Kempen en Keizer [2].
* In de ontwikkelfase van de opdracht is er een agile ontwikkelstrategie gebruikt.
* Er is gebruik gemaakt van deskresearch en fieldresearch.

## De onderzoeksvraag

De onderzoeksvraag luidt als volgt: “*Hoe kunnen identiteitsmanagement systemen van verschillende domeinen op een veilige manier worden samengebracht tot een SSO systeem voor een gebruiker?*”

Om tot een antwoord op de onderzoeksvraag te komen zijn er een aantal deelvragen geformuleerd.

### De deelvragen

De deelvragen die van belang zijn voor het beantwoorden van de onderzoeksvraag zijn in onderstaande opsomming weergegeven met tussen accolades de grit fase die wordt doorlopen tijdens het beantwoorden van de vraag.

1. Aan welke functionaliteitseisen en veiligheidseisen moet een POC voldoen?(definitiefase)
2. Welke technieken en protocollen zijn er op het gebied van SSO? (definitiefase)
3. Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden? (ontwerpfase)
4. Welk framework is het meest geschikt als basis voor het realiseren van een POC? (ontwerpfase)
5. Hoe moet het POC uiteindelijk worden gerealiseerd? (realisatiefase)

## Opbouw van het rapport

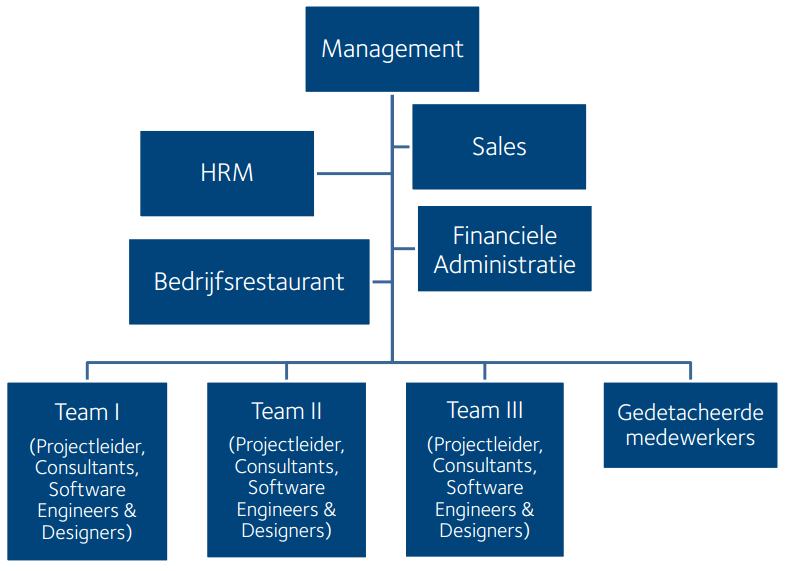
Dit rapport is als volgt opgebouwd: Hoofdstuk 2 vertelt over het bedrijf waarvoor de afstudeeropdracht is uitgevoerd. Hoofdstuk 3 gaat over het onderzoek dat is uitgevoerd in het kader van de afstudeeropdracht en zal in het bijzonder het SAML protocol toelichten. Hiervoor is gekozen omdat het SAML protocol centraal staat in het JOSSO framework, waarvan de karakteristieken en functionaliteiten zullen worden toegelicht in hoofdstuk 4. In hoofdstuk 5 zal het POC worden gepresenteerd waarin de eerder beschreven functionaliteiten van het JOSSO framework worden benut. Hoofdstuk 6 presenteert de conclusies die uit de realisatiefase kunnen worden getrokken. Hoofdstuk 7 is een reflectie op het verloop van de afstudeeropdracht. In hoofdstuk 8 is de literatuurlijst opgenomen gevolgd door de bijlagen. De *verklarende woordenlijst* is opgenomen in bijlage A.

# Het bedrijf

De afstudeeropdracht is beschikbaar gesteld door het bedrijf E-ID internet strategies (E-ID). E-ID is in 2000 opgericht en is van start gegaan met twee FTE. Is 2007 waren dit al 25 FTE geworden en nu, in 2012, telt E-ID 55 medewerkers. E-ID ambieert door te groeien naar 150 medewerkers zodat het de groei in omzet kan aanhouden. Deze omzet was in 2011 € 5.6 miljoen en in 2012 wordt een stijging van minimaal 5% verwacht ten opzichte van 2011.

De missie van E-ID is, om met een korte time-to-market, oplossingen (zoals: webapplicaties, mobile apps en frameworks) te realiseren op het snijvlak van business en IT die slim en innovatief zijn en die de interactie tussen bedrijven en hun klanten verbetert. E-ID richt zicht voornamelijk op bedrijven uit de top500 van Nederland. Het streven is dat alle bedrijven die hun klant interactie willen verbeteren, aankloppen bij E-ID, vanwege de hoge kwaliteit die E-ID levert.

E-ID heeft momenteel één vestiging in Vianen waar de 55 medewerkers werkzaam zijn (als ze niet zijn gedetacheerd). De verhouding van het niveau van de programmeurs is 60% senior-, 20% medior- en 20% junior- niveau[3]. In een informele werksfeer wordt er door deze programmeurs gewerkt aan projecten voor klanten of het leveren van support en advies. De projecten worden uitgevoerd in teams. Voor nieuwe projecten wordt er een nieuw team opgesteld met een nieuwe samenstelling van medewerkers. De teams die in het onderstaande organigram te zien zijn, dienen ter illustratie. Er is dus geen sprake van een afdeling “team1, 2 en 3” binnen E-ID.



Figuur 1 Organigram E-ID

# De opdracht

Het doel van de opdracht is om aan de hand van een onderzoek een duidelijk advies te kunnen geven over hoe Federated Identity oplossingen het beste gerealiseerd kunnen worden.

De opdracht is uitgevoerd in drie fases van de Grit methodiek zoals eerder besproken in hoofdstuk 1. De eerste fase die is doorlopen is de definitiefase, waarin invulling is gegeven aan de andere fases. De ontwerpfase is gebruikt om een duidelijk beeld te krijgen van de werkzaamheden die uitgevoerd gaan worden in de realisatiefase. In de realisatiefase is er aan een POC gewerkt.

Gedurende alle fases van de opdracht is er onderzoek uitgevoerd volgens het TSP. Alles wat is onderzocht tijdens de opdracht is gedocumenteerd in het onderzoeksverslag dat te vinden is in bijlage C.

## Aanpak en planning

In het PID zijn milestones gedefinieerd met een opleverdatum om het project structuur te geven. De aanpak en planning van deze milestones staan in ook detail omschreven in het PID. Globaal is de planning als volgt: eerst de onderzoeksvragen beantwoorden. Hiervoor zal gebruik worden gemaakt van desk- en fieldresearch. Met de kennis die wordt opgedaan door het beantwoorden van de vragen zal een POC worden uitgevoerd. Dit POC bestaan uit gestructureerde kleinere projecten waarbij functionaliteiten worden gedemonstreerd. Tijdens deze projecten zal er ook gebruik worden gemaakt van trail and error, om eventueel snel extra functionaliteiten aan te tonen als zich daar de gelegenheid voor doet; Denk hierbij aan het aanzetten van bepaalde opties om te testen of deze opties ook werken onder de huidige omstandigheden.

## Technische uitdaging/probleem

Een Federated Identity ontstaat door het koppelen van meerdere digitale identiteiten van één persoon. Werken met een Federated Identity heeft weinig tot geen nut als het niet gecombineerd wordt met Cross-Domain-SSO (ook wel “Multi-Domain-SSO” en “internet SSO” genoemd). De Federated Identity moet er voor zorgen dat de gebruiker tussen domeinen kan navigeren, zonder opnieuw in te loggen, terwijl de digitale identiteit die de gebruiker heeft altijd past bij het domein waar de gebruiker zich op bevindt. Er is geen sprake van meerwaarde als een Federated Identity niet wordt gebruikt voor het realiseren van extra functionaliteit. Indien er sprake is van een Federated Identity, dan wordt er automatisch van uit gegaan dat er ook sprake is van Cross-Domain-SSO.

Het probleem met Cross-Domain-SSO is dat standaard SSO oplossingen niet gebruikt kunnen worden. Deze maken namelijk allemaal gebruik van cookies. Het probleem met cookies is dat commerciële browsers afdwingen dat cookies domein specifiek zijn omwille veiligheidsredenen, gebaseerd op RFC 2901 en 2965 [4]. Hierdoor is het niet mogelijk voor domeinen om cookies te delen. De sessie cookie van een domein kan dus niet door een ander domein gebruikt worden.

Federated Identity oplossingen handmatig ontwikkelen kost veel tijd omdat:

* Veel Use-Cases moeten worden opgesteld
* Code architectuur moet worden ontwerpen
* Standaard Use-Cases, zoals wachtwoord wijzigen, zullen ook geïmplementeerd moeten worden
* Veel overhead door besprekingen en administratieve werkzaamheden

Daarbij bestaat ook de kans dat het project niet succesvol zal worden afgerond. Door deze omstandigheden doet zich de gelegenheid voor om:

* Te onderzoeken of bestaande technieken en standaarden een uitkomst kunnen bieden bij het realiseren van een Federated Identity oplossing.
* Te experimenteren met frameworks, die welgekozen technieken en standaarden benutten en gebruiken, die in de toekomst ingezet kunnen worden voor het realiseren van Federated Identity oplossingen.

## Het onderzoek

Tijdens de afstudeeropdracht is een onderzoek uitgevoerd om een goed beeld te krijgen van het probleem en wat de mogelijke oplossingen zijn. De eerder genoemde onderzoeksvraag die centraal staat in dit onderzoek is:

*Hoe kunnen identiteitsmanagementsystemen van verschillende domeinen op een veilige manier worden samengebracht tot een SSO systeem voor één gebruiker?*

Met behulp van het onderzoek zijn er stapsgewijs keuzes gemaakt die de realisatiefase vorm hebben gegeven. Het onderzoek is gestructureerd volgens het TSP. De bevindingen die zijn gedaan door middel van het uitvoegen van het diepteonderzoek, zullen in dit hoofdstuk worden gepresenteerd. Iedere paragraaf van dit hoofdstuk zal een deelvraag centraal stellen en het antwoord erop onderbouwen.

### De requirements

In deze paragraaf wordt de volgende deelvraag beantwoord:

*Aan welke functionaliteitseisen en veiligheidseisen moet een proof of concept (POC) voldoen?*

Deze vraag is beantwoord door gebruik te maken van fieldresearch. Met als resultaat dat er een *SMART* lijst is opgesteld met mogelijke eisen en wensen. Deze lijst is vervolgens besproken en door middel van deze besprekingen was het mogelijk om aan iedere eis een MoSCoW beoordeling gegeven. De MoSCoW-methode is een wijze van prioriteiten stellen waarbij iedere eis een letterwaarde krijgt die de prioriteit van de eis aangeeft [5].

* De waarde M staat voor: **M**UST have.
* De waarde S staat voor: **S**HOULD have.
* De waarde C staat voor: **C**OULD have.
* De waarde W staat voor: **W**ONT have.

Het requirements document bevat een lijst van 42 eisen die zijn opgedeeld in in *functional-* en *non-functional requirements* (Voor de lijst met requirements zie bijlage D). Er zijn in totaal 20 requirements met een M waarde, in deze scriptie wordt aangegeven wanneer aan een dergelijke requirement wordt voldaan. Als in een paragraaf bepaalde M requirements worden aangetoond dan zullen deze aan het einde van de paragraaf worden weergegeven in blauwe blokken. In de praktijk kan het voorkomen dat een requirement met een M beoordeling wordt geschrapt. Dit zou kunnen voorkomen als het implementeren van een requirement veel kosten met zich mee brengt. In de opdracht is wel aangetoond dat in theorie een requirement kan worden ondersteund.

### Technologieën

In deze paragraaf wordt de volgende deelvraag beantwoord:

*Welke technieken en protocollen zijn er op het gebied van SSO?*

Om deze vraag te kunnen beantwoorden is er onderzoek gedaan naar populaire technieken en protocollen die een rol kunnen spelen bij het faciliteren van Cross-Domain-SSO. Voor dit onderzoek is voornamelijk gebruik gemaakt van deskresearch. Met als resultaat dat 5 technieken en protocollen zijn bestempeld als relevant. Er is uitgezocht hoe bruikbaar ze daadwerkelijk zijn en of ze geschikt zijn voor de opdracht. Hieronder een kort overzicht van de onderzochte protocollen en standaarden met een korte samenvatting van de essenties (voor een uitgebreidere samenvatting zie bijlage E):

* **SAML** - een XML standaard die beschrijft hoe tokens, requests en responses er uit moeten zien en hoe bepaalde entiteiten er mee kunnen communiceren [6].
* **WS-Federation** - Een flexibele federated identity architectuur, die token onafhankelijk is maar wel en manier definieert voor het verkrijgen van een token [7].
* **OpenID** - Een techniek waarmee een gebruiker zich door een andere partij laat identificeren (bijvoorbeeld google), de andere partij (google) geeft daarbij de gegevens van de gebruiker aan een andere organisatie [8].
* **OAuth** - een techniek waarmee een gebruiker een partij toegang kan verlenen tot zijn gegevens die zich ergens anders bevinden (bijv. bij google) [9].
* **XRI** - een uitbreiding op de URI syntax, waarmee naar documenten (XRDS) verwezen wordt, die extra informatie bevatten over bijvoorbeeld een gebruiker [10].

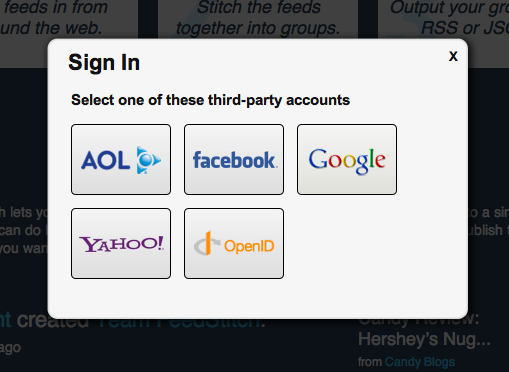
In de rest van dit hoofdstuk zal een 3-tal technieken verder toelichten. De technieken OpendID en OAuth worden toegelicht, omdat deze technieken erg populair zijn en vaak worden genoemd als er gesproken wordt over SSO op het internet. De SAML standaard wordt toegelicht, omdat deze geschikt is voor het realiseren van Federated Identity oplossingen.

#### OpenID en OAuth

OpenID en OAuth zijn technieken die vaak worden genoemd als er gesproken wordt over SSO. Deze technieken zijn echter niet geschikt voor Cross-Domain-SSO. Als er sprake is van Cross-Domain-SSO zal een gebruiker na authenticatie een sessie krijgen. Zo lang als deze sessie geldig is zal de gebruiker autorisatie hebben tot beveiligde content en functionaliteit op verschillende domeinen.

##### OpenID

OpenID is een gedecentraliseerd authenticatiemechanisme dat het mogelijk maakt om met één account op meerdere websites in te loggen [9]. Het werkt als volgt, een gebruiker maakt een OpenID-account aan bij een OpenID provider (bijv. Gmail of Yahoo). Andere websites (bijv. Blog sites) accepteren deze OpenID-account.



Als de gebruiker gebruik wil maken van zijn OpenID account moet hij selecteren wie zijn OpenID provider is en vervolgens moet hij inloggen. Als dat succesvol verloopt dan stuurt de OpenID provider gegevens van de gebruiker door naar website waarvan de gebruiker gebruik wil maken.

Het voordeel van OpenID is dat gebruikers met één account op veel domeinen van het internet welkom zijn, maar het is geen Cross-Domain-SSO oplossing. Een gebruiker moet nog steeds op iedere site zijn OpenID provider selecteren en inloggen, er is dus geen sprake van Cross-Domain-SSO.

##### OAuth

OAuthis een protocol waarmee gebruikers toegang tot hun gegevens kunnen verlenen aan een andere partij [9].

Als een gebruiker een account heeft bij een OAuth ondersteunde site, kan de gebruiker de gegevens die hij beheert op die site beschikbaar stellen aan een andere “consumerende” site. Zodra de gebruiker zijn gegevens beschikbaar wil stellen gaat het volgende proces in werking:

Het voordeel van OAuth is dus dat gebruikers toegang tot gegevens kunnen verlenen die op een ander domein worden beheerd. OAuth is echter geen SSO oplossing; want een gebruiker moet op iedere site zijn OAuth provider selecteren en inloggen, er is dus geen sprake van Cross-Domain-SSO.

#### SAML

**SAML** is een acroniem voor Security Assertion Markup Language, het is een op XML gebaseerde standaard voor het communiceren van identiteitsinformatie tussen organisaties. De standaard is gedefinieerd door OASIS, een organisatie die zich richt op het tot stand brengen van technische standaarden.

 C:\Users\blur5\Dropbox\private\werk documenten\bronnen\logo's\samlXMLorg-logo.png

Het primaire doel waarvoor SAML is ontworpen, is een oplossing te bieden voor SSO op het internet. De doelgroep van SAML zijn organisaties die veilig hun internet applicaties, binnen en buiten de organisatie, met elkaar willen verbinden. De standaard specificeert een manier voor het overdragen van authenticatie, rechten en eigenschappen van een principaal (bijvoorbeeld een gebruiker of service).

Dit hoofdstuk zal dieper ingaan op de technische werking van SAML. Er zal worden toegelicht uit wat voor componenten een SAML oplossing bestaat, hoe deze componenten met elkaar communiceren en hoe met SAML het cookie probleem van Cross-Domain-SSO wordt opgelost.

SAML componenten

SAML kan op vele manieren worden ingezet doordat het platform onafhankelijk is en omdat de standaard ook kan worden aangepast, zodat deze gecombineerd kan worden met andere standaarden. SAML oplossingen hebben vaak wel dezelfde structuur doordat er gebruik wordt gemaakt van entiteiten die een vaste rol hebben:

* Een **Identity Provider** (IDP), bewaart en levert Identiteitsprofielen op en regelt authenticatie.
* Een **Service Provider** (SP), biedt diensten en informatie aan en regelt autorisatie.

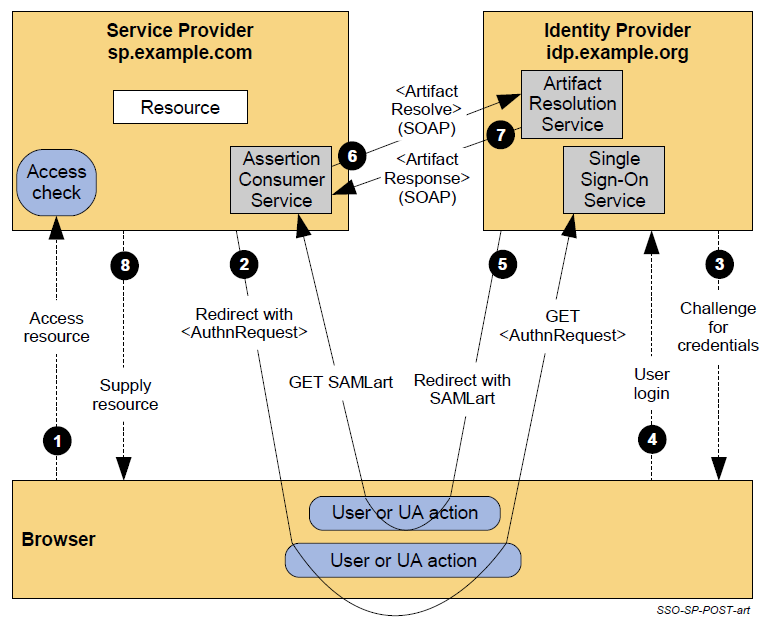


Als minstens één IDP en één SP een overeenkomst hebben om informatie uit te wisselen, is er sprake van een **Circle of Trust** (COT).

Een COT komt tot stand doordat providers *metadata* met elkaar uitwisselen. *Metadata* bevat configuratie informatie die het mogelijk maakt voor providers om met elkaar te communiceren [11]. Met behulp van SAML kunnen ook attributen van gebruikersprofielen van de IDP gekoppeld worden aan attributen van gebruikersprofielen van de SP binnen een COT. Als de IDP in een dergelijke situatie een bewering stuurt naar de SP, is de SP in staat om lokale gegevens van de gebruiker op te halen en deze te benutten. Zo kan er dus een Federated Identity worden gerealiseerd.

##### Message flows

##### Met SAML zijn er een aantal variaties te bedenken wat betreft de berichten die uitgewisseld worden tussen een IDP en een SP en andersom. Een voorbeeld van een mogelijke message flow is te zien in Figuur 2.



Figuur 2 SP initiated SSO met Artifact resolution [12]

In Figuur 2 is het volgende scenario te zien: een gebruiker wil gebruik maken van beveiligde content van een SP zonder dat de gebruiker zich heeft geauthentiseerd. De gebruiker authentiseert zich bij de IDP en gaat terug naar de SP waar hij de beweringen van de IDP presenteert om zo toegang te krijgen tot de beveiligde content. De SP kan nog extra informatie opvragen over de gebruiker aan de IDP met behulp van *artifact resolutioin*, deze stappen (stap 6 en 7) zijn niet noodzakelijk.

###### Twee type message flows

Er zijn twee type messages flows, de message flow die te zien is in Figuur 2 is een **SP initiated** scenario. Omdat het begint met een gebruiker die probeert de SP te benaderen. Als een gebruiker zich eerst zou authentiseren bij een IDP zou er sprake zijn van een **IDP inittiated** scenario. In dat geval zouden stap 1 en 2 worden overgeslagen. Een visuele weergave van het verschil tussen IDP en SP initiated message flows is te zien in Figuur 3.

##### **C:\Users\blur5\Dropbox\private\werk documenten\bronnen\flows\SAML IDP en SP flow.png**

##### 

#### Cross-Domain-SSO met SAML

Figuur IDP en SP initiated scenario’s

Deze paragraaf benadrukt hoe SAML het cookie probleem van Cross-Domain-SSO oplost.

Het “cookie probleem” is dat websites geen cookies van elkaar mogen gebruiken. Dat maakt het onmogelijk voor websites om de sessie van een gebruiker te delen door gebruik te maken van elkaars cookies.

SAML trotseert het cookie probleem door websites de rol van SP te laten vervullen en de websites niet zelf de sessie cookies bij te laten houden. De sessies worden bijgehouden in de cookie van een IDP, welke een eigen domein heeft. De websites communiceren met de IDP om te achterhalen of de gebruiker zich heeft geauthentiseerd en wat de rollen e.d. zijn die de gebruiker heeft. Hierdoor wordt het mogelijk voor een gebruiker om van de ene site naar de andere te gaan zonder opnieuw te hoeven authentiseren (er van uit gaand dat beide sites een COT hebben met dezelfde IDP). Beide sites communiceren namelijk met de IDP en die houdt de sessie van de gebruiker bij.

### Frameworks

In deze paragraaf wordt de volgende deelvraag beantwoord:

*Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?*

Voordat de onderzoeksvraag wordt beantwoord wordt eerst toegelicht waarom het wenselijk is om gebruik te maken van een framework.

**Waarom een framework?**

Gebruik maken van een framework voor het regelen van security gerelateerde aspecten van een project heeft bepaalde voordelen ten opzichte van het zelf ontwikkelen van een oplossing, namelijk:

* Gebruik maken van een framework verkort de tijd die nodig is voor het realiseren van de security aspecten van een project.
* Als er gebruik wordt gemaakt van een framework, kunnen onderhoud en aanpassing worden gedaan door het aanpassen van de configuratie. Het is dus niet nodig om de code aan te passen in de meeste gevallen.
* De kans of fouten is groter bij handgeschreven code en als er een fout zit in het framework is de kans dat deze snel gevonden wordt groter doordat deze software door een groter publiek wordt gebruikt.
* Door regelmatige updates wordt er wordt altijd gebruik gemaakt van de nieuwste technieken rondom beveiliging
* Een framework maakt vaak gebruik van veelgebruikte standaarden en technieken. Hierdoor wordt het makkelijker om technieken die in de toekomst ontwikkeld gaan worden te ondersteunen omdat deze vaak rekening houden met bestaande standaarden en technieken.

**Antwoord op de onderzoeksvraag**

Om de onderzoeksvraag te kunnen beantwoorden is er voornamelijk gebruik gemaakt van deskresearch. Er is uitgebreid onderzoek gedaan naar frameworks die zich richten op het realiseren van Cross-Domain-SSO en Federated Identities. Dit onderzoek had een lijst van 13 frameworks als resultaat. Er is een overzicht met algemene informatie van deze 13 frameworks opgesteld, zodat er beter kon worden geschat hoe bruikbaar en betrouwbaar een framework is. Voor het overzicht van de 13 frameworks zie bijlage F.

### Framework keuzen

In deze paragraaf wordt de volgende deelvraag beantwoord:

*Welk framework is het meest geschikt als basis voor het realiseren van een POC?*

Van de lijst van 13 frameworks, die het resultaat was van het beantwoorden van onderzoeksvraag 2, is gebruikt voor het maken van een shortlist van de vier meest belovende frameworks. Onderstaand overzicht is tot stand gekomen door de frameworks met elkaar te vergelijken. Hierbij is gekeken naar verschillende criteria, zoals bruikbaarheid in de toekomst en stabiliteit. Voor het volledige onderzoek naar frameworks zie bijlage F.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Framework | Shibboleth | JOSSO | openAM | Ping-Identity |
| Geschikt voor het koppelen van sites | ++ | ++ | + | ++ |
| Voorbeeld implementaties/ tutorials beschikbaar | ++ | ++ | ++ | ++ |
| Toegankelijkheid | ++ | ++ | + | + |
| Verwachte implementatie tijd | ++ | ++ | ++ | - |
| Heeft een forum | ++ | ++ | ++ | ++ |
| Duidelijk beeld van de architectuur | ++ | ++ | + | - |
| Start project | 2000 | 2004 | openSSO:  2005-2006 | SourceID:  2001 |

Legenda: ++ = goed

+ = voldoende

- = slecht

#### Afgevallen frameworks

Naar aanleiding van bovenstaande tabel, zijn openAM en Ping-Identity als eerste afgevallen in de selectieprocedure van het kiezen van het meest geschikte framework. Zij hebben de minste plusjes.

**openAM**: Is afgevallen omdat het project instabiel oogt. Het project bouwt voort op het door Oracle afgestote project “openSSO”. De documentatie van de architectuur van OpenAM is ook niet duidelijk genoeg.

**Ping-Identity**: is een commercieel bedrijf dat gebruik maakt van het open source project “SourceID”. Ping-Identity is afgevallen omdat het geld kost om van het product gebruik te maken en omdat informatie over de architectuur niet toegankelijk is.

Central Authentication Service (**CAS**) is een andere grote speler die vaak wordt genoemd als er over Cross-Domain-SSO wordt gesproken [13]. CAS is niet opgenomen in de shortlist omdat het gebruik maakt van één centrale server die alles regelt en het is onduidelijk in welke mate het geschikt is voor het realiseren van een Federated Identity. CAS is ook niet gebaseerd op SAML maar ondersteunt de standaard wel. De SLO faciliteiten zijn bijvoorbeeld gebaseerd op SAML, maar deze zijn gedateerd. Tot op heden (juni 2012) zijn deze gebaseerd op SAML versie 1.1 terwijl de meest recente versie 2.0 is.

#### JOSSO overtreft Shibboleth

Het resultaat van het selectie proces was dat er twee frameworks overbleven, Shibboleth en JOSSO.

Shibboleth is ontworpen zodat een site die een dienst aanbiedt niet de authenticatie van de dienst-afnemers hoeft te doen. De gebruikers worden geauthentiseerd door hun eigen organisatie. Het hoofdscenario met shibboleth begint met een gebruiker die gebruik wil maken van een service. De gebruiker moet dan zelf aangeven van welke organisatie hij of zij afkomstig is en wordt vervolgens doorgestuurd naar zijn eigen organisatie om in te loggen. Daarna wordt de gebruiker weer doorgestuurd naar de service waar hij of zij gebruik van wilde maken.

JOSSO is ontworpen zodat er snel oplossingen kunnen worden ontwikkeld waarmee meerdere organisaties elkaars gebruikers kunnen delen over het internet.

In tegenstelling tot Shibboleth hoeft een gebruiker met het JOSSO framework niet aan te geven van welke organisatie hij afkomstig is, dus is het gebruikersvriendelijker. Daarom is er voor JOSSO gekozen als het framework voor de realisatiefase van de opdracht.

Door het kiezen van het JOSSO framework kunnen de volgende requirements worden afgedekt:

### Uitvoering POC

In deze paragraaf wordt de volgende deelvraag beantwoord:

*Hoe moet het POC uiteindelijk worden gerealiseerd?*

Er is voor gekozen om het realiseren van het POC te verdelen in een aantal sub-taken (ook wel “projecten” genoemd). Zo kan er namelijk systematischer worden gewerkt aan het in kaart brengen van de functionaliteiten van het JOSSO framework. Het product wat voort komt uit een sub-taak wordt een “oplossing” genoemd.

De oplossingen zullen allemaal gebruik maken van het JOSSO framework en zullen worden ontwikkeld op een één laptop waarop een testomgeving wordt opgezet. Voor iedere oplossing wordt er gekeken of onderdelen van eerder gerealiseerde oplossingen kunnen worden hergebruikt om tijd te besparen.

De onderzoeksvraag is verder beantwoord door het framework in gebruik te nemen en bevindingen te documenteren. Doordat de ontwerp- en realisatie-aspecten zo omvangrijk zijn, zullen deze niet worden besproken in deze paragraaf maar in hoofdstuk 5. Eerst zal in hoofdstuk 4 worden omschreven hoe het JOSSO framework werkt.

# Het JOSSO framework

JOSSO is een Open Source internet SSO framework dat wordt ontwikkeld door het bedrijf “Atricore”. Het primaire doel van het JOSSO framework is: het mogelijk maken om, op een veilige manier, efficiënt standaard gebaseerde Cross-Domain-SSO oplossingen te realiseren. Het JOSSO framework maakt hiervoor intensief gebruik van de SAML standaard. Er wordt gebruik gemaakt van IDPs en SPs, deze worden in de documentatie van het JOSSO framework respectievelijk “Gateway” en “Agent” genoemd [14].



Het framework kan gratis gebruikt worden doordat het is uitgegeven onder een GNU-Lesser General Public License (LGPL) [15]. Atricore verdient dus niets aan de gebruikers van het framework. Het onderhouden van het framework wordt gedaan door een “core-team” dat tegen betaling gewenste functionaliteit implementeert. Het is ook mogelijk voor gebruikers van het framework om een bijdrage te leveren aan het project. Deze bijdrage wordt dan gecontroleerd door een core-developer. Dit is een gezonde structuur voor een opensource project want als er teveel mensen aan een project werken vertraagt dat de voortgang van het project doordat er teveel communicatie overhead ontstaat, dit wordt ook wel *Brooks’ law* genoemd [16]. Door het veiligstellen van de voortgang van het project en het laagdrempelig houden van het gebruik van het framework ontstaat er zekerheid dat het JOSSO project in de toekomst wordt voortgezet.

Dit hoofdstuk zal dieper ingaan op de technische werking van het JOSSO framework. Er zal worden toegelicht hoe, het JOSSO framework gebruik maakt van SAML, hoe de componenten van het JOSSO framework opgezet en geconfigureerd kunnen worden en wat de voordelen zijn van bepaalde opties die het JOSSO framework biedt.

## De Gateway (Identity provider)

In het JOSSO framework wordt gebruik gemaakt van op SAML gebaseerde gateway (IDP). Als er van JOSSO1 gebruik wordt gemaakt dan moet de gateway via commando’s op een applicatieserver worden geïnstalleerd. Het JOSSO1 framework ondersteunt alle populaire applicatieservers, hierdoor is er veel vrijheid in de keuze van de applicatieserver.

JOSSO2 maakt gebruik van een ingebouwde Apache Karaf server. Dit is een standaard onderdeel van het framework dat wordt gebruikt voor het hosten van gateways. De belangrijkste feature van een Apache Karaf server is dat deze Open Services Gateway initiative (OSGi) gebaseerde services ondersteunt. Het gebruiken van een dergelijke server betekent dat er meer applicaties naast elkaar kunnen draaien op de server en dat de server niet opnieuw opgestart hoeft te worden als er een nieuwe applicatie op wordt geïntroduceerd [17].

## De Agent (Service provider)

In het JOSSO framework wordt gebruik gemaakt van agents (SPs), deze zijn gebaseerd op de SAML standaard. Een agent kan eenvoudig worden geïnstalleerd op een applicatieserver, dit wordt ook wel “activeren” genoemd. Bij het activeren wordt de configuratie, die is gedaan op het SP element m.b.v. de console, overgezet naar de server met de benodigde applicatie software van de agent zelf en configuratiebestanden met daarin informatie over de IDPs die deel uitmaken van de COT waar de SP aan toebehoord.

**Een applicatie gebruik laten maken van het JOSSO framework.**

Een agent wordt gebruikt voor het beveiligen van een applicatie. Als een applicatie beveiligd is met een agent wordt deze applicatie ook wel “gejossifyd” genoemd. Het proces van het realiseren van een gejossifyde applicatie wordt “*jossifiën*” genoemd. Voor het jossifiën van een applicatie is het volgende noodzakelijk:

De aanpassingen aan het web.xml bestand hebben betrekking tot het instellen van FORM-based authenticatie [18]. De pagina die moet worden gebruikt voor het inloggen kan namelijk handmatig worden ingesteld met behulp van FORM based authenticatie. De best practise van het JOSSO framework is om de FORM based authenticatie zo te configureren dat de pagina die wordt aangeroepen (deze pagina bevindt zich in de applicatie) de gebruiker op een juiste manier te redirect naar de login pagina van de IDP (deze pagina bevindt zich op de gateway). Daar zal de gebruiker zich authentiseren en vervolgens weer worden terug geredirect naar de gejossifiede applicatie.

**Werking van de agent**

De redirects worden afgehandeld door een component dat op de applicatie server van de gejossifyde applicatie draait. Dit component wordt “de agent” genoemd. De agent filtert op bepaalde URLs voor het uitvoeren van redirects. Zo wordt bijvoorbeeld een URL zoals: *“domeinnaam.nl:8080/applicatienaam/****JOSSO\_USER\_LOGIN****”* onderschept door een agent. De agent filtert namelijk op het stuk “JOSSO\_USER\_LOGIN” in de URL. Met “JOSSO\_USER\_LOGIN” wordt aangegeven dat een gebruiker zich wil authentiseren en moet worden geridirect naar de login pagina van de IDP. Er zijn ook alternatieven, bijvoorbeeld dat een geautomatiseerd proces zich wil authentiseren, hiervoor zal dan een andere URL worden gebruikt.

## Beveiliging

Doordat niet iedereen die het internet gebruikt goede bedoelingen heeft moet er rekening gehouden met beveiliging. De gateways van het JOSSO framework zijn beveiligd tegen een aantal aanvallen die kwaadwillende zouden kunnen uitvoeren. De rest van deze paragraaf zal een toelichting geven op wat voor een manier het JOSSO framework is beveiligd tegen de meest voor de hand liggende aanvallen.

### Broken authentication and session management

Broken authentication en session management zijn veiligheidslekken die over het algemeen worden veroorzaakt door slordigheid. Bij een dergelijk lek maakt een aanvaller gebruik van een zwakheid waar een ontwikkelaar geen rekening mee heeft gehouden.

Het JOSSO framework gaat dit o.a. tegen door:

* Standaard authenticatie mechanismes te gebruiken, waardoor de kans op lekken wordt beperkt.
* Wachtwoorden kunnen worden gehashd voordat ze worden opgeslagen.
* Berichtuitwisseling met de SP kan worden gesignd, dat wil zeggen: dat er data wordt toegevoegd aan het bericht waarmee de zender van het bericht geverifieerd kan worden.
* Sessie timeout kan worden ingesteld.

SSL

Het JOSSO framework kan ook gebruik maken van Secure Sockets Layer (SSL). Hiermee wordt de netwerkverbinding tussen de client en de server versleuteld. Een hacker kan dan niet zomaar de informatie die uitgewisseld wordt onderscheppen, onder “informatie” valt ook het sessieID. SSL helpt dus ook tegen man in de middle aanvallen.

### CSRF

Cross site request forgery (CSRF) is een aanval waarmee een kwaadwillige de sessie van een gebruiker misbruikt om zo functies aan te roepen namens de gebruiker. JOSSO maakt het uitvoeren van een dergelijke aanval hoogdrempeliger door POST requests te gebruiken in plaats van GET requests. Dit betekend dat de methode aanroep niet kan worden uitgevoerd door middel van een gebruiker een URL te laten uitvoeren. De aanvaller zal het hele POST request moeten versturen om een functie te laten uitvoeren namens de gebruiker.

## Framework versies

Er zijn twee versies van het JOSSO framework, namelijk JOSSO1 en JOSSO2.

### JOSSO1

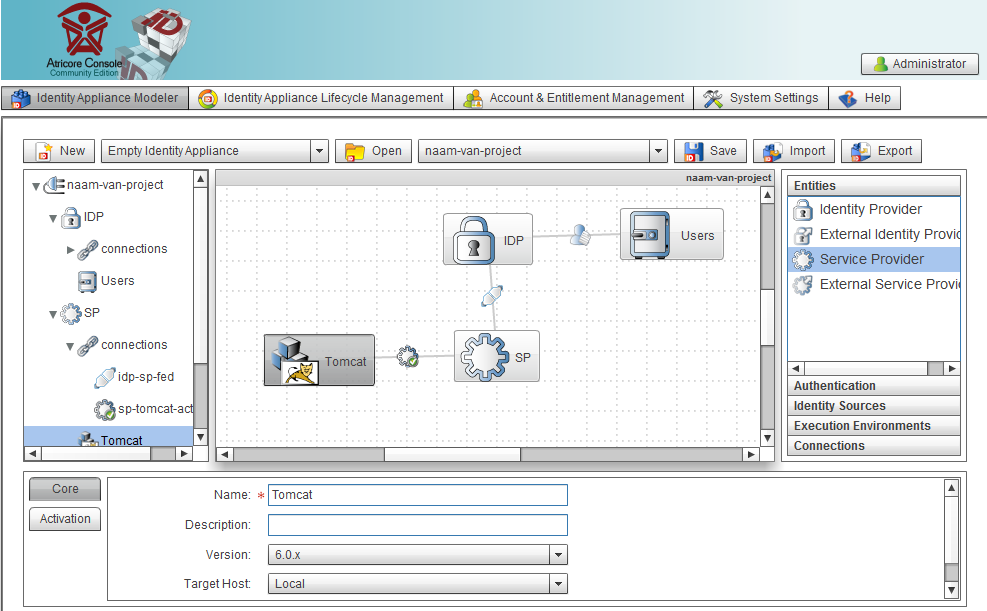
JOSSO1 is minder gebruikersvriendelijk dan JOSSO2. In JOSSO1 moet het gereed maken van een JOSSO gateway met commando’s worden gedaan. Hetzelfde geldt voor het installeren van agents. Het verbinden van deze entiteiten vereist handmatige configuratie in XML bestanden. Deze bevinden zich op de applicatieservers die worden gebruikt in de oplossing.

### JOSSO2

JOSSO2 is voor het grootste deel van de afstudeeropdracht gebruikt. Het voordeel van JOSSO2 is dat er niet meer gewerkt hoeft te worden met commando’s en dat er minder geconfigureerd hoeft te worden in XML bestanden. Een ander groot voordeel is dat JOSSO2 met een webconsole werkt. De webconsole ondersteunt het ontwerpproces van een *Identiteitsarchitectuur*, dit wordt verder toegelicht in de volgende paragraaf.

#### De console

De JOSSO installatie bevat een web console. Met de console kunnen veel aspecten van een project worden doorlopen. Met de console kan namelijk een COT worden ontworpen, gegenereerd en beheerd. Een screenshot van de JOSSO console, met een voorbeeld model er op, is te zien in Figuur 4.



Figuur 4 JOSSO console

De console ondersteunt MDD. Dat betekend dat er een aan de hand van een model een werkende oplossing kan worden gegenereerd.

De MDD ondersteuning van JOSSO is niet meegenomen is de afweging om het framework te gaan gebruiken. Ondanks dat heeft werken met de MDD functionaliteit van de console een aantal noemenswaardige voordelen, namelijk:

Er is deskresearch uitgevoerd waarbij is gekeken hoe werken met MDD in de praktijk wordt ervaren. Er zijn een aantal negatieven argumenten die vaak genoemd worden [19], namelijk:

* De tijd die het kost om code te laten genereren vanuit de modellen is buitenproportioneel.
* Code moet vaak handmatig nog worden aangepast om te kunnen compilen.
* De MDD tool beperkt de keuzevrijheid van technologieën de gebruikt kunnen worden.

Van deze problemen is bij het JOSSO framework geen sprake want er wordt geen code gegenereerd maar configuratiebestanden voor applicaties. Deze applicaties zijn SAML compliant wat betekent dat deze gekoppeld kunnen worden met iedere andere SAML applicatie, ook als deze niet geen onderdeel zijn van het JOSSO framework. Hierdoor ontstaat veel flexibiliteit voor het opzetten van een COT.

##### Identity sources

Een Identity source is de data laag van een provider. Een IDP kan uit een Identity source gegevens van een gebruiker halen die vervolgens in een SAML bewering kunnen worden geplaatst. Een SP kan een Identity source gebruiken om de gegevens SAML beweringen van een IDP te verrijken en zo dus eenvoudig een Federated Identity realiseren.

De JOSSO2 community edition (CE) ondersteunt vier type Identity sources, namelijk: Databases, LDAP directories, XML bestanden en Identity vaults.

Een *Identity Vault* is een derby database, die is inbegrepen in de installatie van het JOSSO framework, waarin gebruikers kunnen worden bijgehouden. Deze database kan vanuit de console worden beheerd en is ideaal voor het testen van het framework, doordat de vault al standaardtabellen heeft voor het opslaan van gebruikers.

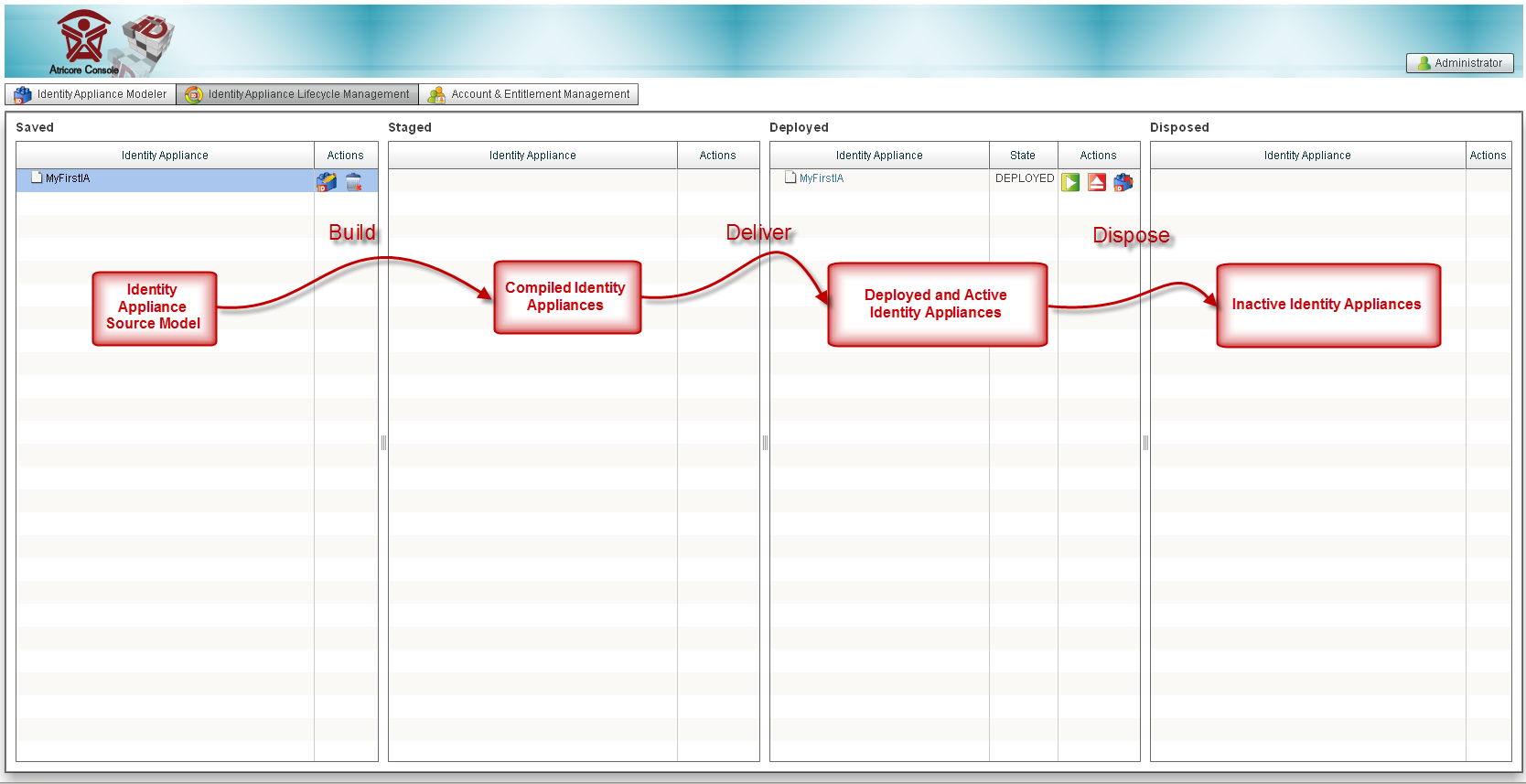
Het is ook mogelijk een *zelf ontworpen database* te gebruiken. Als een database als Identity source gebruikt wordt moeten bepaalde query’s worden opgegeven, zodat het framework met de database kan communiceren. De query’s maken gebruik van parameters, Dit betekent dat er een “?” in de query wordt opgenomen op de plaats waar een variabele wordt ingevoerd. Door parameters op deze manier te gebruiken wordt SQL injection tegen gehouden [20]. Dit komt doordat parameters aan bepaalde eisen moeten voldoen, zoals korter zijn dan de maximale lengte van de kolom waar de parameter een waarde van zou moeten zijn, of het bevatten van SQL syntax zoals “ ; ” of “ ’ ” of “ -- ”.

##### Het opzetten van een COT

De console ondersteunt MDD en maakt het mogelijk om een werkende omgeving te genereren. Er zijn elementen beschikbaar die met hulp van “drag and drop” gecreëerd kunnen worden en aan elkaar verbonden. Onder deze elementen zijn onder andere SAML entiteiten beschikbaar, namelijk de eerder in dit hoofdstuk genoemde IDPs en SPs. Van beide bestaat een internal en external variant. Het verschil daartussen is dat voor een external een metadata-file aangeleverd moet worden om deze in de COT op te nemen [21]. Bij een internal entiteit is er geen metadata-file nodig omdat het JOSSO framework hier volledige controle over heeft en dit genereert zonder de ontwikkelaar van de COT er mee te belasten.

##### Life cycle management

Het JOSSO framework biedt ook life cycle management aan. De life cycle van een project kent 4 fases: *opgeslagen*, *gebouwd*, *gestart* en *afgeschreven*. In Figuur 5 is te zien hoe dit er uit ziet in de JOSSO console.



Figuur 5 life cyle management

Fase 1 wordt afgehandeld met behulp van de *modeler* die te zien is in Figuur 4. Als een model is opgeslagen kan het worden gecompileerd zodat er een uitvoerbaar project ontstaat. Dit uitvoerbare project kan vervolgens worden gestart. Bij het starten van het project worden de SPs niet noodzakelijk beschikbaar omdat deze op andere applicatieservers draaien. Wel wordt de IDP (mits deze niet external is) gestart en zal beschikbaar zijn voor authenticatie requests. De IDP draait op de apache-karaf server die standaard door het JOSSO framework wordt gebruikt.

# Proof of concept

Dit hoofdstuk presenteert de resultaten die behaald zijn in de realisatiefase van de afstudeeropdracht.

Er is voor gekozen om meerdere POC oplossingen te maken in plaats van één grote complexe oplossing omdat dit beter uitvoerbaar bleek. De beweegredenen hiervoor worden uitgelegd in de eerste paragraaf van dit hoofdstuk (5.1).

Het realiseren van één POC oplossing wordt in dit hoofdstuk een POC project genoemd. De POC projecten die besproken gaan worden zijn uitgevoerd met behulp van het JOSSO framework, beschreven in hoofdstuk 4. Iedere paragraaf van dit hoofdstuk zal één POC project toelichten. Hierbij zullen aspecten aan bod komen zoals het doel van het POC en de bevindingen die zijn gedaan tijdens het uitvoeren. De testomgeving, de aanpak en de problemen die naar voren zijn gekomen bij het realiseren van iedere POC oplossing zijn in detail te vinden in hoofdstuk 6.5 van bijlage C.

## Onderzoeksaanpak vaststellen

Het doel van het eerste POC project was aantonen dat het mogelijk is om met behulp van het JOSSO framework een Cross-Domain-SSO oplossing te realiseren welke gebruik maakt van een Federated Identity. Bij deze oplossing zouden alle componenten (SPs en een IDP) op verschillende virtuele machines (VM) draaien zodat ze ieder een eigen IP adres zouden hebben, een weergave van de testomgeving is te zien in Figuur 6.

#### C:\Users\blur5\Dropbox\private\werk documenten\bronnen\plaatjes\josso\test omgeving.pngProblemen

Het plan bleek te ambitieus te zijn om uit te voeren. Het bevatte te veel overbodige complexiteit om mee te werken. Het werken met VMs zorgt namelijk alleen maar voor een complexere omgeving die geen waarde toevoegt voor het demonstreren van de functionaliteit van het JOSSO framework. Cross-Domain-SSO kan ook worden aangetoond door alleen het hosts bestand aan te passen.

Figuur 6 Opzet eerste testomgeving

### Resultaat

Door geconfronteerd te worden met het probleem van een “overbodig complexe omgeving” is besloten om de aanpak voor toekomstige POC oplossingen te veranderen. In plaats van alle mogelijke eisen in één keer proberen aan te tonen, worden deze opgedeeld in meerdere kleinschaligere POC projecten. Met de filosofie dat op deze manier het behalen van resultaten sneller zal gaan.

Voor alle toekomstige oplossingen wordt het gebruiken van VMs vermeden, in plaats daar van wordt het hosts bestand gebruikt. Het hosts bestand is als volgt gebruikt: voor iedere oplossing zijn er andere domeinnamen gebruikt, een voorbeeld van de extra regels die hiervoor aan het hosts bestand zijn toegevoegd:

127.0.0.1 serviceprovider1.nl

127.0.0.1 serviceprovider2.nl

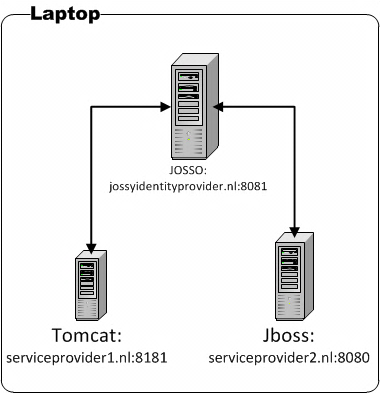
127.0.0.1 jossyidentityprovider.nl

|  |
| --- |
|  |

Hoewel de applicaties op hetzelfde IP adres bereikbaar zijn zal een browser nog steeds niet toestaan dat deze cookies met elkaar delen omdat de domeinnaam niet overeenkomt. Hierdoor kan er, zonder shortcuts, Cross-Domain-SSO worden aangetoond.

## De basisfunctionaliteit testen

Het doel van de deze oplossing, is het aantonen dat er met behulp van het JOSSO framework een Cross-Domain-SSO oplossing gerealiseerd kan worden. Hierbij wordt geen Federated Identity gebruikt maar van één overkoepelende digitale identiteit. Voor het realiseren van de oplossing is de testomgeving die te zien in is Figuur 7 gebruikt. Op de servers zijn demo-applicaties van het framework geïnstalleerd, deze demo-applicaties worden “partnerapp” genoemd.



Figuur 7 Testomgeving voor het aantonen van basisfunctionaliteit.

problemen

Er hebben zich geen problemen voorgedaan tijdens het realiseren van deze POC oplossing. Wel is preventief het port nummer van de tomcat server gewijzigd om conflicten met de Jboss server te voorkomen.

### Resultaat

Met behulp van deze oplossing is aangetoond dat de aanpak van het werken zonder VM’s effectief is en dat het JOSSO framework Cross-Domain-SSO ondersteunt. Ook is er ervaren dat projecten binnen een paar uur gerealiseerd kunnen worden met behulp van de JOSSO-console. Er is sprake van een enorme tijdwinst ten opzichte van zelf een oplossing vanaf de grond opbouwen met behulp van een programmeertaal, omdat dat enkele weken zou duren.

De volgende functional requirements zijn afgedekt met behulp van dit POC project:



Ook zijn de volgende non-functional requirements afgedekt:

## 

## Custom application

Het doel van de deze oplossing is: vaststellen hoeveel moeite het kost om een standaard java web-applicatie deel uit te laten maken van een COT met behulp van het JOSSO framework.

Dit is gedaan door een Hallo-world applicatie te maken die gebruik maakt van BASIC authenticatie [22] en die vervolgens aan te sluiten op een COT. Hiervoor moest de BASIC authenticatie worden vervangen door FORM authenticatie.

### Resultaat

Met behulp van dit POC project is vastgesteld dat het jossifiën van een applicaite binnen een korte tijd kan worden gedaan zonder tegen grote problemen aan te lopen.

Door de hello-world pagina alleen maar toegankelijk te maken voor gebruikers met de rol “role1”, is aangetoond dat JOSSO role based access control (RBAC) ondersteunt. Dit is gunstig omdat RBAC de basis is voor sommige Must-have requirements.

## Behoud van de login pagina

Het doel van de deze POC oplossing was uitzoeken hoe de originele login pagina van een website behouden kan worden (en onder hetzelfde adres beschikbaar kan blijven). Het behouden van de originele login pagina van een website heeft bepaalde voordelen. Zo zorgt het er voor dat gebruikers niet schrikken van een login scherm waarmee ze onbekend zijn, of van een domeinnaam waarmee ze niet vertrouwd zijn. Dit houdt het gebruiken van de website laagdrempelig.

Voor deze POC oplossing is de Hello-world applicatie van de voorafgaande POC oplossing als basis gebruikt. Vanuit berichten op de forums van de JOSSO gemeenschap bleek dat het eenvoudiger is om de gewenste functionaliteit te bereiken met het gebruik van JOSSO1. Dus er is van deze gelegenheid gebruik gemaakt om te experimenteren met JOSSO1 om zo een beter beeld te krijgen van alle mogelijkheden die er zijn, met alle bijbehorende voor- en nadelen.

Het behouden van de login pagina is gerealiseerd door de redirect pagina naar de IDP te vervangen door een pagina zonder redirect. De nieuwe pagina is een gewoon login scherm waar een gebruiker zijn username en wachtwoord moet invullen. Vervolgens wordt er een authenticatie verzoek gedaan aan de IDP met behulp van een http form post submit. Zo wordt een redirect vermeden (de code van de nieuwe login-pagina is te zien in bijlage G).

### Resultaat

Met behulp van dit POC project is aangetoond dat het mogelijk is de originele inlog pagina van een website te behouden. Door het gebruiken van een http post in plaats van http get wordt het aanvallen van de site met behulp van CSRF ook ontmoedigd. Dit komt doordat het vrij eenvoudig is met http GET om een CSRF aanval te doen, bij een http post is dat anders want daar moet een aanvaller gebruik maken van het form wat gebruik wordt in het http post submit.

Met het behouden van de login pagina wordt het mogelijk zelf te bepalen welke content zich bevindt op de loginpagina. Dit maakt het mogelijk rekening te houden met CSRF door bijvoorbeeld alle content weer te geven via HTTPS.

Met behulp van dit project is de volgende non-functional requirement afgedekt:

### **Branding van de login pagina als alternatief.**

Er is ook onderzoek gedaan naar alternatieve mogelijkheden. Een alternatieve mogelijkheid die het JOSSO framework ondersteunt heet “branding”. Het toepassen van branding houdt in dat de login pagina op de IDP zo wordt aangepast dat deze past bij de applicaties/websites die er gebruik van maken [23]. Het voordeel van deze techniek is dat hiermee het “login pagina probleem” voor alle SPs binnen een COT kan worden opgelost.

De “gebrande” login pagina moet gebruikers een vertrouwd gevoel geven. Het enige waar gebruikers nog aan zullen moeten wennen is dat het login scherm zich bevindt op een ander domein als waar ze gebruik van willen maken. Een gebruiker vertrouwd maken met deze techniek kan worden ondersteund door een info pagina te maken voor de gebruiker waarop wordt uitgelegd waarom de gebruiker wordt verzocht in te loggen op een ander domein als waar hij of zij gebruik van wil maken.

## Werken met een Federated Identity

Het doel van de vierdePOC oplossing is bewijzen dat er met behulp van het JOSSO framework een Federated Identity kan worden ondersteund. Ter verduidelijking van de eisen is er een scenario opgesteld waarbij een gebruiker met SSO navigeert tussen twee gekoppelde sites, het uitgeschreven scenario is vinden in bijlage H.

Het aantonen dat het JOSSO framework werken met een Federated Identity ondersteunt is gedaan door een Federated Identity te creëren waarbij drie gebruikersdatabases aan elkaar gekoppeld werden, op basis van de loginnaam van een gebruiker. De IDP had één database met inlog gegevens en de SP’s één database met daarin aanvullende gegevens van een gebruiker.

### Problemen

Tijdens dit project hebben zich problemen voor gedaan met het tot stand brengen van een account-koppeling en er is een observatie gedaan van onwenselijk cashe gebruik.

*Koppeling op basis van Email adres*

Het koppelen op basis van loginnaam is in de praktijk niet wenselijk omdat in de praktijk vaak gebruik wordt gemaakt van verschillende loginnamen. Daarom is er getracht een link te leggen tussen accounts op basis van Email adressen, hiervoor is gekozen omdat:

* Een Email adres behoort maar aan een persoon toe.
* Als er een koppeling gemaakt kan worden met een email adres, betekent dit dat er ook een willekeurig ander attribuut van de gebruiker gebruikt wordt voor het tot stand brengen van een koppeling (zolang als het attribuut uniek is voor iedere gebruiker).

Het is echter niet gelukt om een koppeling op basis van Email adres te realiseren. Het koppelen van accounts op basis van Email adres is wel een configuratie optie voor de “account linkage policy” van een SP. Dit suggereert dat er wel ondersteuning voor is of dat deze wordt verwacht. Omdat de handleiding van JOSSO niet erg diep ingaat op de technische werking van het framework is het niet duidelijk hoe JOSSO het account linken op basis van email ondersteunt.

Mogelijke oorzaken van het probleem zouden kunnen zijn dat de gebruikte query’s niet de return waarde geven die het JOSSO framework verwacht. Andere factoren die van invloed zouden kunnen zijn, zijn waar mogelijk weggehaald. Zo is bijvoorbeeld het databasemodel aangepast zodat deze overeenkomt met de demo-database van JOSSO. Er is ook veelvuldig geprobeerd hulp te zoeken bij de gemeenschap van JOSSO maar deze gaf geen gehoor.

*Observatie van cache gebruik*

Tijdens het testen van de oplossing viel het op dat het cache gebruik van de browser onverwacht gedrag veroorzaakt. Namelijk dat als een gebruiker uitlogt en opnieuw inlogt de beveiligde pagina uit de cache wordt opgehaald, met als gevolg dat de gebruiker niet de pagina te zien krijgt die hij verwacht te zien als hij of zij is ingelogd. Hiervoor is een hard refresh nodig zodat de pagina opnieuw wordt opgehaald van de server. Dit probleem kan worden verholpen door de pagina zelf aan te passen en code toe te voegen die er voor zorgt dat de cache niet wordt gebruikt.

### Resultaat

Met behulp van dit POC project is de volgende functionele requirement aangetoond:

Effect van een Federated IdentityEr is ook getest hoe het systeem omgaat met de attributen van een gebruiker als zijn accounts aan elkaar gekoppeld zijn (de account van de IDP en de lokale account op van de SP). Dit kan op 3 manieren namelijk:

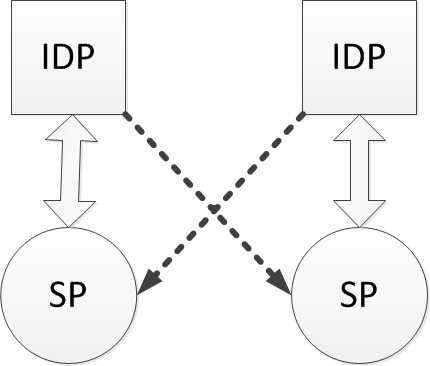
1. De attributen van de josso-account worden gebruikt
2. De attributen van de domein-account worden gebruikt
3. Alle attributen worden gebruikt.

Het omschakelen tussen deze 3 opties kan worden gerealiseerd door de configuratie van een SP aan te passen. Door gebruik te maken van de JOSSO console kan dit binnen enkele minuten.

## Behoud van oude account

In de voorgaande oplossingen werden de accounts (inlog gegevens) van gebruikers niet behouden. De inlog gegevens die een website had voordat deze werd opgenomen in een COT werden niet meer gebruikt. De centraal staande IDP dicteert welke set inlog gegevens gebruikt wordt. Dit is niet wenselijk omdat een plotselinge verandering van login gegevens verwarrend kan zijn voor een eindgebruiker.

Het doel van de dezePOC oplossing is: aantonen dat het mogelijk is om een Federated Identity te introduceren aan een website, waarbij de originele inlog gegevens van gebruikers behouden blijven. In andere woorden: dat iedere website zijn eigen userdatabase heeft, die ook wordt gebruikt voor de authenticatie van gebruikers en niet alleen voor het verrijken van de digitale identiteit van een al ingelogde gebruiker die van een ander domein binnen de COT afkomt.



Deze oplossing is gerealiseerd door de twee mogelijke message flows van SAML (SP initiated en IDP initiated, beschreven in hoofdstuk 3.3.2.2) te combineren. Zo heeft iedere SP zijn originele userdatabase bij zijn “preferred IDP”, dat gebruikt zal worden bij het SP initiated scenario. Maar de SP zal ook gebruikers toelaten die zich hebben geauthentiseerd bij de andere IDP, het IDP initiated scenario.

### Problemen

Een probleem dat naar voren kwam bij deze oplossing was het instellen van de preferred IDP. Het probleem was dat beide IDP’s de database van de andere IDP ook aanspraken om de username te controleren, dit veroorzaakte onverwacht gedrag. Er is hier sprake van een bug in het JOSSO framework. Omdat de preferred IDP instellen met behulp van een checkbox zou moeten gaan. Op de forums wordt er ook gezegd dat de IDP met prioriteit feature niet betrouwbaar is (dit bericht is op 23-1-2012 geplaatst).

### Resultaat

Met behulp van dit POC project zijn de volgende functionele requirements aangetoond:



Wachtwoord encryptie

Doordat de inloggegevens van gebruikers een centrale rol spelen in deze oplossing, is tijdens realiseren van deze oplossing is ook aangetoond dat het JOSSO framework wachtwoord encryptie ondersteunt. Dit is gedaan door de configuratie van de IDP aan te passen. Hiermee is vastgesteld dat het JOSSO framework encryptie ondersteund in de vormen van MD5 SHA1. Dit betekend dat het JOSSO framework minder gevoelig is voor Broken Authentication and Session Management [24] doordat dit Insecure Cryptographic Storage [25] voorkomt.

### Een alternatieve oplossing

Een alternatieve oplossing waarmee ook zou kunnen worden bereikt dat gebruikers hun oude accounts kunnen behouden is: **één complexe externe IDP** maken die van meerdere Identity sources gebruik maakt. Deze oplossing heeft als voordeel dat de IDP precies voldoet aan de eisen die er aan worden gesteld. Er zijn wel een aantal nadelen verbonden aan deze oplossing, namelijk:

Verschillen in onderhoud

Onderhoud gaat nodig zijn in de toekomst, ongeacht de gekozen oplossing. Stel een klant wil nog een website koppelen aan de COT, dan moet er minimaal een Identity Source en een SP aan de COT worden toegevoegd. De rest van deze paragraaf ligt toe wat voor gevolgen een dergelijk onderhoud zou hebben voor een oplossing waarbij één complexe IDP wordt gebruikt en wat de gevolgen zouden zijn voor een oplossing waarbij meerdere JOSSO IDPS worden gebruikt.

Als er gewerkt wordt met *één complexe IDP,* dan zal er programeerwerk nodig zou zijn voor het toevoegen van de nieuwe Identity source. De nieuwe SP zal ook moeten worden aangepast om gebruik te gaan maken van de IDP. De bestaande SP’s hebben deze aanpassing niet nodig doordat ze al een COT vormen met de IDP. De tijd die het kost om deze aanpassing door te voeren is afhankelijk van hoe snel er geprogrammeerd kan worden. Naar verwachting zal dit naarmate de IDP complexer wordt steeds langzamer gaan. Dit komt doordat er niet altijd dezelfde mensen aan het project werken. Dit heeft tot gevolg dat er meer communicatie overhead ontstaat (*Brooks’ law*) en dat de code uiteindelijk moeilijk te begrijpen is door de verschillende programmeerstijlen van mensen. Dit verhoogt de kans op fouten bij het doorvoeren van wijzigen.

Als er gewerkt wordt met *meerdere JOSSO IDPS,* dan zal er een nieuwe IDP met de nieuwe Identity source en een nieuwe SP aan de COT moeten worden toegevoegd. Deze componenten kunnen met behulp van de console worden geconfigureerd en gegenereerd. Het grootste verschil met de eerder genoemde oplossing is dat alle bestaande SPs nu ook de beweringen van de nieuwe IDP moeten gaan honoreren. Dit kan in de console worden gedaan door het toevoegen van een extra connectie, dit zal ongeveer 1 minuut in beslag nemen. Doordat er dus voornamelijk geconfigureerd moet worden en niet geprogrammeerd zal het doorvoeren van een dergelijke wijziging binnen een paar uur gedaan kunnen worden.

Downtime

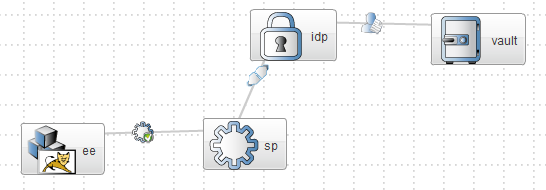
De downtime voor beide oplossingen is effectief hetzelfde. Want of er nou SPs worden gewijzigd of één IDP, het herstarten van de JOSSO server is altijd noodzakelijk.

## Ondersteuning voor spring security

Het doel van de deze oplossing is: Aantonen dat een applicatie die gebruik maakt van Spring Security ook gebruik kan maken van het JOSSO framework.

### Aanpak

Hiervoor is een eenvoudige JOSSO omgeving opgezet, zodat het doel zo snel mogelijk bereikt wordt. De testomgeving bestaat uit: een IDP met een Identity vault, een SP met een tomcat7 server als execution environment met daarop een applicatie draaiende die gebruik maakt van spring security. Het model van de omgeving is te zien in Figuur 8.



Figuur 8 model

Na het jossyfiën van de applicatie, is geverifieerd dat de applicatie werkt zoals bedoeld door te testen of een gebruiker toegang tot de applicatie werd geweigerd als hij niet beschikte over de juiste rol. Dit was het geval, een gebruiker kreeg een *403 error* als hij niet beschikte over de juiste rol; als hij deze rol wel had kreeg de gebruiker wel toegang.

#### Problemen

Voordat de applicatie ontwikkeld kon worden moest er uitgezocht worden hoe spring security in de praktijk gebruikt kan worden. Dit koste enige tijd. Verder zijn er geen praktische problemen geweest tijdens dit POC project.

### Resultaat

Er is een applicatie ontwikkeld die gebruik maakt van spring security voor het beschermen van content.

De applicatie draait op de tomcat server en heeft beschermde content, waar alleen gebruikers met de rol ROLE\_ADMIN toegang tot hebben. Door een gebruiker aan te maken in de Identity vault met de rol ROLE\_ADMIN en vervolgens in te loggen op de app is aangetoond dat spring security applicaties gebruik kunnen maken van JOSSO.

Met behulp van dit POC project kan de volgende non-functional requirement worden vervuld:

# Conclusie en aanbevelingen

Deze opdracht kwam voort uit de wens voor een Federated Identity oplossing die het mogelijk maakt voor gebruikers om tussen websites te navigeren met SSO, oftewel Cross-Domain-SSO. Het realiseren van Cross-Domain-SSO kan niet worden gedaan met standaard SSO oplossingen. Dit komt doordat deze gebruik maken van domein specifieke cookies. Er moet een techniek worden gebruikt die in staat is Cross-Domain-SSO te realiseren waarbij het niet uitmaakt dat cookies domein specifiek zijn. Populaire SSO technieken zoals OpenID en OAuth vallen snel af, omdat deze niet ontworpen zijn voor het realiseren van Cross-Domain-SSO. De aangewezen techniek voor het realiseren van Cross-Domain-SSO oplossingen is SAML. Dit komt doordat SAML nadrukkelijk is ontworpen voor het, op een veilige manier, realiseren van Cross-Domain-SSO oplossingen. SAML is een XML standaard die definieert hoe SP’s en IDP’s met elkaar kunnen communiceren. Doordat SAML alleen communicatie methodes definieert, legt het geen imitaties op aan een oplossing maar creëert het juist veel flexibiliteit, hierdoor kan een SAML oplossing veel vormen aannemen.

SAML lost het cookie probleem op door websites (SP’s) te ontlasten van de taak gebruikers te authentiseren. Het authentiseren wordt namelijk gedaan door een IDP, die vervolgens beweringen over gebruikers communiceert met de SP’s. Het cookie probleem is hier mee opgelost doordat de IDP een eigen domein heeft en daar de cookie op beheert. SP’s van andere domeinen kunnen de sessie en attributen van een gebruiker achterhalen door met de IDP te communiceren.

Doordat SAML geniet van populariteit zijn er veel frameworks die ondersteuning bieden voor de standaard. Het gebruiken van een framework heeft bepaalde voordelen zoals periodieke updates die de veiligheid van het framework up-to-date houden waardoor het framework de nieuwste bedreigingen kan weerstaan. De kans dat er programmeerfouten in een framework zitten is ook kleiner dan wanneer er op eigen kracht een oplossing wordt gerealiseerd. Dit komt doordat een framework zich richt op een specifiek onderdeel en doordat het door veel mensen wordt gebruikt waardoor fouten, als ze er zijn, sneller worden gesignaleerd.

Na een uitgebreid selectieproces is er gekozen om te gaan werken met het JOSSO framework. JOSSO is een opensource project wat in stand blijft door tegen betaling specifieke functionaliteit in het framework te implementeren. Terwijl het gebruiken van het framework kosteloos kan doordat het framework is uitgegeven onder een LGPL. Het JOSSO framework maakt gebruikt van SAML. Hierdoor kan het realiseren van Cross-Domain-SSO probleemloos. Voor het koppelen van user-accounts tussen de domeinen biedt het JOSSO framework ook veel vrijheid, waardoor er altijd voor de best passende methode kan worden gekozen.

Een belangrijke feature van het JOSSO framework is de manier waarop het MDD ondersteuning biedt. Het JOSSO framework genereert namelijk configuratie voor standaard JOSSO componenten. Hierdoor zijn de problemen die zich voordoen met het genereren van code, zoals lange genereertijden en compileerproblemen, niet aanwezig. De MDD functionaliteit biedt noemenswaardige voordelen. Zo kan bijvoorbeeld het koppelen van applicaties (websites) binnen enkele uren worden gerealiseerd in plaats van enkele weken. Een ander voordeel is dat het beheren van een op JOSSO gebaseerde oplossing eenvoudiger is dan het beheren van een oplossing waarvan de code handgeschreven is. Hierdoor heeft een JOSSO gebaseerde oplossing een langere levensverwachting.

# Evaluatie

Dit hoofdstuk wil ik graag gebruiken om te vertellen over hoe ik mijn afstudeerstage bij E-ID heb ervaren.

Mijn eerdere werkervaringen waren bij een grote multinational, E-ID is het eerste MKB waar ik bij heb gewerkt. Het grootste verschil is dat ik bij E-ID meer van mijn collega’s heb leren kennen. Dit maakte het werk een stuk leuker. Ook vond ik de vrijheid die ik kreeg omtrent werktijden erg fijn, zo was het bijvoorbeeld geen probleem als ik een paar uur thuis werkte als er veel file stond.

Zes dagen voor de aanvang van mijn afstudeerstage werd de opdracht die ik had afgekeurd. Dit had tot gevolg dat er binnen drie dagen een vervangende opdracht voor mij was opgezet. De opdracht die ik heb gekregen vond ik erg leuk en interessant omdat deze aansloot met de kennis de ik al had vergaard van security op het gebied van software-engineering, iets wat de afgewezen opdracht niet had.

Ik vind het jammer dat ik mijn onderzoek en ideeën over de beste oplossing niet heb kunnen toepassen in de praktijk, want ik geloof dat daar ongetwijfeld nog een hoop mee te leren valt.

Het schoolonderdeel waar ik het meeste aan heb gehad is de IMS proftaak in het 3e jaar. Daarbij moest ik een verslag opstellen van een onderzoek. De feedback die ik op dit verslag heb gekregen heb ik goed kunnen gebruiken om mijn scriptie te verbeteren.

Alles bij elkaar ben ik tevreden over mijn afstudeerperiode. Ik heb veel kunnen leren van mijn begeleiders en Ik zie ook gelegenheid om de technische kennis die ik heb opgedaan in de toekomst toe te passen, omdat ik echt geloof dat de vraag naar Federated Identity oplossingen de komende jaren zal toenemen. Dit komt doordat op het gebied van gebruikersvriendelijkheid, Federated Identity oplossingen het vervolg zijn van standaard SSO oplossingen.

# Literatuurlijst

1. Grit, R. (2011). *Project Management*. Groningen: Wolters Noordhoff
2. Kempen, P. & Keizer, J. (2011) *Competent afstuderen en stagelopen. Een advieskundige benadering*. Groningen: Wolters Noordhoff
3. Z.a.(z.d.). (online) *Bedrijfsprofiel*. <http://www.e-id.nl/overeid>
4. Tomz, K. (7-7-2008). (online) *Single Sign-on in ASP.NET and Other Platforms*, <http://www.codeproject.com/Articles/27576/Single-Sign-on-in-ASP-NET-and-Other-Platforms>
5. Z.a. (5-2012). (online) *MoSCoW Method*. <http://en.wikipedia.org/wiki/MoSCoW_Method>
6. Madsen, P. & Maler, E. (12-4-2005). *SAML V2.0 Executive Overview*
7. Goodner, M. & Hondo, M. & Nadalin, A. & McIntosh, M. & Schmidt, D. (28-5-2007). (online) *Understanding WS-Federation*. <http://msdn.microsoft.com/en-us/library/bb498017.aspx>
8. z.a. (25-9-2011). (online) *OpenID Explained*. <http://openidexplained.com/>
9. Walther, P. (2011*). Hauptseminar Web Engineering im WS 2010/2011 - Identity & Authentication*. Chemnitz
10. Ragle, D. (31-7-2008). (online) *Simple Comments and OpenID*. <http://www.webreference.com/programming/perl/comments/openid/2.html>
11. Cantor, S. & Moreh, J. & Philpott, R. & Maler, E. (15-3-2005). *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*
12. Ragouzis, N. & Hughes, J. & Philpott, R. & Maler, E. & Madsen, P. & Scavo, T. (25-3-2008). *Security Assertion Markup Language (SAML) V2.0 Technical Overview*
13. z.a. (31-5-2007). (online) Central Authentication Service. <http://onlamp.com/onlamp/2007/05/31/cas-single-sign-on-with-jifty.html>
14. z.a. (2011). (online) JOSSO 2.2 Reference *Guide*. <http://www.josso.org/docs/en-US/html/>
15. Free Software Foundation, inc. (2-1999). (online) *GNU LESSER GEBERAL PUBLIC LICENSE V 2.1.*<http://www.gnu.org/licenses/lgpl-2.1.html>
16. Brooks, F. (1995) *The Mythical Man-Month*
17. Kuipers, J. (2009).(online) *OSGi: modulariteit zonder herstarts voor Enterprise Java applicaties* <http://www.weloveit.nl/osgi-enterprise-java-2009-1.htm>
18. Mitchell, S. (14-1-2008). (online), *An Overview of Forms Authentication*. <http://www.asp.net/web-forms/tutorials/security/introduction/an-overview-of-forms-authentication-cs>
19. Cliff (20-2-2006). (online) *Pros and Cons of MDA Code Generators?* <http://ask.slashdot.org/story/06/02/21/0126210/pros-and-cons-of-mda-code-generators>
20. Post, E. Scheffers, E. (4-6-2012). (online, niet publiekelijk toegankelijk) *Security checklist*
21. Rasmusson, S. (22-2-2012). (online) *SAML Metadata.* <http://mylifewithjava.blogspot.nl/2012/02/saml-metadata.html>
22. Z.a. (15-4-2012)(online) *Configure Basic Authentication (IIS 7).* <http://technet.microsoft.com/en-us/library/cc772009(v=ws.10).aspx>
23. Brigandi, G. Oyuela, S.G. (16-1-2009). (online) *Branding JOSSO.* <http://www.josso.org/confluence/display/JOSSO1/Branding+JOSSO>
24. OWASP Foundation. (2-5-2010). (online) *Top 10 2010-A3-Broken Authentication and Session Management* <https://www.owasp.org/index.php/Top_10_2010-A3>
25. OWASP Foundation. (24-10-2011). (online) *Top 10 2010-A7-Insecure Cryptographic Storage* <https://www.owasp.org/index.php/Top_10_2010-A7>

Bijlagen

Indeling:

Bijlage **A** bevat de verklarende woordenlijst.

Bijlage **B** bevat het PID.

Bijlage **C** bevat het onderzoeksdocument.

Bijlage **D** bevat een lijst van de vastgestelde requirements.  
Bijlage **E** bevat een lijst met toelichting van de onderzochte protocollen.  
Bijlage **F** bevat onderzochte frameworks met keuze argumentatie.  
Bijlage **G** bevat de code van een domein-eigen loginpagina.  
Bijlage **H** bevat een scenario wat is gebruikt om de afstudeeropdracht levendiger te maken.

1. Verklarende woordenlijst

**Account** - De combinatie van een gebruikersnaam en wachtwoord.

**Artifact binding** - is een manier van communiceren waarbij de bewering van de IDP niet meteen in zijn geheel wordt gecommuniceerd. De IDP stuurt de gebruiker terug naar de SP met een referentie (een artifact). Een artifact wordt normaal gesproken gecommuniceerd als een parameter in een HTTP GET request waarbij de artifact in de URL wordt opgenomen. De SP vraagt dan weer aan de IDP voor de volledige bewering die bij het artifact hoort. De volledige bewering wordt dan gecommuniceerd (resolved) met behulp van een SOAP call tussen de SP en de IDP [12].

**Authenticatie** – is het proces van vaststellen of iemand of iets, is, wat hij, zij of het beweerd te zijn.

**Autorisatie** – is het proces van iets of iemand toestemming geven om iets te hebben of doen.

**Broken authentication and session management** - het niet goed afschermen van gebruikersnamen en wachtwoorden of sessies waardoor hackers er misbruik van kunnen maken.

**Brook’s law** – Het toevoegen van programmeurs aan een software project stagneert de voortgang.

**Browser** - een software applicatie waarmee content op het internet kan worden weergegeven. Bijvoorbeeld: Internet Explorer of Firefox

**Cache** – een snel toegankelijk geheugen wat wordt gebruikt voor veelgebruikte functies en gegevens in op te slaan.

**Checkbox** – een vakje op een computerscherm dat door een gebruiker kan worden aangevinkt om aan te geven dat de gebruiker iets wel of niet wil.

**Cookie** – een bericht van een webserver naar een webbrowser dat door de browser wordt opgeslagen in een tekst bestand. Dit bestand (de cookie) wordt bij ieder verzoek aan de webserver meegestuurd. Cookies kunnen allerlei soorten informatie bevatten zoals surf gedrag, informatie over de gebruiker en informatie over de sessie van de gebruiker.

**Console** – een venster in een beeldscherm waarmee direct met een achterliggend systeem kan worden gecommuniceerd.

**Content** – De inhoud van een webpagina, zoals tekst en afbeeldingen.

**COT** – Circle of Trust: Een keten van SPs en IDPs waarbij beweringen worden uitgewisseld en vertrouwd.

**Cross-Domain-SSO** – SSO functionaliteit over meerdere domeinen.

**CSRF** – Cross Site Request Forgery: een hacker aanval waarbij de hacker de browser van een gebruiker manipuleert om zo de sessie van de gebruiker te misbruiken. Bij CSRF wordt dus het vertrouwen wat een website heeft in een gebruiker misbruikt.

**Deskresearch** – het verzamelen van informatie die al beschikbaar is, bijvoorbeeld op papier of op het internet.

**Diepteonderzoek** – Een onderdeel van het TSP waarin antwoord wordt gegeven op een onderzoeksvraag.

**Digitale identiteit** - is de online weergave van een individu binnen een domein, zoals vastgesteld door het individu zelf of door anderen. Een individu kan meerdere digitale identiteiten bezitten in meerdere domeinen.

**Domein** - een unieke aanduiding van een locatie op het internet

**Downtime ­**– De tijd dat een server niet bereikbaar is.

**Federated Identity**: een digitale identiteit waarmee een persoon van meerdere losstaande systemen gebruik kan maken.  
  
**Federated SSO**: Single Sign On functionaliteit in combinatie met een federated identity.

**Fieldresearch** – iedere activiteit die er op is gericht informatie te verzamelen, met het gebruik van methodes zoals interviewen, bespreken en directe observaties.

**Framework**: een softwarepakket dat het realiseren van andere software ondersteund.

**Functional requirement** – Een eis aan een systeem/applicatie dat deze in staat moet zijn om een bepaalde taak te verrichten.

**Gebruiker** – een persoon die gebruikt maakt van een domein, applicatie, website of ander product.

**Gebruikersvriendelijkheid** – de hoogte waarin een eindgebruiker een product efficiënt, effectief en naar tevredenheid kan gebruiken.

**Hard refresh** – een verversing van de content op een website waarbij de cache niet wordt gebruikt.

**Hash** - het resultaat van een hash functie. Een hash functie laat een onomkeerbare berekening los op data, met als resultaat een code die altijd hetzelfde is voor een bepaalde set data, met een vaste lengte.

**Hello world applicatie** – is een simpele applicatie die als die gerund wordt het bericht “hello world” laat zien.

**Hosts bestand** – Een bestand dat in bepaalde operating systemen wordt gebruikt om domein namen te koppelen aan IPadressen.

**Identiteitsarchitectuur** – een architectuur die omschrijft hoe er met de identiteit van personen of objecten wordt omgegaan.

**IDP** – Identity provider – een entiteit die een principaal kan authentiseren voor service providers.

**Insecure Cryptographic Storage** – Het niet geencrypt opslaan van gegevens, met als gevolg dat het makkelijker wordt voor onbevoegden om deze gegevens te bemachtigen en te gebruiken.

**Jossyfiën** – Het proces van het aanpassen van een applicatie zodat deze kan aansluiten op een SP component van het JOSSO framework.

**JSP pagina –** Java Server Pages: Een technologie waarmee design en functionaliteit van elkaar gescheiden kunnen worden. Op deze manier kan er meer functionaliteit beschikbaar worden gesteld op een pagina.

**LGPL** - GNU Lesser General Public License: Software ontwikkeld onder deze licentie mag geheel vrijblijvend worden aangepast en gebruikt. Zelfs in combinatie met software die wordt ontwikkeld onder een commerciële licentie.

**Man-in-the-middle** – Een aanval waarbij een hacker tussen een client en server in gaat zitten en alle informatie die wordt uitgewisseld onderschept.

**MDD** - model driven development: een techniek waarbij, vanuit een model, code kan worden gegenereerd.

**MD5** – Message-Digest algorithm 5: een veel gebruikte crypto grafische hash functie waar een 128-bit hash waarde mee kan worden berekend.

**Metadata** – wordt gebruikt in het SAML protocol om een service te omschrijven. Metadata bevat informatie zoals welke services beschikbaar zijn en hoe deze benaderd kunnen worden. Metadata is gedefinieerd is XML en zowel IDPs als SPs gebruiken het om zichzelf benaderbaar te maken.

**MySQL database** – een database management systeem wat gebruik maakt van SQL.

**Non-functional requirement** – Een eis aan een systeem/applicatie dat deze over bepaalde eigenschappen moet beschikken, denk hierbij aan snelheidseisen en dergelijken.

**OASIS** - is een acroniem voor Organization for the Advancement of Structured Information Standards (OASIS). Deze organisatie richt zich op het definiëren van standaarden zoals SAML.

**Onreduceerbaar complex systeem** – Een systeem bestaande uit meerdere componenten die samenwerken om een functionaliteit te realiseren, waarbij het verwijderen van een component er voor zocht dat het hele systeem niet meer kan functioneren.

**PID** – Project Initiation Document: een document wat er voor dient het project te definiëren zodat het uitvoervaar wordt. Het document bevat onder andere de doelstelling van het project, een rolverdeling en een plan van aanpak.

**POC** – Proof Of Concept: is een kleinschalig project met het doel een bepaalde functionaliteit of eigenschap vast te stellen / demonstreren.

**Principaal** – de leidende identiteit.

**Profiel** - de attributen die bij een account horen (bijv. een Email adres) en dus een digitale identiteit bepalen.

**Query** – een vraag (een *SQL query* is een vraag of commando naar een database toe gericht).

**RBAC** – Role Based Access Control: Autorisatie op bases van rollen.

**Redirect** – een doorverwijzing (deze kan geautomatiseerd worden).

**Requirement** – een eis die wordt gesteld aan een product. Dit kan betrekking hebben tot functionaliteiten en eigenschappen.

**RFC** – Request For Comment: is een omschrijving van een standaard of een nieuw/gewijzigd internet of netwerking protocol. Als een standaard wordt voorgesteld wordt deze publiekelijk beschikbaar gesteld. Zo kan er nog commentaar worden gegeven en kunnen eventuele wijzigingen worden doorgevoerd voordat de standaard definitief wordt gemaakt.

**SAML** - is een acroniem voor Security Assertion Markup Language, het is een op XML gebaseerde standard voor het communiceren van identiteitsinformatie tussen organisaties.

**Scenario** – een omschrijving van achtereenvolgende gebeurtenissen.

**Screenshot** – een afbeelding van een beeldscherm.

**Sessie** – de periode waarin een gebruiker een verbinding met een systeem in stand houdt.

**SHA1** – Secure Hash Algorithm 1: een onomkeerbare hash functie, vergelijkbaar met MD5 alleen is SHA1 iets veiliger maar zwaarder om uit te voeren.

**SLO** – single logout: voor meerdere applicaties in een keer uitloggen.

**SQL** – Structured Query Language: een programmeertaal die is ontworpen voor het beheren van databases.

**SMART** – is een bepaalde eis die gesteld kan worden aan een definitie. Een SMART definitie is Specifiek, Meetbaar,Haalbaar, Realistisch en Tijdsgebonden.

**SP** – Service provider: een entiteit die een dienst aanbiedt.

**SQL injection** – het invullen van SQL query’s in datavelden waardoor de achterliggende database wordt gemanipuleerd.

**SSL** - Secure Sockets Layer: een protocol wat het mogelijk maakt versleutelde verbindingen op te zetten op het internet.

**SSO** – single sing on: eenmalig inloggen voor het gebruik van meerdere diensten.

**Standard** – een erkende schriftelijk vastgestelde beschrijving van set regels en eigenschappen.

**Stub applicatie** – een kleine applicatie die geen andere eigenschappen heeft als het bezitten van basis functionaliteiten.

**URL** – Uniform Resource Locator: is het adres van een website. Bijvoorbeeld: “*www.voorbeeld.nl/homepage*”

**Use-Case** – Een methodische omschrijving van een scenario. In een Use-Case worden alle mogelijke sequenties, tussen een systeem en een gebruiker, die doorlopen kunnen worden bij het behalen van een specifiek doel in kaart gebracht.

**XML** – eXtensible Markup Language is een taal die een set regels die een bepaald formaat voor documenten definiëren, zodat het document door mensen en machines gelezen kan worden.

**VM** – Virtual Machine: Een volledig geïsoleerde operating systeem installatie op een normaal operating systeem.

**VMware** – een applicatie die virtuele machines kan draaien.

**403 error** - Forbidden, deze error wordt opgegooid als een gebruiker toegang wil tot een webpagina maar dat toegang wordt geweigerd.

1. Project Initiation Document

|  |  |  |
| --- | --- | --- |
| PROJECT:  ( ) | | |
|  |  |  |
|  |  |  |

Documenthistorie

Revisies

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Status | Datum | Wijzigingen |
| 0.1 | Concept | 05-03-2012 | Eerste invullingen |
| 0.2 | Concept | 06-03-2012 | Verwerking mondelinge feedback |
| 0.3 | Concept | 13-03-2012 | Verwerking schriftelijke feedback (Edvard Scheffers) |
| 0.4 | Concept | 16-03-2012 | Verwerking schriftelijke feedback (Frank Haverkort) |
| 1.0 | Concept | 20-03-2012 | Verwerking schriftelijke feedback (Frank Haverkort) |

Goedkeuring

Dit document behoeft de volgende goedkeuringen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versie | Datum goedkeuring | Naam | Functie | Paraaf |
| 1.0 | 21-03-2012 | Frank Haverkort | Docent begeleider |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Distributie

Dit document is verstuurd aan:

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum verzending | Naam | Functie |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Managementsamenvatting

Doel van dit document

Dit document heeft tot doel het project te definiëren, als basis te dienen voor het management ervan en de beoordeling van het succes van het project mogelijk te maken.

De twee belangrijkste redenen voor gebruik van dit document zijn:

* om er zeker van te zijn dat het project een gezonde basis heeft voordat het projectteam gevraagd wordt zich aan het project te committeren;
* om te dienen als basisdocument op grond waarvan het projectteam en de Projectmanager de voortgang en wijzigingen kunnen toetsen en bewaken en vragen omtrent geldigheid van het project tijdens de uitvoering ervan kunnen beoordelen.

Aanleiding

De aanleiding voor dit project is dat klanten van E-ID, die in het bezit zijn van meerdere domeinen (bijvoorbeeld websites). Klanten willen dat deze domeinen aan elkaar gekoppeld worden. Door middel van deze koppeling moet het mogelijk worden voor eindgebruikers om zonder opnieuw in te hoeven loggen, te navigeren tussen de verschillende domeinen. Het grote probleem wat zich hierbij voordoet is het identificeren van gebruikers als ze van het ene domein naar het andere navigeren. Hiervoor moet een oplossing worden gevonden die in veel omgevingen inzetbaar is zodat het koppelen van domeinen in de toekomst sneller gedaan kan worden.

In dit project wordt de volgende onderzoeksvraag centraal gesteld:   
*Hoe kunnen identiteitsmanagement systemen van verschillende domeinen op een veilige manier worden samengebracht tot een SSO systeem voor een gebruiker?*

Om dit te onderzoeken is de onderzoeksvraag opgedeeld in een aantal deelvragen:

* Aan welke functionaliteiteisen en veiligheidseisen moet een POC voldoen?
* Welke technieken en protocollen zijn er op het gebied van SSO?
* Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?
* Welk framework is het meest geschikt als basis voor het realiseren van een POC?
* Hoe moet het POC uiteindelijk worden gerealiseerd?

Globale aanpak

*Aanpak:*

Eerst zullen de eisen worden vastgesteld waaraan het eindresultaat moet voldoen. Als dat is gedaan zal er een ontwerp voor een POC gemaakt worden, daarna zal er onderzocht worden of er frameworks zijn die mogelijk kunnen helpen in het realiseren van een POC. Tegen het einde van de contractperiode zullen alle werkzaamheden en bevindingen worden gedocumenteerd.

*Methode:*

Voor het project zal de faseringsmethode van Grit worden gebruikt. Deze methode is binnen dit project als volgt ingedeeld:

Voor het project is een globale planning gemaakt, dit lijkt er veel op een waterval methodiek. Echter zal tijdens het project ook gebruik worden gemaakt van een agile ontwikkelstrategie. In een agile strategie wordt er “as we go” continu gekeken wat de beste manier is om het project voort te zetten. Op deze wijze kan er makkelijk worden omgegaan met eventuele wijzigingen die naar voren komen uit onderzoeken of problemen. Deze agile strategie wordt niet in alle fases van het project toegepast.

Globale kosten en doorlooptijd

Er is geen beeld van de totale kosten voor dit project. Het project zal lopen van 05-03-2012 t/m 13-8-2012

Inhoudsopgave

[1 Inleiding 13](#_Toc321319890)

[2 Achtergrond 14](#_Toc321319891)

[3 Projectdefinitie 15](#_Toc321319892)

[3.1 Projectdoelstellingen 15](#_Toc321319893)

[3.2 Gekozen oplossing of aanpak 16](#_Toc321319894)

[3.3 Scope van het project 16](#_Toc321319895)

[3.4 Productdecompositiestructuur (PDS) op projectniveau 17](#_Toc321319896)

[3.5 Productstroomschema (PSS) op projectniveau 18](#_Toc321319897)

[3.6 Productbeschrijvingen op projectniveau 18](#_Toc321319898)

[3.7 Projectbudget 18](#_Toc321319899)

[3.8 Wijzigingsbudget 18](#_Toc321319900)

[3.9 Benodigde Resources 18](#_Toc321319901)

[3.10 Uitsluitingen 19](#_Toc321319902)

[3.11 Afhankelijkheden 19](#_Toc321319903)

[3.12 Randvoorwaarden 19](#_Toc321319904)

[3.13 Aannames 19](#_Toc321319905)

[4 Projectorganisatiestructuur 20](#_Toc321319906)

[4.1 Opdrachtgever (E-ID) 20](#_Toc321319907)

[Rolbeschrijving 20](#_Toc321319908)

[Projectgerelateerde taken 20](#_Toc321319909)

[Specifieke verantwoordelijkheden 20](#_Toc321319910)

[4.2 Projectborging, Projectmanager, Projectsupport (Edvard Scheffers) 20](#_Toc321319911)

[Rolbeschrijving 20](#_Toc321319912)

[Projectgerelateerde taken 20](#_Toc321319913)

[Specifieke verantwoordelijkheden 20](#_Toc321319914)

[4.3 Teammanager (Frank Haverkort) 21](#_Toc321319915)

[Rolbeschrijving 21](#_Toc321319916)

[Projectgerelateerde taken 21](#_Toc321319917)

[Specifieke verantwoordelijkheden 21](#_Toc321319918)

[Bemiddelen bij problemen. 21](#_Toc321319919)

[4.4 Uitvoerende kracht (Hugo van der Zandt) 21](#_Toc321319920)

[Rolbeschrijving 21](#_Toc321319921)

[Projectgerelateerde taken 21](#_Toc321319922)

[Specifieke verantwoordelijkheden 21](#_Toc321319923)

[4.5 Klant (Erik Post) 21](#_Toc321319924)

[Rolbeschrijving 21](#_Toc321319925)

[Projectgerelateerde taken 21](#_Toc321319926)

[Specifieke verantwoordelijkheden 21](#_Toc321319927)

[5 Projectbeheersing 22](#_Toc321319928)

[5.1 Rapportage 22](#_Toc321319929)

[5.2 Voortgangsbewaking 22](#_Toc321319930)

[5.3 Toleranties 22](#_Toc321319931)

[5.4 Risicomanagement 22](#_Toc321319932)

[5.5 Kwaliteitsbewaking 23](#_Toc321319933)

[5.6 Wijzigingsprocedure 23](#_Toc321319934)

[5.7 Escalatieprocedure 23](#_Toc321319935)

[6 Planning 24](#_Toc321319936)

[6.1 Milestones 24](#_Toc321319937)

[6.2 Werkwijze 25](#_Toc321319938)

[Bijlage A: Communicatieplan 26](#_Toc321319939)

[Bijlage B: Scenario’s 27](#_Toc321319940)

# Inleiding

**Doel van dit document**

Dit document is opgesteld om alle relevante basisinformatie en uitgangspunten van het project vast te leggen om het op de juiste wijze te kunnen besturen. Het heeft tot doel het project te definiëren, als basis te dienen voor het management ervan en de beoordeling van het succes van het project mogelijk te maken.

Dit Project-initiatie-document (of PID) behandelt de volgende fundamentele aspecten van het project:

* Wat beoogt men met het project te bereiken?
* Waarom is het belangrijk om deze doelstellingen te bereiken?
* Wie zijn er betrokken bij het managen van het project en wat zijn hun rollen en verantwoordelijkheden?
* Hoe en wanneer zullen de maatregelen die in dit PID besproken worden gerealiseerd worden?

Het document wordt gebruikt:

* om er zeker van te zijn dat het project een gezonde basis heeft voordat de Stuurgroep gevraagd wordt zich aan het project te committeren;
* om te dienen als basisdocument op grond waarvan de Stuurgroep en de Projectmanager de voortgang en wijzigingen kunnen toetsen en bewaken en vragen omtrent geldigheid van het project tijdens de uitvoering ervan kunnen beoordelen.

# Achtergrond

De context van het project

E-ID heeft klanten met eindgebruikers en die de behoefte hebben om tussen domeinen te navigeren, zonder opnieuw te moeten inloggen.

Een case die goed gebruikt kan worden is de case van een klant die twee websites heeft. Een website om boeken te verkopen en een website om proefexamens te maken, beide websites (ofwel domeinen) hebben hun eigen identiteitsmanagement systeem waar in principe dezelfde gebruikers zitten. In de ideale situatie kan deze groep gebruikers tussen de websites navigeren zonder opnieuw in te moeten loggen.

Er is behoefte aan een manier om identiteitsmanagement systemen aan elkaar te koppelen en zo een vorm van SSO mogelijk te maken.

Hiervoor is geen kant en klare oplossing beschikbaar. Er zijn wel frameworks beschikbaar die ondersteuning bieden voor het ontwikkelen van een systeem en/of techniek die SSO over meerdere domeinen (die wellicht wel aan bepaalde criteria moeten voldoen) mogelijk maakt. Het is de bedoeling dat het systeem en/of techniek die ontwikkeld wordt platform onafhankelijk ingezet kan worden dus het blijft maatwerk.

Concrete aanleiding voor het project

De aanleiding van dit project is dat E-ID klanten met eindgebruikers heeft, en deze eindgebruikers hebben behoefte aan een SSO oplossing voor het navigeren tussen domeinen, onder een domein wordt verstaan, een omgeving met een eigen groep geregistreerde gebruikers en een aantal services waar iemand met de juiste autorisatie gebruik van kan maken. (bijvoorbeeld een website).

E-ID wil een systeem en/of techniek ontwikkelen die platform onafhankelijk kan worden ingezet. Het doel van het project is het leveren van een POC van een systeem en/of techniek die verschillende identiteitsmanagement systemen aan elkaar kan koppelen en zo een SSO oplossing realiseert tussen domeinen.

Redenen, waarom voor een projectmatige aanpak is gekozen

Het grote voordeel van projectmatig werken is dat je kunt faseren en tussenresultaten vast kunt leggen. Als eerste wordt onderzocht wat de problemen zijn, vervolgens worden de problemen geanalyseerd, er wordt voor mogelijke oplossingen gezocht en de beste oplossing zal worden uitgewerkt.

De fasering van dit project is verder uitgewerkt in “Productstroomschema (PSS) op projectniveau”

Beschrijving van de huidige situatie en de eventuele problemen of tekortkomingen van deze situatie

In de huidige situatie zijn er klanten met meerdere domeinen, die ieder een identiteitsmanagement systemen hebben. Het is mogelijk dat eindgebruikers bij beide domeinen in het systeem staan. De eindgebruiker kan niet tussen domeinen navigeren zonder opnieuw in te loggen. Efficiënter zou zijn als eindgebruikers maar een keer hoeven inloggen om tussen domeinen te kunnen navigeren. Ze zijn immers bij beide bekend en beide domeinen zijn van dezelfde organisatie. Het nadeel nu is dat oplossingen die niet SSO zijn, klant onvriendelijk zijn. Dit komt doordat de klant opnieuw moet inloggen en dit verhoogt de drempel tot het gebruik maken van het andere domein (website).

# Projectdefinitie

## Projectdoelstellingen

**Doel van het project**

Het doel van dit project is, onderzoek naar en het bouwen van een prototype.

**Het onderzoek**

In dit project wordt de volgende onderzoeksvraag centraal gesteld:

Hoe kunnen identiteitsmanagement systemen van verschillende domeinen op een veilige manier worden samengebracht tot een SSO systeem voor een gebruiker?

Om dit te onderzoeken is de onderzoeksvraag opgedeeld in een aantal deelvragen:

1. Aan welke functionaliteiteisen en veiligheidseisen moet een POC voldoen?
2. Welke technieken en protocollen zijn er op het gebied van SSO?
3. Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?
4. Welk framework is het meest geschikt als basis voor het realiseren van een POC?
5. Hoe moet het POC uiteindelijk worden gerealiseerd?

**De gewenste situatie**

De gewenste situatie ziet er als volgt uit: er een POC gerealiseerd waarin een realisatie van SSO is gedaan en gebruikers op een veilige manier tussen verschillende domeinen kunnen navigeren. Het POC zal ook een aantal scenario’s ondersteunen, deze zijn beschreven in “PID-Bijlage B: Scenario’s”. Ook zal er documentatie zijn van de gebruikte technieken en protocollen met waar toepasselijk een verantwoording waarom bepaalde keuzes gemaakt zijn.

**Voordelen en mogelijkheden**

Het grote voordeel van dit project is dat er een generieke oplossing gevonden wordt op een probleem waar mogelijk meerdere klanten mee zitten. Met behulp van dit POC is het mogelijk om deze klanten binnen zeer korte tijd van een oplossing te voorzien.

## Gekozen oplossing of aanpak

De volgende werkwijze wordt gehanteerd: Er wordt gewerkt volgens de faseringsmethode van Grit, deze methode houdt de volgende fases aan:

* Definitie fase, wat moet er gebeuren
* Ontwerp fase, hoe moet het gebeuren
* Realisatie fase, uitvoering

De definitie fase zal bestaan uit desk research en field research, bij desk research zal er via literatuur en internet gezocht worden naar omschrijvingen van het probleem en hoe de oplossing er uit moet zien. Bij field research worden klanten benaderend en worden hun wensen in beeld gebracht.

De ontwerp fase, deze fase zal bestaan uit een analyse van de oplossingsmethodes. Het doel is in deze fase het vaststellen van de vorm die de oplossing zal krijgen.

In de realisatie fase zal het prototype worden gemaakt en de documentatie die daar bij hoort.

Iedere fase zal een aantal tussenproducten opleven, hierover is meer te vinden in hoofdstuk 6 “Planning”.

## Scope van het project

Het project zal in zijn huidige vorm binnen E-ID blijven. Erik Post zal de rol van (interne) klant vervullen.

Onder de scope van dit project valt onder andere: Onderzoeken, Ontwerpen, realiseren, testen en documenteren.

Buiten de scoop valt: Het implementeren van het systeem/de techniek die ontwikkeld gaat worden, en onderzoek naar de juridische aspecten valt ook buiten de scope van dit project.

## Productdecompositiestructuur (PDS) op projectniveau

PDS Project

POC

B

Documentatie

A

Applicatie

C Onderzoek + Rapporten

A1

Ontwerp

A2

Product

B1

PID

B2

Eindverslag

C1Sub-onderzoeken

C2

TSP

A1.1

architectuur

A1.2

Usecases

## Productstroomschema (PSS) op projectniveau

Het project zal door een aantal fasen gaan.

Definitiefase:

1. Functioneel afbakenen, wat is het gewenste resultaat en wat is mogelijk binnen de tijd die beschikbaar is voor het project.
2. Onderzoeksvraag/vragen gaan beantwoorden/afbakenen (bevat o.a. onderzoek naar frameworks)
3. Vaststellen van de beveiligingseisen (Non functionals)

Ontwerpfase:

1. Selectie voor te gebruiken framework of extra design (Als niets past, zal een framework opgezet moeten worden (Extra design))
2. Architectuur (De vorm van het eindresultaat definiëren)

Realisatiefase:

1. POC die koppelingen tussen domeinen mogelijk maakt.
2. Het documenteren van bevindingen, conclusies en advies.

Producten die hier uit voortkomen zijn vastgelegd in hoofdstuk 6 “Planning”.

## Productbeschrijvingen op projectniveau

Het eindproduct is een POC van een veilig systeem en/of techniek, die tussen domeinen een SSO oplossing kan realiseren. Bij het POC komt een uitgebreide Nederlandstalige documentatie over: wat er is gemaakt, waarom voor bepaalde keuzes is gekozen, hoe het is gemaakt en hoe problemen zijn opgelost die naar voren zijn gekomen tijdens het project (hieronder valt ook het onderzoeksrapport).

## Projectbudget

Er is geen budget vastgesteld voor dit project.

Wel zijn er begeleidingsuren voor de stagiair, een huurauto en een laptop en een onkostenvergoeding voor de stagiair beschikbaar gesteld.

## Wijzigingsbudget

N.v.t.

## Benodigde Resources

In onderstaande tabel staan de resources die nodig zijn bij dit project.

|  |
| --- |
| Een werkplek met stroom en internet voorzieningen |
| Contact uren met Edvard Scheffers (mentor/projectleider) |
| Contact uren met Erik Post (interne klant) |
| Contact uren met Frank Haverkort (docent begeleider) |

## Uitsluitingen

Het eindproduct is niet voor een externe klant bedoeld en het zal ook niet gebruikt worden voor interne productie systemen binnen E-ID.

Er is geen sprake van geheimhouding binnen dit project.

Er zal geen onderzoek worden gedaan naar de mogelijke juridische kant van dit project.

## Afhankelijkheden

Interfaces en relaties

met programma’s,

n.v.t.

met andere projecten,

Als er gebruik wordt gemaakt van open source projecten moet gecontroleerd worden of de licentie van het open source project geschikt is.

met bedrijfsonderdelen,

n.v.t.

met andere organisaties.

n.v.t.

## Randvoorwaarden

De randvoorwaarde voor dit project is dat er eens in de twee weken een bespreking is van de voorgang van het project.

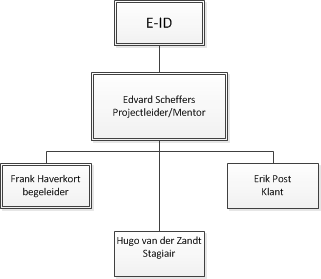
Ook zal er eens in de twee weken een voortgangsrapportage naar de docent begeleider worden gestuurd.

## Aannames

Een aanname voor het POC is dat als er meerdere domein aan elkaar gekoppeld zijn dat er tussen alle domeinen genavigeerd mag worden.

Ook wordt aangenomen dat het koppelen van accounts op basis van persoonlijke gegevens juridisch gezien mag. Dit mag omdat de gegevens allemaal vrijwillig gegeven zijn aan dezelfde klant door de eindgebruikers. De gegevens bevinden zich alleen niet in hetzelfde domein.

# Projectorganisatiestructuur

**

## Opdrachtgever (E-ID)

### Rolbeschrijving

E-ID is het bedrijf waarvoor het project wordt uitgevoerd.

### Projectgerelateerde taken

Beschikbaar stellen van een werkplek, een laptop en een auto.

### Specifieke verantwoordelijkheden

Betaald eventuele kosten die bij het project naar voren komen.

## Projectborging, Projectmanager, Projectsupport (Edvard Scheffers)

### Rolbeschrijving

Stage mentor.

### Projectgerelateerde taken

Managen en Adviseren. Technische ondersteuning, documenten reviewen

### Specifieke verantwoordelijkheden

Bijsturen wanneer dat nodig is.

## Teammanager (Frank Haverkort)

### Rolbeschrijving

Stage begeleider.

### Projectgerelateerde taken

Aanspreek punt bij conflicten & voortgang bewaking.

### Specifieke verantwoordelijkheden

Bemiddelen bij problemen.

## Uitvoerende kracht (Hugo van der Zandt)

### Rolbeschrijving

Realiseren van het project.

### Projectgerelateerde taken

Onderzoeken, Ontwerpen, realiseren, testen en documenteren.

### Specifieke verantwoordelijkheden

Rapporteren over voorgang

## Klant (Erik Post)

### Rolbeschrijving

Interne klant.

### Projectgerelateerde taken

Controle uitvoeren op de functionele eisen en de kwaliteit van het product.

### Specifieke verantwoordelijkheden

Feedback geven.

# Projectbeheersing

## Rapportage

Hiervoor is “PID-Bijlage A: Communicatieplan” opgesteld.

## Voortgangsbewaking

De voortgang zal worden bewaakt door middel van overleg. De onderstaande overlegmatrix ligt dit verder toe.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Overleg | Aanwezig | Frequentie | Tijdstip | Doel | Onder­werpen | Notulen |
| **Projectteamoverleg** | Mentor, klant en stagiair | 1 x per 2 weken | flexibel | Afstemming deelprojecten | Voortgang, verwachte problemen en Open Issues | In de vorm van aantekeningen |
| **Tussentijdse product beoordeling** | Docent begeleider, stagiair en mogelijk mentor | Faseovergang | nader te bepalen | Verkrijgen van feedback | Fase-eindrapport, Faseplan, Projectplan, PID, Risico's | In de vorm van aantekeningen of Email |

## Toleranties

Er wordt agile gewerkt dus er is van te voren geen vaste planning die als richtlijn kan worden gebruikt voor toleranties met betrekking tot uitloop. Wel zal er in de realisatie fase gebruikt worden gemaakt van JIRA, een project tracking tool. Dit wordt gedaan zodat de stagiair ervaring op kan doen met taakgericht werken en het inschatten van de hoeveelheid tijd die nodig is voor het uitvoeren van bepaalde taken.

## Risicomanagement

De risico’s die een rol spelen bij dit project zullen hier worden toegelicht.

Het grootste risico van dit project is dat er een teamlid uitvalt en niet meer in staat is zijn functie te vervullen. De kans hierop is echter erg klein dus hier zullen geen maatregelen voor getroffen worden.

Het is mogelijk dat een laptop buitengebruik geraakt en dat er bestanden verloren gaan. Om te voorkomen dat documenten verloren gaan zullen onderzoeksdocumenten in confluence worden gezet en overige documenten opgeslagen worden in een extern systeem, Dropbox.

Het is mogelijk dat een fase in van het project langer duurt dan verwacht, Mocht deze achterstand niet in te halen zijn door het sneller af te ronden van een volgende fase zal het project mogelijk verlengd worden, met bijvoorbeeld als gevolg dat de einddatum niet 13-08-2012 maar 23-08-2012 is.

Mocht het project sneller afgerond zijn dan verwacht met als gevolg dat er tijd over is dan zal de scope van het project worden uitgebreid. Hoe dit zal worden gedaan is beschreven in hoofdstuk 5.6 “Wijzigingsprocedure”.

Voor bepaalde documenten die moeten worden opgeleverd in de loop van dit project, moet een template worden gehanteerd. Het zou kunnen dat de vorm van deze template totaal niet past bij de praktijk. Als dit het geval is zal overlegt worden met de instantie die te template heeft opgelegd hoe hier mee moet worden omgegaan.

## Kwaliteitsbewaking

Kwaliteitsbewaking zal worden gedaan door middel van voortgangsvergaderingen en verzoeken voor feedback aan de docent begeleider, mogelijk zal er ook worden bijgestuurd door de opdrachtgever.

## Wijzigingsprocedure

Wijzigingen worden naar voortschrijdend inzicht doorgevoerd. Dit houdt in dat naarmate er meer bekend wordt het verdere verloop van het project wordt aangepast.

## Escalatieprocedure

Mochten zich erg grote problemen voor doen tijdens het project zal er tijd worden vrijgemaakt voor het vinden van een oplossing, de docent begeleider zal hiervan op de hoogte worden gebracht.

Mochten er problemen zijn binnen het project team dan zal de docent begeleider bemiddelen en zal er naar een oplossing worden gezocht.

# Planning

## Milestones

Binnen de fases van Grit wordt er naar verschillende milestones toegewerkt.

De onderstaande tabel bevat een ruwe planning van wanneer een milestone gehaald moet zijn. Er wordt aan meerdere milestones tegelijkertijd gewerkt, zo wordt er onder andere al gekeken naar passende frameworks terwijl de requirements nog worden opgesteld. Ook wordt gedurende het project informatie die interessant is en wellicht bruikbaar is voor de scriptie gearchiveerd zodat deze gemakkelijk te vinden is als de scriptie geschreven wordt.

*D = definitiefase O = ontwerpfase en R = realisatiefase*

|  |  |  |  |
| --- | --- | --- | --- |
| Nr. | Fase | Omschrijving | Opleverdatum |
| 1 | D | Vast stellen wat de wensen van de klant zijn | 13-03-2012 |
| 2 | D | Definitieve requirements vast stellen. | 20-03-2012 |
| 3 | D | Opstellen van high level architectuur | 27-03-2012 |
| 4 | O | Een passend framework uitzoeken (zowel open-source als commercieel), met gedocumenteerde onderbouwing. | 01-04-2012 |
| 5 | O | Onderzoeken hoe het framework gebruikt dient te worden. (best practises) | 10-04-2012 |
| 6 | O | POC applicatie ontwerpen. | 16-04-2012 |
| 7 | R | JIRA in gebruik nemen. | 18-04-2012 |
| 8 | R | Een applicatie bouwen die gebruik maakt van het framework. | 21-05-2012 |
| 9 | O/R | Evalueren of het gekozen framework geschikt is of dat er stappen herhaald moeten worden. (*deze taak wordt zo snel mogelijk uitgevoerd*) | 30-05-2012 |
| 10 | R | Stub-applicatie uitbreiden tot een demonstreerbare toepassing. | 03-07-2012 |
| 11 | R | Documenteren, archiveren en presenteren. | 13-08-2012 |

## Werkwijze

In deze paragraaf wordt verteld welke werkwijze wordt gebruikt voor het behalen van een milestone. Ook zal er worden toegelicht welke deelvragen een rol spelen tijdens het werken aan een milestone.

De deelvragen:

1. Aan welke functionaliteiteisen en veiligheidseisen moet een POC voldoen?
2. Welke technieken en protocollen zijn er op het gebied van SSO?
3. Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?
4. Welk framework is het meest geschikt als basis voor het realiseren van een POC?
5. Hoe moet het POC uiteindelijk worden gerealiseerd?

Werkwijze voor het behalen van een milestone:

1. Scenario’s opstellen en in overleg met mentor en klant vast stellen welke scenario’s ondersteund moeten worden en welke niet. (deelvraag 1)
2. De requirements worden samengesteld op basis van overleg met mentor en klant. (deelvraag 1)
3. Opstellen high level architectuur. (deelvraag 1 & 2)
4. Een groot aantal frameworks onderzoeken en vervolgens een shortlist maken. (deelvraag 2 en 3)
5. Kijken of er applicaties te vinden zijn die gebruik maken van het framework en ervaringen van andere gebruikers doorlezen. (deelvraag 4)
6. Opstellen detail architectuur. (deelvraag 5)
7. Vaststellen en inplannen ontwikkelactiviteiten met behulp van Jira. (deelvraag 5)
8. Starten met applicatie ontwikkeling op basis van planning en architectuur.
9. Zo snel mogelijk beslissen of het gekozen framework geschikt is voor het realiseren van een demonstreerbare toepassing. (deelvraag 4 & 5)
10. De stub applicatie als basis gebruiken van de demonstreerbare toepassing en verder uitbreiden. Denk hierbij aan het uitbreiden van de functionaliteit en het verbeteren van de uitstraling. (deelvraag 5)
11. Het opstellen van de nodige documenten zoals handleidingen en de scriptie, en deze documenten archiveren in de wiki van E-ID.

# PID-Bijlage A: Communicatieplan

**Revisies**

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Status | Datum | Wijzigingen |
| 0.1 | concept |  | Eerste invulling |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Inleiding**

Dit communicatieplan benoemt alle partijen die een (positief of negatief) belang hebben bij het project en de wijze waarop zij bij het project zullen worden betrokken en welke communicatievormen daarbij gebruikt worden. Het gaat hierbij om partijen en communicatie buiten de formele projectmanagementstructuur zoals beschreven in het PID.

**Belanghebbenden bij het project**

|  |  |  |  |
| --- | --- | --- | --- |
| Wie | Namens | Belang | **Communicatievorm(\*)** |
| persoon | groep, afdeling | Welk belang bij project? | Hoe wordt deze persoon betrokken bij het project? |
| Frank Haverkort | Fontys | Stage begeleider | informeren, accepteren |
| Edvard Scheffers | E-ID | Stage mentor | overleggen, adviseren, informeren, besluiten, accepteren, aansturing |
| Hugo van der Zandt | Fontys en E-ID | Stagiair | overleggen, adviseren, informeren, besluiten, accepteren, uitvoeren |
|  |  |  |  |

**Communicatiekanalen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Van | Naar | Informatie | Medium | Frequentie of data |
| Persoon of groep | Persoon of groep | Soort informatie | Email, telefoon, rapport |  |
| Frank Haverkort | Hugo van der Zandt | Informatie rondom stage | Email, Telefoon | Als er veranderingen vanuit Fontys te melden zijn. |
| Edvard Scheffers | Hugo van der Zandt | Informatie rondom de Opdracht | Besprekingen, Email | Zoveel als er gevraagd wordt. |

# PID-Bijlage B: Scenario’s

In de onderstaande tabel staan scenario’s omschreven die voor kunnen komen in de praktijk. Er wordt in deze scenario’s van uit gegaan dat een account maar bij een domein hoort.

Bij ieder scenario is er sprake van navigatie van een domein (domein A) naar een ander domein (domeinB).

|  |  |
| --- | --- |
| Scenario Nummer | Omschrijving van het scenario |
| Scenario 1 | Een gebruiker is ingelogd op domein A en navigeert naar domein B waar hij nog geen account heeft. |
| Scenario 2 | Een gebruiker is ingelogd op domein A en navigeert naar domein B waar hij ook een account heeft (zonder dat deze al gekoppeld zijn) |
| Scenario 3 | Een gebruiker is ingelogd op domein A en navigeert naar domein B, waar hij automatisch ingelogd is met behulp van de SSO koppeling. |
| Scenario 4 | Een gebruiker maakt een nieuwe account aan op domein A of B |
| Scenario 5 | Een gebruiker logt uit op domein A of B |

1. Onderzoeksdocument

|  |
| --- |
| E-ID |
| Tienstappenplan |
| Onderzoek naar SSO navigatie |

|  |
| --- |
| Hugo van der zandt  05-03-2012 |

# Voorwoord

Dit Verslag gaat over het onderzoek voor een afstudeer stage die uitgevoerd is in de periode 05-03-2012 t/m 13-08-2012 door Hugo van der Zandt. De Volgende hoofdstukken zullen allemaal betrekking hebben tot voorbereidingen die getroffen zijn voor de afstudeerstage bij E-ID. E-ID is het bedrijf waar de afstudeer stage gelopen gaat worden.

In dit verslag zal telkens een onderdeel van het tien stappenplan van Kempen en de Keizer uitgewerkt worden aan de hand van een eigen casus. De casus zal zijn: “SSO navigatie”.

In elk hoofdstuk zal ingegaan worden op het verloop, of het eventuele toekomstige verloop. Verder zullen hierbij ook de competenties beschreven worden die van toepassing zijn op het desbetreffende hoofdstuk.

Inhoudsopgave

[Externe oriëntatie 30](#_Toc320195635)

[Intakegesprek 31](#_Toc320195636)

[Oriënterende activiteiten 32](#_Toc320195637)

[Analyse & Terugkoppeling/contractering 33](#_Toc320195638)

[Werkplanning en project organisatie 34](#_Toc320195639)

[Diepteonderzoek 35](#_Toc320195640)

[Oplossingsplan 48](#_Toc320195641)

[Invoering 48](#_Toc320195642)

# Begrippenlijst

SSO – single sign on: eenmalig inloggen voor het gebruik van meerdere diensten.  
SLO – single logout: voor meerdere applicaties in een keer uitloggen.  
framework: een softwarepakket dat als basis dient voor het realiseren van andere software.  
federated identity: het linken van een elektronische identiteit van een persoon over meerdere systemen.  
federated SSO: single sign on mogelijkheden met een Federated Identity.  
PID – project initiation document: een plan van aanpak.  
CoT – Circle of Trust: Een keten van domeinen die authenticatie en autorisatie van gebruikers van elkaar honoreren   
MDD - model driven design: een techniek waarbij, vanuit een model, code kan worden gegenereerd.

# Externe oriëntatie

Er is onderzocht wat voor een bedrijf E-ID is door de site grondig door te nemen en door met een medewerker in gesprek te gaan.

Hieruit kwam onder andere naar voren dat:

* Het bedrijf één vestiging heeft in Vianen
* Dat er +-50 mensen werken maar dat het ambieert om binnen 5jaar uit te breiden naar 150 medewerkers (60% senior programmeurs, 20% medior programmeurs en 20%junior programmeurs)
* Ze maken software op maat voor bedrijven uit de top500 en ze geven levenslange garantie op de software die ze maken.
* Zeer goede arbeidsvoorwaarden voor stagiaires (zelfde als de vaste medewerkers)
* Alle gewenste begeleiding tijdens de gehele stage duur

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Samenwerken
* Analyseren

# Intakegesprek

Bij de intake gesprekken is optimaal gebruik gemaakt van de kennis die is opgedaan bij het solliciteren bij Brunel IT.

Het intakegesprekken werd afgenomen door een HR manager en senior programmeur, het gesprek duurden ongeveer 90minuten. Wat heel duidelijk in het gesprek naar voren kwam, was dat er veel begeleiding aangeboden wordt door E-ID. Voor de afstudeer stage wordt één stagebegeleider toegewezen en één klant. De klant zal gewoon een medewerker zijn. Verder wordt alle kennis die beschikbaar is binnen het bedrijf beschikbaar gesteld. Zo wordt er toegang verleend tot de wiki van E-ID en mag er bij iedere collega worden aangeklopt voor hulp als dat nodig is.

Toebehoren competenties:

* Professioneel handelen
* Analyseren

# Oriënterende activiteiten

De oriënterende activiteiten bestonden uit activiteiten zoals: een tweede gesprek, Email contact en telefonisch contact.

In het tweede gesprek zijn een paar algemene dingen besproken en de afstudeeropdracht. Echter is deze opdracht door fontys hogescholen 6dagen voor de aanvang van de afstudeerstage afgewezen. Met als gevolg dat er in zeer korte tijd een vervangende opdacht moest worden opgesteld. Deze vervangende opdracht is als volgt geformuleerd:

|  |
| --- |
| Het betreft een SSO oplossing voor meerdere applicaties met een eigen user/authenticatie base. Voorbeeld: Het komt er op neer dat als er vanuit applicatie A wordt genavigeerd naar applicatie B dat inloggen automatisch moet gebeuren. Aangezien de applicaties niet op hetzelfde domein draaien en allebei een eigen ldap (oid) gebruiken is er geen gemeenschappelijke factor. Hierdoor vallen zo'n beetje alle standaard SSO oplossingen af.  Oplossingsrichting: De oplossingsrichting zou te vinden moeten zijn in een soort van keychain. In zo'n keychain leg je vast welke account van applicatie A gekoppeld is aan applicatie B. Hierbij is het natuurlijk ook belangrijk hoe je daadwerkelijk zo'n koppeling legt. In dit verhaal kunnen aspecten als OpenID en OAuth worden bekeken.  Er zijn enkele Opensource frameworks die een gedeelte van deze functionaliteit bieden echter een kant en klare oplossing is er niet.  De oplossing moet draaien in de Cloud zodat externe systemen er gemakkelijk aan kunnen koppelen. Dit moet natuurlijk met hoge beveiligingseisen gebeuren. |

De algemene informatie die naar voren kwam is uit deze oriënterende fase is:

* E-ID stelt hardware beschikbaar voor stagiairs (laptops en dergelijke).
* E-ID zorgt voor een lease auto omdat het kantoor moeilijk te bereiken is met het openbaar vervoer.

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Samenwerken
* Analyseren
* Adviseren
* ontwerpen

# Analyse & Terugkoppeling/contractering

Dit hoofdstuk gaat in op de afspraken die gemaakt zijn rondom de afstudeeropdracht en hoe deze uitgevoerd zal worden.

**De afspraken:**

Er is een afstudeerovereenkomst opgesteld door E-ID met algemene voorwarden, deze overeenkomst is getekend door E-ID, Fontys hogescholen en de afstudeerder.

**De uitvoering:**

Bij de uitvoering van de afstudeeropdracht zijn de volgende mensen betrokken:

Edvard Scheffers: stage begeleider

Frank Haverkort: docent begeleider

Erik Post: interne klant

Hugo van der Zandt: afstudeerder

Voor de afstudeeropdracht zal gebruik worden gemaakt van de faseringsmethode van Grit, deze methode houdt de volgende fases aan:

1. Definitie fase, wat moet er gebeuren
2. Ontwerp fase, hoe moet het gebeuren
3. Realisatie fase, uitvoering

Er is ook een globale planning gemaakt, dit lijkt er veel op een waterval methodiek. Echter zal tijdens het project ook gebruik worden gemaakt van een agile ontwikkelstrategie. In een agile strategie wordt er “as we go” continu gekeken wat de beste manier is om het project voort te zetten. Op deze wijze kan er gemakkelijk worden omgegaan met eventuele wijzigingen die naar voren komen uit onderzoeken of problemen. Deze agile strategie wordt niet in alle fases van het project toegepast.

Voor het project is een PID opgesteld waarin de methodiek en aanpak die in het project gebruikt worden in detail omschreven zijn. Voor dat het PID werd goedgekeurd is het door alle begeleiders gecontroleerd en waar nodig aangepast.

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Samenwerken
* Analyseren
* Realiseren
* Ontwerpen
* Beheren

# Werkplanning en project organisatie

Dit hoofdstuk gaat over hoe het project georganiseerd is en hoe het project uitgevoerd gaat worden.

Voor het project stelt E-ID een stagebegeleider (Edvard Sheffers) ter beschikking en een interne klant (Erik Post). Verder zal Fontys Hogescholen een docent begeleider (Frank Haverkort) ter beschikking stellen. Voor de uitvoering van de stageopdracht is er een PID worden opgesteld met daarin een gedetailleerdere planning en werkwijze, een korte samenvatting van deze planning en werkwijze is:

Voor de uitvoering van de stageopdracht zal de faseringsmethode van Grit worden gebruikt. Deze methode is binnen dit project als volgt ingedeeld:

Voor het project is een globale planning gemaakt, dit lijkt er veel op een waterval methodiek. Echter zal tijdens het project ook gebruik worden gemaakt van een Agile ontwikkelstrategie. Dit betekend dat er “as we go” gekeken wordt wat de beste manier is om het project voort te zetten. Op deze wijze kan er gemakkelijk worden omgegaan met eventuele wijzigingen die naar voren komen uit onderzoeken of problemen. Deze agile strategie wordt niet in alle fases van het project toegepast (de agile strategie zal met namen gebruikt worden in de realisatie fase).

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Analyseren
* samenwerken

# Diepteonderzoek

In het diepteonderzoek wordt de volgende onderzoeksvraag centraal gesteld:

*Hoe kunnen identiteitsmanagement systemen van verschillende domeinen op een veilige manier worden samengebracht tot een SSO systeem voor een gebruiker?*

Om deze onderzoeksvraag te kunnen beantwoorden is deze opgedeeld in een aantal deelvragen:

1. Aan welke functionaliteiteisen en veiligheidseisen moet een proof of concept (POC) voldoen?
2. Welke technieken en protocollen zijn er op het gebied van SSO?
3. Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?
4. Welk framework is het meest geschikt als basis voor het realiseren van een POC?
5. Hoe moet het POC uiteindelijk worden gerealiseerd?

De deelvragen zullen in dit hoofdstuk beantwoordt worden in een eigen paragraaf.

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Realiseren
* Samenwerken
* Analyseren
* Ontwerpen
* Adviseren

## Deelvraag 1 eisen

Aan welke functionaliteiteisen en veiligheidseisen moet een proof of concept (POC) voldoen?

Om deze vraag te kunnen beantwoorden is er vastgesteld wat de wensen zijn van de interne klant van E-ID.

Om dit duidelijk te krijgen is er een requirements document opgesteld. Dit is als volgt gedaan: Eerst is er gebrainstormd over alle functionele en non functionele eisen. Vervolgens zijn deze eisen doorgesproken met het project team en is iedere eis beoordeeld met behulp van de MoSCoW methodiek. De MoSCoW methodiek werkt als volgt, een eis krijgt een M, S, C of W als waarde.

* De waarde M staat voor **M**UST have.
* De waarde S staat voor **S**HOULD have.
* De waarde C staat voor **C**OULD have.
* De waarde W staat voor **W**ONT have.

## Deelvraag 2 technieken

Welke technieken en protocollen zijn er op het gebied van SSO?

Om deze vraag te kunnen beantwoorden is er onderzoek gedaan naar populaire technieken, protocollen die mogelijk een rol kunnen spelen het gebied van SSO. Voor dit onderzoek is voornamelijk gebruik gemaakt van het internet.

Zijn uiteindelijk 5 technieken en protocollen geïdentificeerd als relevant op het gebied van SSO:

* SAML
* WS-Federation
* OpenID
* OAuth
* XRI

Van ieder is een korte samenvatting gemaakt die de essenties van de techniek/ het protocol in kaart brengt.

In een notendop:

**SAML**: Is een XML standaard die beschrijft hoe tokens, requests en responses er uit moeten zien en hoe er mee omgegaan moet worden.

**WS-Federation**: Is een flexibele federated identity architectuur, die token onafhankelijk is maar wel en manier definieert voor het verkrijgen van token.

**OpenID**: Is een techniek waarmee een gebruiker zich door een andere partij laat identificeren (bijv. google), de andere partij (bijv. google) geeft daarbij de gegevens van de gebruiker aan een andere organisatie.

**OAuth**: Is een techniek waarmee een gebruiker een partij toegang kan verlenen tot zijn gegevens die zich ergens anders bevinden (bijv. bij google).

**XRI**: Is een uitbreiding op de URI syntax, waarmee naar documenten (XRDS) verwezen wordt die extra informatie bevatten over bijv. een gebruiker.

Voor een uitgebreidere omschrijving zie bijlage E

## Deelvraag 3frameworks

Welke frameworks zijn er die mogelijk deze technieken en protocollen ondersteunen en de benodigde functionaliteit bieden?

Om een antwoord te kunnen geven op deze vraag is er uitgebreid onderzoek gedaan naar SSO frameworks. Hier bij is voornamelijk gebruik gemaakt van het internet. Dit onderzoek had als resultaat een lijst van 13 mogelijke frameworks, deze waren zowel open source als commercieel. Van ieder framework is onderzocht hoe lang het al bestaat, hoe het onderhouden wordt, welke standaarden ondersteund worden e.d.

\*note: Als meteen duidelijk was dat een framework niet geschikt was voor het maken van een POC werd er ook geen verder onderzoek naar gedaan.

Voor de volledige lijst zie bijlage F.

## Deelvraag 4 framework keuzen

Welk framework is het meest geschikt als basis voor het realiseren van een POC?

Om deze vraag te kunnen beantwoorden is van de lijst van 13 frameworks een shortlist gemaakt, met daarin de 4 meest belovende frameworks: Shibboleth, JOSSO, openAM en PingFederate. Er is voor deze vier gekozen omdat deze 4 allemaal erg volwassen zijn en allemaal van elkaar verschillen. Een kort overzicht van de achtergrond van de frameworks:

* **PingFederate** is een commercieel product (gebaseerd op een open source project, genaamd SouceID).
* **OpenAM** is een oud project van Oracle, dat toen het nog door Oracle ontwikkeld werd de naam OpenSSO droeg.
* **Shibboleth** is een groot open source project wat ook andere open source projecten gebruikt en overneemt (zoals OpenSAML).
* **JOSSO** is net zoals Shibboleth een populaire open source oplossingen voor federated SSO.

Naar deze vier frameworks is meer onderzoek gedaan, zoals het globaal uitzoeken van hoe het framework technisch gezien werkt, hoeveel tijd het kost om informatie te krijgen over het framework en wat de ervaringen zijn van mensen die met het framework hebben gewerkt.

Uit de shortlist kwamen twee frameworks naar voren die zeer geschikt leken voor het realiseren van een POC, dit waren SHibboleth en JOSSO. Deze frameworks zijn uitgebreid met elkaar vergleken. Met het uiteindelijke resultaat dat het JOSSO framework is gekozen als basis van het POC.

De hoofdreden voor het kiezen van JOSSO was dat de architectuur van JOSSO beter paste dan die van Shibboleth. Shibboleth ging namelijk meer uit van een scenario waar één domein gebruikers bij houdt en een ander domein een service aanbied. Er was dus geen sprake van een “circle of trust”(er is sprake van een “cirle of trust” als alle deelnemende partijen van een federation elkaar vertrouwen). JOSSO is echter ontworpen voor het realiseren voor een circle of trust en daardoor dus geschikter. Wat een bijrol speelde in het kiezen voor JOSSO is dat de gemeenschap van JOSSO toegankelijker is voor nieuwkomers. Dit is belangrijk omdat er zich mogelijk problemen gaan voordoen bij het realiseren van een POC waarbij een klein beetje input van een ervaringsdeskundige erg veel kan helpen.

## Deelvraag 5 POC

*Hoe moet het POC uiteindelijk worden gerealiseerd?*

Het realiseren van het POC wordt gedaan met behulp van JOSSO2. De “best practise” voor het gebruik van JOSSO is om gebruik te maken van de console die inbegrepen is in de installatie van JOSSO. Met de console kan er met behulp van MDD een CoT worden opgezet.

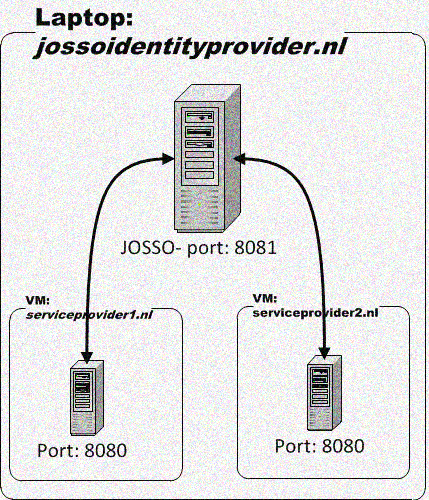
**Planning & uitvoering:**

Als eerste moet er aangetoond worden dat het met JOSSO mogelijk is om tussen domein een SSO koppeling te maken. Vervolgens moeten specifiekere functionaliteiten worden getest in vervolg projecten.

Vaststellen realisatie methode

Het doel van het eerste POC project was aantonen dat het mogelijk is om met behulp van het JOSSO framework een Cross-Domain-SSO oplossing te realiseren die gebruik maakt van een Federated Identity. Bij deze oplossing zouden alle componenten (SPs en een IDP) op verschillende machines draaien zodat ze ieder een eigen IP adres zouden hebben.

### ****Aanpak****

De testomgeving voor deze oplossing maakte gebruik van VM’s. De omgeving zou er als volgt uitzien: Een laptop met daarop JOSSO en VMware geïnstalleerd. In het hosts bestand van de laptop wordt het IP adres van de laptop gekoppeld aan “*jossoidentityprovider.nl*” en het IP adres van twee VMs aan “*serviceprovider1.nl*” en “*serviceprovider2.nl*”. Op beide VMs/domeinen zou een tomcat server draaien. Deze zouden te bereiken zijn op port 8080 en applicaties hosten die, met behulp van JOSSO, Cross-Domain-SSO en Federated Identity ondersteuning zouden krijgen.

### ****resultaat****

De omgeving bleek te complex om mee te kunnen werken. Voor alle toekomstige oplossingen wordt het gebruiken van VMs vermeden, in plaats daar van wordt het hosts bestand gebruikt. Het hosts bestand is als volgt gebruikt: voor iedere oplossing zijn er andere domeinnamen gebruikt, een voorbeeld van de extra regels die hiervoor aan het hosts bestand zijn toegevoegd:

|  |
| --- |
| 127.0.0.1 serviceprovider1.nl  127.0.0.1 serviceprovider2.nl  127.0.0.1 jossyidentityprovider.nl |

Hoewel de applicaties op hetzelfde IP adres te bereikbaar zijn zal een browser nog steeds niet toestaan dat deze cookies met elkaar delen omdat de domeinnaam niet overeen komt. Hierdoor kan er, zonder shortcuts, Cross-Domain-SSO worden aangetoond.

## De basis functionaliteit testen

Het doel van de deze oplossing, is het aantonen dat er met behulp van het JOSSO framework een Cross-Domain-SSO oplossing gerealiseerd kan worden. Hierbij wordt geen Federated Identity gebruikt maar van één overkoepelende digitale identiteit.

### ****C:\Users\blur5\Dropbox\private\werk documenten\bronnen\plaatjes\josso\eerte stap.pngAanpak****

Figuur Test omgeving

Voor de het realiseren van de oplossing is een test omgeving op een laptop ingericht, deze ziet er als volgt uit: een tomcat server, een JBoss server en een JOSSO server, de test omgeving met adressen en port nummers is te zien in Figuur 9.

Als het JOSSO framework wordt geïnstalleerd worden er standaard demo-applicaties mee geïnstalleerd die via de JOSSO console op een applicatie server gezet kunnen worden. Deze demo-applicaties worden “partnerapp” genoemd. Op de Tomcat en Jboss servers is een partnerapp geplaatst. Vervolgens zijn met behulp van de JOSSO console alle componenten aan elkaar gekoppeld.

### ****resultaat****

Met behulp van deze oplossing is aangetoond dat het JOSSO framework Cross-Domain-SSO ondersteund.

Custom application

Met deze oplossing moet worden aangetoond hoeveel moeite het kost om een standaard java web-applicatie deel uit te laten maken van een COT met behulp van het JOSSO framework.

### ****Aanpak****

Deze oplossing maakt gebruik van de test omgeving van de voorafgaande oplossing. Voor deze oplossing is de partnerapp op de tomcat server vervangen door een van de grond af gebouwde hello-world web-applicatie. Waarvan de hello-world pagina is gedefinieerd als beveiligde content waar alleen gebruikers met de rol “*role1*” toegang tot hebben.

Eerst is aangetoond dat de hello-world applicatie goed functioneert door gebruik te maken van BASIC authenticatie, omdat dit snel kan worden gerealiseerd en ondersteund wordt door alle browsers [17]. Vervolgens is de hello world applicatie gejossifiëd.

Voor het jossifiën van de Hello-word applicatie moet de BASIC authenticatie worden vervangen door FORM authenticatie. Dit wordt gedaan door het aanpassen van de web.xml van de hello-world applicatie. Ook is er ook in de web.xml file geconfigureerd welke pagina wordt gebruikt voor de FORM authenticatie. Voor deze oplossing is er naar een .jsp pagina gelinkt die de gebruiker redirect naar de IDP. Deze .jsp pagina bevat de volgende code:

|  |
| --- |
| **<%**@page contentType**=**"text/html; charset=iso-8859-1" language**=**"java" session**=**"true" **%>**  **<%** response**.**sendRedirect**(**request**.**getContextPath**()** **+** "/josso\_login/"**);** **%>** |

Door aan de redirect “/josso\_login/” toe te voegen wordt de redirect onderschept door de agent op de tomcat server, zoals eerder beschreven in hoofdstuk 4.2. Waarna de agent er voor zorgt dat de gebruiker naar de loginpagina van de IDP wordt geridirect.

### ****resultaat****

Met behulp van dit POC project is vast gesteld dat het jossifiën van een applicaite, binnen een korte tijd, kan worden gedaan zonder tegen grote problemen aan te lopen.

Door de hello-world pagina alleen maar toegankelijk te maken voor gebruikers met de rol “role1”, is aangetoond dat JOSSO role based access control (RBAC) ondersteund. Dit is nodig omdat RBAC de basis is voor sommige Must-have requirements.

## Behoud van de login pagina

Het doel van de deze POC oplossing is uitzoeken hoe de originele login pagina van een website behouden kan worden (en onder het zelfde adres beschikbaar kan blijven).

**Aanpak**

Voor de deze POC oplossing is de Hello-world applicatie van de voorafgaande POC oplossing als basis gebruikt. Vanuit berichten op de forums van de JOSSO gemeenschap bleek dat het eenvoudiger is om de gewenste functionaliteit te bereiken met het gebruik van JOSSO1. Dus er is van deze gelegenheid gebruik gemaakt om te experimenteren met JOSSO1, om zo een beter beeld te krijgen van alle mogelijkheden die er zijn, met alle bijbehorende voor en nadelen.

Er is op twee manieren geprobeerd deze POC oplossing te realiseren, dit was nodig doordat de eerste oplossing ongewenste bijwerkingen had.

#### ****Oplossing 1, aanpassen van de IDP****

Door het gebruiken van JOSSO1 waren er meer configuratie bestanden beschikbaar op de applicatie server die als IDP werd gebruikt. Wat tot gevolg had dat er meer configuratie opties beschikbaar werden. Een van de bestanden die beschikbaar werd was de “*josso-gateway-web.xml*”. In dit bestand kan een “*customLoginURL*” worden gespecificeerd.

Dit zorgt er voor dat alle login verzoeken doorgestuurd worden naar een andere login pagina, dus niet de standaard login pagina van de IDP. Het bijeffect wat dit had was dat niet alleen de Hello-word applicatie van dit speciaal gedefinieerde login scherm gebruik maakte. De andere applicaties die op de IDP voor authenticatie vertrouwden maakte nu ook allemaal gebruik van de login pagina van de Hello-world applicatie. Dit was niet de bedoeling, het is de bedoeling dat iedere applicatie een eigen login scherm kan hebben.

#### ****Oplossing 2, aanpassen van de applicatie****

Voor de tweede oplossing is de configuratie van de IDP is hersteld naar zijn originele staat en is niet meer aangepast, omdat deze methode erg gevoelig blijkt voor onbedoelde bijwerkingen.

Voor de twee oplossing is de redirect pagina naar de IDP vervangen in de applicatie door een pagina met een http from post submit. Dit is gedaan door in de web.xml de web-pagina, die gebruikt wordt voor de FORM authenticatie, te vervangen door een nieuwe login-pagina waar een gebruiker een username en wachtwoord kan invullen. Met de “login” knop wordt het verzoek voor authenticatie naar de IDP verstuurd zonder dat de gebruiker iets ziet van de communicatie met de IDP. De code van de nieuwe login-pagina is te zien in bijlage G.

Voor de volledigheid, is er ook onderzocht wat de mogelijkheden zijn betreffende het configureren van de pagina waar de gebruiker naartoe wordt geredirect nadat hij of zij een inlog poging heeft gedaan. De pagina waar een gebruiker naartoe wordt geredirect kan worden ingesteld met behulp van hidden input velden. Door deze met een specifieke naam aan het form toe te voegen, is de IDP in staat de gebruiker naar een pagina naar wens te redirecten.

### ****resultaat****

Met behulp van dit POC project is aangetoond dat het mogelijk is de originele inlog pagina van een website te behouden.

## Werken met een Federated Identity

Het doel van de dezePOC oplossing is bewijzen dat er met behulp van het JOSSO framework een Federated Identity kan worden ondersteund.

### ****Aanpak****

Voor deze oplossing is de test omgeving aangepast ten opzichte van de test omgevingen die gebruikt zijn in de voorgaande oplossingen. De test omgeving van deze oplossing draait op een laptop en bevat de volgende componenten:

* Twee applicaties
* Twee tomcat servers
* Een josso2 server met een Identity Vault
* Twee mysql-databases

De omgeving is te zien in **Fout! Verwijzingsbron niet gevonden.**.

Applicaties

Er zijn applicaties gemaakt voor deze oplossing, dit zijn simpele stub web-applicaties. Beide websites bezitten publieke pagina’s en afgeschermde pagina’s. De afgeschermde pagina’s kunnen alleen maar bezocht worden door geauthentiseerde gebruikers met de juiste rol.

Applicatie servers

Er is bewust voor gekozen om de applicaties te hosten op twee tomcat servers in plaats van op een tomcat server en een Jboss server.

Het opzetten van twee naast elkaar draaiende tomcat servers kost tijd. De “nieuwe” tomcat server moet namelijk zo geconfigureerd worden dat deze niet afhankelijk is van de eerste en de eerste tomcat server ook niet verstoord. De tijd die met configureren verloren gaat wordt echter snel terug verdiend doordat:

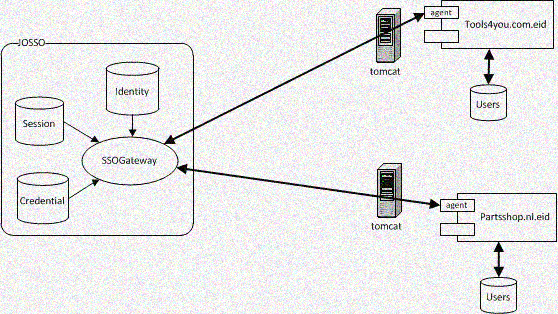
* een tomcat server makkelijker te configureren is als een Jboss server.
* sneller kan herstarten als een Jboss server.

Het sneller herstarten is een opmerkbaar voordeel, omdat herstarten veel wordt gedaan bij het realiseren van een oplossing, om er zeker van te zijn dat eerdere tests geen invloed hebben op de test die wordt uitgevoerd.

DatabasesDe IDP van josso2 die gebruikt wordt in deze oplossing maakt geen gebruik van deze mysql-databases voor het authentiseren van de gebruikers. Voor het authentiseren van de gebruikers maakt de IDP gebruik van een Identity vault.

Iedere webapplicatie heeft een eigen mysql-database die waarin gebruikers gegevens worden bijgehouden (een totaal overzicht is te zien in Figuur 10). Er is gekozen voor het gebruiken van mysql-databases omdat deze eenvoudig op te zetten zijn en omdat mysql-databases niet standaard worden ondersteund door het JOSSO framework. Zo kon er getest hoeveel moeite het kost om componenten te gebruiken waar geen standaard ondersteuning voor is. Dit bleek eenvoudig, wat er moest gebeuren om ondersteuning te realiseren voor mysql-databases was: het toevoegen van de juiste drivers aan de libraries van het JOSSO framewrok.

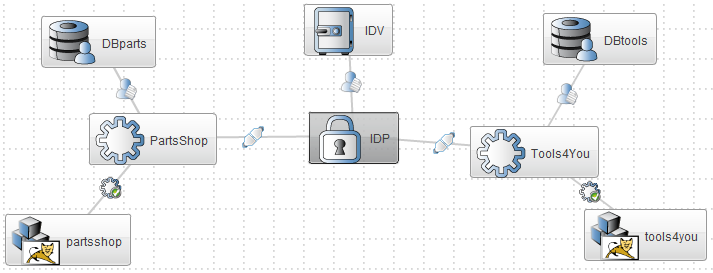
Domeinnamen

Zoals te zien is in **Fout! Verwijzingsbron niet gevonden.** zijn de domein namen waarop de applicaties benaderd worden ook aangepast ten opzichte van eerdere oplossingen. Dit is gedaan omdat het aanpassen van het hosts bestand een kleine moeite is voor het project levendiger maken van het project.

Figuur test omgeving

Federated Identity

In **Fout! Verwijzingsbron niet gevonden.** is goed te zien hoe de Federated Identity is gerealiseerd. De SPs maken gebruik van de mysql-databases voor het opzoeken van een lokale identiteit van een gebruiker. Dit betekent dus dat de SP de digitale identiteit van de gebruiker beheerd. De applicaties zelf hebben hier geen zorg aan.



Figuur 11 model

De koppeling

Bij de configuratie opties van een SP is onder andere een account link policy inbegrepen. Hiermee wordt aangegeven op wat voor een manier de accounts van de IDP aan die van de SP zijn gekoppeld.

Het koppelen van accounts is gedaan door een account te koppelen op basis van de loginnaam. Dit was voldoende om te demonstreren dat een Federated Identity gerealiseerd kan worden met behulp van het JOSSO framework.

#### ****Problemen****

Tijdens dit project hebben zich een probleem voor gedaan met het tot stand brengen van een koppeling en er is een observatie gedaan van onwenselijk cashe gebruik.

*Koppeling op basis van Email adres*

Het koppelen op basis van loginnaam is in de praktijk niet wenselijk omdat in de praktijk vaak gebruik wordt gemaakt van verschillende loginnamen. Daarom is er getracht een link te leggen op basis van Email adressen. Het kiezen van een Email adres had twee redenen:

* Een Email adres behoort maar aan een persoon toe.
* Als er een koppeling gemaakt kan worden met een email adres, betekend dit dat er ook een willekeurig ander attribuut van de gebruik gebruikt worden voor het tot stand brengen van een koppeling (zolang als het attribuut uniek is voor iedere gebruiker).

Het is echter niet gelukt om een koppeling op basis van Email adres te realiseren. Het koppelen van accounts op basis van Email adres is wel een configuratie optie voor de “account linkage policy” van een SP, dit suggereert dat er wel ondersteuning voor is of dat deze wordt verwacht. Omdat de handleiding van JOSSO niet erg diep ingaat op de technische werking van het framework is het niet duidelijk hoe josso het account linken op basis van email ondersteunend.

Mogelijke oorzaken van het probleem zouden kunnen zijn dat de gebruikte query’s niet de return waarde geven die het JOSSO framework verwacht. Andere factoren die van invloed zouden kunnen zijn, zijn waar mogelijk weggehaald. Zo is bijvoorbeeld het databasemodel aangepast zodat deze overeen komt met de demo-database van JOSSO. Er is ook veelvuldig geprobeerd hulp te zoeken bij de gemeenschap van JOSSO maar deze gaf geen gehoor.

*Observatie van cache gebruik*

Tijdens het testen van de oplossing viel het op dat het cache gebruik van de browser onverwacht gedrag veroorzaakt. Namelijk dat als een gebruiker uitlogt en opnieuw inlogt de beveiligede pagina uit de cache wordt opgehaald, met als gevolg dat de gebruiker niet de pagina te zien krijgt die hij verwacht te zien als hij of zij is ingelogd. Hiervoor is een hard refresh nodig zodat de pagina opnieuw wordt opgehaald bij de server. Dit probleem kan worden verholpen door de pagina zelf aan te passen en code toe te voegen die er voor zorgt dat de cache niet wordt gebruikt.

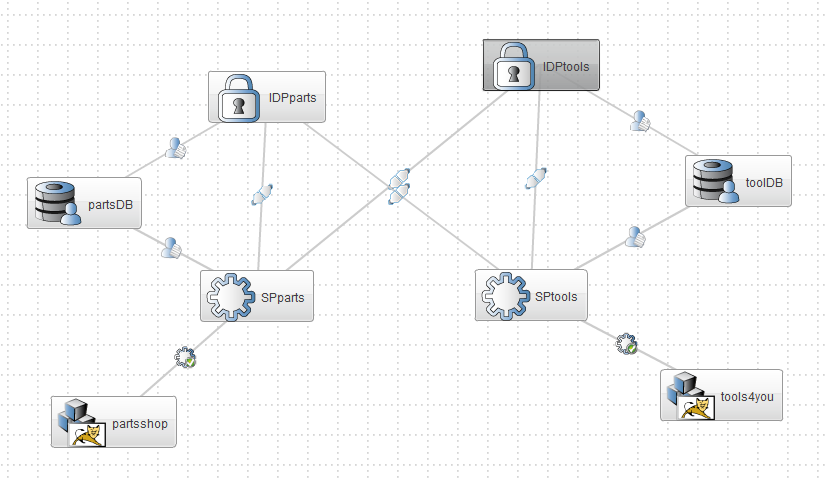
### ****resultaat****

Met behulp van dit POC project is aangetoond dat het mogelijk is een Federated Identity te realiseren met behulp van het JOSSO framework.

## Behoud van oude account

Het doel van de dezePOC oplossing is: aantonen dat het mogelijk is om een Federated Identity te introduceren aan een website, waarbij de originele inlog gegevens van gebruikers behouden blijven. In andere woorden: Dat iedere website zijn eigen userdatabase heeft, die ook wordt gebruikt voor de authenticatie van gebruikers en niet alleen voor het verrijken van de digitale identiteit van een al ingelogde gebruiker die van een ander domein binnen de COT afkomt.

### ****Aanpak****

In **Fout! Verwijzingsbron niet gevonden.** is het model te zien waarmee geprobeerd is account behoud te realiseren. Zoals te zien is, is er een tweede IDP aan de test omgeving toegevoegd. Dit is gedaan zodat iedere SP een eigen IDP heeft waar prioriteit aan wordt toegekend, deze IDP wordt dan voor de SP “preferred -IDP” genoemd. Dit betekend dat als een gebruiker gebruik wil maken van een service van de SP als hij niet is ingelogd, dat hij naar de bijbehorende IDP wordt geridirect voor authenticatie (dit is dus niet een willekeurige IDP).

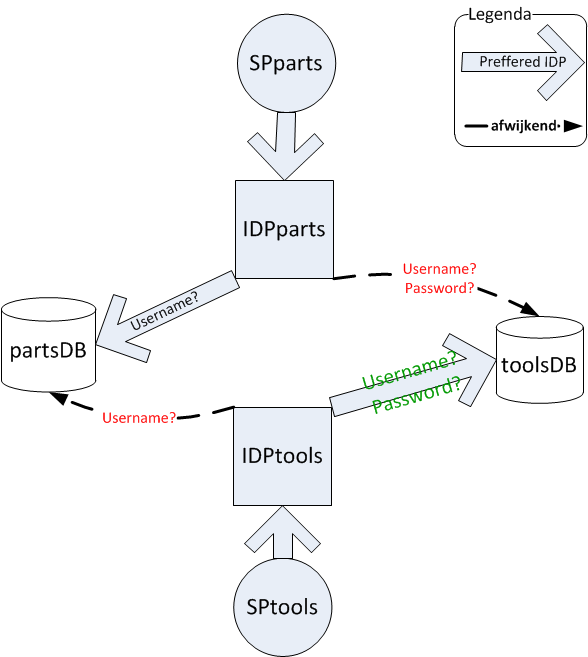
Figuur model

De SP zal wel van beide IDP’s ingelogde gebruikers toegang geven tot beschermde services en content. De SP zal ook proberen de digitale identiteit van een gebruiker te verrijken op basis van de gegevens die in de bijbehorende lokale gebruikers database aanwezig zijn.

#### ****Problemen****

Een probleem wat zich naar voren kwam bij deze oplossing was het instellen van de IDP met prioriteit. Wat er zou moeten gebeuren zonder fout is het volgende:

* Stap1: Een SP redirect een gebruiker naar de preferred -IDP
* Stap2: De preferred -IDP controleert één database voor de username en password combinatie die de gebruiker invoert.
* Stap 3: gebruiker wordt terug naar de SP geridirect met een geldige sessie.

Stap 2 werd niet goed uitgevoerd (zie ter illustratie **Fout! Verwijzingsbron niet gevonden.**). In plaats van dat een IDP de database waar aan deze is gekoppeld gebruikt voor de controle van een gebruikers username en password, controleerde de IDP een database voor username en password en een database voor alleen de username.

Figuur Credential check fout

Eén database had de rol voor het controleren van de username en password en de andere database had de rol voor het controleren of de username wel bestaat. Als aan een van deze twee voorwaarden niet werd voldaan dan kreeg een gebruiker geen geldige sessie. Door het uitvoeren van meerdere tests in een poging een oplossing te vinden bleek dat de databases ook van rol wisselen (dit gedrag deed zich voor op de Enterprise Edition en de Community Edition van JOSSO).

Er is hier sprake van een bug in het JOSSO framework. Omdat de IDP met prioriteit instellen met behulp van een checkbox zou moeten gaan. Op de forums wordt er ook gezegd dat de IDP met prioriteit feature niet betrouwbaar is (dit bericht is op 23-1-2012 geplaatst).

### ****resultaat****

Met behulp van dit POC project is aangetoond dat het mogelijk is gebruikers accounts aan elkaar te koppelen.

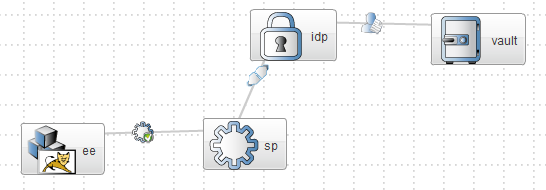
## Ondersteuning voor spring security

Het doel van de deze oplossing is: Aantonen dat een applicatie die gebruik maakt van Spring Security ook gebruik kan maken van het JOSSO framework.

### ****Aanpak****

Hiervoor is een eenvoudige JOSSO omgeving opgezet, zodat het doel zo snel mogelijk bereikt wordt. Dit kan worden gedaan omdat het in deze oplossing draait om de applicatie die wordt gejossyfiëd en niet een koppeling tussen verschillende applicaties.

De test omgeving bestaat uit: een IDP met een Identity vault, een SP met een tomcat7 server als execution environment met daarop een applicatie draaiende die gebruik maakt van spring security. Het model van de omgeving is te zien in Figuur 8.



Figuur 14 model

Na het jossyfiën van de applicatie, is geverifieerd dat de applicatie werkt zoals bedoeld door in te loggen met een gebruiker die niet de juiste rol heeft, waardoor een *403 error* wordt opgegooid. Met een gebruiker met de juiste rol werd er wel toegang verleend.

#### ****Problemen****

Voordat de applicatie ontwikkeld kon worden moest er uitgezocht worden hoe spring security in de praktijk gebruikt kan worden, dit koste tijd. Verder zijn er geen praktische problemen geweest tijdens dit POC project.

### resultaat

Met behulp van dit POC project is aangetoond dat applicaties die van springsecurity gebruik maken ook deel uit kunnen maken van een COT die is opgezet met behulp van het JOSSO framework.

# Oplossingsplan

Het is belangrijk dat alle bevindingen die worden gedaan tijdens dit project goed worden gedocumenteerd zodat als er ooit een externe klant behoefte heeft aan een SSO oplossing, al het vooronderzoek al gedaan is.

Het is ook een doel van het project om een POC op te leveren dat bevestigt dat wat er gedocumenteerd is ook in de praktijk toepasbaar is.

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Realiseren
* Analyseren
* Beheren
* Documenteren
* Realiseren
* Ontwerpen

# Invoering

De invoering is nog erg ver in de toekomst. De manier en waarop E-ID de ontwikkelde software in gebruik gaat nemen gaat afhangen van de resultaten die behaald zijn in de voorgaande stappen.

Het zou ook kunnen dat het project nog niet binnen de tijd die beschikbaar is voor de afstudeerstage uitgevoerd kan worden. Als dit het geval is wordt het project mogelijk voortgezet na de stage.

Toebehoren competenties:

* Professioneel handelen
* Methodisch handelen
* Realiseren
* Beheren
* Samenwerken
* Analyseren
* Ontwerpen
* Adviseren

1. Requirements

# Doel en scope

## Doel

Het doel van dit project is, het realiseren van een systeem en/of techniek in de vorm van een Proof Of Concept (POC) die identiteitsmanagement systemen kan koppelen en zo Single Sign On (SSO) tussen domeinen kan realiseren.

## Scope

De scope van dit project omvat het realiseren van een POC, met alle bijbehorende documentatie.

## Product omschrijving

Het product wat ontwikkeld gaat worden zal in staat zijn om:

* Identiteitsmanagement systemen, die zich bevinden op verschillende domeinen, aan elkaar te koppelen.
* SSO navigatie tussen domeinen te realiseren maken.

# Requirements

In dit hoofdstuk zijn de requirements waar het eindproduct aan moet voldoen gedocumenteerd. Het hoofdstuk is opgedeeld in twee delen, functional requirements en non functional requirements.

Aan iedere requirement wordt een belangrijks graad gegeven met behulp van de MoSCoW methodiek. De MoSCoW methodiek werkt als volgt, een requirement krijgt een M, S, C of W als waarde.

* De waarde M staat voor **M**UST have.
* De waarde S staat voor **S**HOULD have.
* De waarde C staat voor  **C**OULD have.
* De waarde W staat voor **W**ONT have.

## Functional requirements

In dit hoofdstuk worden requirements weergegeven in tabellen en gekoppeld aan een MoSCoW waarde. Aan iedere requirement wordt een ID gekoppeld bestaande uit een letter en een cijfer zodat er gemakkelijk naar een requirement kan worden gerefereerd. Eventueel wordt er onder een tabel ook nog een toelichting gegeven voor bepaalde requirements.

\*note: Als er bij een requirement een M staat betekend dat, dat de oplossing (het framework) het moet ondersteunen. Dit betekend niet dat als de oplossing in de praktijk wordt geïmplementeerd de requirement ook verplicht moet worden ondersteund. Dit komt doordat voor het in gebruik nemen van sommige requirements niet kan zonder wijzigingen aan te brengen in een domein en dat is niet altijd wenselijk.

Begrippen die in dit hoofdstuk worden gebruikt:

* Account = De combinatie van een gebruikersnaam en wachtwoord.
* Profiel = de attributen die bij een account horen (bijv. een Email adres).
* Gebruiker = iemand die gebruikt maakt van een domein.
* FIF = Federated identity framework.
* SSO = Single Sign On
* SLO = Single Log Out
* Her-authenticatie = Als een gebruiker van een ander domein af komt hij zijn wachtwoord moet invoeren op het moment dat hij zijn profiel data wil bewerken. Zo is er zekerheid van dat het de gebruiker zelf is de bewerkingen wil uitvoeren en niet iemand anders.
* Als in de kolom **gebruiker is bekend bij** “Domein A & Domein B” staat dan is er een koppeling tussen de accounts van de gebruiker.

Authenticatie

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Gebruiker is bekend bij | Requirement | MoSCoW |
| a1 | Domein A & Domein B | Gebruiker kan met zijn account van Domein A inloggen op Domein B | S |
| a2 | Alleen Domein A | Gebruiker kan met zijn account van Domein A inloggen op Domein B | S |
| a3 | Domein A & Domein B | Gebruiker is ingelogd op domein A en kan via SSO naar domein B navigeren | M |
| a4 | Alleen Domein A | Gebruiker logt in op Domein A en logt via SSO in op Domein B | W |
| a5 | Alleen Domein A | Gebruiker logt in op Domein A en navigeert naar domein B en wordt gevraagd of hij een account wil aanmaken en koppelen | M |
| a6 | Domein A & Domein B zonder koppeling | De gebruiker moet in staat zijn om zijn accounts aan elkaar te koppelen | M |
| a7 | Domein A & Domein B | als een gebruiker inlogt op een standaard of niet standaard manier moet dit bijgehouden worden. | M |

Toelichting op ID a7: Met de standaard manier wordt bedoel dat een gebruiker is ingelogd op een domein met de gebruikersnaam en wachtwoord die bij dat domein horen. Een voorbeeld van een niet standaard manier is als een gebruiker vanuit een ander domein via SSO binnen komt. De reden waarom dit bijgehouden zou moeten worden is omdat er extra veiligheidsmaatregelen kunnen worden geïmplementeerd als dit wordt bijgehouden. Een voorbeeld van een extra veiligheidsmaatregel is her-authenticatie.

### Accounts koppelen

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Gebruiker is bekend bij | Requirement | MoSCoW |
| k1 | Domein A & Domein B zonder koppeling | De gebruiker moet in staat zijn om zijn accounts aan elkaar te koppelen | M |
| k2 | Domein A & Domein B zonder koppeling | Gebruiker logt in bij Domein A en navigeert naar Domein B, Domein B vraagt of de gebruiker of hij zijn accounts wil koppelen | M |
| k3 | Alleen Domein A | Gebruiker logt in op Domein A en navigeert naar Domein B, Domein B vraagt of de gebruiker een account wil aanmaken die meteen gekoppeld is aan die van Domein A | M |
| k4 | Geen Domein | Het Domein vraagt of hij een account aan wil aanmaken dat meteen gekoppeld is aan Domein A en Domein B | W |

Autorisatie

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Gebruiker is bekend bij | Requirement | MoSCoW |
| o1 | Domein A & Domein B | Gebruiker logt in op Domein A en logt via SSO in op Domein B, en kan zijn B-profiel **inzien**. (uitzondering her-authenticatie) | S |
| o2 | Domein A & Domein B | Gebruiker logt in op Domein A en logt via SSO in op Domein B, en kan zijn B-profiel **wijzigen**. (uitzondering her-authenticatie) | C |
| o3 | Domein A & Domein B | Gebruiker logt in op Domein A en logt via SSO in op Domein B, en kan zijn B-profiel **verwijderen**. (uitzondering her-authenticatie) | W |
| o4 | Domein A & Domein B | Gebruiker logt in op Domein A en logt via SSO in op Domein B, en verkrijgt zijn toegekende rollen en groepen voor domein B (specifiek per rol koppelen) | M |
| o5 | Domein A & Domein B | Gebruiker is ingelogd op Domein A en Domein B, als hij uitlogt op een domein zal hij ook op het andere domein uitloggen (SLO) | M |

Als er bij een requirement “uitzondering her-authenticatie” staat betekend dit dat, dat de requirement een M waarde zou moeten hebben als her-authenticatie mogelijk is.  Het probleem is echter dat om her-authenticatie mogelijk te maken er wijzigen moeten worden doorgevoerd op een domein, dat niet altijd wenselijk is.

Administrators

Er wordt hier over administrators gesproken, daaronder wordt verstaan: de administrator van een domein. Dit is dus iets anders dan een gebruiker met extra privileges.

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| m1 | Administrator accounts kunnen aan elkaar gekoppeld worden. (uitzondering her-authenticatie) | W |
| m2 | Als een gebruiker administrator is op Domein A heeft hij deze rechten ook op Domein B | W |
| m3 | Een administrator mag accounts van aderen gebruikers aan elkaar koppelen | C |
| m4 | Een administrator van Domein A moet op Domein A kunnen inloggen met zijn account van Domein B en gebruik kunnen maken van zijn administrator rechten. (uitzondering her-authenticatie) | W (S) |
| m5 | Het FIF heeft zijn eigen administrator accounts, deze staan los van de domeinen. | M |

Toelichting op ID m1: Het is niet wenselijk dat administrator accounts aan elkaar gekoppeld worden. Dit is omdat zelfs als een administrator account maar rechten heeft tot een domein, en de account wordt gecomprimeerd de schade voor het domein erg groot kan zijn. Door een administrator account op meerdere domeinen rechten te laten hebben word de impact van een gecomprimeerd account vele malen groter. Dit is ook de reden dat aan ID m5 een MUST is toegekend, het is niet wenselijk dat er veel mensen zijn met administrator rechten voor het FIF. Dit is omdat misbruik van deze rechten grote gevolgen kan hebben voor ieder domein dat is aangesloten op het FIF.

## Non functionals

De non functionals zijn verdeeld over drie paragraven: algemeen, profiel data en communicatie.

De paragraaf “profiel data” gaat over hoe met profiel data moet worden omgegaan. De paragraaf “communicatie” gaat over hoe informatieoverdracht plaats vindt tussen systemen.

### Algemeen

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| n1 | Navigeren van domein A naar B m.b.v. SSO duurt maximaal 5seconden | M |
| n2 | Er moet rekening worden gehouden met het beveiligen tegen “broken authentication and session management” | M |
| n3 | Er moet rekening worden gehouden met het beveiligen tegen “Cross-site request forgery (CSRF)” | M |
| n4 | Als er gebruik wordt gemaakt van een framework moet deze een actieve community hebben | S |
| n5 | Er moet gelet worden op de licenties van de software die gebruikt wordt | M |
| n6 | Een framework moet minimaal 5jaar oud zijn. | M |
| n7 | Een framework moet nog steeds actief ontwikkeld worden | M |
| n8 | Tijdens de ontwikkelfase mogen er geen kosten verbonden zijn aan het gebruik van een framework. | M |
| n9 | Applicaties die gebruik maken van spring security moeten ook aangesloten kunnen worden op het framework | M |
| n10 | Het moet mogelijk zijn voor websites om hun originele loginpagina te behouden | M |

Toelichting op ID n1: Het realiseren van een bepaalde performance is niet hoofddoel van dit project, het gaat om het realiseren van SSO tussen domeinen. De standaard voor het laden van een pagina is 2 seconden. Voor Dit project wordt daar 3seconden bij opgeteld omdat er achter de schermen een hoop gebeurd, dit brengt de maximale tijd die het mag kosten om m.b.v. SSO van domein A naar domein B te navigeren op 5 seconden.

Toelichting op ID n2 & ID n3: Uit de OWSAP top 10 zijn dit twee meest grootste security risico’s voor een FIF. Dit is omdat een FIF juist zorgt voor het opzetten en managen van sessies en hierbij wordt mogelijk gebruik gemaakt van een parameter in een URL.

Profiel data

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| p1 | Er zal specifieke profiel data tussen Domein A en Domein B gedeeld worden | M |
| p2 | Het FIF houdt gemeenschappelijke profiel data bij | W |
| p3 | Domein A mag gebruik maken van alle profiel data van Domein B als de profielen aan elkaar gekoppeld zijn. | W |
| p4 | Het FIF heeft leesrecht tot alle profiel data bij de domeinen die aangesloten zijn op het FIF. | S |
| p5 | Het FIF heeft schrijfrecht op alle profiel data bij de domeinen die aangesloten zijn op het FIF. (mag dus nieuwe dingen toevoegen) | W |
| p6 | Het FIF heeft wijzigingsrecht op alle profiel data bij de domeinen die aangesloten zijn op het FIF. (mag dus bestaande data van gebruikers wijzigen) | W |
| p7 | Het FIF heeft verwijderingsrecht op alle profiel data bij de domeinen die aangesloten zijn op het FIF. | W |

Communicatie

|  |  |  |
| --- | --- | --- |
| ID | Requirement | MoSCoW |
| c1 | Responses en requests zullen het SAML protocol aanhouden | S |
| c2 | Ieder domein gebruikt hetzelfde type security token | C |
| c3 | Alle security policies zijn in het bezit van het FIF (het FIF regelt dus alle autorisatie in dit geval) | W |
| c4 | Gebruikers moeten in ieder domein hetzelfde ID hebben | W |

Toelichting ID c1: Het is wenselijk dat het SAML protocol wordt aangehouden omdat dit heeft meest ondersteunde protocol op de markt is. Dit betekend dat de kans groot is dat als er een nieuw domein op het FIF moet worden aangesloten, de kans relatief groot is dat het nieuwe domein de standaard ondersteund.

1. Protocollen

Dit document presenteert technieken die in nou verband staan tot SSO, met een korte omschrijving en een bijbehorende bron.

SAML

**Omschrijving:**

The "Web SSO" profile of SAML is a commonly deployed protocol in enterprises/ISPs/schools/etc. to allow users to login once, and automatically be able to authenticate to other systems both within the organization, and externally.  More generally SAML is an XML-based framework for identity assertions & protocols, such as Web SSO, but also SOAP web service security tokens.

*“defines the format of XML messages that applications can use to exchange information about network identities and access rights”*

**Bron:** [OASIS](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security)

WS-federation

**Omschrijving:**

“WS-Federation extends WS-Trust to provide a flexible Federated Identity architecture with clean separation between trust mechanisms, security token formats, and the protocol for obtaining tokens. This architecture enables a reusable security token service model and protocol to address the identity requirements of both webapplications and web services in a variety of trust relationships.”

“WS-Federation does not restrict users to a specific security token format. Instead, WS-Federation builds on the WS-Trust encapsulation mechanism, the RST/RSTR, which allows protocol processing to remain agnostic of the type of token being transmitted. This enhances the interoperability and migration of customer deployed products as the industry introduces new and better security token formats.”

**Bron:** [microsoft](http://msdn.microsoft.com/en-us/library/bb498017.aspx)

OpenID

**Omschrijving:**

OpenID is een gedecentraliseerd authenticatiemechanisme dat het mogelijk maakt om met één account op meerdere websites in te loggen.

Het werkt als volgt, een gebruiker maakt een openID account aan bij een openID provider (bijv. Gmail of Yahoo). Andere websites (bijv. Blog sites) accepteren deze openID-account.

Als de gebruiker gebruik wil maken van zijn OpenID account moet hij selecteren wie zijn OpenID provider is, vervolgens moet hij inloggen. Vervolgens stuurt de OpenID provider gegevens van de gebruiker door naar website waarop de gebruik wil inloggen.

Het voordeel van OpenID is dus dat gebruikers met één account op veel domeinen van het internet welkom zijn. Maar het is geen multi-domain-SSO oplossing, een gebruiker moet nog steeds op iedere site zijn OpenID provider selecteren en inloggen, er is dus geen sprake van multi-domain-SSO.

\*note: OpenID Identifiers--or the identification string the end user claims is his or her own unique identifier--can, at least as of OpenID 2.0, take one of two forms. They can be either a standard Web URL, or an XRI.

**Bron:** [openID](http://openidexplained.com/)

OAuth

**Omschrijving:**

OAuth is een protocol waarmee gebruikers toegang tot hun gegevens kunnen verlenen aan een andere partij. Het werkt als volgt, een gebruiker heeft een account bij een website die OAuth ondersteund voor haar gebruikers (bijv. google). Als de gebruiker naar een andere site (voorbeeld.nl) gaat en gebruik wil maken van een service waarbij de service gegevens van de gebruiker nodig heeft; dan zal de service de gebruiker de mogelijkheid bieden om deze gegevens met gebruik van OAuth beschikbaar te stellen. Dat gebeurt als volgt, de gebruiker wijst aan welke OAuth ondersteunende site zijn gegevens beheerd. De gebruiker wordt vervolgens naar die site gebracht en moet daar inloggen en bevestigen dat zijn gegevens door voorbeeld.nl gebruikt mogen worden. De gebruiker wordt teruggebracht naar voorbeeld.nl en kan nu gebruik maken van de service.

Het voordeel van OAuth is dus dat gebruikers toegang tot hun gegevens kunnen verlenen die op een ander domein worden beheerd. Maar OAuth is geen SSO oplossing, een gebruiker moet nog steeds op iedere site zijn OAuth provider selecteren en inloggen, er is dus geen sprake van SSO.

**Bron:** [Link](http://vsr.informatik.tu-chemnitz.de/edu/2011/webe-seminar-ss/drafts/03/) & [workflow](http://hueniverse.com/2007/10/beginners-guide-to-oauth-part-ii-protocol-workflow/)

XRI

**Omschrijving:**

XRI (Extensible Resource Identifier) is een uitbreiding op de syntax van IRI (Internationalized Resource Identifier, RFC 3987) dat een uitbreiding is op de URI (Uniform Resource Identifier) syntax.

Als er aan een URL (Uniform Resource Locator) een stuk XRI syntax wordt toegevoegd kan een proxy XRI resoluties doen, het resultaat daar van is een XRDS (Extensible Resource Descriptor Sequence).

Een XRDS is een XML document met daar in meer informatie over een person/organisatie maar het kan ook andere dingen bevatten zoals identifiers.

**Bron:** [Link](http://www.webreference.com/programming/perl/comments/openid/2.html) & [OASIS-FAQ](http://www.oasis-open.org/committees/xri/faq.php)

1. Frameworks

Een overzicht van identity management frameworks**(samengesteld in maart 2012)**, bron: de link die vermeldt staat bij het framework.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Naam | Kosten/ License | Notes | bouw jaar | Link | Laatste release en forum activiteit | Ernstigheid van recente bugs |
| SourceID | Open Source  (commercieel) | (SAML, WS-Federation) | 2001 | <http://www.sourceid.org/about/index.cfm> | **release:** 9-2-2011**,**  **issue post:** geen inzicht op issues **,** | Geen community, deze bevindt zich bij ping-Identity. |
| ID-ON-PROD | Open Source  (commercieel) | Gebaseerd op JOSSO(JEE en SAML2), verder is er op de site een hoop verkoop praat maar weinig inhoudelijks. | 2004 | <http://www.atricore.com/solutions/> | **release: -,**  **issue post: - ,** | Het is een commerciële versie van JOSSO. |
| TridentHE Access Manager | - | Based on **Shibboleth**.  (geen downloadbaar product beschikbaar) | 2004 | <http://www.aegisusa.com/tridentHE.html> | **Latest news:** 15-9-2011 | Deze zijn het zelfde als die van shibboleth zelf. |
| Shibboleth | Open Source  ([apache](http://www.apache.org/licenses/LICENSE-2.0.html) 2.0) | Uitgebreide documentatie  **(**SAML 2.0, WS-Federation, OpenID, OAuth [link](https://wiki.shibboleth.net/confluence/display/DEV/Supported+Protocols)**)** | Project: 2000  Code: 2003 | <http://shibboleth.internet2.edu/> | **release:** 27-2-2012 **,**  **issue post:** 12-3-2012**,**  **(**small but global dev-team**)** | Lijken allemaal op minor issues. |
| openSAML | Open Source  ([apache](http://www.apache.org/licenses/LICENSE-2.0.html) 2.0) | **Shibboleth** werkt met dit project. Verdere ontwikkeling van dit project gebeurt ook vanuit shibboleth. | 2005 | <http://www.unicon.net/opensource/shibboleth> | **update:** 27-2-2012 **,**  **issue post:** 22-2-2012**,** | openSAML valt onder shibboleth. Issues m.b.t. openSAML lijken medium. |
| Force.com | Dev-open source | Geen passend product, het is een ontwikkel platform. (Wel Fijne documentatie over algemene begrippen.) | - | <http://wiki.developerforce.com/page/About> | **release: - ,**  **issue post: - ,** | - |
| PicketLink | Open source (LGPL) | Voormalig **JBoss SSO,** (SAML 1.0, 1.1, 2.0; WS-Federation 1.2; WS-Trust 1.4; OpenID 2.0; OAuth 1.0A, 2.0, OAuth WRAP) | 2005 | <http://www.jboss.org/picketlink> | **release**: 29-2-2012,  **issue** **post**: 6-3-2012, (oogt kleiner dan bijv. openAM) | Bijv: Error parsing security token request (dit gaat over XACML) |
| JOSSO | Open source ([LGPL](http://www.josso.org/confluence/display/JOSSO1/GNU+LESSER+GENERAL+PUBLIC+LICENSE)) | Uitgebreide documentatie, “Dutch Railway Services” maakt o.a. gebruik van josso  (SAML, SPML, WS-Trust en WS-Security) | August 2004 | [JOSSO-Home](http://www.josso.org/confluence/display/JOSSO1/JOSSO+-+Java+Open+Single+Sign-On+Project+Home) | **release**: 18-10-2011 ,  **forum post**: 12-3-2012,  (actief forum) | Weinig issues (laatste uit 2011), oude issues lijken medium |
| openAM | Open source (CDDL) [lijkt op LGPL] | Bouwt voort op een oud Oracle project, genaamd **openSSO** (SAML, WS-Federation) | openSSO:  2005-2006 | <http://www.forgerock.com/openam.html> | **release**: 7-11-2011,  **issue post**: 12-3-2011. (250 mensen in de wiki) | Bijv: problemen bij het gebruik van meerdere passwords. |
| Ping-Identity | 90days free trial. | Based on SourceID, groot klanten bestand.  ([SAML](https://www.pingidentity.com/resource-center/SAML-Tutorials-and-Resources.cfm), [OpenID](https://www.pingidentity.com/resource-center/openid.cfm), WS-Fed) | SourceID: 2001 | [www.pingidentity.com](https://www.pingidentity.com/our-solutions/pingfederate.cfm) | **release**: 1-11-2011 ,  **issue post**: issues hebben geen datum ,  (actief forum) | Minor (commerciële organisatie, mogelijk worden ernstige issues afgeschermd) |
| Trivoli FIM | Eigen license | Er is weinig informatie te vinden over het product. Er is wel een overvloed aan documentatie. (Liberty, SAML, WS-Federation, WS-Security and WS-Trust) | 2006 (gebaseerd op een copyright datum) | <http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/> | [Last-release:](https://www-304.ibm.com/support/docview.wss?uid=swg24026593) 26-4-2010  Er is een community maar die is niet specifiek voor dit product. | Niet beschikbaar |
| CAS | <http://www.jasig.org/cas/license> lijkt op apache 2.0 | Geen goede ondersteuning van bekende protocollen en/of standaarden. | 2004 | <http://www.jasig.org/cas/about> | **Last update:** 15-5-2012 | Medium issues |
| CROWD | Commercieel | Niet geschikt: “All your Crowd-connected applications must be in the same domain.” | - | <http://confluence.atlassian.com/display/CROWD/Overview+of+SSO> | **-** | - |

# Shortlist

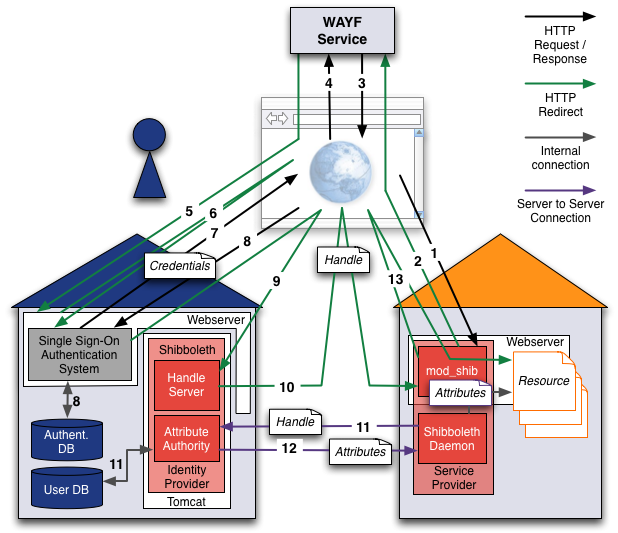
(G= goed , V = voldoende en O = onvoldoende)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Framework | Shibboleth | JOSSO | openAM | Ping-Identity |
| Geschikt voor het koppelen van sites | G [link](http://shibboleth.internet2.edu/uses.html) | G [link](http://www.josso.org/confluence/display/JOSSO1/JOSSO2+Birds-eye+View) | V [link](http://forgerock.com/sites/default/files/file/ProductSheet-OpenAM-A4.pdf) | G [link](https://www.pingidentity.com/customers/case-study-Ping-Identity-Powers-Internet-SSO-to-SURFnet's-Research-and-Education-Network-for-the-Netherlands-6757.cfm) |
| Voorbeeld implementaties beschikbaar / tutorials etc. | Er is veel documentatie beschikbaar voor het installeren. | Tutorial [link](http://www.josso.org/confluence/display/JOSSO1/Video+Tutorial+-+Internet+SSO+Rollout+using+JOSSO2) | Tutorial [link](https://wikis.forgerock.org/confluence/display/openam/3+OpenAM+Server+Configuration) | Sample java app van ping-Identiy zelf. [link](https://www.pingidentity.com/support-and-downloads/product-documentation/java-integration-kit/2-4-2/loader.cfm?csModule=security/getfile&pageid=14314) |
| Toegankelijkheid | G | G (2007 note\*: let op, het kost tijd als je applicatie server niet direct ondersteund wordt door JOSSO) | V | V |
| Algemene indruk | Overvloed aan mogelijkheden, Documentatie is duidelijk en zeer makkelijk te vinden. | Het lijkt een erg resultaatgericht product, maar het is onduidelijk wanneer JOSSO 1 en wanneer JOSSO 2 gebruikt moet worden. | Het framework geeft een erg complete indruk, er is ook voldoende documentatie maar deze is onduidelijk. | Overvloed aan mogelijkheden |
| Verwachte implementatie tijd | 1-2 weken (schatting) | 1-2 weken (schatting) mits Tomcat als applicatie server wordt gebruikt. | 1-2 weken (schatting) | 4 weken (voor meer dan 90% van de gevallen) |
| Heeft een forum | G [link](http://shibboleth.1660669.n2.nabble.com/) | G [link](http://sourceforge.net/search/?group_id=116854&type_of_search=forums&words=Enter+Keyword&search=Search) | G [link](https://forums.oracle.com/forums/forum.jspa?forumID=876) | G [link](https://www.pingidentity.com/support/answers/) |
| Duidelijk beeld van de architectuur | G | G | V | O |
| Impact op domein | Ja | Ja | Ja | Ja |

**Shibboleth**:

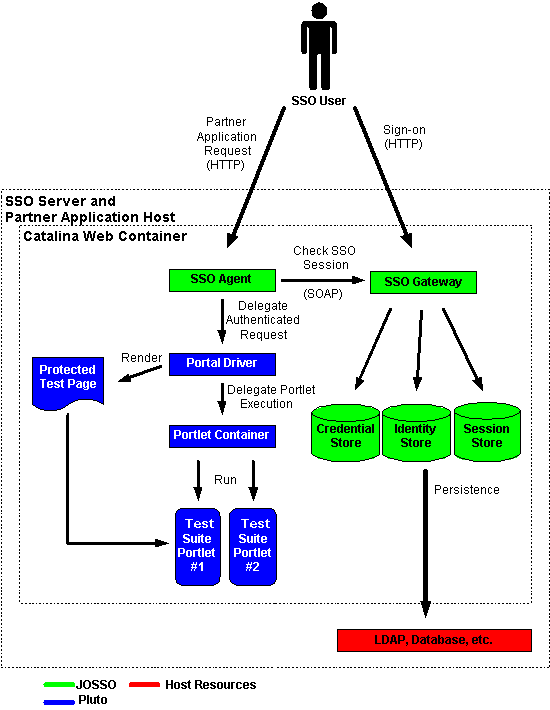
<http://www.switch.ch/aai/demo/expert.html>

(user is niet ingelogd) domein A is de home organisatie en domein B en een service provider

1. Gebruiker navigeert naar domein B (zegt, Ik wil Service-B ik kom van Domein-A)
2. Redirect naar een WAYF server (where are you from)
3. De WAYF laat een pagina zien waar de gebruiker moet invullen waar hij vandaan komt (domein A, B, C of D etc.)
4. De gebruiker bevestigd dat hij een keuze heeft gemaakt
5. De gebruiker krijgt een cookie die zijn keuze voor de sessie bij houdt en de browser stuurt een request naar de shibboleth handle server die zich bevind in domein A
6. Browser vraagt de login pagina aan.
7. De browser krijgt de login pagina.
8. De gebruiker vult zijn naam en WW in en het systeem (wat los staat van shibboleth) controleert het.
9. De browser krijgt een cookie en een redicrect naar de shibboleth server die zich bevend in domein A.
10. De shibboleth server op domein A ziet dat de gebruiker geauthenticeerd is en geeft de gebruiker een “handle” en stuurt deze handle door naar de shibboleth plungin van domein B.
11. De shibboleth server op domein B vraagt, m.b.v. de handle, de shibboleth server op domein A wat voor een attributen de gebruiker allemaal heeft.
12. De shibboleth server op domein A verifieert de handle en stuurt een response met de attributen naar domein B.
13. De gebruiker krijgt een shibboleth cookie en krijgt toegang tot de service op domein B
14. 

**JOSSO (2005):**

<http://today.java.net/pub/a/today/2005/02/18/josso.html>

1. The user requests a protected resource from a partner application.
2. The SSO Agent protecting the partner application intercepts the request and, since it is unidentified, redirects the user to the single sign-on form on the SSO Gateway.
3. The user enters his/her credentials, which, depending on the selected authentication scheme, they may request either a user/password pair or a X.509 Client Certificate.
4. The SSO Gateway processes the form submission, loads the user's credentials from the configured stores (e.g., LDAP, database) and checks their validity.
5. If the credentials are valid, the user is authenticated and an SSO session token is generated and saved for his/her session in the session store. The user is then redirected back to the protected resource originally requested.
6. The SSO Agent protecting the partner application intercepts the request, and using the SSO Gateway JAAS Login module, checks the validity of the session and obtains the authenticated subject from the SSO Gateway using SOAP.
7. The SSO Gateway picks up the session identifier supplied from the session store and obtains the associated user information from the identity store.
8. The SSO Agent introduces the authenticated subject returned by the SSO Gateway web service in the HTTP request and handles it to the target webapplication.
9. The webapplication processes the HTTP request and eventually uses the getUserPrincipal() and isUserInRole() standard calls on the HttpServletRequest instance to obtain the identity of the authenticated user to perform additional security assertions.
10. 

**openAM:**

<https://wikis.forgerock.org/confluence/display/openam/OpenAM+and+ADFS2+configuration>

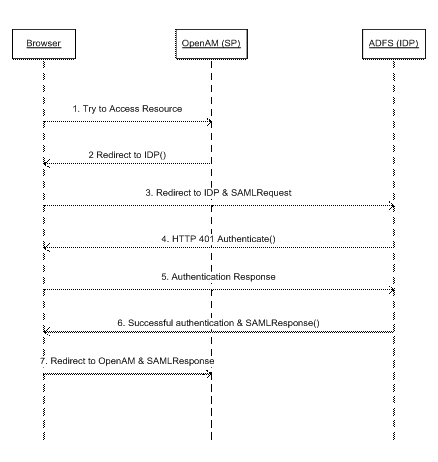
*Step* 1: User tries to access a resource on the SP (OpenAM server). This webpage will try to detect if the user has already got an OpenAM session, if not the user will be redirected to the IDP (ADFS) server.  
A SAML Authentication Request message is generated and sent along with the redirection (*Step* 2 and 3)

In Step 3 the IDP server receives the request. If the client browser has already has an ADFS session, step 4 and 5 are omitted, otherwise an HTTP 401 will be issued to the client which is basically an HTTP authentication request.

*Step* 5 will return the authentication response. In case of NTLM/Kerberos this is an automated process via the SPNEGO protocol.

*Step* 6: the IDP server receives the authentication response and when successful it will generate a SAML response.

*Step* 7: Client browser is redirected back to the OpenAM server where is decodes the SAML response and issues an OpenAM token.



**Ping-Identity:**

<http://documentation.pingidentity.com/display/AIK11/SP+Overview+and+Configuration>

1. PingFederate receives a SAML assertion from an IdP partner. The assertion is validated and parsed into the user attributes, which are temporarily maintained within PingFederate.
2. The PingFederate server redirects the user to the target application with a reference to the user attributes. The reference is included in the URL query string. For example: [https://target.example.com?REF=ABC123](https://target.example.com/?REF=ABC123)
3. The target application makes an authenticated direct HTTP(S) call to PingFederate to retrieve the user attributes. For example: <https://pingfederate.example.com:9031/ext/ref/pickup?REF=ABC123>
4. PingFederate looks up the attributes (in the above example, referenced by ABC123) and provides them to the target application in the HTTP response.
5. The target application uses the attributes to create a user session, enabling access to the target resource.

# JOSSO vs Shibboleth

Shibboleth is ontworpen zodat een site die een dienst aanbied niet de authenticatie van de afnemers hoeft te doen, de gebruikers worden gewoon doorgestuurd naar hun “home” site waar ze inloggen en zo een token krijgen die hun de rechten geeft om van de dienst van een andere site gebruik te maken. JOSSO is ontworpen zodat meerdere partijen elkaars gebruikers kunnen delen.

Bij shibboleth begint het hoofd scenario met een gebruiker die gebruik wil maken van een service en moet aangeven van welke organisatie hij of zij afkomstig is en wordt vervolgens doorgestuurd naar de organisatie om in te loggen en wordt daarna weer doorgestuurd naar de service. Bij JOSSO hoeft een gebruiker niet aan te geven van welke organisatie hij afkomstig is, dus is de gebruikers ervaring beter.

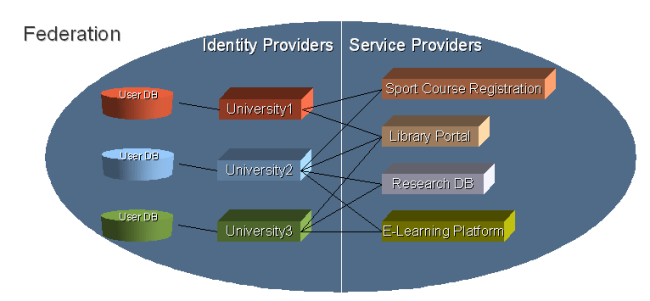


Figure 1 shibboleth

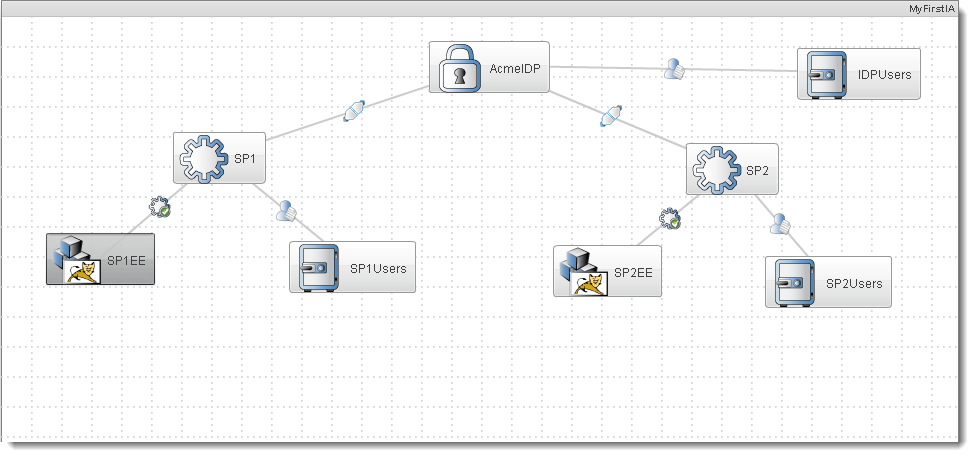


Figure 2 JOSSO

De JOSSO gemeenschap is toegankelijker dan die van shibboleth, als nieuwkomer op de JOSSO forums worden gestelde vragen snel beantwoord en op de forums van shibboleth wordt je als nieuwkomer genegeerd.

Verder lijkt de Enterprise Edition (EE) versie van JOSSO sneller te implementeren dan shibboleth. De EE versie van JOSSO hoeft niet gebruikt te worden maar om een eerste indruk te krijgen kan gebruik worden gemaakt van een trail van 15dagen. Als de EE versie geschikt blijkt kan gekeken worden naar het eventueel gebruiken van de Community Edition (CE) versie. (de gemeenschap van JOSSO raad aan met de CE versie te beginnen maar hier zijn geen tutorial ’s van.)

Tot nu toe lijkt het er op dat het grootste verschil tussen de CE en EE versie is dat er bij de EE gebruik kan worden gemaakt van installatie-wizards om de installatie te versnellen en te vergemakkelijken.

**Conclusie**

Het eerste framework wat geprobeerd gaat worden is JOSSO, omdat dit framework het meest geschikt lijkt voor het oplossen van het probleem dat leidend is voor dit project. Wat een bijrol speelt is de toegankelijkheid van de gemeenschap, het kan nuttig zijn komen als vragen die naar voren komen kunnen worden beantwoord door mensen die ervaring hebben met de software.

JOSSO maakt gebruik van een Lesser General Public License, een quote van wiki de belangrijkste aspecten hiervan weergeeft is: “Wijzigingen in de broncode van de LGPL-delen van een product moeten ter beschikking worden gesteld aan de gebruikers. Echter nieuwe geschreven code die alleen maar gelinkt is aan de LGPL-code hoeft niet ter beschikking te worden gesteld.”

1. Loginpagina code

|  |
| --- |
| <%@page contentType="text/html" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <body>  <% // Check if error information has to be displayed  if (request.getParameter("josso\_error\_type") != null) { %>  <font color="red">Invalid login information</font>  <% } %>  <form name="usernamePasswordLoginForm"  method="post" action="<%=request.getContextPath()%>/josso\_authentication/"  <input type="hidden" name="josso\_back\_to"  value="http://domeina.nl:8080/basicdemo/geheime.html">  <input type="hidden" name="josso\_on\_error"  value="http://domeina.nl:8080/basicdemo/loginerror.html">  <input type="hidden" name="josso\_cmd" value="login">  <table border="0" cellpadding="0" cellspacing="5">  <tr><td>username:</td><td><input type="text" name="josso\_username" size="10">  </td></tr>  <tr><td>password:</td><td><input type="password" name="josso\_password" size="10">  </td></tr>  <tr><td colspan="2" align="center"><input type="submit" value="Login" >  </td></tr>  </table>  </form>  </body>  </html> |

Wanneer de gebruiker dit form submit zal de josso agent de redirect opvangen doordat deze filterd op “/josso\_authenticate/”. Omdat de velden een specifieke naam hebben, kan de IDP deze gebruiken als parameters. Hierdoor is het mogelijk is voor de IDP om het authenticatie proces uit te voeren en een sessie voor de gebruiker aan te maken.

1. Scenario

##### Het Henk scenario:

Henk Bossen is een hobbyist auto monteur en hij koopt auto onderdelen en gereedschap in via webshops. Voor auto onderdelen wordt gebruik gemaakt van **partsshop.nl.eid**en voor gereedschap wordt gebruik gemaakt van **tools4you.com.eid.** Henk heeft een account voor persoonlijke account bij beide webshops en deze accounts zijn aan elkaar gekoppeld.

|  |  |
| --- | --- |
| **Domein** | **Account naam van Henk** |
| Partsshop.nl.eid | Henk\_Bossen |
| Tools4you.com.eid | Packman42 |

**Henk gaat een band vervangen:**

Henk logt in op Partsshop.nl.eid, als hij dit heeft gedaan dan ziet hij staan dat hij ingelogd is onder de naam “Henk\_Bossen”. Henk bestelt m.b.v. zijn creditcard gegevens die hij heeft toevertrouwd aan partsshop.nl.eid een nieuwe band. Meteen nadat hij de aankoop heeft bevestigd komt in een reclame balk bijpassend gereedschap tevoorschijn. Henk klinkt op een link voor een gereedschap item dat hij wil aanschaffen. Zo navigeert hij naar Tools4you.com.eid en ziet dat hij daar is ingelogd als “packman42” en kan daar meteen beginnen met het bestellen van het gereedschap dat hij wil aanschaffen.