



Uitweg wil een volgende stap maken in het Uitweg programma, met de ontwikkeling van een opvolger van de protocolmodule. In dit document wordt een product breakdown structure van de Talaris gegeven en worden extra diensten die Uitweg op de producten wil leveren nader beschreven.

In de business case is de geschiedenis, aanleiding, context en afbakening van Talaris al gegeven.

SYSTEEMBESCHRIJVING

Ketens

In een (informatie)**keten** delen twee of meer **ketenpartners** (KP's) informatie die invloed uitoefent op hun processen. Een Uitweg-project resulteert in een aantal punten waarop de behoefte bestaat (deels) geautomatiseerd gegevens uit te wisselen:

- Een analyse van de processen waarin meerdere ketenpartners samen een rol spelen, al dan niet geautomatiseerd.
- Een analyse van de systemen van ketenpartners die een rol spelen in die processen en die daartoe berichten willen uitwisselen (produceren of consumeren).
- Analyse van de punten in het proces waarop eindgebruikers direct gegevens willen uitwisselen met eindgebruikers van een andere ketenpartner, of met een systeem van een ketenpartner.

Uit de procesanalyse zijn vaak een aantal **berichten**, of in sommige gevallen uitgebreider **transacties** van meerdere berichten en enige interne logica af te leiden.

Voorbeeld: een jeugdzorgaanbieder schrijft een jeugdige in en vraagt zich af of andere zorgaanbieders meer gegevens van deze jeugdige kunnen en willen delen. De zorgaanbieder zou zelf alle zorgaanbieders die zij kent kunnen vragen om deze gegevens, of een centrale partij die ze allemaal kent (zoals een Bureau Jeugdzorg) daarom kunnen vragen. Een transactie zou dan de vorm kunnen nemen van: 1. vraag het BJz om de gegevens op basis van identificerende gegevens van de jeugdige; 2. het BJz vraagt alle andere ZA's om de gegevens die zij kunnen en willen delen; 3. als het BJz voldoende responses binnen heeft, of als er een bepaalde tijd verstreken is, rapporteert het BJz aan de oorspronkelijke ZA; 4. de oorspronkelijke ZA combineert de ontvangen gegevens en rapporteert het resultaat ook weer terug aan het BJz; 5. De ZA's die reageerden worden ook door het BJz voorzien van het eindresultaat.

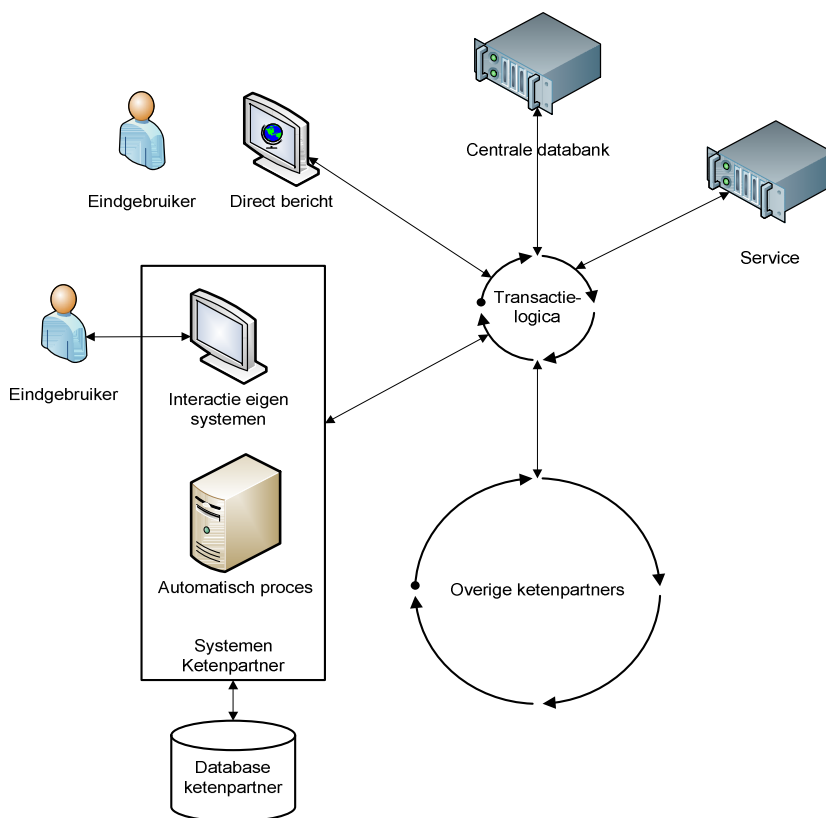
Uit de systeemanalyse komt naar voren welke **dataset** het **ketenpartnersysteem** kan leveren of opslaan, die interessant is binnen een transactie en welke handelingen daartoe in de toepassing moeten worden verricht – automatisch of door de gebruiker. Verder wordt vastgesteld voor welk platform relevante systeemonderdelen zijn geschreven en over welke protocollen/interfaces de systeemontwikkelaar eventueel kan communiceren.

Voorbeeld: het BJz gebruikt IJ, een toepassing op Oracle en met een interface in ASP.Net. De toepassing kan niet automatisch een overzicht van relevante ZA's genereren en zou moeten worden aangepast om dergelijke gegevens te registreren. Vanuit ASP.Net is het wenselijk met een andere .Net-toepassing te communiceren of een SOAP service te consumeren. De toepassing bevat geen faciliteiten voor het opslaan van alle gegevens m.b.t. jeugdigen, maar een subset is wel interessant om te registreren op een eigen dossier. Bij het ontvangen van gegevens van ZA's zou daarom een update van de gegevens van de jeugdige kunnen worden uitgevoerd met akkoord van een daartoe gemachtigde gebruiker.

De analyse van de behoefte naar directe gegevensuitwisseling komt neer op momenten in het proces waarop (nog) geen handeling wordt verricht in een systeem, omdat die handeling pas betekenisvol wordt op het moment dat het systeem gekoppeld is met het systeem van een ketenpartner, of een medewerker van een ketenpartner direct kan worden bereikt.

Voorbeeld: als een ZA uiteindelijk een set resultaten van andere ZA's ontvangt en op basis daarvan een resultaat bepaalt, worden deze gegevens in het eigen systeem opgenomen. De ZA moet melding maken van de zorgverlening bij Zorg voor Jeugd, die daarvoor een webservice aanbieden. Daarbij wil de zorgaanbieder graag aan collega's laten weten wat zij heeft bepaald dat de actuele dataset is, danwel op een centraal toegankelijke plek voor ketenpartners, danwel door ketenpartners direct te berichten.

Dit levert een aantal soorten informatieproducenten en consumenten op:



Figuur 1

Vanaf linksonder, met de klok mee:

- **Automatische processen** kunnen berichten verzenden en verwerken zonder tussenkomst van een gebruiker. Dit kan een (nieuw) onderdeel zijn van bestaande systemen, een alleenstaand proces dat de database met bestaande systemen deelt, een alleenstaand proces dat interacteert (via andere services) met bestaande systemen of een geheel nieuwe toepassing, los van de bestaande systemen van de ketenpartner. In alle gevallen gaat het om software buiten het domein van Talaris en in het domein en beheer van de ketenpartner.

Voorbeeld: een ZA krijgt een verzoek om gegevens over een jeugdige van een vertrouwde partij. De registratieapplicatie van de ZA is uitgebreid met een module die 's nachts dergelijke verzoeken verwerkt en de gegevens bij elkaar queryt en verzendt. Deze module wordt door de softwareleverancier van de ketenpartner geschreven op basis van vanuit Talaris aangeleverde componenten voor het eigen platform.

- De **UI van de systemen van de ketenpartner** (dezelfde categorieën als hierboven) kan ook interactie met de gebruiker bieden voordat of tijdens dat berichten worden verwerkt en verzonden. Ook hier gaat het om software en UI's die buiten het domein van Talaris vallen.

Voorbeeld: de ontvangende ZA krijgt tegenstrijdige gegevens van andere ZA's en wil een gebruiker laten beslissen welke gegevens de juiste zijn voordat deze worden opgenomen in de eigen database. Pas als dat gelukt is, stuurt deze ZA het resulterende record terug naar het BJz.

- Sommige eindgebruikers kunnen **direct** worden bereikt, zonder tussenkomst van de eigen systemen op een (web)UI. Deze interface valt binnen het Talaris domein.

Voorbeeld: een kleine ZA heeft wellicht geen systemen die aan te passen zijn voor een redelijk budget, maar wil mogelijk wel de nodige gegevens leveren of opvragen middels een webinterface.

- Berichten die door alle ketenpartners kunnen worden ingezien, maar niet door een specifieke ketenpartner hoeven worden beheerd, kunnen worden opgeslagen in een **centrale databank**. Deze centrale databank (of tenminste de automatische interactie daarmee) valt binnen het Talaris domein.

Voorbeeld: de uiteindelijk aan het BJz teruggemelde resultaatrecords zouden kunnen worden opgeslagen in een centrale databank, zodat niet alle ZA's hoeven te worden bijgewerkt met een up to date record, maar ze dit op eigen gelegenheid kunnen opvragen vanuit de centrale databank.

- **Services** zijn vergelijkbaar met een centrale databank, in dat ze berichten kunnen consumeren of produceren zonder tussenkomst van een ketenpartner of systemen

van een ketenpartner, maar de service zelf valt buiten het domein van Talaris. Talaris voorziet in een adapter om berichtenuitwisseling met een specifieke service mogelijk te maken met Uitweg ketenpartners.

Voorbeeld: Na het samenvoegen van de resultaten en het vaststellen van een definitief record voor de jeugdige, meldt het vragende ZA aan Zorg voor Jeugd dat er zorg wordt geleverd aan de jeugdige. Vanuit de ZA bekeken is dit een vergelijkbaar bericht als alle andere Talaris berichten. Talaris verzorgt de omzetting naar het formaat dat Zorg voor Jeugd vereist. Als dat later verandert, merkt een ZA daar – in ieder geval vanuit de techniek – niets van.



Transactielogica

Naast het simpele transport van berichten van en naar ketenpartners is het wenselijk dat er op basis van het berichtenverkeer en de inhoud (het vrij toegankelijke deel) van berichten simpele beslissingen kunnen worden genomen over het berichtenverkeer.

Daarbij moeten ketenpartners kunnen ingrijpen in het berichtenverkeer als een transactie anders verloopt dan gepland of niet langer kan worden uitgevoerd. De volgende functionaliteit moet beschikbaar zijn:

- **Annulering**

Een transactie annuleren, of er althans geen deel meer aan nemen. Dit betekent o.a. dat Talaris ook niet op acties van deze partij in een transactie zou moeten wachten. Een dergelijke handeling moet worden gecommuniceerd aan andere partijen in een transactie, met een mogelijkheid daarvoor een reden op te geven.

Voorbeeld: Een ZA die om gegevens van een jeugdige heeft gevraagd heeft deze gegevens niet langer nodig en annuleert de transactie. Het BJz en eventueel andere betrokken ZA's worden hiervan op de hoogte gesteld. Talaris verwerkt eventueel aangeboden berichten nog wel, maar de annulerende ketenpartner is hiervoor niet meer nodig.

- **Conditie**

Specifieke stappen in een transactie kunnen worden gestart op basis van condities over de inhoud van direct daarvoor ontvangen berichten. Vanzelfsprekend heeft Talaris hierbij geen toegang tot versleutelde gegevens en kan alleen de metadata van een bericht of de onversleutelde inhoud hierbij een rol spelen.

Voorbeeld: een transactie zou kunnen plaatsvinden tussen 10 ZA's en een BJz. Het is mogelijk dat men tot de conclusie komt dat een reactie van 3 ZA's al wel voldoende is, zolang die binnen een uur worden ontvangen. Als het ontvangen langer dan een uur duurt of er binnen een uur minder dan 3 reacties worden ontvangen, wordt de transactie als mislukt beschouwd en wordt dat aan de vragende ZA gemeld. Anders worden de binnen een uur ontvangen reacties verzonden¹.

- **Anonimisering**

Vanwege privacy-issues of door technische beperkingen is het soms niet mogelijk voor verschillende partijen in een keten om dezelfde identificerende kenmerken van entiteiten waarover wordt gecommuniceerd te registreren. Daarbij is het niet altijd wenselijk om een entiteit met al zijn kenmerkende gegevens te communiceren, terwijl men toch later wil kunnen vaststellen om welke entiteit het ging. Als deze omstandigheden zich voordoen is het zaak dat ketenpartners van een gedeelde dienst gebruik kunnen maken voor het uniek identificeren van objecten².

Voorbeeld: bij de communicatie over jeugdigen zou gebruikt gemaakt kunnen worden van het BSN voor de identificatie van een jeugdige. Maar zolang het BSN niet bekend is, of wellicht niet geregistreerd mag worden voor dit doel, is een ander gedeeld kenmerk nodig om jongeren te identificeren – mede om te voorkomen dat eindgebruikers telkens weer de match moeten (helpen) maken tussen ontvangen gegevens en gegevens in het eigen systeem.

Beveiliging, autorisatie en authenticatie

Over de communicatie tussen ketenpartners moeten bepaalde kwaliteitsgaranties kunnen worden gegeven. Een aantal (deels *niet-functionele*) eisen op een rij:

- **Zekerheid**

Berichten die worden verzonden moeten ook aankomen en als dat niet gebeurt,

¹ Conditie van toenemende complexiteit zijn denkbaar. In dit voorbeeld zou men bijvoorbeeld zeker 10 minuten kunnen willen wachten, in plaats van direct met de eerste drie resultaten te reageren, maar na 10 minuten wel z.s.m. te zenden na het ontvangen van genoeg reacties. Etc.

² Hiervoor zijn zowel bestaande technische oplossingen denkbaar, zoals het genereren en opslaan van GUIDs of UUIDs (zie <http://www.ietf.org/rfc/rfc4122.txt>) als (ook bestaande) meer functionele oplossingen, zoals het gebruik van een systeem als Handle (zie <http://www.handle.net/>).

moet te achterhalen zijn door een systeem, de eindgebruiker of een beheerder wat er gebeurd is.

- **Veiligheid**

Onderschepte berichten moeten geen gevaar vormen voor de privacy van de ketenpartners of hun cliënten. Berichten moeten dus goed³ versleuteld zijn en alleen te ontsleutelen door de bedoelde ontvanger(s) of tenminste een vertrouwde groep ketenpartners.

Ook is het mogelijk dat binnen de organisatie van een ketenpartner slechts enkele gebruikers toegang hebben tot de gecommuniceerde berichten, of dat medewerkers van de ketenpartner verschillende rollen binnen een transactie innemen.

Voorbeeld: bij een ZA is het denkbaar dat slechts enkele medewerkers inzage mogen hebben in de gegevens die van andere ZA's worden ontvangen, terwijl andere medewerkers alleen toegang hebben tot de eigen gegevens om op te leveren op verzoek. Duidelijk is dat een BJz een andere rol heeft in een transactie dan een ZA en het is dus ook zaak dat ZA's zich niet (abusievelijk of anderszins) kunnen uitgeven voor elkaar of het BJz.

- **Betrouwbaarheid**

Van een ontvangen bericht moet met zekerheid te zeggen zijn van wie het afkomstig is, om er in het eigen proces acties aan te kunnen koppelen.

Tezamen betekenen deze eisen dat het berichten verkeer buiten het applicatiedomein van Talaris versleuteld moeten worden behandeld, alleen geautoriseerde toegang mogelijk moet zijn tot specifieke gegevens en dat er een methode voor authenticatie op Talaris nodig is die kan worden vertaald naar alle soorten gebruikers van het systeem (zie ook Rapportage en beheer).

Rapportage en beheer

In de hierboven beschreven ketens zijn veel systemen betrokken en komen tal van interfaces tussen deze systemen tot stand. Om deze samenwerking mogelijk te maken is het nodig dat er goede rapportage- en beheersmogelijkheden zijn voor de systemen en interfaces.

Daarbij is het voor alle betrokkenen (management van eindgebruikers, beheerders, Uitweg organisatie, ontwikkelaars van ketenpartners, bouwer van Talaris) relevant om bepaalde informatie over de stroom van berichten op te kunnen vragen. Ook is het mogelijk, dat men in ketens informatie over de communicatie met een client wil delen.

Voorbeeld: een BJz wil graag weten hoeveel gebruik gemaakt wordt van de communicatiemogelijkheden en wil daar graag per kwartaal overzicht op. De ontwikkelaar van software bij een ketenpartner wil, bij wijzigingen aan de software, graag inzicht of de berichtenstroom nog correct functioneert en wil tests kunnen runnen tegen de communicatie-software zonder daarmee de productieomgeving te verstoren. Uitweg wil bij problemen die mogelijk ontstaan bij de communicatie inzicht kunnen verkrijgen in waar het probleem optreedt.

Hierbij spelen een aantal extra issues:

- **Toegang**

Niet alle partijen hebben dezelfde toegangsrechten tot informatie in transacties. Waar een systeembeheerder alleen hoeft te weten of alle software naar behoren werkt en of communicatie mogelijk is, wil het management meer weten over de inhoud en aantallen van verzonden berichten, terwijl eindgebruikers of systemen toegang tot alle gegevens in berichten nodig hebben. De bouwer van de software heeft mogelijk toegang nodig tot metadata over de berichten die juist weer niet zinvol is voor de andere betrokkenen.

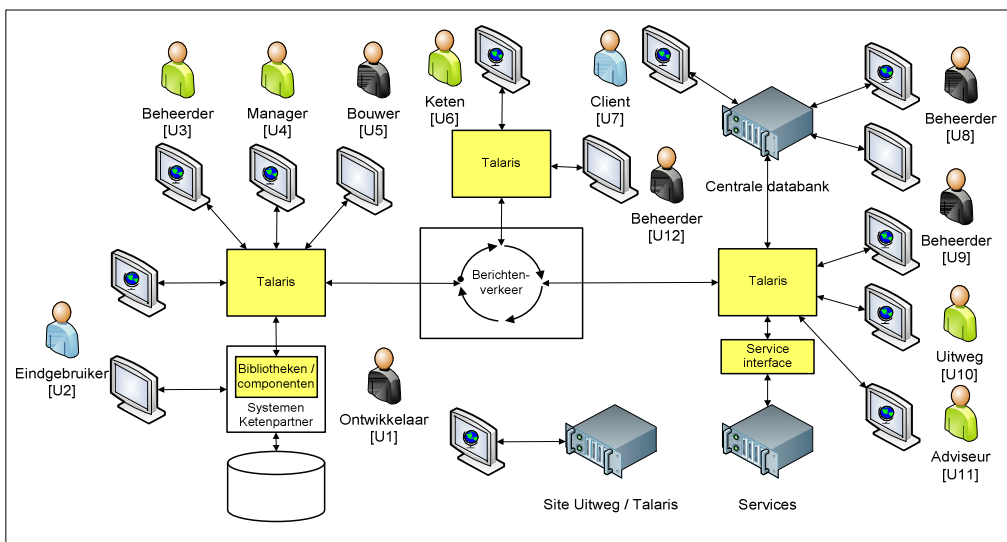
- **Distributie van informatie**

Afhankelijk van het soort informatie dat toegankelijk moet zijn, moeten er gegevens over het berichtenverkeer centraal, over alle partijen of bij een specifieke partij worden geregistreerd. De distributie van deze informatie vormt een aparte berichtenstroom, die geheel intern is aan het Talaris platform.

De behoefte aan rapportage en beheer levert extra informatieproducenten en consumenten op en het totale plaatje komt er zo uit te zien:

³ Volgens gangbare normen "goed" versleuteld, dus bijvoorbeeld op basis van SSL met signed certificates (TTP) en een sterk versleutelingsalgoritme. Of over een sterk beveiligde VPN connectie met een andere Talaris direct communiceren. Tal van mogelijkheden zijn hier denkbaar, een specifieke oplossing heeft (nog) niet direct de voorkeur.





Figuur 2

Beginnend bij de 'Ontwikkelaar', met de klok mee:

- De **ontwikkelaar** [U1] van systemen bij de ketenpartner maakt gebruik van *bibliotheken of componenten* voor het eigen platform. Downloads, tekst en uitleg en mogelijk zelfs communitytools (forum, blog, etc.) zijn beschikbaar op de *site van Uitweg / Talaris*.
- De **eindgebruiker** [U2] is dezelfde als die (beide) in Figuur 1.
- De (systeem)**beheerder** [U3] bij de ketenpartner wil inzicht kunnen verkrijgen in de toestand en het functioneren van geïnstalleerde software in de eigen omgeving en mogelijk aanpassingen aan de configuratie kunnen maken.

Voorbeeld: het Talaris platform zou kunnen vereisen dat op een poort in de firewall van de ketenpartner andere locaties toegang wordt verschaft tot de server. Deze poort zou configurabel moeten zijn zonder tussenkomst van de bouwer van Talaris en de systeembeheerder moet zelf kunnen vaststellen dat de aanpassing succesvol is, zonder impact op lopende transacties.

- Een **manager** [U4] bij de ketenpartner wil overzicht hebben op het verloop van berichten, de activiteit van de eigen eindgebruikers en mogelijk informatie met betrekking tot licenties of andere operationele kosten van de Talaris software (hosting, etc.)
- De **bouwer** van Talaris [U5] moet de benodigde software bij de ketenpartner kunnen installeren en beheren. Deze interactie met het platform gaat verder dan de interactie die de eigen beheerder [U3] met het systeem moet hebben. In de figuur is deze interface niet voorgesteld als een web interface, omdat het voorstelbaar is dat een deel van de beheerstaken alleen maar kunnen worden uitgevoerd als het platform zelf niet operationeel is.
- Een vertegenwoordiger van de **keten** [U6] als geheel wil overzicht hebben op het verloop van berichten in de gehele keten. Vergelijkbaar met de informatie die de managers [U4] nodig hebben, maar duidelijk niet per sé dezelfde subset van metadata.
- **Clienten** [U7] van ketenpartners kan beperkte toegang tot de eigen informatie worden gegeven. Ze treden hierbij effectief ook op als een ketenpartner, maar bezitten natuurlijk geen eigen systemen of organisatie.
- Ook op een centrale databank is natuurlijk een **applicatiebeheerder** [U8] nodig.
- Daarnaast is het nodig dat de bouwer van Talaris een rol van **beheerder** [U9] op zich kan nemen over het gehele Talaris-netwerk op basis waarvan transactie-, database- en systeemupdates kunnen worden verspreid.

Voorbeeld: er zijn verschillende scenario's waarin centrale updates nodig/gewenst zijn. Denk aan het bijwerken van transacties in een specifieke keten, of in meerdere ketens ineens om te kunnen omgaan met een verandering van protocol of om gebruik te kunnen maken van nieuwe functionaliteit. Of een verzoek om onderhoudsinformatie te verzamelen van verschillende systemen in het Talaris platform. Of de distributie van een update aan software van het Talaris platform zelf⁴.

⁴ Hierbij is het belang dat dergelijke updates kunnen worden gemaakt, niet dat deze allemaal op dezelfde wijze beschikbaar worden gemaakt. Zo is het goed mogelijk dat



- Een vertegenwoordiger van **Uitweg** [U10] moet, vergelijkbaar met de rol van de vertegenwoordiger van de keten [U6], inzicht kunnen krijgen in het functioneren van de verschillende ketens, transacties en evt. licenties.
- **Adviseurs** [U11] die nieuwe ketens definiëren en inrichten op het Talaris platform moet centraal toegang worden verschaft om nieuwe transacties te implementeren en te testen en uiteindelijk uit te laten rollen naar nieuw geïnstallerede Talaris ketenpartners. Naast het soort functies dat ook de beheerder [U9] nodig heeft, moet de adviseur ook beschikken over instrumenten om zonder intieme kennis van de techniek aanpassingen of uitbreidingen aan transacties te maken.
- Tenslotte is in de figuur ook de rol van de bouwer als **beheerder** [U12] van Talaris-locaties voor de keten aangegeven. Deze rol is vergelijkbaar met de rol van bouwer [U5] en beheerder [U9] zoals hierboven beschreven.

Zoals uit de figuur blijkt, is het Talaris platform op drie *soorten* locaties toegankelijk. Bij de ketenpartner hebben gebruikers [U1] tot en met [U5] toegang tot Talaris. De keten als geheel [U6] heeft mogelijk op een andere locatie toegang tot Talaris. En gebruikers [U7] tot en met [U8] hebben toegang tot Talaris op een locatie die kan worden gedeeld tussen Uitweg en de providers van infrastructuur en ondersteuning.

Het centrale 'berichtenverkeer' suggereert dat ook Talaris van een onderliggende berichtenservice gebruikt maakt en dat Talaris met name de service access points en de gebruikersinterfaces hieraan toevoegt, samen met transactielogica en servicekwaliteiten zoals autorisatie, authenticatie, etc.

Kwaliteit

Aan het berichtenverkeer en de Talaris software moeten bepaalde kwaliteitseisen kunnen worden verbonden, over het algemeen *niet-functionele* eisen:

- **Snelheid**
Over alle interfaces moet gelden dat duidelijk is hoe lang een respons ten hoogste op zich laat wachten. De maximale wachttijd voor een respons zou overeen moeten stemmen met de verwachtingen die passen bij een bepaald type interface. Daarnaast moet vooraf duidelijk zijn welke eisen door toepassingen van ketenpartners en de eindgebruikers worden gesteld voor de totale duur van een transactie en de berichtenuitwisselingen waaruit die transactie bestaat. Dit verschilt per keten, maar wordt natuurlijk beperkt door de technische mogelijkheden.
Voorbeeld: de wachttijd voor een call op een .Net-klasse is naar verwachting aanmerkelijk lager dan de wachttijd op voor een SOAP-service. Bij een SOAP-service zal e.e.a. zeker binnen de normale sessie-timeout van de client en server moeten worden afgehandeld, wat weer sneller is dan de verwachte wachttijd voor de reactie op een bericht dat in een wachtrij wordt gezet of in een mailbox wordt gedeponeerd.
- **Duurzaamheid**
Als een transactie eenmaal is geïmplementeerd en werkt, zouden toekomstige uitbreidingen en veranderingen geen gevolgen moeten hebben voor partijen die niet mee willen (of kunnen veranderen). Of het nu gaat om uitbreidingen of aanpassingen aan de onderliggende software, of uitbreidingen of aanpassingen aan de proceslogica, lopende transacties en werkende ketens zouden er niet onder moeten leiden. Natuurlijk is het mogelijk dat alle ketenpartners samen besluiten tot een aanpassing die er uiteindelijk toe leidt dat een eerdere transactie niet meer nodig is, maar dat doet niets af aan het voorgaande.
Voorbeeld: als de transactie uit de bovenstaande voorbeelden in eerste instantie wordt uitgevoerd zonder de terugmelding van de resultaten naar het BJz, maar in een later stadium tot deze uitbreiding wordt besloten, dan moeten berichten die al onderweg zijn (binnen de tijd die daarvoor staat) gewoon worden verwerkt, zonder het gehele netwerk stil te leggen. Daarnaast moet het zo zijn dat ketenpartners die geen behoefte hebben aan deze aanpassing de oude transactie kunnen blijven gebruiken, als de andere ketenpartners daarmee instemmen. Als hiervoor wijzigingen aan het Talaris platform moeten worden gemaakt, moeten deze wijzigingen kunnen worden doorgevoerd zonder effecten op lopende berichten en transacties.
- **Openheid**
Hoewel het interne protocol van Talaris gesloten kan (en waarschijnlijk moet) zijn, is het wenselijk dat het platform zelf zo toegankelijk mogelijk is. Dit vertaalt zich naar een breed aanbod van bibliotheken/componenten voor gebruik in software van ketenpartners, maar zou zich ook moeten uiten in een open interface (API) voor de aansturing van de toepassing. Functionaliteit zoals logging, rapportage en beheer zouden ook voor andere toepassingen beschikbaar gemaakt moeten worden.

voor systeemupdates van een bestaand update- of patchsysteem gebruik gemaakt wordt.

REQUIREMENTS

Onderstaande is een indeling naar systeemonderdelen en interfaces van de functionaliteit zoals die in de systeembeschrijving is gegeven. De punten zijn volgende de MoSCoW methodiek gewaardeerd.



Code	Requirement	
T1 Engine		
T1.1	Talaris moet gebruikers ⁵ kunnen authenticeren	M
T1.1.1	Talaris moet een gebruiker kunnen authenticeren op basis van gebruikersnaam + wachtwoord.	M
T1.1.1.2	Authenticatie moet kunnen worden beheerd in Talaris.	S
T1.1.2	Talaris moet een gebruiker kunnen authenticeren op basis van bestaande Windows credentials.	C
T1.1.3	Talaris moet een gebruiker kunnen authenticeren op basis van bestaande Linux credentials.	C
T1.2	Talaris moet autorisatie kunnen verlenen voor de eigen functies	M
T1.2.1	Alle ingebouwde functies van Talaris hebben een vastgestelde autorisatie	S
T1.2.2	Alle stappen in een transactie hebben een vastgestelde autorisatie (of zijn beschikbaar voor allen)	M
T1.2.3	Ongeautoriseerde handelingen in Talaris worden gelogd en de niet geautoriseerde gebruiker wordt gemeld dat deze niet geautoriseerd is.	C
T1.3	Talaris heeft een Access control mechanisme voor het toelaten van gebruik van functies	M
T1.3.1	Gebruikers worden per KP georganiseerd in gebruikersgroepen, die toegang worden verleend tot rollen.	M
T1.3.2	Functies worden gekoppeld aan (één of meer) rollen die ze mogen uitvoeren.	M
T1.3.2.1	Welke rol welke functie mag uitvoeren is te beheren in Talaris.	C
T1.3.2.1	Welke stap in een transactie mag worden uitgevoerd door welke rol is te configureren in de definitie van een transactie.	S
T1.3.3	Gebruikers bij een KP hebben alleen toegang tot stappen van transacties waartoe die KP toegang heeft.	M
T1.3.3.1	Welke KP's toegang hebben tot een transactie is te configureren in de definitie van een transactie.	S
T1.4	Berichtenverkeer in Talaris is geordend in transacties die worden gedefinieerd als XML-documenten met bijbehorende schema's die de vorm van de berichten vastleggen.	M
T1.4.1	Nieuwe transactietypes kunnen aan Talaris worden toegevoegd.	M
T1.4.2	Bestaande transactietypes kunnen worden gescheduled voor verwijdering.	M
T1.4.2.1	Als een transactietype is gescheduled voor verwijdering kunnen er geen nieuwe transacties van worden gestart.	C
T1.4.3	Transacties hebben een voorgedefinieerde time to live.	M
T1.4.3.1	De ttl van een transactie is aan te passen in de definitie van een transactie.	C
T1.4.3.2	Iedere stap van een transactie heeft een eigen ttl	C
T1.4.3.2.1	Als een ttl wordt overschreden, wordt de stap afgebroken en op basis van de transactielogica een andere stap gestart.	C

⁵ NB: 'Gebruikers' zijn in deze context vaak andere programmaonderdelen of systemen die een bepaalde identiteit aannemen, niet per sé gebruikers van vlees en bloed.



T1.4.4	Transacties bestaan uit stappen die onder een combinatie van condities kunnen worden gestart.	S
T1.4.4.1	Stappen kunnen worden gestart op basis van het aantal ontvangen berichten in de vorige stap.	C
T1.4.4.2	Stappen kunnen worden gestart op basis van de evaluatie van een Xpath expressie over de inhoud van ontvangen berichten.	C
T1.4.4.3	Stappen kunnen worden gestart op basis van de tijd die is verstreken sinds de vorige stap werd gestart.	C
T1.4.5	Stappen kunnen worden verbonden aan een gebruiker die daarna de vervolgstappen ook moet uitvoeren binnen de rol voor die stap –of- aan een rol, waarbij alle gebruikers die die rol kunnen aannemen vervolgstappen kunnen uitvoeren.	S
T1.4.5.1	Gebruikers moeten een specifiek aan hen toegewezen stap kunnen teruggeven aan de rol waarbinnen ze hem hebben gestart.	C
T1.5	Berichtenverkeer n.a.v. transactiestappen heeft de vorm van XML documenten.	M
T1.5.1	De toegestane structuur van berichten wordt gevalideerd tegen schema's (XSD) door Talaris en kan verschillen tussen verschillende transacties	S
T1.5.2	Transactietypen kunnen in meerdere versies aan Talaris worden toegevoegd. Voor een nieuwe versie van een transactietype is dezelfde access control van toepassing als op de vorige	C
T1.6	Rapportage en logging over berichtenverkeer is op een API beschikbaar	W
T1.7	Een gebruiker kan een overzicht worden geboden van beschikbare transacties (en stappen) op basis van access control.	M
T1.7.1	Het overzicht van beschikbare stappen kan worden gefilterd op lopende of nieuwe transacties.	S
T1.7.2	Het overzicht van beschikbare stappen kan worden geordend op urgentie (resterende TTL van stap of transactie)	C
T1.8	Een gebruiker kan een stap van een transactie starten.	M
T1.8.1	De gebruiker krijgt feedback over het succes van het starten van een stap, of over de reden voor het evt. niet starten	M
T1.8.2	Talaris is volledig verantwoordelijk voor het bewaren van de state van een transactie, iedere operatie met een gebruiker moet atomair en in principe stateless zijn.	S
T1.8.3	Een gebruiker kan het starten van een stap valideren tegen Talaris, zonder de stap daadwerkelijk te starten, om te verifiëren dat de geboden parameters valide zijn.	S
T1.8.3.1	Indien de gestarte stap niet valide is (of om andere redenen niet zou kunnen starten), wordt dit aan de gebruiker gemeld.	S
T1.9	Een gebruiker kan een lopende transactie annuleren.	M
T1.9.1	Talaris stelt alle betrokkenen bij de transactie op dat moment op de hoogte van het annuleren. Alle gebruikers die op respons wachten in hun respons en alle gebruikers die berichten verwachten in een annuleringsbericht.	S
T1.10	Talaris kan een unieke identifier genereren voor unieke identificatie over heel Talaris (anonimisering)	C
T1.10.1	Talaris herkent dergelijke identifiers bij validatie en verzekert dat ze bestaan als ze worden gebruikt.	C
T1.11	Talaris kan binaire content in berichten verzenden (berichten zijn dus niet beperkt tot platte xml-content).	M
T1.11.1	Aan de maximale omvang van berichten in Talaris wordt in principe geen bovengrens gesteld. Indien nodig is het mogelijk dat Talaris content daarom in onderdelen verzend, maar dergelijke opdeling is voor gebruikers transparant.	S



T1.12	Talaris kan patches (op zichzelf en de eigen database) toepassen	M
T1.12.1	Als een patch moet worden toegepast, kunnen gebruikers hun handelingen afmaken zonder verdere restricties, maar kunnen geen nieuwe handelingen worden gestart totdat de patch is voltooid.	S
T1.12.2	Als de patch niet voltooid kan worden, wordt deze teruggedraaid.	C
T1.12.3	Lopende transacties worden nooit als gevolg van een patch afgebroken. Voor de geldige duur van de transactie blijft het mogelijk deze af te ronden ⁶ .	S
T1.12.4	Patches op de database die geen invloed hebben op bestaande functionaliteit (zoals het toevoegen van nieuwe transacties) moeten kunnen worden toegepast zonder het functioneren van Talaris te onderbreken.	C
T1.12.5	Voor een patch is geen menselijke tussenkomst nodig, Talaris beschikt over de mogelijkheid willekeurige onderdelen te stoppen, patchen en weer te starten. (tenzij het natuurlijk nodig is systeemonderdelen buiten Talaris te herstarten)	C
T1.12.6	Met het patchen van Talaris kunnen ook andere systeemonderdelen worden bijgewerkt, zoals ondersteunende platformonderdelen en -services.	W
...		
T2 Transport		
T2.1	Talaris moet berichten die het gevolg zijn van transactiestappen versleuteld naar de desbetreffende KP kunnen transporteren.	M
T2.1.1	Het transport van berichten moet versleuteld gebeuren, op een wijze waarop alleen ontsleuteling door de bedoelde ontvanger(sgroep) mogelijk is.	S
T2.1.1.1	Naast versleuteling op basis van een vaststaand sleutel/certificaat moet aan een bericht nadere versleuteling in de vorm van een wachtwoord of certificaat kunnen worden toegevoegd.	C
T2.2	Talaris moet te verzenden berichten bufferen totdat deze verwerkt kunnen worden door de Talaris bij de bedoelde KP.	M
T2.2.1	Talaris moet kunnen rapporteren over berichten in de wachtrij [U3, U5]	S
T2.3	Talaris moet ontvangen berichten bufferen totdat deze verwerkt kunnen worden door de bedoelde gebruiker.	M
T2.3.1	Talaris moet kunnen rapporteren over berichten in de wachtrij [U3, U5]	S
T2.4	Indien berichten niet onmiddellijk kunnen worden verzonden, moet Talaris het een (instelbaar) aantal keren opnieuw proberen met een (instelbaar) interval.	M
T2.5	Talaris moet verschillende protocollen kunnen gebruiken voor het verzenden van berichten, al naar gelang beschikbaarheid bij de betrokken KP's	M
T2.5.1	Talaris moet berichten kunnen verzenden aan andere Talaris over https.	S
T2.5.2	Talaris moet berichten kunnen verzenden aan andere Talaris over een https op basis van 'firewall hole punching'	C
T2.5.3	Talaris moet berichten kunnen verzenden aan andere Talaris over	W

⁶ Hierbij is het natuurlijk mogelijk dat een transactie pas als gevolg van de patch kan worden afgerond, als de patch een bug fixt die het afronden eerder voorkwam. 1.12.3 heeft daarom vooral betrekking op patches die functionaliteit van Talaris veranderen t.o.v. eerdere specificatie, het volgens bestaande specificaties laten functioneren van Talaris valt hier niet onder. E.e.a. verandert niets aan de eis dat gestarte transacties altijd moeten kunnen worden afgerond als dat onder de geldende specificaties op het moment waarop een transactie werd gestart mogelijk zou moeten zijn.



	PGP-encrypted mail	
T2.5.4	Talaris moet berichten kunnen verzenden aan andere Talaris met gebruikmaking van AMQP	C
...		-
T3 Service interfaces		
T3.1	Talaris moet een virtuele ketenpartner aanbieden voor een test service	M
T3.1.1	Talaris moet een test service endpoint (op basis van SOAP) implementeren	M
T3.1.1.1	De test service moet alle relevante elementen van T1 implementeren en rapporteren over het succes van de uitvoering.	C
T3.2	Talaris moet een virtuele ketenpartner aanbieden voor de Zorg voor Jeugd service ⁷	C
T3.2.1	Een transactie voor het maken van meldingen op deze service moet worden ingericht	C
...		-
T4 Bibliotheken / Componenten		
T4.1	Talaris heeft een bibliotheek / set componenten voor gebruik van alle voor een gebruiker beschikbare functies en stappen voor gebruik in .Net-toepassingen.	M
T4.2	Talaris heeft een bibliotheek / set componenten voor gebruik van alle voor een gebruiker beschikbare functies en stappen voor gebruik in PHP-toepassingen.	C
T4.3	Talaris heeft een bibliotheek / set componenten voor gebruik van alle voor een gebruiker beschikbare functies en stappen voor gebruik in SOAP-capable toepassingen.	S
S1 Centrale databank (CD)		
S1.1	Talaris heeft een CD die als een anonieme KP deel kan nemen aan transacties, voor automatische opslag van entiteiten uit berichten.	W
S1.1.1	Voor de CD is een speciale rol beschikbaar en kunnen identiteiten worden toegevoegd voor gebruik binnen specifieke transacties	W
S1.1.2	<i>// nadere definitie van het gedrag van de CD is nodig</i>	W
S1.2	De CD is database-agnostisch en kan op een willekeurige XML database (XML-enabled of native XML) worden geïmplementeerd.	W
...		
I User interfaces		
I1	Talaris heeft een webbased UI voor eindgebruikers [U2]	M
I1.1	[U2] kan alle voor deze [U2] beschikbare functies en stappen (als in T1) middels formulieren bedienen.	M
I1.1.2	Voor een transactie kunnen, op basis van de definitie van de transactie, templates voor alle bijbehorende formulieren automatisch worden gegenereerd.	S
I1.1.3	Talaris moet deze formulieren zelf kunnen hosten en verwerken.	S
I1.2	Nieuwe [U2] kan zich aanmelden.	C
I1.2.1	[U3] kan nieuw aangemelde gebruikers accepteren en rollen toekennen.	C
I1.2.2	[U2] kan hun persoonlijk gegevens (wachtwoord en evt. personalia) bewerken	C
I1.3	[U2] kan een visuele weergave (spoorkaart) krijgen van	W

⁷ De Zorg voor Jeugd service is nader gedocumenteerd als <http://producten.interaccess.nl/zvj/services/SignaleringWebService?wsdl>



	beschikbare stappen en positie in een transactie van lopende transacties.	
I1.4	[U3] kan stappen en functies voor rol A zichtbaar maken voor [U2] in rol B	W
I1.4.1	[U2] kan optioneel stappen en functies opsommen die niet beschikbaar zijn in de eigen rol, maar wel onder een andere rol.	W
I1.5	[U2] kan zich aanmelden met gebruikersnaam en wachtwoord	M
I1.6	[U2] kan een beschikbare rol selecteren	M
I1.6.1	[U2] kan een rol selecteren zonder opnieuw in te loggen	S
I1.6.2	[U2] kan meerdere rollen selecteren en bij het uitvoeren van een stap waarbij het niet evident is, de juiste rol selecteren	C
I2	Talaris heeft een webbased UI voor beheerders bij KP's [U3]	M
I2.1	[U3] kan vaststellen dat Talaris correct is geïnstalleerd	M
I2.1.1	[U3] kan de engine van Talaris starten en stoppen	M
I2.1.2	[U3] kan zien welke van de KP's (per transactie) kunnen worden bereikt.	S
I2.2	[U3] kan een geautomatiseerde testtransactie uitvoeren met een willekeurige KP die kan worden bereikt. Deze testtransactie wordt onder een speciale beheersrol uitgevoerd en vereist geen interactie aan de kant van de desbetreffende KP.	S
I2.2.1	[U3] kan een variant van de testtransactie uitvoeren die de [U3] van de desbetreffende KP een tegenstap beschikbaar maakt en de data van de eerste [U3] toont, ter bevestiging van correcte communicatie.	C
I2.2.2	De testtransactie test alle aspecten van de engine (T1) en rapporteert per requirement of eraan kan worden voldaan.	C
I2.3	[U3] kan een overzicht krijgen van alle geregistreerde [U2]	S
I2.4	[U3] kan een overzicht krijgen van recente (instelbaar) succesvolle logins door [U2] en de daarbij gekozen rollen	S
I2.4.1	Van specifieke transacties kan in de transactiedefinitie worden opgenomen dat dergelijk gebruik niet moet worden opgenomen in de overzichten onder I2.4	C
I2.4.2	Van specifieke gebruikersgroepen kan door [U3] worden aangegeven dat ze buiten de overzichten onder I2.4 moeten worden gehouden.	C
I2.5	[U3] kan alle instellingen die bij de rol van [U3] horen aanpassen	M
I2.5.1	// nadere specificatie van deze instellingen nodig	-
I2.6	[U3] kan de testservice transactie uitvoeren en rapporteren over het resultaat.	S
I3	Talaris heeft een webbased UI voor managers bij KP's [U4]	C
	// nadere specificatie van deze interface nodig	-
I4	Talaris heeft een UI voor beheer door de bouwer [U5, U9, U12]	M
I4.1	[U5, U9, U12] kan alle instellingen op Talaris wijzigen, inclusief instellingen voor [U3] (I2.5)	S
I4.2	[U5, U9, U12] kan alle vormen van logging beheren	M
I4.2.1	Logging kan worden uitgevoerd naar externe databases (formaat nader te bepalen)	S
I4.2.2	Logging kan worden gereduceerd tot recente entries, op datum	S
I4.2.3	Logging kan worden gefilterd op relevante attributen (type melding, voorkomen termen in log entries, periode)	C
I4.2.4	Loggin kan worden doorgevoerd op in het systeem bestaande	C

	logging faciliteiten	
I4.2.4.1	Loggin kan worden doorgevoerd op het Windows eventlog	C
I4.3	[U5, U9, U12] kan van alle prerequisieten voor Talaris verifiëren dat ze correct functioneren en door Talaris kunnen worden benut	S
I4.4	Individuele componenten van Talaris kunnen worden geïnstalleerd, verwijderd, gerepareerd.	M
I4.5	[U5] kan patches en updates (zoals bedoeld onder I8.2) accorderen of weigeren. De verzender wordt hiervan op de hoogte gesteld.	C
I5	Talaris heeft een webbased UI voor de gehele keten [U6]	C
	// nadere specificatie van deze interface nodig	-
I6	Talaris heeft een webbased UI voor klanten van KP's [U7]	W
	// nadere specificatie van deze interface nodig	-
I7	Talaris heeft een webbased UI voor beheer van CD's [U8]	W
	// nadere specificatie van deze interface nodig	-
I8	Talaris heeft een webbased UI voor beheer door de bouwer [U9]	S
I8.1	Deze interface biedt [U9] op afstand alle functionaliteit die I4 lokaal biedt, op basis van authenticatie van de beheerder bij een specifieke ketenpartner, waar mogelijk.	S
I8.2	[U9] kan systeem- en databaseupdates aanbieden die lokaal kunnen worden toegepast of met Talaris kunnen worden verzonden aan andere (groepen) ketenpartners om daar te worden toegepast.	S
I9	Talaris heeft een webbased UI voor beheer door Uitweg [U10]	M
I9.1	[U10] kan nieuwe transacties invoeren bij KP's waarop de transactie van toepassing is.	
I9.2	[U10] kan rollen aanmaken en toekennen, zoals per I2.5, op basis van authenticatie van de beheerder bij een specifieke ketenpartner	C
I9.3	[U10] kan een overzicht krijgen van alle berichtenverkeer (echter niet de inhoud van het verkeer) per transactietype.	
I10	Talaris heeft een webbased UI voor beheer door adviseurs [U11]	C
	// nadere specificatie van deze interface nodig	-
...		

