

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

1	Algemeen	Antwoord
1	<i>Wat is uw beeld bij het scheiden van de werkzaamheden in meerdere kavels.</i> <i>Geef daarbij vanuit uw expertise uw beeld. Bent u bijvoorbeeld ontwikkelaar schets dan vanuit deze rol/expertise daarover uw beeld.</i> <i>Kunt u zich vinden in een verdeling? Hoe kunnen wij ervoor zorgen dat alle partijen zich conformeren aan 'hun deel' van de verantwoordelijkheid?</i>	1. Waar, ons inziens, goed over nagedacht moet worden is de migratie naar de integratie van nieuwe componenten. Wij zien bij diverse gemeentes dat de weg naar de common ground gedachten en de ZGW standaard niet in een keer realistisch is. Daarom helpen wij diverse gemeentes nu ook met een brug tussen de nieuwe en de oude wereld. In de verdeling van de componenten is ons inziens nu nog te weinig rekening gehouden met deze transitie. Waardoor het vaak niet mogelijk zal zijn om nieuwe componenten toe te voegen in een bestaand landschap. 2. De opsplitsing in de genoemde kavels is te begrijpen vanuit het oogpunt van flexibiliteit over alle aangesloten Dimpact organisaties en herbruikbaarheid van componenten. Wij zijn van mening dat dit ook op andere manieren te organiseren is. SaaS-toepassingen, waaronder die van ons, worden uitgesloten door de kavelverdeling waarbij volgens kavel 2 componenten via Haven gehost dienen te worden te hanteren. SaaS-toepassingen worden veelal door de leverancier zelf gehost en dit past niet binnen het genoemde in kavel 2. In lijn met de Common Ground visie zullen er op niveau 1+2 registercomponenten ontwikkeld worden voor het ontsluiten van een bron en op niveau 4+5 gebruikerstoepassingen. Hosting zou ook in deze context georganiseerd kunnen worden waarbij het ook mogelijk wordt dat een SaaS leverancier de diensten zelf host en beschikbaar stelt. 3. Wij zien onszelf primair als ontwikkelaar en systeemintegrator, en regelen in de regel zelfstandig met een derde partij de hosting van de door ons ontwikkelde oplossingen. Zo kunnen wij een volledige ontzorging bieden. Het uitgangspunt wat Dimpact hanteert om verschillende componenten door verschillende partijen te laten ontwikkelen onderschrijven wij ook. Een variatie aan ontwikkelaars en componenten zorgt ervoor dat een palet van oplossingen mogelijk is. Echter, als systeemintegrator maakt de gekozen aanpak het kunnen afsluiten van een SLA een stuk complexer. De componenten waar wij onze oplossingen uit opbouwen zijn in andere situaties door onszelf gekozen en daar nemen wij dan ook de verantwoordelijkheid voor. Daar waar onze klant vereist dat een ander component ontwikkeld door een derde partij ingezet moet worden ligt in onze ogen de verantwoordelijkheid van deze keuze bij de klant, niet bij de integrator. Wij zouden aanraden om als de behoefte is om de verantwoordelijkheid bij de systeemintegrator neer te leggen ook de ruimte te geven dat de systeemintegrator 'the best tool for the job' kan inzetten, wat soms anders kan zijn dan hoe Dimpact dat ziet. Zo niet, dan moet er geaccepteerd worden dat niet alles valt binnen de scope van de systeemintegrator. Ook vinden we dat het eigenlijk aan de systeemintegrator is om de keuze te maken over welke hostingpartij het meest geschikt is. Des de meer er wordt voorgeschreven, des de meer de keuzes worden beperkt. Ook hier kiezen wij liever voor 'best supplier for the job' afhankelijk van de functionele en niet functionele eisen. Des de meer vooraf door de klant bepaalt wordt des de meer dit ook de verantwoordelijkheid van de klant wordt. In onze ervaring bij andere vergelijkbare casussen leiden dit soort

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	beperkingen in de regel tot een sub-optimale dienstverlening. 4. De verdeling van de verschillende werkzaamheden over meerdere kavels lijkt ons een goede keuze. Door een duidelijke scheiding in taken, rollen en verantwoordelijkheden creëert u een situatie waarin enerzijds specialisten ingezet worden in het vakgebied waarin zij goed zijn en anderzijds voorkomt u een soort van leveranciersafhankelijkheid. Doordat de kavels enige raakvlak met elkaar hebben, creëert u wel een spanningsveld tussen de leveranciers van de verschillende kavels doordat belangen van leveranciers, gebruikers en leden tegenstrijdig kunnen zijn. Met de juiste regie vanuit Dimpact kan er gestuurd worden op functionerende ketens. De uitdaging waar u in deze opzet voor staat is ervoor zorgen dat alle partijen zich conformeren aan hun verantwoordelijkheid. Dit heeft niet alleen betrekking op de leveranciers, ook de deelnemende gemeenten dienen hierin hun verantwoordelijkheid te nemen. Wij adviseren u om de klantbehoefte (de eindgebruiker) centraal te stellen. Dit gaat hier niet alleen om de belangen van de deelnemende gemeenten, maar juist ook om de belangen van de inwoners van die gemeenten. Deelnemende partijen dienen te allen tijde het belang van de klant te dienen. Dit betekent dat er wordt gedacht en gekeken naar passende oplossingen in plaats van oorzaken. Hierin ligt er op strategisch/tactisch gebied een rol weggelegd voor Dimpact en op tactisch/operationeel gebied voor de deelnemende leveranciers. 5. Vanuit onze ervaring als system-integrator hebben wij een positief beeld bij het scheiden van het werk in meerdere kavels. Uiteraard merken we hierbij op dat we in de praktijk zien dat de granulariteit tussen de kavels vaak net wat anders ligt dan de hier geschetste uitgangspunten, maar dat zijn wat meer nuance punten die we bij onderstaande vragen en tijdens een eventueel gesprek graag toelichten. We merken dat wij een breed en pro-actief beeld hebben bij de invulling van de rol van system-integrator zowel naar binnen toe (richting de afnemende gemeente) met betrekking tot support, als naar buiten toe met betrekking tot het ondersteunen van andere partijen (andere leveranciers of partijen). Dat vraagt om een serieuze tijdsinvestering en duidelijke lijnen. Zeker als er bij overige kavels voor meerdere leveranciers wordt gekozen is het belangrijk om één helder aanspreekpunt te hebben, we zouden daarom voorstellen om voor dat kavel wel degelijk voor één primaire leverancier te kiezen waarbij er een tweede leverancier klaar zou kunnen staan als uitvalsmogelijkheid. Wel merken we op dat er binnen het kavel development ook nog een gedachte loopt om ontwikkeling en code-beheer uit elkaar te trekken. Dat is een complex vraagstuk, waarbij het de vraag is of daar niet te veel wordt geregeld binnen één kavel. We hebben onze gedachten daarover toegelicht bij de desbetreffende vragen, maar adviseren om hiervoor een separaat gesprek te voeren met de partijen die op dit moment voor Dimpact ontwikkelen en beheren en de daarbij behorende ervaringen op te halen. 6. Omdat de verkaveling voortkomt uit de wensen en doelstellingen van Dimpact, zou elke leverancier zich in de basis moeten kunnen vinden in de huidige opzet. Vanuit onze huidige klantenbestel kunnen we wel
--	--

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		adviseren wat we nog meer in de markt zien. Zo gebeurt het vaak dat wat nu kavel 1 en 2 is als een gecombineerde verantwoordelijkheid wordt uitgevraagd. Door de “run” verantwoordelijkheid in de volledige stack te beleggen bij één partij die tevens ook de regie voert over de andere ketenpartijen, kan Dimpact maximaal ontzorgd worden. Er zijn overigens ook voorbeelden waar deze verantwoordelijkheden wel gescheiden zijn. Dit kan goed werken maar houd er dan wel rekening mee dat meer partijen leidt tot meer afstemming en dus complexere processen. 7. Deze indeling van de kavels lijkt ons vanuit onze expertise logisch. Het is verstandig om het systeemintegratiekavel te gebruiken voor een partij die de regie voert over alle interacties tussen partijen en gemeenten, en tevens ook de overall service level beheert en monitort. Ons advies: • kavel 1: gebruik één partij die de centrale interacties voert tussen alle partijen en gemeenten. Dit is één partij die alles regelt; • kavel 2: gebruik hier één partij die de infrastructuur op basis van Common Ground beheert en monitort, inclusief de onderliggende publieke cloud resourcing; • kavel 3: gebruik hiervoor de best passende partij per deelproject ter vervanging van de onderdelen van de Atos e-Suite. Kortom: gebruik dus voor kavel 1 één partner en voor kavel 2 één partner. Alleen op kavel 3 adviseren wij u te werken met verschillende partners. 8. De verdeling in meerdere kavels is wat ons betreft een goede aanpak. Onze samenwerkstrategie is bestand tegen samenwerking tussen autonome partijen. Wij voorzien dan ook geen problemen met samenwerking in en tussen kavels. Om partijen te laten conformeren aan hun deel van de verantwoordelijk stellen we een samenwerking voor op basis van Holacracy en Scrum. 9. a. Wij antwoorden vanuit softwareontwikkelaar perspectief. Wij vinden het sterk om tenminste perceel 1 en 3 te hanteren. De vraag is of hosting een apart perceel dient te zijn, of dat dit bij Systeem Integrator profiel kan worden geïntegreerd. b. Wij vinden het opzetten van een poule voor perceel 3 duidelijk. Een duidelijke verdeling helpt marktpartijen een goede inschatting te maken of deze uitvraag bijdraagt aan hun strategie. Het kwantificeren hiervan maakt dit nog meer SMART
--	--	--

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

2	<i>Hoe kijkt u aan tegen de samenwerking binnen een vereniging als Dimpact alsmede de samenwerking in een (open source) community?</i>	1. Dimpact is in onze visie verantwoordelijk voor het invullen van de rol van Product Owner als het gaat om software ontwikkeling in opdracht van Dimpact. Wij leveren het Scrum team met daarin alle expertise die nodig is voor het productie waardig maken van de software en alle andere deliverables zoals opgesomd in het antwoord hierboven. Dit partnerschap vereist dat de Product Owner minimaal 2 dagen per week - liever meer - samen met de ontwikkelaars op de - virtuele - werkvloer samenwerkt. 2. In een dergelijke samenwerking is het zeer belangrijk om de verantwoordelijkheden van alle partijen goed te beschrijven en onder alle partijen te delen. Elke deelnemer moet weten wat van hem verwacht wordt. De marktconsultatie geeft nog te weinig inzicht hoe Dimpact dit voor zich ziet. Regievoering is in ieder geval zeer noodzakelijk om met elkaar binnen de gestelde afspraken en verantwoordelijkheden de gewenste doelen te behalen. 3. We juichen het toe dat Dimpact ervoor kiest om de kwaliteit te borgen door een opdeling in kavels waarin marktpartijen met ieder hun eigen expertise conform standaarden (Common Ground, maar hopelijk ook bijv. ISO 9001 en ISO 27001) acteren, zodat er een Single Point Of Contact is, er voldoende inhoudelijke expertise is bij onze counterparts en bovendien de eigen medewerkers van de gemeente zich bezig kunnen houden met hun kerntaken. 4. De kracht van samenwerking met een vereniging als Dimpact ligt volgens ons vooral op het gebied van versnelling, kwaliteit en herbruikbaar van componenten. Dankzij de regie van een organisatie zoals Dimpact kunnen wensen/behoefte vanuit de deelnemende gemeenten worden verzameld en vertaald naar een gemeenschappelijke/generieke behoefte. De kracht van open source ligt ook op het gebied van samenwerking, waardoor modules en componenten schaalbaar worden gerealiseerd. 5. Wij verwachten dat de coöperatieve aard van Dimpact uitstekend past bij een (open-source) community. Gemeenten ontwikkelen samen, met behulp van de markt software en bepalen de richting en de koers. Dimpact is al gewend om gemeenten samen te brengen, kennis te delen en regie te voeren. Dit is precies wat nodig is om Common Ground te versnellen. Dimpact kan hierbij het verschil maken (met behulp van leveranciers) door middel van het integreren van componenten tot werkende software. Want uiteindelijk vragen gemeenten om werkende software en niet om technische componenten, libraries, etc. Dimpact kan hierin een centrale rol spelen en de markt enerzijds verbinden met ontwikkeling van open-source software en anderzijds de behoefte van gemeenten adequaat invullen met moderne werkende software oplossingen waar tevens een strik om heen zit.
---	--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		6. Het is erg fijn om als software partij samen te werken met een vereniging die de belangen behartigt van een groot aantal afnemers. De vereniging kent de onderliggende afnemers goed, en kan daardoor met kennis en autoriteit zoeken naar overeenkomsten en verschillen tussen de afnemers. Common ground is bedoeld om meer uniformiteit aan te brengen in het software landschap. Hoe goed je oplossing ook is, er zullen altijd individuele afnemers zijn die een ander proces hebben of andere wensen hebben. Over het samenwerken met de open source community bestaat een hoop onduidelijkheid en hoge verwachtingen die lang niet altijd kloppen met de werkelijkheid. Vanuit onze ervaring is het belangrijk voor een vereniging als dimpact een aantal zaken te realiseren: - Het "open sourcen" van een software product er absoluut niet automatisch voor zorgt dat de open source community actief gaat bijdragen, controleren, corrigeren of input leveren. - Het hergebruiken van reeds beschikbare (common ground) open source componenten lijkt een grote tijdwinst, maar het vakgebied is nog dermate jong dat het hergebruik in de praktijk vooralsnog minimaal blijkt en soms zelfs onderzoek en integratietijd kost. - Zorgen voor adoptie in de open source community is een project op zich, waarbij het lanceren en in gebruik nemen van een kwalitatief goed product dat naar tevredenheid gebruikt wordt door de afnemers de belangrijkste eerste stap is. Het open source aspect van een common ground project is van enorme waarde bij de transparantie, auditability, documentatie en overdraagbaarheid van de geschreven softwarecomponenten. Ook is de inzetbaarheid van de software bij een onbeperkt aantal afnemers een belangrijk pluspunt. Door vanaf de start van het project alle componenten publiek en open source te ontwikkelen dwing je partijen om diepe inzage in het ontwikkelproces te geven en wordt het veel gemakkelijker om input en feedback te geven op werk van een andere partij. 7. Het is goed dat Dimpact vraagstukken van gemeenten centraliseert, beheert en tevens de regie voert voor het betrekken van de juiste partners uit de markt. Schaalbaarheid zorgt ervoor dat het voor alle gemeenten behapbaar wordt om gebruik te maken van open standaarden. Daarbij werkt dit op termijn kostenverlagend. De samenwerking in een open source community houdt de verschillende partijen op de kavel scherp om open en transparant te zijn. 8. Een open source community is enerzijds spannend aangezien marktpartijen vaak erg gehecht zijn aan intellectueel eigendom en hierop (financiële) bestaansrecht baseren. Daarom dient goed gekeken te worden naar verdienmodellen van marktpartijen. Hierin zijn meerdere mogelijkheden: - ga uit van (of ontwikkel) een referentie architectuur die te allen tijde basis vormt voor uitvragen - volledige betaling van softwareontwikkelingen: intellectueel eigendom komt volledig bij Dimpact te liggen. - licentiemodel zodat marktpartijen hun ontwikkelingen ook bij andere partijen buiten Dimpact kan toepassen. Mogelijk is dit niet gewenst voor Dimpact, dan dient hier duidelijk over te worden verschaft in aanbesteding. Het voordeel hiervan is dat Dimpact lagere initiële kosten heeft. Nadeel van een open source community is dat veiligheid en up to date zijn van data en modellen niet is belegd waardoor er veiligheids- en kwaliteitsrisico's kunnen ontstaan.
--	--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		U loopt het risico dat overdracht niet gaat zoals gehoopt. Bij een gedwongen samenwerking dient daarom duidelijk te zijn wat de verwachtingen zijn, hoe de overdracht zal plaatsvinden en dit dient ook duidelijk te worden (bij voorkeur gestandaardiseerd) te zijn bij een uitvraag onder perceel 3.
2	Kavel 1 Systeemintegratie	Antwoord
3	<i>Wat is uw overstijgende beeld voor invulling van deze opdracht?</i>	1. Het is een hele goede gedachtegang dat alle applicaties soepel samenwerken, dat staat immers centraal in het Common Ground principe. Echter is dat nu nog niet het geval. De vraag wordt dan ook, hoe gaan de oude applicaties, met de oude API's (zoals soap) met de nieuwe applicaties communiceren. Wat ons betreft hoort dit bij het integratie-vraagstuk en dat missen wij nog in deze kavel. Dit is wel onze expertise, en bij deze willen wij ook hierbij een helpende hand uitreiken om te helpen dit eventueel uit te werken. 2. Veelvuldig direct contact met de eindgebruikers van de software helpt succesvolle systeemintegratie. Uiteraard is governance tussen de verschillende betrokken partijen breder dan alleen contact met de eindgebruikers, maar helaas wordt deze laatste vaak onvoldoende ingevuld. Succesvolle systeemintegratie stelt een aantal voorwaarden aan de software. Alle data wordt bij voorkeur maar eenmalig opgeslagen en alle data heeft een component eigenaar die een API biedt aan andere componenten die deze data willen raadplegen. Vanwege de privacy gevoelige aard van de gegevens streven we naar maximale borging van de informatiebeveiliging door security by design toe te passen. Wij hanteren hier de volgende principes: • Vertrouwelijkheid, geef alleen toegang tot gegevens waarvoor de gebruiker is toegestaan • Integriteit, zorg ervoor dat er niet met gegevens wordt geknoeid of gewijzigd door onbevoegde gebruikers • Beschikbaarheid, zorg ervoor dat systemen en gegevens beschikbaar zijn voor geautoriseerde gebruikers wanneer ze die nodig hebben Omdat er verschillende partijen zijn betrokken bij de realisatie en gebruik van het portfolio is het naleven van bovenstaande principes niet eenvoudig en zal er een regiefunctie moeten worden ingevuld. Dit is de gezamenlijke verantwoordelijkheid van systeemintegratie, Dimpact en gemeenten. Om deze goed in te vullen zal er een langdurige relatie moeten worden opgebouwd waar naast formele relaties, ook informele relaties moeten worden opgebouwd op de verschillende werkvloeren.

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	3. Op termijn zullen de integraties volgens de Common Ground visie op basis van API's plaatsvinden. Zodoende is het relevant om met name in te zetten op een goede API gateway en ondersteuning van NLX. Vooralsnog heb je te maken met een transitiefase waarbij ook de 'oude' integratiestandaarden ondersteund moeten worden. 4. Wij leveren software in de vorm van SaaS-diensten. Door de voorwaarde dat de systeemintegrator geen eigen software mag leveren is het onwaarschijnlijk dat wij op dit kavel inschrijven. Wij zien meer in een model waarin leveranciers SaaS-diensten leveren inclusief hosting en support. Dit maakt het mogelijk de kwaliteit van dienstverlening hoog te houden omdat afstemmingsproblemen kleiner zijn (en binnen één bedrijf liggen) in tegenstelling tot een opdeling in drie kavels waarbij altijd minimaal drie leveranciers betrokken zijn. 5. Voor de deelnemende gemeenten en voor Dimpact is dit de kavel die het succes van de software en componenten bepaald. Als geen ander weten we dat het gebruik van standaard of ontwikkelde software geen garantie voor succes is zolang de implementatie ervan niet goed georganiseerd is. Een juiste implementatie vraagt kennis en inzicht van de organisatie, de te realiseren doelstellingen en de mogelijkheden (en onmogelijkheden) die de software en bijbehorende componenten bieden. Systeem integratie is wat dat betreft een vorm van maatwerk, waarin de beschikbare software optimaal ingezet wordt. 6. De systeemintegrator heeft de regiefunctie om een zo goed mogelijke dienst beschikbaar te maken voor de gemeenten. Wij vinden wel dat de samenwerking tussen systeemintegrator en ontwikkelaar het beste werkt als dit onder 1 dak valt, dus wij zijn het er niet mee eens dat 'onafhankelijkheid' (het niet zelf leveren van software) als uitgangspunt verstandig is. Juist het snel kunnen schakelen en goed op elkaar ingespeeld zijn is iets wat als het over organisaties heen ingericht moet worden lang niet altijd goed gaat. Wij zouden dan ook aanraden om meerdere beheer/systeemintegratiepartijen te contracteren (1 per dienst, ipv 1 voor alles), en te zorgen dat per beheerpartij verschillende oplossingen mogelijk zijn waarbij nauw met de ontwikkelende partij kan worden samengewerkt. 7. In de kavel waarin de Systeemintegratie en ondersteuning wordt geboden is de kavel waarin de toegevoegde waarde van de software tot uiting komt. De ontwikkelde software en componenten die op generieke wijze is ontwikkeld, wordt voor de
--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	deelnemende gemeenten ingericht. Ook biedt de systeem integratie ondersteuning bij het gebruik van de software en modules. Ons beeld is dat hier maatwerk gevraagd wordt, niet alleen bij de 'onboarding' van nieuwe software, maar ook bij de ondersteuning van de medewerkers van de deelnemende gemeenten. Iedere deelnemende gemeenten beschikt over een gebruikersorganisatie die haar eigen dynamiek, eisen en wensen ten aanzien van de implementatie hebben. De software kan vanuit technisch en gebruikersperspectief briljant zijn, maar valt en staat met een juiste implementatie. Om tot een succesvolle implementatie te komen is kennis nodig van de gewenste situatie, de huidige situatie en de wijze waarop er de gebruikersorganisatie (de deelnemende gemeente) met de vereniging samenwerkt om de gemeenschappelijke doelstelling(en) te realiseren. 8. Ook al hebben we ervaring met systeemintegratie, toch zien wij ons niet als een aanbevolen bedrijf voor deze kavel (zie ons antwoord op vraag 1, gedeelte expertise). Wel zijn wij van mening dat de partij die deze opdracht op zich zal nemen, zich ter dege zal moeten inwerken in de functionaliteiten, opbouw en documentatie van de verschillende ontwikkelde componenten. Deze partij zal een verbindende rol moeten spelen tussen de verschillende betrokken partijen. 9. De systeemintegratie is de meest complexe kavel en overziet de hele keten. In dit kavel worden partijen verantwoordelijk gehouden voor een werkende oplossing, in plaats van een werkende infrastructuur of een werkend component. Omdat wij ontwikkelen met het low-code platform van Mendix zijn wij gewend om niet alleen componenten te ontwikkelen, maar ook werkende software geïntegreerd met andere systemen en componenten. 10. Waar ik in mijn antwoorden spreek over "technisch governance", is mijn aanname dat deze belegd zou worden bij de partij die inschrijft op kavel 1. Een zorg daarbij zou zijn dat het daadwerkelijk realiseren van integraties met het bestaande software landschap, derde partijen en andere systemen ook een specialisme op zich is dat moet worden gecontroleerd en gemanaged, terwijl het inderdaad belangrijk is dat de partij met governance onafhankelijk is. Een andere zorg voor deze inschrijver is dat er zeer nauw samen moet worden gewerkt met de partij die het solution design uitvoeren voor de applicaties, daar de solution designers het meeste kennis hebben van het domein en de applicaties, en er voldoende kennis van de inhoud moet zijn om te kunnen bijsturen bij integratieproblematiek.
--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		11. Voor ons is de scope van kavel 1 op detailniveau nog onvoldoende helder. Als we kijken naar de specifieke vraag, zien we taken en afspraken die goed bij een systeembeheerder zouden passen, en taken die juist beter tot hun recht zouden komen bij een functioneel beheerder, die het een en ander coördineert en aanstuurt. Vraag 5 lijkt te wijzen op een product owner-achtige rol, die het een en ander samenbrengt. Specifiek genoemd staat een 24/7 beschikbaarheid bij verstoringen. Het lijkt ons dat dit bij incidentbeheer zeker ook voor de andere kavels en partijen geldt. Wellicht is het beter dat de ontwikkelaars van de producten de trainingen verzorgen, omdat hier de meeste inhoudelijke expertise ligt. 12. Zeer sterk concept door Common Ground (lees vaste referentie architectuur) principe te hanteren voor gemeenten. Wij voorzien hoge complexiteit in samenwerkingen als de uitgangspunten vooraf niet duidelijk zijn. Via API's en architectuur kunnen applicaties met elkaar gaan communiceren. Hierin dient de data laag gescheiden te worden van de applicatie laag zoals staat beschreven in common ground. Hoe de data toegankelijk wordt gemaakt maar ook beveiligd is, is een uitdaging.
4	<i>Hoe kijkt u aan tegen 'verplichte' samenwerking met andere partijen, bijvoorbeeld hosting of software-ontwikkelaars?</i>	1. Dat systemen goed samen kunnen werken en volgens standaarden zijn ontwikkeld zoals 'haven' is niets meer dan normaal in 2021. Daar hoort ook een goede samenwerking bij met andere partijen om bij uitdagingen samen tot een mooie oplossing te komen. In de gedachtengoed van Opensource zou het normaal moeten zijn dat je op verschillende vlakken samenwerkt. Vaak gestimuleerd vanuit de community soms als investering vanuit een leverancier. Zie onze actieve deelname op diverse slack kanalen. 2. Aangezien er binnen Common Ground software wordt ontwikkeld door verschillende leveranciers en verschillende opdrachtgevers is samenwerking een gegeven. Effectieve samenwerking tussen de verschillende partijen kan worden gefaciliteerd door de volgende spelregels: • Scrum demo's worden bijgewoond door alle betrokken stakeholders • Systeemintegratie, hosting en software-ontwikkelaars hebben allemaal toegang tot de gezamenlijke documentatie benodigd voor het beheer van de software • Systeemintegratie, hosting en software-ontwikkelaars delen een informeel communicatiekanaal zoals een slack-channel • Alle betrokken partijen kunnen agendapunten opvoeren voor het reguliere overleg Een kleine kanttekening bij bovenstaande: het is een valkuil om alle (technische) stakeholders deel te maken van het Scrum team. Alleen die personen die dagelijks aan het project / component werken nemen plaats in het Scrum team. Andere personen die deelactiviteiten uitvoeren wonen de demo bij en hebben regulier overleg met het Scrum team, maar blijven stakeholder en zijn geen leden van het Scrum team.

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	Vanuit onze rol als softwareaanbieder zien wij het volgende aandachtspunt bij samenwerking met een systeemintegrator. Als het om meldingen van gebruikers in een app (laag 4+5) gaat is vaak directe communicatie met de gebruiker nodig. Er moet bijvoorbeeld doorgevraagd worden op details van de melding, zonder tussenkomst van de systeemintegrator. 3. Een keuze voor 'verplichte' samenwerking betekent dat de klant (Dimpact) ook haar verantwoordelijkheid moet nemen dat dit tot een sub-optimale dienstverlening kan leiden. Het kan goed gaan, maar de mogelijkheden om als integrator te sturen en te switchen worden beperkt dus dit zal tot wrijving leiden. 4. Binnen ons vakgebied is dit gebruikelijk. Wij werken voor grotere organisaties waarbinnen sprake is van een bestaand ICT-landschap dat 'naadloos' met elkaar dient samen te werken. Dit geldt niet alleen voor de systemen, maar ook voor de medewerkers die in multidisciplinaire teams moeten kunnen werken. 5. Dit zou voor de partij die de rol van system integrator op zich zal nemen geen probleem mogen zijn. 6. We zijn gewend om met andere software leveranciers, cloud-leveranciers samen te werken. De oplossingen die we gebruiken zijn cloud-agnostisch. De basis van common ground is open source waardoor er sprake is van transparantie in de samenwerking. De openheid van open source componenten sluit tevens goed aan op de openheid van het Mendix low-code platform wat onze ontwikkelaars gebruiken om hun software mee te ontwikkelen. 7. In onze optiek kunnen partijen prima samenwerken onder de volgende voorwaarden: • de partij beschikt over de juiste expertise voor een bepaalde kavel; • er wordt gebruiktgemaakt van wereldwijd geaccepteerde open source standaarden (de nadere invulling daarvan voor Nederland vindt plaats via Haven/Common Ground); • er vindt uitgebreide documentatie plaats via portalachtige omgevingen. Zo is overdraagbaarheid geborgd. 8. Wij zijn zelf niet zo'n voorstander van een verplichting. Door te verplichten ontstaat er een risico dat er bij partijen onvoldoende motivatie is om producten en diensten te blijven innoveren. Bij elke opdracht zou er wederom voor de beste combinatie gekozen moeten worden. 9. Onze ervaring leert ons dat verplichte samenwerking, en overdracht van software vaak niet gemakkelijk gaat (not invented here syndrom). In een roulerende poule van perceel 3 wordt dit complexer. U dient in
--	---

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		nadere uitvragen duidelijk randvoorwaarden te scheppen zodat overdracht simpel gaat omdat standaard uitgangspunten worden gehanteerd door alle samenwerkingspartijen. Dit is de rol van de system integrator of Dimpact zelf.
5	<i>Hoe ziet u de rol van verbinder tussen alle gecontracteerde leveranciers? Hoe kan Dimpact borgen dat er een echte relatie ontstaat tussen eenieder? En dan niet de leverancier-klant relatie, maar die van partners?</i>	1. Is nog heel moeilijk aan te geven. Het feit dat Dimpact al de regie heeft genomen voor meerdere gemeentes is al positief. Verder ontstaat er partnerschap bij gezamenlijke transparante doelen en kan de olie daarvoor een gezamenlijke beloning zijn. 2. De relatie tussen de verschillende gecontracteerde leveranciers kan alleen tot een partnerschap uitgroeien door wederzijds voordeel op basis van wederzijds vertrouwen. Een hoger doel dat gedeeld wordt door alle leveranciers kan daarbij helpen. Het hogere doel is in deze situatie het klant eerst principe. Concreet betekent dit dat de tevredenheid van de klant - e.g. de gebruikers van de software en het platform - leidend is voor de leveranciers. Heel concreet betekent dit dat alle leveranciers worden betrokken bij de behaalde successen en ook daadwerkelijk de gebruikers ontmoeten. Zolang de echte gebruikers anoniem blijven voor de betrokken leveranciers is bovenstaande lastiger te bereiken. 3. Dimpact kan een partner-relatie bevorderen door ruimte te laten voor leveranciers, bijvoorbeeld door ook SaaS-diensten binnen de kavels mogelijk te maken. Als leveranciers vanuit hun eigen expertise en visie kunnen samenwerken met respect voor hun expertise en visie, zal de bereidheid om samen te werken groot zijn. 4. De uitdaging waarmee u in deze aanpak voor staat is de optimale samenwerking tussen Dimpact als regisseur, leveranciers voor de verschillende kavels en de (medewerkers van de) deelnemende gemeenten. Een partnerschap is niet iets wat je afspreekt, maar is iets wat je creëert of laat ontstaan door de juiste randvoorwaarden te scheppen voor de verschillende deelnemers. In de beschrijving van uw marktconsultatiedocument is er sprake van meerdere leveranciers voor de drie verschillende kavels. Daarbij is er ook sprake dat er verschillende modules of componenten ontwikkeld worden. Hierdoor kan de situatie ontstaan dat leveranciers zich focussen op dat deel waar zij verantwoordelijk zijn. Partnerschap gaat verder dan dat. Partnerschap betekent dat partijen elkaars belangen kennen, het gemeenschappelijke, overstijgende doel kennen, en zich inspannen om die doelstellingen te realiseren. Dit betekent dat de betrokken partijen goed geïnformeerd moeten zijn voor de te leveren (deel)producten en inzicht hebben in de mogelijkheden en onmogelijkheden van de oplossing en de te hanteren roadmap. Hierdoor kunnen verwachtingen gecreëerd en gerealiseerd worden.

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		5. Dit is afhankelijk van hoe Dimpact verwacht dat de rollen contractueel worden geregeld. Sluit Dimpact de contracten af met de verschillende soorten leveranciers en heeft de systemintegrator een coördinerende rol, of wordt aan de systemintegrator gevraagd om de contracten met ontwikkelaars en hosters te regelen? Echte relaties ontstaan over tijd bij een goede samenwerking en na opbouwen van wederzijds begrip. Dit kan alleen plaatsvinden als de verantwoordelijkheden glashelder zijn en alle partijen er op dezelfde manier in staan. 6. In een omgeving met meerdere leveranciers, meerdere organisaties en een veelheid aan stakeholders is verbinding misschien wel de grootste uitdaging waarmee u te maken krijgt. De deelnemende organisaties kennen bovendien een eigen dynamiek. Bovendien dienen de verschillende leveranciers samen te werken bij de ontwikkeling van componenten die gebaseerd zijn op een generieke behoefte. Hier ligt een natuurlijk spanningsveld. Onze visie hierop is als volgt: - gemeenschappelijke belang vormt de basis voor een goede samenwerking; - inzicht in elkaars belangen draagt bij aan een constructieve samenwerking en - respect voor elkaars belangen leidt tot een partnerschap De realisatie van een partnerschap vraagt een zekere inspanning van alle partijen. Dit betekent onder meer dat er inzicht is in: - behoefte vanuit de afnemers; - overzicht in mogelijkheden en onmogelijkheden van de oplossing (gap-analyse) - de roadmap voor doorontwikkeling Hiermee kunnen de leveranciers enerzijds de deelnemende gemeenten ondersteunen en volledig ontzorgen en anderzijds begrip / respect hebben voor elkaar (on)mogelijkheden op dusdanige wijze dat de lid gemeenten hierin geholpen worden. 7. De partij die kavel 1 uitvoert, neemt de rol van verbinder op zich. Wij adviseren deze partij als verbinder tweemaal per jaar chemistry meetings te laten organiseren, waarin de samenwerking vanuit alle betrokken partijen wordt geëvalueerd. Daarbij is het belangrijk dat partijen podium krijgen om de laatste technische ontwikkelingen te delen die relevant zijn voor de partners van de andere kavels. Zo wordt een optimale samenwerking geborgd. 8. Naar onze mening bestaat een echte samenwerking uit relatie, proces en inhoud. Vaak wordt er veel geïnvesteerd in het proces, voldoende op de inhoud maar zeer weinig op de relatie. Uiteindelijk hebben verschillende bedrijven toch verschillende belangen. Ons advies zou hierbij zijn om als verbinder met name te investeren op de relatie. Denk aan diverse online en offline meetings, kennissessies, awards, wedstrijden en ga zo maar door.
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		9. Een relatie tussen leverancier en Dimpact lijkt ons randvoorwaardelijk. Een relatie tussen leveranciers onderling wordt complexer. - opties zijn: alle partijen 'skin-in-the-game', lange termijn belangen afspreken. Financiering hiervan is ook complex. Partners ontstaan door co-creatie. Leveranciers die samen met klanten het probleem identificeren, oplossing ontwerpen en testen, en uiteindelijk via system integrator implementeren voor alle dimpact partners.
3	Kavel 2 Hosting	Antwoord
6	<i>Wat is uw beeld bij de keus voor Haven hosting?</i> <i>Heeft u hier ervaring mee?</i>	1. Wij zijn het eens met de keuze voor Haven Hosting en hebben hier ook ervaring mee. 2. Wij zijn een heel groot voorstander van Haven hosting. Voor een succesvolle multi-cloud strategie zijn standaarden cruciaal. Wij zijn AWS partner en hebben veel ervaring met het migreren en beheren van applicaties gehost op AWS alsmede Kubernetes, cloud load-balancing en clustered databases. 3. Aangezien wij een SaaS-toepassing leveren is hosting op basis van Haven niet direct voor ons van toepassing. Het idee is om flexibel de aangeboden software over te kunnen zetten van de ene hosting partij naar de ander. Wij zien als SaaS-leverancier meerwaarde in het zelf beheren van de software en platform. 4. We hebben wel ervaring met Kubernetes, maar niet met de Haven. Eén van de ideeën achter Haven is dat klanten "hun" software naar een andere hoster kunnen overbrengen. Is dit ook het geval bij Dimpact? Dat heeft dan impact op de softwarearchitectuur. In onze software worden veel modules gedeeld tussen klanten (en bevatten data van meerdere klanten) zoals integratiesoftware of multi-tenant applicaties. Als een klant "zijn" software wil oppakken en ergens anders neer zetten is dat niet zonder meer mogelijk. Daarnaast willen wij onze eigen software niet extern laten hosten op bijvoorbeeld een Haven-platform. Wij zien het zelf beheren van onze software en platforms als essentieel onderdeel van onze dienstverlening. Wij zien meer in het benadrukken van het gebruik van standaard-API's en in aanvulling daarop standaard import- en exportmogelijkheden. Zodat klant altijd zijn data kan exporteren en vernietigen. Hiermee is het eigenaarschap van de data voor de gemeente verzekerd, terwijl de gemeente tegelijkertijd wel profiteert van de kwaliteit en het schaalvoordeel van SaaS-diensten die door leveranciers worden geleverd. 5. Steeds meer ontwikkelaars volgen de ontwikkelingen of starten met ontwikkelen volgens het 5 lagen principe van Common Ground. Door de schaalbare opzet van Haven betekent dit vaak ook dat applicaties geschikt zijn voor gebruik op een Haven compliant hosting infrastructuur. Het gebruik van Haven als clusterimplementatie stelt gemeenten in staat, snel en eenvoudig applicaties te

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	gebruiken, ongeacht de onderliggende infrastructuur. De Haven implementatie is geoptimaliseerd voor gebruik door overheid en biedt uitgebreide mogelijkheden om te voorzien in de veiligheid van de omgeving en bijbehorende data. 6. Hosting is een vakgebied waarvoor wij graag met gecertificeerde specialisten samenwerken. De term "Haven Hosting" is vrij nieuw, maar het concept is vrij traditioneel. Hosting wordt op zo een manier ingeregeld dat deze aan specifieke standaarden voldoet. Dit biedt voordelen (voorspelbaarheid en beheersbaarheid), maar ook nadelen (geen flexibiliteit). Voor het concept dat u voor ogen heeft, is de keuze voor Haven Hosting een voor de hand liggende keuze. Wel adviseren wij u enige mate van flexibiliteit in te bouwen. Dit kan door uitzonderingen alleen te accepteren, mits dit goed onderbouwd is en de kwaliteit en veiligheid ten goede komt. 7. Wij hebben ervaring met het inzetten van Haven/Kubernetes hosting voor meerdere klanten, wel altijd als technisch applicatiebeheerder/integrator. Wij kiezen er bewust voor geen eigen fysieke hosting of Kubernetes clusters zelf te managen, dit beleggen wij bij een leverancier of cloud-platform (Azure, GCP, AWS). 8. Haven hosting houdt in dat websites, applicaties en registers op homogene wijze worden gehost. Dit concept biedt voor- en nadelen. De voordelen liggen op het gebied van duidelijkheid vooraf, een uniforme en daarmee voorspelbare situatie. De nadelen zijn dat een homogene inrichting beperkend kan werken bij de doorontwikkeling van nieuwe componenten en technieken. De keuze voor Haven Hosting is uitstekend, zeker in een omgeving waarin standaardisatie plaats vindt. Wel adviseren wij hierbij het 'pas toe of leg uit'-principe toe te passen. 9. Haven hosting en de daarmee samenhangende verplichte een aanbevolen processen zien wij zeker als een goede keuze voor het type applicaties dat ontwikkeld en ondersteund zal worden. Aangezien iedereen toegang zal hebben tot de modules zijn de checks die worden aangehaald zeker een must om de vereiste kwaliteit te garanderen (zoals bv. uptime). 10. Ja, wij voorzien geen enkel probleem om Mendix op een Haven-compliant cluster te draaien (zowel op basis van Operator als op basis van Docker buildpack). Onze ontwikkelaars gaan uit van een werkende infrastructuur op het moment dat zij software gaan ontwikkelen. Wij hebben geen specifieke voorkeur hierin. 11. We onderschrijven de keuze voor Haven-hosting. Wij beheren momenteel voor verschillende gemeenten Haven- omgevingen met al dan niet productie-applicaties. We zijn daarbij zeer tevreden over de
--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		mogelijkheden die Haven biedt. Wel merken we op dat Haven echt dient te worden gezien als een migratie standaard, het faciliteert een basis laag voor applicaties waarmee gegarandeerd wordt dat applicaties kunnen draaien op een omgeving. Dit creëert Cloud neutraliteit en biedt een beeld op de toekomst, waarbij gemeenten zonder problemen van Cloud-leveranciers zouden kunnen wisselen. Haven behelst echter niet het doel de omgeving volledig productiewaardig te maken, daar komen andere zaken bij kijken zoals security, wet bio, back-up etc. waar de standaard geen of te weinig uitspraken over doet en bewust ruimte creëert voor leveranciers om zich te onderscheiden. We zouden Dimpact bij een tender adviseren hierover na te denken. Enerzijds dienen juridische eisen te worden geborgd, maar anderzijds zal een uitvraag ook de ruimte moeten laten aan Haven-partijen om zich te kunnen onderscheiden. 12. Wij vinden het verstandig dat u kiest voor Haven complied hosting gegeven de discussie omtrent privacy shield in de markt. U zorgt zo voor een abstractielaag tussen publieke cloud resources en de daadwerkelijke applicatie deployments. Onze mening is dat de inkoop van publieke cloud resources en Haven-werkzaamheden in principe bij dezelfde partij zouden moeten liggen. 13. Wij zijn bekend met Haven. De positieve kant hiervan is dat dit gemeenten stimuleert om haar applicaties te moderniseren zodat zij met Docker kunnen werken. Helaas hebben wij al wel vastgesteld dat een aantal technische principes van Haven alweer is ingehaald door de praktijk. Als leverancier maken wij al meer dan 5 jaar gebruik van Docker in productie en hebben al een aantal lessons learned. Er is dus een risico dat Dimpact en/of gemeenten halsstarrig vasthouden aan principes en daardoor de vorm belangrijker vinden dan het resultaat. Dus zover het een pas-toe-of-leg-uit principe is, lijkt het ons een logische en zeer goede keuze.
7	<i>Hoe ziet u uw rol ten opzichte van de andere kavels? Tot waar rijkt uw expertise en/of verantwoordelijkheid?</i>	1. Data-integratie en migratie komt eigenlijk voor op alle 3 de kavels. Het is in ieder zijn belang dat hier goed over nagedacht wordt. En dat er een IT-landschap komt waarbij je overzicht hebt over je data en integraties. Wat makkelijk te managen en configureren is voor de IT-teams bij verschillende gemeentes en waarbij problemen met duidelijke error-handling goed op het netvlies komen. Je zal snel zien dat als je dit nieuwe landschap uittekent dat er heel veel pijltjes van de ene component naar de andere component en ook vaak weer terug komen te staan. Die connecties dat is de benodigde data-integratie. Wij hebben verschillende architecten sessies verzorgd bij verschillende gemeentes waarbij wij inzichtelijk gemaakt hebben welke afhankelijkheden er zijn binnen het landschap en hoe je hier mee om kan gaan. 2. Hosting is verantwoordelijk voor de stabiliteit en continuïteit van de test-, acceptatie- en productieomgevingen en de sleutelwoorden daarvoor zijn automatisering en standaardisatie. Door moderne technieken zoals terraform en ansible is het mogelijk om geautomatiseerd heel snel een

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		nieuwe omgeving in te richten - voor bijvoorbeeld een nieuwe integratie of audit omgeving. Ook alle proactieve monitoring kan worden geautomatiseerd, maar dat vergt wel nauwe samenwerking tussen de software ontwikkelaars, hosting en systeemintegratie. • De software ontwikkelaars zijn verantwoordelijk voor de ontwikkeling van monitoring endpoints, geautomatiseerde testen en runtime aan te passen logging niveaus. • Systeemintegratie is verantwoordelijk voor geautomatiseerde ketentesten • Hosting is verantwoordelijk om bovenstaande functionaliteit in te zetten voor de borging van zo stabiel mogelijke omgevingen Hosting is niet verantwoordelijk voor de kwaliteit van de software en het functionele beheer. 3. De keuze tussen unmanaged en managed hosting is niet een of, maar een en. Voor bepaalde componenten en omgevingen zullen unmanaged hosting voldoende zijn en andere - het merendeel vermoeden wij - software producten zullen baat hebben bij managed hosting. Het scala aan services dat bijvoorbeeld Amazon biedt op AWS is inmiddels zo groot, dat per software product / component zal moeten worden bepaald welke services worden afgenomen. Uitgangspunt is dat het Haven omgevingen zijn om cloud-agnostische infrastructuur te garanderen. Wij zijn een groot voorstander van een multi-cloud strategie en de verschillen tussen de verschillende infrastructuren bij de gemeenten zijn te groot voor een one-size-fits-all oplossing. Het zal in de praktijk dus een scala van oplossingen zijn zoals cloud en on-premise, managed en unmanaged, maar met als grote gemene deler de Haven standaard. 4. Voor ons is het essentieel dat de hosting-leverancier de randvoorwaarden levert die noodzakelijk zijn. Als ontwikkelaar kunnen wij de benodigde specificaties aanleveren. Voor het inrichten van een ontwikkelomgeving en ontwikkelstraat veronderstellen wij dat dit een verantwoordelijkheid is van de ontwikkelaar. 5. Onze verantwoordelijkheid rijkt tot OS of cluster-niveau. Het beheren van fysieke infrastructuur of het beheren van een Haven cluster valt buiten onze expertise. 6. De rol van hosting leverancier richt zich op het faciliteren van betrouwbare omgevingen die voldoen aan de gevraagde specificaties, het monitoren van uptime en performance, het signaleren van beveiligingsrisico's en verdacht verkeer, het treffen van maatregelen in het kader van informatiebeveiliging en de performance en het bieden van ondersteuning en informatie waarmee andere leveranciers inzicht hebben/krijgen in mogelijke (performance) verbeteringen van de applicatie. 7. Wij vinden dat de partij die verantwoordelijk zal zijn voor de hosting-kavel zijn expertise zowel
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		pro-actief (op eigen initiatief) als re-actief (op vraag van bv. de ontwikkelingspartner) beschikbaar moet stellen van de andere partijen. Als expertise denken we aan bv. security, server dimensioning, performantie, het mee werken aan het creëren van een stabiele omgeving, 99.99% uptime, ...maar ook aan het delen van kennis over de vele (technologische) mogelijkheden en innovaties die bedrijven zoals bv. Amazon tegenwoordig aanreiken. Veel van deze technologische mogelijkheden en innovaties hebben immers een directe impact op / relatie met softwareontwikkeling. 8. Wij begeven ons met name op het vlak van het ontwikkelen van componenten en software producten en daarmee deels ook op een stuk systeem integratie. Dat wil zeggen, ontwikkelen van API's, workflows, data-integratie, ontwikkelen van een presentatielaag, connecteren met voorzieningen zoals digid/e-herkenning, etc. Een werkende cloud c.q. infrastructuur is hierbij voor ons een randvoorwaarde die reeds ingevuld dient te zijn. 9. We zien een sterke mate van overlap tussen de rol van hosting en het ondersteunen van partijen en een samentrekking van managed-hosting en systeemintegratie. Vanuit onze rol als system integrator installeren wij applicaties die wij niet zelf hebben gebouwd . Dit leidt in de praktijk vaak tot vragen en uitdagingen, zo heeft lang niet iedere Common Ground- applicatie de benodigde installatie bestanden voor Haven. Juist hier kan een system integrator waarde toevoegen door partijen te helpen met kennis en ondersteuning. Op die manier verlagen we de drempel van Common Ground en Haven, waarbij we partijen meer ruimte gunnen om zich te focussen op innovatie. In de praktijk ervaren wij dan ook een ondersteunende rol waarbij we ontwikkelaars helpen hun applicatie niet alleen te "Haven"-proofen, maar ook productiewaardig te maken. Waarbij gedacht kan worden aan Bio-compliance, AVG vraagstukken over logging en dergelijke. We ervaren dan ook geen strakke scheiding tussen beheer en ontwikkeling en dienen regelmatig kleine pr's in op applicaties die wij hosten. Vanuit de een werkend ecosysteem denken we ook dat het belangrijk is dat we aspecten zoals veiligheid en compliance als een gemeenschappelijke verantwoordelijkheid zien. Uiteraard zal hier een firstline verantwoordelijke voor moeten zijn in de vorm van een beheerder op de codebase en system integrator op een omgeving, maar pragmatisch gezien willen we ook dat er wordt versneld en wordt geïnnoveerd. Dat betekent juist dat we als ecosysteem partijen willen aantrekken die wellicht (nog) niet over de juiste kennis en kunde van de zeer specifieke Common Ground en overheidsmarkt vraagstukken beschikken. Die dienen we dus te faciliteren. De hosting partij heeft vanuit zijn klantcontact en overzicht op de omgeving hier goed zicht op en kan dit samen met de system integrator faciliteren. Er is dus enige overlap nodig tussen hosting en systeemintegratie, wat vraagt om partijen die elkaar kennen en vertrouwen. 10. Verbinding hebben wij hoog in onze kernwaarden staan. We zetten erop in Dimpact verder te helpen in een steeds veranderende markt door in de periferie van dit soort opdrachten samen met andere partners kennisdelingssessies te organiseren. Hiermee ontstaat er een betere verbintenis tussen partijen.
--	--	--

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		Onze kennis en kunde sluiten perfect aan op kavel 2 (inclusief het beheer van publieke cloud resources) en gedeeltelijk op kavel 3 (afhankelijk van de uitvraag).
8	<i>Welke keus ligt Dimpact voor? Kiezen we voor unmanaged of managed hosting? Zoeken we een ASP of sluiten we enkel overeenkomsten met cloud computing services partijen?</i>	1. Als de hosting maar haven compliant, leverancier onafhankelijk en cloud-based is! 2. De keuze tussen unmanaged en managed hosting is niet een of, maar een en. Voor bepaalde componenten en omgevingen zullen unmanaged hosting voldoende zijn en andere - het merendeel vermoeden wij - software producten zullen baat hebben bij managed hosting. Het scala aan services dat bijvoorbeeld Amazon biedt op AWS is inmiddels zo groot, dat per software product / component zal moeten worden bepaald welke services worden afgenomen. Uitgangspunt is dat het Haven omgevingen zijn om cloud-agnostische infrastructuur te garanderen. Wij zijn een groot voorstander van een multi-cloud strategie en de verschillen tussen de verschillende infrastructures bij de gemeenten zijn te groot voor een one-size-fits-all oplossing. Het zal in de praktijk dus een scala van oplossingen zijn zoals cloud en on-premise, managed en unmanaged, maar met als grote gemene deler de Haven standaard. 3. Zoals eerder genoemd biedt Kubernetes veel voordelen wanneer het optimaal ingezet wordt. Het biedt aan de andere kant ook veel uitdagingen als het aankomt op het kennisniveau, beheer en beveiliging. Updates volgen elkaar snel op en het 'Cloudnative landschap' is continu in ontwikkeling. Het volgen van alle ontwikkelingen en het maken van de juiste keuzes om in te investeren is een dagtaak voor specialisten. In de Common Ground opzet en de samenwerking die Dimpact nastreeft is het verstandig een specialist als partner te hebben om op dit vlak ontzorgt te worden. Gezien de continue ontwikkelingen op het gebied van updates en beveiliging is het te adviseren om gebruik te maken van een 'Managed' provider in plaats van een eenmalige of periodieke opzet vanuit een Consultancy organisatie. Technical debt kan snel optreden wanneer de Kubernetes omgeving niet passend onderhouden wordt. Wanneer er gebruikt wordt gemaakt van een managed provider kunnen inschrijvers van Kavel 1 & 3 zich volledig focussen op het ontwerpen, bouwen en bij elkaar brengen van de juiste componenten aan de applicatiezijde. 4. Het antwoord op deze vraag kan alleen goed beantwoord worden als Dimpact een duidelijk beeld en visie heeft ten aanzien van de verantwoordelijkheden, taken en bevoegdheden. In aansluiting op een van onze eerdere antwoorden geldt dat hosting een randvoorwaarde is die in orde moet zijn. Dit betekent ook dat de leverancier hierin goed moet kunnen ondersteunen. Uit ervaring met een vergelijkbare opdracht hebben wij ervaren dat managed hosting uitstekend werkt, tenzij de randvoorwaarden niet in orde zijn. Randvoorwaarden liggen in dit geval op het gebied van reactie- en oplostijden.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		5. Het hangt af van de definities van 'managed' of 'unmanaged', maar in de regel wil je naast overeenkomsten met een clouddienst (zoals Azure AKS voor het draaien van een Haven-compliant cluster) ook een partij laten aansluiten binnen deze kavel die gespecialiseerd is in het inrichten en beveiligen van Haven-omgevingen en daar de verantwoordelijkheid willen beleggen om de configuratie van Azure op een adequate en veilige manier in te richten. Dus unmanaged is prima als optie maar daar moet een leverancier bij voor de inrichting daarvan. Deze noodzaak zien wij met name vanwege de keuze voor Haven/Kubernetes, wat additionele 'infra' complexiteit met zich meebrengt . 6. De vraag heeft betrekking op de wijze waarop taken en verantwoordelijkheden worden georganiseerd. Met de gekozen opzet van verschillende kavels, verschillende leveranciers en een regiefunctie vanuit de organisatie Dimpact kan deze vraag alleen goed beantwoord worden als duidelijk is in hoeverre Dimpact vanuit operationeel oogpunt de regie wil voeren. Onze advies in deze is laat de leverancier die verantwoordelijk is voor de implementatie een voorstel uitbrengen voor de wijze hoe dit te organiseren. Hierdoor is het mogelijk om de verdeling van taken en verantwoordelijkheden op juiste wijze te organiseren. 7. Naar de klanten toe van Dimpact (de gemeenten), menen wij dat er gekozen moet worden voor managed hosting. Bij hedendaagse ontwikkeling is er immers heel vaak een sterke afhankelijkheid en correlatie tussen de server set-up / infrastructuur en de ontwikkelde applicatie. Het is dan ook belangrijk dat dit geheel wordt beheerd door specialisten ter zake. Wij zijn van mening dat deze specialisatie niet verwacht mag worden van de gemeenten. Onze ervaring leert ons dat het vertrekpunt voor het leveren van hostingdiensten verschillend is bij een ASP en een cloud computing service partij. Een ASP levert de hostingdiensten als onderdeel van een totaaloplossing inclusief de applicatie. De hosting is voor een ASP noodzakelijk voor het kwalitatief kunnen aanbieden van de applicatie en vaak is de hosting sterk verweven met de applicatie. De core business van de ASP is dan ook niet de hosting, maar het totaalpakket van de applicatie, de hosting en aanverwante diensten. De core business van een cloud computing service partij is dan weer wel de hosting. Wij menen dat geen van beide type leveranciers moet uitgesloten worden. Beiden hebben immers hun voordelen en toegevoegde waarde en dit afhankelijk van de dienst die geleverd moet worden naar de klant. Bij bepaalde applicaties/componenten zal de verwevenheid met de hosting en server set-up zeer groot zijn. In dat geval is het inzetten van een ASP een betere optie. Bij andere componenten met een mindere of geen verwevenheid van de applicatie met de hosting en server set-up is een cloud computing service bedrijf dan weer een betere optie. Onze aanbeveling is om de keuze tussen een ASP en een cloud computing service bedrijf te bepalen afhankelijk van de te leveren component of combinaties van componenten naar de klant toe. Is er een sterke verwevenheid tussen de component(en) en de hosting / server set-up, dan bevelen wij een ASP aan. Is die verwevenheid er niet, dan bevelen wij de keuze voor een cloud
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		computing service bedrijf aan. 8. Wij hebben hier geen voorkeur in. Voor onze klanten kunnen wij Mendix Cloud hosting inzetten, maar we kunnen ook prima uit de voeten met Azure/AWS, private cloud of hybrid cloud. 9. Voor wat betreft unmanaged of managed hosting is het veel eerder de vraag: welk gedeelte willen jullie waar inkopen? Het kan ook de uitvraag zijn om de systeembeheerder (ASP) de hosting te laten inregelen en be-heren (en een gedeelte van kavel 1 samen te voegen met kavel 2). In verband met de vertrouwelijkheid van gegevens is het wellicht beter om data in Nederland op te slaan in plaats van in de Cloud bij een Amerikaans bedrijf die daarbij ook verplicht is om desgewenst de Amerikaanse overheid inzicht te geven in gegevens. Haven is een algemene structuur die ook goed op basis van OpenStack op eigen ingerichte VPS-systemen te draaien is. In dat geval kan in Nederland worden ingekocht bij een van de grotere hosting providers. Bijv. CloudVPS ondersteunt OpenStack. Op deze manier blijft kennis in Nederland en ontstaat er geen afhankelijkheid van een Amerikaanse partij. Tenslotte, qua kostenaspect, het zelf in laten richten van OpenStack servers geeft lagere kosten dan inkoop bij een Cloudpartij. In een recent gemaakte prijsvergelijking door Cybersprint kwam het zelf inrichten van een omgeving een factor 2-2,5x zo goedkoop uit vergeleken met de Cloud. 10. In onze ervaring zit er binnen een hosting propositie een scheiding tussen de hardware ijzerkant (ofwel alles tot en met Kubernetes) en de inrichting en beheerkant van de applicaties die op Kubernetes draaien. Dit lijkt in lijn te liggen met managed- en unmanaged- hosting en de gedachte om van systeemintegratie een los perceel te maken. Voor diverse gemeenten leveren wij nu het beheer van applicaties (inclusief installatie, configuratie en beheer) op de gehoste omgeving. Daarbij merken wij dus dat er in de praktijk vaak een "knip" zit tussen managed- hosting en het management van hoe het cluster gebruikt wordt, dit laat zich meer vergelijken met wat in deze consultatie "systeemintegratie" wordt genoemd. Dat laatste heeft dan weer te maken met inhoudelijke kennis en expertise op zowel overheidsvlak (wet BIO etc.), als Common Groundvlak, daarbij valt dan weer een onderscheid te maken tussen technische kennis (zoals Haven) en community kennis (welke applicaties zijn er en hoe verhouden die zich tot elkaar) Dat gezegd hebbende zouden we managed-hosting adviseren. Kubernetes en Haven zijn specialistische kennis gebieden die bij heel veel gemeenten niet beheerst worden. Door het managed uit te vragen weet je als gemeente dat de onderlaag (tot en met Haven) geregeld is en up-to-date blijft. Je wordt hierdoor ontzorgd en je behoudt je onafhankelijkheid omdat je Haven als standaard afdwingt in je uitvraag en dus kan switchen tussen hosting partijen (managed of unmanaged). 11. We adviseren Dimpact zelf een contract af te sluiten met een cloud computing services-partij (zoals Microsoft Azure, AWS of Google). Alle managed services daarop adviseren wij u te laten uitvoeren door een geaccrediteerde partner van dit platform. Ons advies is om dit samen te brengen in kavel 2, inclusief
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		het volledige beheer, de monitoring en de support van Haven/Common Ground. De managed hosting partner heeft zo volledig toegang tot de publieke resources van Dimpact. De grote cloudpartijen wensen ook op deze manier samen te werken.
3.	Kavel 3 Ontwikkeling componenten en softwareproducten	Antwoord
9	<i>Een van de uitgangspunten is om ontwikkelaars in projecten een verschillende rol te laten vervullen. Hoe kijkt u hier tegenaan? Kunt u een kritische doch constructieve rol innemen? Heeft u ervaring met het vervullen van verschillende rollen?</i>	1. Ja wij zijn gewend om onze opdrachtgevers en partners op te leiden en vervolgens nog kritisch maar constructief mee te denken. Vanuit onze devops gedachten zijn wij prima in staat om samen te werken. 2. Een moderne software ontwikkelaar heeft altijd een aantal verschillende rollen binnen een project: • Analist, user stories moeten worden opgeknipt in taken en behoeven soms extra analyse • Architect, goede architecturen komen voort uit zelforganiserende teams. • Tester, (geautomatiseerde) testen van opgeleverde software • Auditeer, peer- reviews op de code van je collega's De kern van Agile werken met Scrum is dat het hele team verantwoordelijk is voor het opleveren van de stories conform de definition of done. Het ene Scrum lid heeft meer affiniteit met back-end code, de ander met testen en weer een ander is een database expert. Maar het hele team werkt aan de dingen die op dat moment belangrijk zijn. Er is geen wezenlijk verschil tussen de initiële ontwikkeling van een software component of het doorontwikkelen ervan. Op de product backlog kunnen zowel nieuwe features staan als taken voor operationele ondersteuning. Ofwel wij onderkennen verschillende werkzaamheden binnen een scrum team, maar we vinden het niet verstandig om een apart bouwteam te hebben en een apart testteam van verschillende leveranciers. Dan staat haaks op hoe Scrum bedoeld is en zal niet bijdragen aan het beoogde doel om betrokkenheid en kwaliteit van de gehele levenscyclus te optimaliseren. Dit doel kan wel worden bereikt door ontwikkelaars en gebruikers samen te laten werken en ontwikkelaars te motiveren door ze innovatieve vraagstukken toe te vertrouwen en ze verantwoordelijkheid voor het einddoel te geven. Het team dat de de initiële ontwikkeling van een software component doet, hoeft niet het team te zijn dat de doorontwikkeling doet, maar het is natuurlijk wel het meest efficiënt als dat wel het geval is. Wij onderkennen de wens van Dimpact om niet te afhankelijk te worden van maar een paar partijen. Door eisen te stellen aan de documentatie, voorwaarden voor gebruikte frameworks en tooling, standaarden te hanteren, code publiek beschikbaar maken, hoge test-coverage en door regelmatig software audits te laten uitvoeren kan je de overdraagbaarheid van software enorm verhogen en daarmee de afhankelijkheid op specifieke leveranciers verkleinen. 3. Het samenwerken in projecten is iets dat we al jaren doen. Juist vanuit die ervaring stellen we dat het in projectvorm ontwikkelen van software veel van de diverse partijen vraagt. Bij het ontwikkelen van software stopt het niet na afronden van het project. Juist daarna heb je te

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		maken met doorontwikkeling en support. Wij weten wat hiervoor nodig is voor onze software. De vraag is of dit ook te organiseren is als er meerdere partijen verantwoordelijk zijn voor de opgeleverde software tijdens en na de ontwikkeling. Heldere afspraken hieromtrent zijn dan cruciaal. Common Ground biedt volgens ons juist ook een perfecte basis voor gestandaardiseerde onderlagen en gebruikerstoepassingen. Daarbij zijn de rollen prima verdeeld en is duidelijker wat van eenieder verwacht wordt. 4. Samenwerking tussen softwareleveranciers is belangrijk en mogelijk, maar bij dit model hebben we vraagtekens omdat met het wisselen van de rol ook het verantwoordelijkheidsgevoel voor de doorontwikkeling verdwijnt. Wij zien meer in een SaaS-model waarin leveranciers een goed gestandaardiseerde onderlaag leveren en Apps die worden doorontwikkeld en steeds betere functies krijgen. De leveranciers zorgen dan voor continuïteit. Hierbij werken leveranciers ook samen. We hebben hier veel ervaring mee omdat we al vaak onze applicaties hebben gekoppeld aan die van andere leveranciers. Daarbij nemen we een constructieve doch kritische rol in met als uitgangspunt altijd een combinatie van resultaatgerichtheid, gebruik van standaarden en gezamenlijk optrekken tussen leveranciers en gemeente. 5. Het is ons niet geheel duidelijk wat u hiermee voor ogen heeft. Bedoelt u hiermee ontwikkelaars (als zijnde de leveranciers) of de medewerkers van de bureaus (de developers). Als het gaat om de leveranciers naar verloop van tijd in een andere rol te zetten, dan kan dat verfrissend werken, mits het een rol is die bij de leverancier past. Keerzijde hiervan is dat opgebouwde kennis en ervaring op een andere manier moet worden ingezet. Als het gaat om developers in een andere rol, dan hebben wij hier onze bedenkingen bij. Immers, wij zetten specialisten in die een specifiek doel voor u gaan realiseren. 6. Als wij u goed begrijpen, dan wilt u ontwikkelaars in verschillende projecten in een verschillende rol laten fungeren. Heeft u hiermee voor ogen dat ontwikkelaars (medewerkers of de leveranciers) in verschillende rollen worden ingezet. Wij begrijpen dat dit uitgangspunt ondersteunend is aan het doel om generieke, modulaire oplossingen te genereren, waarbij voorkomen wordt dat leveranciers en/of ontwikkelaar in een soort van tunnelvisie of silo-denken terecht komen en hiermee een bijdrage leveren aan de uitgangspunten en doelstellingen van Dimpact. Het uitgangspunt is goed en kan, mits de juist randvoorwaarden geschapen zijn, een zeer constructieve bijdrage leveren aan de uitgangspunten en doelstelling van de (deel)projecten. De randvoorwaarden en aansturing hierin zijn essentieel. Immers een leverancier kan enerzijds ingehuurd worden voor kennis en capaciteit, in dit geval ligt de sturing op het eindresultaat elders. Anderzijds kan de opdrachtgever een deel- of eindproduct inkopen tegen een vooraf gestelde prijs, in dat geval ligt de verantwoordelijkheid (en daarmee de in te zetten kennis en capaciteit) bij de leverancier. De wijze of dit conform de uitgangspunten en/of de doelstellingen van het (deel)project ten goede komt
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	dient per situatie beoordeeld te worden. 7. Wij staan positief tegenover het feit om verschillende (softwareontwikkelaars)bedrijven te laten samenwerken op verschillende deeldomeinen zoals bv. ontwikkeling (bouwer), testing, Dit brengt immers verschillende expertises samen, zorgt vaak voor interessante en soms onverwachte kruisbestuivingen en verhoogt ook de efficiëntie, de opleversnelheid en de kwaliteit. Voorwaarde is wel een uitstekende organisatorische aanpak met de juiste afspraken over o.a. documentatie, verantwoordelijkheden, een projectopvolgingstool die in staat is om alles te beheren/op te volgen, etc. De eerder vermelde IPLS-development train (station) – aanpak kan hier zeker aan bijdragen. Wij menen zelfs dat hier ver in gegaan kan worden. Het is bv. perfect mogelijk om na de ontwikkeling een code review proces te laten uitvoeren door een andere partij. 8. Wij werken vooral op basis van principes als SAFE, DevOps en Agile. Dit zouden we zeker als norm adviseren voor software ontwikkeling/ Een verschillende rol laten invullen kan daarin samenwerking verder versterken. Andere belangrijke principes welke we hanteren zijn; Design Thinking/UX, Privacy by design, security by design . We adviseren Dimpact goede UX, privacy en security expertise in huis te halen in de gerelateerde kavels. Ook adviseren we goed na te denken over kwaliteitsmanagement en testen van de software, mogelijk is dit nog een vierde kavel waardig, om te voorkomen dat de slager zijn eigen vlees keurt. 9. Het onderverdelen van specifieke rollen binnen de ontwikkeling van software producten is wat ons betreft een verstandige en belangrijke keuze. Naast volledige oplossingen hebben we hebben ervaring in allerlei verschillende sub-verantwoordelijkheden. Denk bijvoorbeeld aan projecten waar we alleen requirements onderzoeken en interfaces ontwerpen, projecten waarin we alleen front- of back-end ontwikkelen, of juist projecten waarin we andermans werk testen / controleren op kwaliteit of doorontwikkelen op een bestaand product. Als de verantwoordelijkheden goed zijn verdeeld en er sterke overkoepelende governance is geloven wij dat op deze manier een best of breed oplossing kan worden gemaakt. 10. De wens om in de loop van een project te kunnen wisselen van rollen tussen partijen is een bijzonder uitdagende en mogelijk kostbare aanpak. In onze ervaring zijn de meest voorkomende gevolgen van een (harde) rolwissel: - Het willen/moeten herschrijven van grote stukken van de codebase om beter aan te sluiten op het ontwikkel paradigma van de nieuwe partij. - Een grote (vaak permanente) verlaging van de productiesnelheid.
--	--

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	- Een vergroting van de onzekerheid in estimates wegens onvoldoende zicht op impact bij wijzigingen. 11. Onze ervaring leert dat een soepele rolwissel kansrijker wordt als de partijen van te voren weten dat ze eventueel rollen van elkaar overnemen en de mogelijkheid hebben gehad tot afstemming van bijvoorbeeld technische keuzes voor talen, frameworks en best practices. Juist een vroege betrokkenheid bij elkaars ontwikkeling zou een rolwissel mogelijk kunnen maken maar dit blijft nog steeds een zeer experimentele aanpak. Als de partijen van te voren niet bekend zijn is het van belang dat er een substantieel deel van de ontwikkeltijd besteed wordt aan het documenteren en toegankelijk maken van toetreders in de codebase. In grote open source projecten wordt dit omschreven als de zogenaamde Developer Experience (lees hier meer over op bijv https://hackernoon.com/the-best-practices-for-a-great-developer-experience-dx-9036834382b0). Bij een dergelijke rolwisseling geniet het de voorkeur om warm over te dragen en de afwisselende teams geruime tijd geven om elkaar te leren kennen, te leren, vragen te kunnen stellen en vervolgplannen af te kunnen stemmen zodat ook de rol-verlatende partij kan helpen bij het onboarden / warme overdracht. 12. Wij vinden dit een goede manier van werken, hebben hiermee ervaring en geven hieraan zelfs de voorkeur. Een developer is zelf verantwoordelijk voor de kwaliteit van zijn software, en tevens in staat om zijn software te deployen in een (cloud)omgeving. In moderne ontwikkelmethodieken is een korte time to market steeds belangrijker. Daarnaast is flexibiliteit in de ontwikkeling van software erg belangrijk. Doordat het ontwikkelteam zelf zijn eigen software kan deployen, ontstaat er een betere kwaliteit, beter testbare software en een kortere ontwikkeltijd van de totale oplossing. 13. Wij hebben ervaring met het ontwikkelen van software met verschillende rollen, maar ook met verschillende teams. Zowel interne teams als externe teams middels een samenwerking met andere bedrijven. Het ene project is hierbij iets meer succesvol dan het andere. Wij hebben gemerkt dat het ontwikkelteam het meeste succes heeft wanneer de ontwikkeling van Backend en Frontend gescheiden wordt. Doorgaans zien wij dat deze verschillende rollen toch een eigen discipline zijn en het niet vaak voorkomt dat deze verwisseld worden. We ondervinden veel succes aan het invullen van ondersteunende rollen. Denk hierbij aan ontwikkelaars die een rol hebben voor het voorbereiden en in goede banen leiden van een refinement, of zelfs (tijdelijk) de rol van Scrum Master vervullen. Het testen en ontwikkelen houden wij bij voorkeur gescheiden in verband met de slager die anders zijn of haar eigen vlees keurt.
--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		14. Het is niet erg om een andere rol te dienen binnen een ander project. Wisselingen in 1 ontwikkeltraject zijn niet wenselijk. Ook gewenste wijzingen na het testen door een andere leverancier geeft zeker vertraging. Wij hebben ervaring met alle verschillende rollen.
10	<i>Hoe ziet u de samenwerking met andere leveranciers in de ontwikkelfase, in de implementatiefase en in de gebruikers-/beheerfase?</i>	1. Het grote voordeel van een architectuur gebaseerd op componenten is dat het werk dat moet gebeuren in kleinere stukjes kan worden geknipt dan een architectuur met silo-applicaties. Hierdoor kan je in theorie met meerdere leveranciers samen werken aan software producten. Alleen wil je ook werken met een goed ingewerkt Scrum team van 3 á 4 personen. Die twee doelstellingen staan op gespannen voet met elkaar en daar zal een balans in moeten worden gevonden. De samenwerking met andere leveranciers in de ontwikkelfase kan in een aantal vormen. Of medewerkers van verschillende leveranciers zitten in hetzelfde Scrum team, of er moet worden aangesloten op een bestaande applicatie door een externe API te consumeren. In dat laatste geval zal veelvuldig overleg noodzakelijk zijn. In het geval dat meerdere Scrum teams van verschillende leveranciers parallel werken aan een groter geïntegreerd software product is het vooral noodzaak om geïntegreerde demo's te geven en zal er veelvuldig afstemming nodig zijn tussen de verschillende teams en de PO's. De samenwerking met de hostingpartij is vooral gericht op het veelvuldig kunnen releasen van nieuwe versies. Hierover moeten werkafspraken worden gemaakt waarin de verschillende verantwoordelijkheden duidelijk zijn. Deze samenwerking zal vooral tijdens de eerste paar releases intensief zijn, maar als het deployment proces meerdere keren is doorlopen kan deze samenwerking minder intensief. De samenwerking met systeemintegratie is complexer en erg afhankelijk van de context. Betreft het een nieuw component, betreft het doorontwikkeling of zijn de werkzaamheden onderdeel van een groter softwaresysteem. Voor alle samenwerking met de andere leveranciers geldt dat de hoeveelheid afstemming groter wordt naarmate de situatie complexer is en er meer afhankelijkheden zijn. Het samenwerken met andere leveranciers vereist dat er meer vooruit moet worden gekeken in de Scrum teams om de onderlinge afhankelijkheden te identificeren en vervolgens te adresseren. Hiervoor is een belangrijke rol weggelegd voor de Scrum Masters en Product Owners. De gebruikers- / beheerfase verschilt in principe niet van de ontwikkelfase zolang het Scrum team nog steeds bij elkaar is. Incidenten en kleine verbeteringen in bestaande features kunnen naast nieuw te ontwikkelen features op de backlog komen. Dit kan zelfs over componenten en systemen heen in hetzelfde Scrum team. 2. Zoals hierboven ook vermeld is samenwerking sterk afhankelijk hoe je de ontwikkeling van software wil organiseren als Dimpact. Een samenwerking in welke fase ook vraagt om goede afspraken. Naar onze mening is inzetten op standaardisatie wenselijk. Er zijn nauwelijks marktpartijen te vinden die willen inzetten op maatwerk. Dit trekt een zware last op de dienstverlening tijdens de beheerfase. Common Ground biedt hier voldoende keuzevrijheid en mogelijkheden. Hoe wij de samenwerking met andere leveranciers zien is sterk afhankelijk van de gekozen perceelinrichting alsmede de keuze van de te ontwikkelen softwarecomponenten. Wanneer wij in staat zijn

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	om SaaS-applicaties op common ground gebaseerde techniek te leveren, zijn we in staat om zeer nauw samen te werken met andere partijen, zowel voor wat betreft de ontwikkel-, implementatie- en beheerfase. Maar ook hier geldt dat vooraf sprake moet zijn van een duidelijke afbakening van verantwoordelijkheden. 3. We zien steeds vaker dat projecten worden uitgevraagd waarin een samenwerking tussen specialistische partijen en overdraagbaarheid tussen teams een belangrijke rol speelt. Dat in combinatie met de nog vrij nieuwe common ground software architectuur is het verstandig gebruik te maken van de geringe ervaring die sommige marktpartijen al hebben. 4. In een samenwerking draait het niet zozeer om de samenwerking tussen leverancier, maar om de samenwerking tussen de medewerkers van de leveranciers (de mensen). Samenwerking is mensenwerk. In ons vak is het gebruikelijk om samen te werken, niet alleen met specialisten vanuit andere disciplines, juist ook met collega's vanuit onze eigen discipline. 5. Wij zouden met name andere ontwikkelpartijen willen laten aansluiten in een adviserende of reviewende rol tijdens de ontwikkel- en implementatiefase. Elke partij heeft specialistische kennis en kunde, dit is hun kracht en die moet zo goed mogelijk benut worden. Buiten dat zouden wij geen voorstander zijn van gedwongen samenwerken. Software ontwikkelen is al een uitdaging op zich en wat Dimpact voorstelt werkt in onze ogen vertragend en risicoverhogend. Wij snappen de reden achter het voorstel (risicospreiding en multi-vendorship) maar vinden dat er verstandigere en realistischere manieren zijn om een 'separation of concerns' te creëren. Met deze aanpak ben je in onze ogen verzekerd van langlopende softwaretrajecten die niets opleveren en ruzie in de tent, waardoor je het tegenovergestelde bereikt. 6. Belangrijk is dat er heldere afspraken zijn wie welke verantwoordelijk heeft en hoe de overdracht tussen de fases plaats vindt. Daarbij is de omschrijving van het 'gewenste eindresultaat' vanuit organisatieperspectief belangrijker dan technische vereisten, die randvoorwaardelijk zijn. We bevelen verder aan om eisen te stellen aan technische documentatie volgens de geldende normen. Indien deze afspraken niet op voorhand helder en eenduidig zijn geformuleerd, bestaat het risico dat iedere partij afzonderlijk oplevert wat is gevraagd, maar dat het eindresultaat niet is wat er nodig is. Wij zijn gewend om tijdens een traject te schakelen met meerdere partijen op verschillende mensen om een product op te leveren dat voldoet aan de eisen van onze stakeholders. 7. Onze ervaring is dat tijdens de ontwikkelfase uitstekend samengewerkt kan worden met andere leveranciers in de ontwikkel-, implementatie- en beheerfase. Afspraken over samenwerking, afstemming en verantwoordelijkheden dienen duidelijk vastgelegd te worden.
--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		Ontwikkelfase: Bij het volgen en correct toepassen van het IPLS-development train-concept (of uiteraard een soortgelijke aanpak), zijn verschillende vormen van samenwerking mogelijk tijdens de ontwikkelfase. Het is in dit concept perfect mogelijk om teams (wagons) van verschillende leveranciers te laten samenwerken en te laten ondersteunen/begeleiden door experts van verschillende leveranciers (in de rol van bv. IPLS-consultant). Wij staan dan ook open om – op basis van de te ontwikkelen component/applicatie en de daarvoor nodige expertise – de meest aangewezen samenwerking tussen verschillende leveranciers aan te gaan. Meer concreet (als voorbeeld) kan er bv. samengewerkt worden (waarbij elk vermeld deel door een andere leverancier wordt uitgevoerd) op het vlak van design (UI/UX) creatie, designimplementatie, back-end ontwikkeling, front-end ontwikkeling, integraties met externe systemen, testing, code reviewing, ... Uiteraard zal er tijdens de ontwikkelfase ook de nodige aandacht besteed moeten worden aan bv. de vereiste server-omgevingen, de server dimensioning, kennisoverdracht ter voorbereiding van M&S, ... Hierdoor overstijgt de samenwerking kavel 3 en zal er ook samengewerkt moeten worden met leveranciers uit kavel 1 en kavel 2. Implementatiefase: Tijdens de implementatiefase is vanzelfsprekend een intensieve samenwerking noodzakelijk tussen leveranciers uit de 3 kavels. De leverancier uit kavel 1 moet op de juiste wijze voorbereid worden om de ontwikkelde integraties te activeren, eventuele trainingen te kunnen geven, M&S te kunnen uitvoeren, etc. De leverancier uit kavel 2 moet dan weer alle voorbereidingen m.b.t. de servers kunnen treffen voor bv. het live gaan. Bij IPLS-development train concept vindt deze samenwerking vooral plaats op het niveau van de locomotief. Gebruikers-/beheersfase: Het IPLS-concept schrijft voor dat het team dat de ontwikkeling heeft uitgevoerd ook betrokken blijft en de verantwoordelijk op zich neemt voor de opgeleverde ontwikkeling. Concreet betekent dit dat het team dat een module ontwikkeld heeft ook verantwoordelijk blijft (of een belangrijke rol op zich neemt) voor bv. bugfixing. Op deze manier is een directe toegang verzekerd tot de kennis van een module en kunnen tickets efficiënt worden afgehandeld. Vooral bij blocking en kritische issues heeft dit zijn meerwaarde al meermaals aangetoond. In lijn met het IPLS-concept, zien wij dan ook tijdens deze fase nog steeds een nauwe samenwerking tussen de verschillende kavels. Kavel 1 staat in voor een correct uitvoeren van o.a. de supportcontacten, kavel 2 zorgt voor een stabiele hosting, kavel 3 voorziet de kennis en expertise voor de bugfixing. Extra: Om de samenwerking in de 3 fases te ondersteunen en te faciliteren, zien wij ook een noodzaak aan het gebruiken van een projectopvolgingstool die doorlopend in alle fases kan ingezet worden zodat op elk moment de juiste informatie snel teruggevonden en opgevolgd kan worden. 8. Omdat wij ontwikkelen met het Mendix low-code platform zijn wij gewend om heel dicht tegen
--	--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	onze eindgebruikers aan te zitten tijdens de ontwikkeling. We ontwikkelen dan ook interactief en op Agile werkwijze. In een vroeg stadium kan worden bijgestuurd. Leveranciers werken onder een product owner gezamenlijk aan het desbetreffende product. Uiteindelijk betreffen dit ontwikkelaars en is het niet relevant voor welk bedrijf je werkt, maar gaat het vooral om de expertise die je inbrengt, de relevante ervaring in het specifieke domein waar je product voor wordt ontwikkeld en je enthousiasme voor het eindresultaat. Dimpact blijft (onze eerste aanname) regie houden en is aanspreekpunt voor de gemeenten en verantwoordelijk voor het proces. Ik zie Dimpact in feite ook als 'hoofdleverancier' en gemeenten als gezamenlijke afnemer. Dimpact houdt de regie op de toeleveranciers (de ICT leveranciers), de toeleveranciers leveren werkende software op waar de gemeenten samen de wensen/eisen voor opstellen. Bij implementatie begeleidt Dimpact (de product owner binnen Dimpact) de gemeente functioneel en de toeleverancier (software ontwikkelaar, SI) technisch. Omdat het open source software betreft, ontwikkeld onder regie van Dimpact, betalen de gemeenten een bijdrage voor de beheerfase. Dimpact pakt dit 1e lijns op en besteedt dit aan de achterkant uit aan leveranciers die de componenten of applicaties hebben ontwikkeld. 9. Voordat de ontwikkelfase start herkennen wij ook nog een solution design / sprint 0 / backlog voorbereidingsfase. Ondanks dat we in korte iteraties werken is het belangrijk om de al dan niet reeds omschreven requirements te vertalen, visualiseren, valideren en vervolgens "development ready" te krijgen. In deze solution design / sprint 0 / voorbereidingsfase is het qua samenwerking belangrijk dat alle betrokken partijen de kans hebben gehad om hun voorbereidingen te treffen voor hun specialisme: - Een front-end team zou bijvoorbeeld al graag feedback geven op interface ontwerpen, en design-teams bijsturen om een reeds ontwikkeld component te gebruiken ipv iets dat ontwikkeld moet worden. - Een back-end team zou graag al op de hoogte worden gebracht van het domein datamodel zodat herbruikbare registers kunnen worden geïdentificeerd. - Een integratieteam zou al betrokken kunnen worden om toekomstige behoeften voor integraties af te stemmen met te integreren producten, of samen met het solution design team API contracten af te spreken voor op te leveren koppelingen. In de ontwikkelfase is er een continue intensieve samenwerking tussen front-end, back-end en devops/hosting engineers. Waar het mogelijk is treft één gezamenlijk samengesteld scrum team de voorkeur waarbij alle partijen en product owner via standups, reviews en andere agile events gesynchroniseerd blijven. Als om wat voor reden dan ook het opstellen van 1 scrum team niet mogelijk is is een synchronisatie tussen verschillende scrum teams onvermijdelijk. Een structurele, overkoepelende "scrum-of-scrums", of dependency mapping afspraak tussen de verschillende product owners + team leads is hierbij een uitkomst om onderlinge afspraken te maken over naderende integratiemomenten en fallback scenario's als integratiemoment niet gehaald worden. In de gebruikers-/beheerfase is het belangrijk om de verantwoordelijkheid voor eerstelijns support duidelijk bij een partij te leggen, logischerwijs bij de partij die het meeste inzicht heeft welke vragen en issues bij welke leverancier vandaan komen, of opgelost kunnen worden. De samenwerking zal dan afhankelijk zijn van de onderhoudscontracten die worden gesloten met de verschillende leveranciers, waarbij er rekening
--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		moet worden gehouden met de verschillende reactietijden. In een ideaal scenario wordt een soortgelijk scrum team als tijdens de eerste bouwphase aangewezen om door te ontwikkelen in de run-fase, zodat er een hoge continuïteit bestaat, kennis in het team vers blijft en er niet alleen wordt beheerd maar ook wordt doorontwikkeld op basis van voortschrijdend inzichten in gebruiker- en stakeholder-feedback. 10. Over onze visie op samenwerking en onze ervaringen hierin, hebben we hierboven en bij vraag 2 al het een en ander geschreven. Kortgezegd, we denken dat samenwerking een grote meerwaarde heeft. In de ontwikkelfase kunnen we ons voorstellen dat meerdere ontwikkelpartijen tegelijkertijd aan dezelfde applicatie werken, bijvoorbeeld door het werk in te delen in front-end- en back-end werk. Daarnaast is het mogelijk (geen must) dat een aparte partij het testwerk doet. Ook zou het kunnen dat partijen elkaar afwisselen, dus bijvoorbeeld dat een partij verder bouwt op code die door een eerdere partij is geschreven. We zien hierbij wel mogelijke haken en ogen. 1. Beide partijen dienen met dezelfde programmeertaal te werken, en hier een hoge beheersing in te hebben. 2. Ook binnen een programmeertaal en ontwikkelplatform zijn er veel vrijheden die verschillende partijen anders kunnen invullen. Aanvullend kunnen er gebreken aan het licht komen, of dingen die minder handig geprogrammeerd zijn, of organisch zo gegroeid zijn. En hoe ga je daarmee om? Opnieuw bouwen en daarmee het fundament goed leggen? Of pragmatisch doorgaan met oude code: wat op de korte termijn leidt tot veel snellere doorontwikkeling, maar op de lange termijn... Het is goed om hier van tevoren over na te denken. Tenslotte, voor de gebruikers-/beheerfase is het belangrijk is dat er met alle betrokken partijen goede afspraken gemaakt worden in de SLA over beschikbaarheid bij zaken die spoed vereisen. 11. In de ontwikkelfase van projecten is er een DevOps Engineer vanuit kavel 2 betrokken om te zorgen voor een goede landing van de individuele softwareprojecten op de generieke Haven Common Ground-omgeving. Dit borgt de standaarden en flexibiliteit in partnerkeuze voor de software. De toetsing en borging van schaalbaarheid, security-aspecten en performance borgt de partner van kavel 2. Bij oplevering van de projecten dient er een goede overdracht plaats te vinden van operations naar de servicedesk van de systeemintegratiepartner. Mochten er alsnog vragen of issues zijn vanuit de gemeenten, dan pakt de servicedesk deze op. 12. Wat wij voor ons zien, is dat wij met meerdere Scrum Teams samenwerken en dat onze mensen in verschillende teams kunnen deelnemen. Een Scrum Team zou hierbij kunnen bestaan uit één, maar ook meerdere leveranciers. Zo hebben wij een aantal projecten uitgevoerd met zo'n gecombineerd ontwikkelteam. De eerlijkheid gebiedt ons te zeggen dat het in het begin altijd even zoeken is naar een effectieve en efficiënte vorm, maar dat dit na een aantal sprints goed kan functioneren. Hierbij kunnen verschillende partijen elk een specifieke rol pakken. Bijvoorbeeld het ontwikkelen van alle backend, of juist de frontend of het testen, en ga zo maar daar.
--	--	---

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		In de beheerfase ligt dit anders. Hierbij zijn er meestal geen sprints waarin de teams actief samenwerken, maar wordt er veelal gewerkt op basis van een melding en een proces. Er zal dan ook vooral geïnvesteerd moeten worden in het gezamenlijk ontwikkelen van beheerprocessen inclusief een SLA. 13. Ontwikkelfase: Een andere leverancier betrokken maar bij voorkeur een klant (lees gemeente) is wenselijk. Er dienen duidelijke afspraken tussen leveranciers over scope te worden gemaakt. Er dient 1 integraal team te worden neergezet die kan bestaan uit mensen van verschillende leveranciers. Implementatiefase: Hierin dient te worden samengewerkt met system integrator. Gebruik/beheerfase: Vanwege single-point-of-contact dient hier bij voorkeur 1 van de ontwikkelende leveranciers in de lead te zijn. Hierin dient de system integrator te zorgen dat de feedback bij deze leverancier terecht komt. Een sprint om doorontwikkeling te starten wordt idealiter met hetzelfde team aan mensen te worden gedaan. Hiermee kan je bouwen met bestaande samenwerking.
11	<i>Hoe ziet u uw rol bij het inpassen van (nieuw) ontwikkelde Componenten in de bestaande Dimpact-stack? Wat is uw ervaring met het afstemmen van software op het totale Portfolio?</i>	1. Wij staan geheel achter de 5 lagen architectuur van Common Ground. Het principe dat iedere applicatie in theorie foutloos zou moeten communiceren is natuurlijk nog niet het geval. Zolang nog niet alle applicaties met de ZGW standaard werken zal er een brug moeten komen tussen oud en nieuw. 2. Bij de ontwikkeling van nieuwe software componenten in een bestaand portfolio is het heel belangrijk om de uniformiteit te bewaken en standaarden op te stellen waaraan de componenten moeten voldoen. Interoperabiliteit tussen applicaties is alleen mogelijk door standaardisatie van de verschillende koppelvlakken tussen de applicaties. Wij onderschrijven de uitgangspunten van de Common Ground visie. Door landelijk gestandaardiseerde APIs te gebruiken kan hergebruik plaatsvinden. 3. Onze ontwikkelaars hebben veel ervaring met het ontwikkelen van software die binnen een ICT landschap gebruikt wordt, modules die in samenhang met andere (open source) modules gebruikt worden. Juist door het gebruik van Open Standaarden is het mogelijk om oplossingen te realiseren die in iedere stack gebruikt kunnen worden. Om dit te realiseren is een functioneel en een technisch ontwerp essentieel om de samenwerking tussen de software en componenten te realiseren. 4. Wij zien onze rol als een ontwikkelaar-op-basis-van-standaarden. Hergebruik van componenten is in een complexe landschap alleen mogelijk bij het zo zuiver mogelijk inzetten van standaarden, en waar de standaarden niet voldoen om die samen met gemeenten en andere overheden die standaarden vast te leggen of te verbeteren. Dit is in onze ogen de enige mogelijke weg om Common Ground succesvol te maken en tot een werkbaar alternatief te komen.

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	5. Wij denken op alle lagen van de Common Ground architectuur een bijdrage te kunnen leveren. Implementatie zou in eerste instantie gericht moeten zijn op het bouwen van een Minimum Viable Product waarna, door middel van feedback uit de markt, doorontwikkeling kan plaatsvinden. Voor het bepalen van het MVP is het belangrijk dat we dit in gezamenlijke sessies met zoveel mogelijk verschillende stakeholders (techneuten, process owners, MT) en een vastomlijnd proces doen. Onderhoud en support zien wij als een integraal onderdeel en verantwoordelijkheid na oplevering. Voor vrijwel ieder nieuw project dat wij ontwikkelen, doen wij dit zo slim en efficiënt mogelijk, waarbij een groot deel van de applicatie bestaat uit herbruikbare componenten dat door teams wereldwijd worden gebouwd, onderhouden en toegepast. Ook hebben wij zelf diverse open-source componenten opgeleverd in het verleden. 6. Als leverancier van open source software hebben wij veel kennis en ervaring opgedaan met de ontwikkeling van modules, waarbij de werking op een bestaande omgeving of core ontwikkeld wordt. De aanpak is hierin helder, aan de hand van een analyse volgt een functioneel en technisch ontwerp. Deze wordt getoetst en gevalideerd. Vervolgens de software of het component ontwikkeld. Waar nodig vindt er tussentijds afstemming andere leverancier plaats. Tot slot wordt de software of component getest en in gebruik genomen. Software en componenten die wij ontwikkelen maken altijd onderdeel uit van een geheel. 7. Hier kunnen wij zeker een rol op ons nemen. Wij zien deze rol op verschillende niveaus: - Het mee nadenken over de vereiste architectuur (als ondersteuning voor het Dimpact architectuurteam) om het inpassen mogelijk te maken. - Afhankelijk van de nodige kennis en expertise kunnen wij ingezet worden om bestaande componenten uit de Dimpact-stack door te ontwikkelen of eventueel aan te passen of te optimaliseren. - Het leggen van connecties tussen de verschillende componenten of het mee nadenken over de meest geschikte wijze om de connecties te leggen. - We kunnen ingezet worden voor het testen van de ingepaste componenten. 8. De kracht van het Mendix low-code platform zit hem juist in het inpassen van nieuwe componenten op bestaande technology stacks, waaronder ook oudere legacy systemen. Zo bouwen we nieuwe innovaties op oudere SAP-systemen, maar ook op andere ERP-oplossingen. 9. Zoals de kavelverdeling nu is opgesteld zien wij een belangrijke rol voor de systeemintegratie partij in het integreren van nieuwe oplossingen in bestaande landschappen. Het is zoals eerder al vermeld van groot belang dat bij de bouw van nieuwe componenten rekening wordt gehouden met de benodigde aansluitingen op andere systemen. De applicatieontwikkelteams maken gebruik van het common ground
--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		lagenmodel, waarin op laag 2 de services en API's worden gedefinieerd. Als er voorkennis is over de te integreren applicaties treft het de voorkeur om al bij de inrichting van de register interfaces rekening te houden met de te integreren applicaties. Onze ervaring met het afstemmen van software op het totale Portfolio of landschap is dat de verwachtingen van integreerbaarheid vaak veel groter en rooskleuriger zijn dan de realiteit, zeker als er geïntegreerd wordt met kant en klare oplossingen waarbij er beperkte of geen invloed is op de ontwikkel roadmap en de mogelijkheden van de beschikbare API's. Dankzij abstractie / lagenstructuur van common ground is een aansluiting van verschillende (toekomstige) common ground compliant applicaties een veel betrouwbaarder vooruitzicht. 10. Wij zien het als onze rol om Dimpact te adviseren bij het optimaal inzetten van de gevraagde functionaliteit, en kijken daarbij naar stabiliteit, onderhoudbaarheid, flexibiliteit en kwaliteit. Daarnaast is het als softwarepartner belangrijk Dimpact op de hoogte te brengen van de laatste ontwikkelingen op dit specifieke deelgebied. Als Europese speler op het gebied van digitale end-to-end oplossingen hebben wij veel ervaring met het realiseren van een verschillende soorten oplossingen, zoals web interfaces, mobile apps, integratie-oplossingen, single sign-on en security-vraagstukken. 11. Wij zien onze rol als analist over interacties tussen componenten, functioneel ontwerper, full-stack ontwikkeling, implementatie en beheer. Het is essentieel om voor ontwikkeling duidelijk te hebben welke relaties binnen het portfolio moeten kunnen worden gelegd. Hiermee kan het componentontwerp worden aangepast op basis van de behoefte. Doordat Common Ground principes worden gebruikt en communicatie via API's gaat kunnen componenten relatief gemakkelijk worden geïmplementeerd als uitgefaseerd.
12	<i>Welke rol hebben ontwikkelaars nadat het Softwareproduct/ Component is opgeleverd? Denk hierbij aan kennisoverdracht, training of beheer van documentatie. Maar bijvoorbeeld ook aan de rol bij bugfixing of het beheer/bewaken van broncode. Hoe zorgt Dimpact ervoor dat Componenten altijd voldoen aan de meest recente en up-to-date technieken en beveiligingsniveaus.</i>	1. Wij zijn tegenwoordig gewend om vanuit Devops AGILE te werken. Dit staat haaks op waterval teams in leven houden. Wij hebben met allebei ervaringen en kunnen uiteindelijk op allebei de methodes werken. Echter zijn wij van mening dat werken volgens de agile devops methode beter de kennisoverdracht, kennis, beheer en verantwoordelijkheid van a-z per component waarborgt. 2. Wij zijn een groot voorstander van component eigenaren analoog aan core committers van open source projecten. Als een component is opgeleverd inclusief de documentatie benodigd voor installatie en doorontwikkeling zal het life-cycle beheer moeten worden georganiseerd. 3. Welke rol ontwikkelaars hebben na het opleveren van de component is sterk afhankelijk van de keuze hoe de software ontwikkeld zal worden en gehost. Als dit wordt vormgegeven volgens kavel 2 en 3 (Haven en Open source) dan is er veel behoefte aan kennisoverdracht, training en documentatie. Hier is naar onze mening dan Dimpact of de community verantwoordelijk. Met

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	betrekking tot up-to-date technieken en beveiligingsniveaus ligt de verantwoordelijkheid ook bij de community. Dimpact zal goed moeten nadenken hoe dit te organiseren zodat de diverse partijen in de samenwerking, ieder in zijn eigen rol, weten wat van hen verwacht wordt. Een keuze voor een SaaS-dienst biedt een hoop ontzorging op dit gebied. 4. Wij denken dat alle rollen geheel of gedeeltelijk bij de leverancier van het component horen. Wanneer er wordt gekozen voor Haven-architectuur in combinatie met open source, dan lijkt ons dat zaken als kennisoverdracht, training en beheer primair bij de gebruikerscommunity liggen. De ontwikkelaar kan dan hooguit nog een ondersteunende rol vervullen. Het zorgdragen voor bugfixes en beveiliging ligt in zo'n situatie ook bij Dimpact, waarbij voorstellen voor verbetering van techniek en/of beveiliging vanuit de gebruikers dienen te komen. 5. Dit is een kwestie van heldere afspraken die wij in samenspraak met u en met eventuele andere ontwikkelaars maken. Ons uitgangspunt is dat er altijd sprake is van het toepassen van open standaarden, geautomatiseerde testen, collegiale review en het vastleggen van documentatie, niet alleen in code, maar ook in een technisch ontwerp. Voor ons is het gebruikelijk dat wij na oplevering van software, modules of componenten het onderhoud en beheer verzorgen, denk hierbij aan preventief, correctief en adaptief onderhoud. 6. Normaal gesproken (bij projecten buiten Common Ground om, vaak punt-oplossingen) wordt er een support-en-hosting overeenkomst afgesloten waarin hosting en onderhoud van de applicaties wordt overeengekomen. Waar er voor eigen hosting wordt gekozen wordt dit beperkt tot een supportovereenkomst (technisch applicatiebeheer en onderhoud van de applicatie). Binnen zo'n overeenkomst wordt (interne) kennisoverdracht en documentatiebeheer afgesproken. Trainingen worden bij het in productie nemen van een applicatie gegeven en na livegang op ad-hoc basis (bijvoorbeeld voor het inwerken van een nieuwe functioneel beheerder). Wij zouden adviseren voor een 'instandhoudingsovereenkomst' die specifiek toeziet op communitysupport, bugfixes en het up-to-date houden van een component. Dan heeft Dimpact een basis gelegd voor beheer, en waar er training of kennisoverdracht nodig is zouden wij aanraden om leveranciers dit onderling te laten regelen, dan is er ruimte voor marktwerking. Doorontwikkeling is zeer afhankelijk van verschillende factoren (nieuwe standaarden, nieuwe wensen en eisen, verbreding van de doelgroepen gebruikers etc). Dit is op voorhand niet in te schatten en spreken we in de regel op ad-hoc sprint-basis af. 7. Het ontwikkelteam blijft verantwoordelijk voor het product/component ook na oplevering. Onderhoud en support zijn wezenlijke onderdelen hiervan. Technische documentatie wordt automatisch gegenereerd tijdens ontwikkeling. Daarnaast adviseren wij dat functionele documentatie ook wordt opgezet en opgenomen in de (publiek toegankelijke) repository om (her)gebruik van het product/componenten te
--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		bevorderen. Door gebruik te maken van tooling (zoals bijvoorbeeld Dependabot) kan veiligheid van de software gewaarborgd worden. Kwetsbaarheden worden op deze manier actief gepatched. Onderhoud wordt door ons actief doorgevoerd op producten en componenten die wij leveren. Denk hierbij aan het upgraden van frameworks die worden toegepast of het met terugwerkende kracht doorvoeren van best practices. 8. Onderdeel van het ontwikkel en implementatiefase is het beschikbaar stellen van documentatie en het verzorgen van kennisoverdracht door middel van een demonstratie, training of handleiding. Technische documentatie wordt volgens vooraf vastgestelde standaarden vastgelegd. 9. Wij zien deze rol op verschillende domeinen: Kennisoverdracht: Ontwikkelaars (en in het verlengde van de ontwikkelaars de test engineers / business analisten) spelen een cruciale rol bij de kennisoverdracht naar zowel kavel 1, kavel 2, maar ook andere leveranciers in kavel 3. Deze kennisoverdracht gebeurt op het niveau van de documentatie, maar ook via (face-to-face) trainingen en meetings. Ter info: als IPLS gecertificeerd bedrijf zijn wij verplicht om voor elk project een PBP op te stellen en te volgen. PBP staat voor People Back-up Plan, wat inhoudt dat voor elke sleutelfiguur een stand-in (back-up) wordt aangewezen die zich mee inwerkt. Valt de sleutelfiguur weg, dan neemt de stand-in over. Dit zorgt automatisch voor een grote focus op kennisoverdracht. Wij raden het opstellen van een PBP (of een soortgelijk plan) per te ontwikkelen component aan. Bijhouden en up-to-date houden van de documentatie: In het geval de ontwikkelaar betrokken is bij de doorontwikkeling, dan is het aanbevolen dat dit team ook instaat voor het bijhouden van de documentatie over deze doorontwikkeling en het informeren (bv. middels een training) van de andere betrokken kavels en leveranciers. Wij raden aan om hiervoor vaste meeting-momenten in te bouwen in de planning. Bewaken van de broncode: Indien de ontwikkelaar ook instaat voor de doorontwikkeling, dan is het logisch dat dit team ook instaat voor het beheer en bewaken van de broncode. Indien de ontwikkelaar niet betrokken is bij de doorontwikkeling, dan bevelen wij aan dat de initiële ontwikkelaar nog steeds de verantwoordelijkheid blijft dragen voor de broncode. Dit kan ingebouwd worden door dit team in te schakelen in het code reviewing proces, maar ook middels DevOps tools en de daarbijhorende rapporten. Bugfixing: Het IPLS-concept schrijft voor dat het team dat de ontwikkeling heeft uitgevoerd ook betrokken blijft en de verantwoordelijk op zich neemt voor de opgeleverde ontwikkeling. Concreet betekent dit dat het team dat een module ontwikkeld heeft ook verantwoordelijk blijft (of een belangrijke rol op zich neemt) voor bv. bugfixing. Op deze manier is een directe toegang verzekerd tot de kennis van een module en kunnen tickets efficiënt worden afgehandeld. Vooral bij blocking en kritische issues heeft dit zijn meerwaarde al meermaals aangetoond.
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		Concreet betekent dit dat wij voor het ontwikkelingsteam ook een belangrijke rol weggelegd zien bij de bugfixing. Updates en upgrades: Ook op dit vlak zien wij een rol weggelegd voor de initiële ontwikkelaar. Ook al wordt de M&S aangestuurd vanuit kavel 1, toch is het aangewezen dat het initiële ontwikkelingsteam ook een bijdrage levert bij het uitvoeren van updates en upgrades. Deze rol kan een leidende rol zijn of een ondersteunende en begeleidende rol. Per slot van rekening is dit team de specialist op het vlak van de code en de applicatie en kunnen zij wellicht beter inschatten welke risico's en eventuele regressie-issues er zouden kunnen optreden bij een update/upgrade. Ons inziens door hier actief beleid rond te voeren en hiervoor een duidelijk plan op te stellen. Dit plan zou volgende punten minimaal moeten omvatten: 1. Het aanstellen van een security officer. 2. Duidelijke richtlijnen en processen voor updates/upgrades en security. 3. Een controlemechanisme of controleproces. 4. Een security test plan. 5. Het uitbouwen van een cross-kavel team dat zich toelegt op update- en securitybewaking. Als we voor een laatste keer mogen verwijzen naar IPLS, dan durven wij ook hier een aantal van de te volgen richtlijnen van IPLS toelichten/voorstellen. - IPLS vereist zogenaamde non-negotiable processen voor het implementeren van updates en security maatregelen. Dit zijn processen die door alle teams zonder fout moeten gevolgd worden. Wij raden dit ook aan in de set-up die Dimpact voor ogen heeft. Alle IPLS- development trains volgen dan steeds dezelfde update en security maatregelen en processen. Samen met deze maatregelen kunnen ook checklists voorzien worden die de teams moeten bijhouden (ter info: het kunnen aanmaken van deze checklists is een ingebouwd feature van Lienion, een projectopvolgingstool). - IPLS vereist een security testplan inclusief het uitvoeren van regelmatige en periodieke security tests zoals bijvoorbeeld voor en na een release maar ook minimaal 1 keer (onaangekondigd) per kwartaal. Deze security tests moeten uitgevoerd worden door een team dat niet betrokken is bij de eigenlijke ontwikkeling en zo op een onafhankelijke wijze deze tests kan uitvoeren. In de set-up van Dimpact zou er een team kunnen opgezet worden (bestaande uit betrokkenen van de verschillende kavels en aangestuurd door de security officer) die deze tests uitvoert. 10. Onderdeel van open-source is dat die verantwoordelijkheid ook deels bij 'de community' ligt. Van belang is dus om onderhoudbaarheid van code op een hoog niveau op te leveren. Dimpact zou ook zelf kennis kunnen opbouwen. In hetzelfde voorbeeld van gemeente Rotterdam, hebben zij ook eigen (Mendix low-code) developers in dienst die de kennis van de nieuwe stack deels borgen. In het Mendix low-code platform dat wij gebruiken om software te ontwikkelen bouw je samen met de business prototypes uit naar werkende oplossingen. Het gebruik is dermate eenvoudig dat iedereen met enige kennis van software ontwikkeling heel snel met low-code aan
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		de slag kan. Door te ontwikkelen binnen het low-code platform zijn veel zaken rondom gebruikte technieken (denk bijvoorbeeld aan Mobile op basis van ReactNative, Cloud Native architectuur, Single Sign On, etc.) up-to-date doordat dit in het platform doorontwikkeld wordt. Daarnaast is de leverancier verantwoordelijk om "vulnerabilities" op te lossen en beveiligingsniveaus up-to-date te houden (volgens SLA en certificeringen). 11. Ontwikkelteams blijven betrokken nadat een component is opgeleverd voor het uitvoeren van software-updates, -upgrades en -iteraties. Daarnaast zorgen de ontwikkelteams voor goede documentatie, kennisoverdracht en training. Dit gebeurt op een gestandaardiseerde manier, zodat andere ontwikkelaars eenvoudig door kunnen ontwikkelen of de software goed kunnen onderhouden. Om te borgen dat Dimpact altijd voldoet aan de meest recente en up-to-date technieken en beveiligingsniveau adviseren wij standaard Software Lifecycle-processen in te richten. Hierbij sluit men op applicatieniveau (dus niet op hostingniveau) een gestandaardiseerde SLA af met de desbetreffende partner. Wij raden u aan om rekening te houden met de volgende aspecten: • proactief onderhoud voor het monitoren van updates, upgrades en patches; • onderhoud van de applicatie; • als onderdeel van de SLA heldere afspraken over dependency management tussen Dimpact en de partners van kavel 3. Hierin speelt Dimpact een leidende rol voor het stellen van kaders; • afspraken over eventuele doorontwikkeling. Verder is het belangrijk dat de partner van kavel 1 inzicht heeft in, monitort op en rapporteert op de gestandaardiseerde SLA's van de partners van kavel 2 en 3. 12. Voor ons is het goed documenteren en zorgdragen van een juiste kennisoverdracht onderdeel van de oplevering. Ontwikkelaars begroten dus ook tijd om dit als onderdeel van de softwarewijziging te doen. Anders gezegd, het zit wat ons betreft in de acceptatie-criteria. Het komt uiteraard voor dan na een aantal sprints blijkt dat er toch een aantal bugs zit in de opgeleverde software. Wij zijn gewend om ook bugs middels een refinement te behandelen. Tijdens deze refinement zal blijken wie de beste ontwikkelaar is voor het oplossen van de bug binnen een sprint. In de meeste gevallen is dit de ontwikkelaar die het project heeft opgeleverd. Het kan ook voorkomen dat die ontwikkelaar om redenen niet inzetbaar is. Dan zal zijn collega het moeten overnemen. Om ook deze situatie goed te kunnen verzorgen is een goede documentatie en refinement van belang. Onze oplossingen, en daarmee ook de ontwikkeling daarvan, zijn onderhevig aan verschillende kwaliteitssystemen die onafhankelijk worden getoetst. Denk hierbij aan ISO9001, ISO27001 en de ENSIA. Als onderdeel van het beheer en onderhoud wat wij uitvoeren, betekent dat dat wij procesmatig onze softwarecomponenten langsgaan en een impactanalyse maken van de eventuele noodzakelijke updates die uitgevoerd moeten worden. Denk hierbij aan het updaten van diverse libraries. Ook dit zal vervolgens via een refinement in een sprint worden gepland.
--	--	--

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		13. Oplevering documentatie met hierin de (technische) kennis over de component. Zoals eerder gesteld zien wij de ontwikkelaar als essentiële partij om ook tijdens beheer betrokken te zijn. Partijen die zelf de code hebben geschreven zijn sneller in staat om bugs op te lossen dan een derde partij die de code eerst eigen dient te maken. Wij verwachten van de system integrator kwaliteitschecks en feedbackloops over de componenten.
4	Overige vragen	Antwoord
13	Wat spreekt u aan, aan de beoogde aanbestedingen? Wat niet? Wat maakt het voor u interessant om zich in te schrijven op een mogelijke aanbesteding?	1. De migratie daadwerkelijk mogelijk maken om het Common Ground principe. En hierbij koploper zijn van de transparante werkmethodes die wij eigenlijk over de hele industrie hopen te gaan zien bij leveranciers. 2. De aspecten die ons aanspreken zijn: • Open source software ontwikkeling: soms een klein beetje eng, want alles is transparant, maar wel uitdagend en geeft veel voldoening • Werken middels de common ground principes: wij zijn de grondleggers van Agile werken in Nederland en hebben er zelfs een boek over geschreven: Scrum for Dummies • Werken voor het publieke domein. Het is altijd prettig als je de maatschappij een stap verder kan helpen. De aspecten die ons minder aanspreken zijn: • Minder DevOps werkzaamheden dan we gewend zijn. • De noodzakelijke inspanning die vereist is om aan aanbestedingen mee te doen. De aspecten die ons aanspreken zijn voldoende om het voor ons interessant te maken op mee te doen met een mogelijke aanbesteding. 3. Wat ons aanspreekt is de innovatieve wijze waarop Dimpact dit wil gaan vormgeven. Het brengt uiteraard ook risico's met zich mee, maar het zal ervoor zorgen dat er (versneld) een hoop gestandaardiseerde componenten binnen de visie Common Ground ontwikkeld gaan worden. Hosting op basis van Haven spreekt ons niet aan als leverancier van SaaS-toepassingen. Deze marktconsultatie geeft ons nog te weinig inzicht in hoe de aanbesteding zal uitzien. De aanbesteding zal op een hoop vlakken, zeker de samenwerkingsvorm en afspraken daarbinnen, meer duidelijk moeten verschaffen. 4. Wat ons aanspreekt is de keuze voor de common ground architectuur en standaarden omdat deze meer duidelijkheid geven dan voorgaande standaarden. Wat ons niet aanspreekt is de exclusieve keuze voor Haven en Open Sourceproducten. Deze keuze sluit producten van ons en mogelijk andere leveranciers uit, terwijl die op basis van Common ground wel uitstekend zouden passen.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

	5. Wij zien het als een belangrijke stap van Dimpact om van individuele overeenkomsten naar bredere raamovereenkomsten te gaan waarbinnen opdrachten kunnen worden uitgezet. Wij maken ons wel zorgen over de mate waarin wij als leverancier nog in onze kracht worden gezet en de ruimte krijgen om met een zo goed mogelijke oplossing te komen waarbij Dimpact wordt ontzorgd. Teveel voorwaarden of restricties kunnen een reden zijn voor ons om niet in te schrijven, waarbij het gedwongen samen ontwikkelen nog wel het meest belemmerend is in onze ogen. Interessante aanbestedingen voor ons zijn concreet met een heldere scope waarin wij onze creativiteit en expertise zo goed mogelijk kunnen benutten, wat na ontwikkeling tot een inzetbaar component leidt. Een waar er tevens breed behoefte aan is binnen de overheid. 6. Dimpact en haar lid gemeenten hebben grote ambities. Wat ons hierin met name aanspreekt is de visie op het gebied van open source, herbruikbaarheid, uitwisselbaarheid en toepasbaarheid. Ons beeld hierin is dat Dimpact voor gemeenten additionele toegevoegde waarde kan creëren. Aan de hand van de beschrijving van deze marktconsultatie zien wij ook een aantal risico's ontstaan. Hoe zijn verantwoordelijkheden daadwerkelijk geregeld, hoe wordt de generieke oplossing gerealiseerd en hoe zorg je ervoor dat leveranciers samenwerken en niet naar elkaar gaan wijzen. Deze risico's zijn beheersbaar mits de visie, doelstellingen en verwachtingen helder zijn en er gewerkt kan worden in een SAFE Agile omgeving. Daarmee kom je samen echt verder. 7. Algemeen vinden wij de set-up die Dimpact voor ogen heeft de 'way-to-go' om op een kwalitatieve en efficiënte wijze de nodige componenten te ontwikkelen voor de doelgroep en zodoende de Dimpact-stack verder uit te bouwen. De aanpak vereist wel een degelijke, weldoordachte organisatorische set-up maar indien Dimpact erin slaagt om deze set-up te verwezenlijken, dan zal dit resulteren in een zeer krachtige ontwikkelingsorganisatie. Wij zouden graag meewerken aan het welslagen van deze set-up en menen dat wij door onze ervaring met het ontwikkelen van op maat gemaakte applicaties en het IPLS-concept, hieraan kunnen bijdragen. 8. Het feit dat er al veel gemeenten samen georganiseerd zijn onder Dimpact maakt dat je voldoende slagkracht en budget hebt om professionele en moderne software applicaties te bouwen die voldoen aan de meest moderne wensen en eisen van vandaag de dag. Het innovatieve karakter van het type trajecten binnen Common Ground en de enorme complexiteit die gepaard gaan met het bouwen van applicaties voor gemeenten op domeinen zoals werk & inkomen, maar ook handhaving en andere domeinen, maakt dat wij onze medewerkers van uitdagende projecten kunnen voorzien die tevens ook gelijk een maatschappelijk karakter hebben. Dat spreekt onze mensen enorm aan en dat maakt dat deze aanbesteding extra interessant is voor ons als software ontwikkel bedrijf.
--	--

Resultaten marktconsultatie Dimpact.

Dit document bevat de geanonimiseerde (niet vertrouwelijke) antwoorden van de geïnteresseerde partijen. Antwoorden waarin bijvoorbeeld concrete voorbeelden (met namen van organisaties benoemd) en antwoorden welke doorspekt zijn met de bedrijfsnaam zijn weggelaten om de anonimiteit te borgen.

		9. Wat ons aanspreekt is de filosofie om open source te ontwikkelen. Het open karakter (op dit moment) van de uitvraag biedt kansen om als organisatie vanuit onze expertise mee te denken en invulling te geven aan de opdracht. Dat laatste is ook een risico om alle meewerkende partijen op één pad te houden. Daarvoor dient de scope helder te zijn en raden wij aan om (samen) te werken op basis van Holacracy met Scrum.
14	<i>Wilt u nog andere zaken met ons delen?</i>	1. Ofschoon wij het Common Ground principe toejuichen, attenderen wij Dimpact er graag op dat het essentieel is om voldoende oog te hebben voor versiebeheer en security. Niet alleen zijn wij daarom ISO 27001 gecertificeerd, ook zijn wij gewend om vooraf aan de aanbesteding met IT-manager de veiligheidsmaatregelen te bespreken. Nu zijn we er ons van bewust dat Dimpact een verscheidenheid aan gemeentes vertegenwoordigd, Desondanks lijkt het ons verstandig om als Dimpact zijnde vanaf het begin oog te hebben voor deze veiligheidsmaatregelen. Aangezien mogelijke opdrachten ook de burger kunnen raken, is in onze optiek een focus op “security by design” en “privacy by design” essentieel. Dit wordt onder andere geborgd door de ISO 27001 certificering. De focus op deze marktconsultatie ligt echt op de technische componenten. Uiteraard zijn deze essentieel om tot een succesvol project te komen. In onze ervaring is dit echter niet de enige factor wat bijdraagt tot een succesvolle oplevering. Misschien nog wel belangrijker is het komen tot een volledige backlog waarbij er vanuit de gebruiker wordt bepaalt wat hij/zij écht nodig heeft. Onze ervaring leert dat veelal wordt gedacht dat de opdrachtomschrijving helder is, maar blijken er enorm veel changes te zijn tijdens het project, of nog erger: Het eindproduct is anders dan wat de eindgebruiker in gedachten had. Om dit te voorkomen is het noodzakelijk om ook oog te hebben voor welk “probleem” er daadwerkelijk wordt opgelost.