

## Discussie over het gebruik van DEMO in het IT-Deltaplan voor de overheid

[Adosh van der Heijden](#), [Wouter Schmitz](#) en [Martin Op 't Land](#) (18 december 2019)

### Adosh:

In [Aanvulling aanzet IT-Deltaplan](#) beschrijft Wouter Schmitz de functionaliteit van de vereiste IT-oplossing als volgt:

- succesvolle vernieuwing van de IT bij de overheid hangt af van het opbreken in behapbare brokken;
- de verantwoordelijkheid voor ieder brok moet zo laag mogelijk worden belegd en voorzien van middelen om veranderingen binnen het brok op te lossen, zonder andere brokken te hoeven veranderen.

Zo kan, volgens hem, er in plaats van een groot programma een groot aantal kleine projecten worden gedraaid met een veel hogere kans op succes.

Hij zoekt de praktische aanpak in het opbreken van "capabilities" en "microservice-architectuur" met "ontkoppelpunten voor zo stabiel mogelijke microservices" en noemt dat "niet te onderschatten zware projecten".

Mijn ervaring is dat DEMO-transacties prima kunnen dienen om de gevraagde microservices te identificeren.

### Reactie Wouter:

DEMO is een grondige methode om de vereiste services goed uit te modelleren. Dat kan helpen bij de stabiliteit van de services. De grootste change impact (blast radius) zit in de services. Als die regelmatig moeten worden aangepast dan leidt dat tot aanpassingen in andere componenten. Daarom acht ik de stabiliteit van de services als belangrijkste eigenschap. Dan worden de wijzigingen binnen een component zoveel mogelijk afgeschermd van de buitenwereld.

De aanpak voorziet in het opdelen in onafhankelijke brokken. Het klopt dat daarna ieder brok afzonderlijk moet worden aangepakt en daar kunnen weer andere methoden voor worden gebruikt, afhankelijk van de situatie binnen het brok. De services vormen dan de randvoorwaarde van de wijziging binnen een brok.

Voordat we de diepte ingaan wat te doen met ieder brok moeten we focus houden op het kernprobleem bij de overheid, namelijk de onbeheersbaar grote projecten. Dat heeft niet gewerkt. Het gaat niets oplossen dit te blijven proberen. Om kans te hebben op een oplossing zal men een andere aanpak moeten gaan volgen. Het opknippen in behapbare brokken is daarin cruciaal.

### Adosh:

Martin Op 't Land heeft daar wetenschappelijk onderzoek naar gedaan in zijn PhD-thesis (Delft University of Technology Netherlands, defended in public Friday June 13th, 2008) onder de titel [Applying Architecture and Ontology to the Splitting and Allying of Enterprises](#) en een concrete en praktische oplossing aangedragen voor die

uitdaging, op basis van DEMO-transacties (het maken en nakomen van één type (bijvoorbeeld overheids-) afspraak).

Ik citeer uit de 'Conclusions' op pg 57:

"What did we experience as limitations of the method applied?

First of all, we could perfectly link applications used in our primary business to the DEMO Construction Model. It was however not so easy to link our generic applications, such as the support for Asset Management or the Traffic Management Data Layer.

For a first order rationalization of the application portfolio – phasing out or replacing complete applications – this method worked. However, it was felt that for more radical architectural improvements, such as the introduction of Service Oriented Architectures (SOA) or the introduction of one corporate data warehouse, the method needs extension. Such an extended method could start with the DEMO transactions delivering Business (B-)services, and use that to identify Information (I-) services and Application (A-)services. In such a way one line of reasoning can be followed from a Service Oriented Enterprise (SOE) to SOA. This could start simple by first deriving I-services and use that for a data consolidation, based on understanding of the meaning of the data."

Enfin, een suggestie om te overwegen en mee te nemen

### **Reactie Martin:**

Over de stabiliteit van services: inderdaad geloof ik dat diensten vele malen stabiel zijn dan hun implementatie. Ik vind dan ook dat de transformatie van een gebied zou moeten beginnen met een overzicht van diensten en wie daarvoor verantwoordelijk zijn. Niet gedetailleerd, maar wel precies: het is een immens verschil of we als service-verantwoordelijkheid benoemen "het vaststellen van een vonnis" versus "het vastleggen van een vonnis". DEMO draagt hiervoor passende en theoretisch doordachte en overdraagbare concepten aan, juist op dat stabiele (implementatie-onafhankelijke) niveau, en beginnend met bedrijfs-services.

Naast dat stabiele (implementatie-onafhankelijke) niveau is het wezenlijk over concepten te beschikken voor zowel organisatie- als IT-implementatie, zodat daarin snel en slim gevarieerd kan worden. De two-pager [DEMO-OIV-lowcode vision](#) legt aan de hand van het pizzeria-voorbeeld uit wat DEMO is, wat Organisatie-ImplementatieVrijheidsgraden (OIV's) zijn, en hoe dit kan bijdragen aan een *RUN-time adaptive enterprise*. De voetnoot op p2 van die flyer vermeldt de onderliggende artikelen, resultaat van onderzoek naar OIV's aan Universiteit Antwerpen en Radboud Universiteit Nijmegen sinds 2012. Uit een recente Proof-of-Concept met ICTU bleek hoe een lowcode applicatie stapsgewijs en agile per DEMO-transactie kan worden opgebouwd. Daarbij konden, zonder herprogrammeren, veranderende organisatie-implementatiekeuzes via een OIV-preference panel worden geregeld.

Daarnaast is recent een andere Proof-of-Concept met ICTU uitgevoerd waarin systematisch vanuit DEMO-modellen business APIs worden afgeleid. Daar is ingegaan op het systematisch afleiden vanuit de grotere brokken (processen) naar de kleinere brokken (microservices). Op de schaal van de overheid is dat een vereiste om het beheersbaar te houden (cf [Normalized Systems theorie](#)). Dit is een

belangrijke bijdrage aan het IT Deltaplan, waar ik wel aandacht zie voor de aanpak (agile) en het gebruik van microservices, maar niet hoe die er dan uit zouden moeten zien of gedefinieerd moeten worden. Als agile teams dat zelf gaan doen, is redelijkerwijs chaos te verwachten – en bij managed chaos gaat DEMO erg helpen.

Ik zou hier wel graag even willen uitzoomen. Een architectuur van ontkoppeling in stabiele services is namelijk wel een nodig, maar zeker niet voldoende antwoord om grote projectissues op te lossen - want die hebben ook sterke organisatorische, financiële en juridische componenten. Heel kort, zonder enige pretentie op volledigheid:

- Wet- en regelgeving zal moeten worden aangepast naar “microservices wetten”, dat wil zeggen dusdanige definities en opbouw van wetten dat deze ruimte bieden en mee kunnen gaan in veranderingen op de schaal van services. Het promotie-onderzoek [Wendbare wetgeving \(Mariette Lokin, VU, 2018\)](#) geeft daarvoor m.i. bruikbare handvatten.
- Belangrijke praktische component van de wetgeving rond overheidstaken is dat deze veelal het financiële arrangement voor de uitvoering bepalen. Daarin zitten nogal eens perverse prikkels die ontkoppeling tegenwerken. Vaak is er geen financiële case voor de verbetering van een afzonderlijke stabiele service, en ook te weinig voor de toekomst-vaste continue verbetering en instand-handhouding ervan.
- Organisatorisch. De overlegstructuren en governance rond overheidsuitvoering zijn in hun aard nogal waterval georganiseerd en kennen een lange cyclus. Dit leidt samen met de niet ontkoppelde complexiteit van de bestaande systemen tot een enorme afstand tussen aantal releases per jaar in de publieke sector en in klantgerichte marktsectoren (ordegrootten 100 à 1000 verschil). Stabiele ontkoppelde services moeten onafhankelijk van elkaar *ge-released* kunnen worden op basis van andere overlegvormen over de verwachtingen van afnemers. Dat is een forse change die naast IT-architectuur ook cultuur raakt.

Het goede nieuws is dat er in de nu lopende brede maatschappelijke heroverwegingsgesprekken van de relevante beleidsambtenaren serieus gesproken wordt over de vraag hoe we meer vernieuwing (of als je het modewoord wilt: innovatie) kunnen inbedden in de overheidsstelsels van wetgeving en daaruit voortvloeiende ICT systemen. Dit “inbouwen in” biedt ruimte om het vervolgens ook verder naar het niveau van een concrete service “in te bouwen”. Het anti-pattern van een beheersmatige uitvoering waar op kosteneffectiviteit gestuurd wordt (en waar van lieverlee weinig leuk en hoogwaardig werk over blijft) enerzijds en dan grote meeslepende (in het begin leuke) en dan vaak vastlopende veranderprojecten zou daarmee doorbroken kunnen worden.

## **Adosh vervolgt:**

### ***AMvB en stelselinformatie-architectuur***

De vraag, die moet zijn beantwoord, om de gewenste uitkomst van een ICT-project te kunnen bepalen en beoordelen is: “Welke stelsel- of bedrijfsrol moet over welke bedrijfsobject(typ)en welke bedrijfsinformatie kunnen registreren en ontsluiten en welke bedrijfsberichten kunnen verzenden en ontvangen.”

Bij het ontwerpen van architecturen voor de Omgevingswet en de USB-wet heb ik dankbaar gebruik gemaakt van de AMvB om het antwoord op die vraag af te leiden. Het zou echter beter zijn als het antwoord op die vraag in de toekomstige AMvB's al integraal zou worden opgenomen.

***Verantwoorde kwaliteit is pas verantwoord nadat deze is verantwoord.***

Veelal voorziet een AMvB ook in de rol van een toezichthouder of inspectie. Een terugkerende klacht van onder-toezicht-staanden (OTS), zoals banken, ziekenhuizen, boeren en horeca, is, dat het toezicht (te) zware administratieve lasten zou veroorzaken. Het toezicht zou daardoor ten koste gaan van de stelselprestaties zelf, zoals zorg, integriteit, CO2-reductie, voedselkwaliteit.

Door mijn ervaring in DNB-Toezicht en contacten met de IGZ heb ik het inzicht verkregen, dat een toezichthouder eigenlijk geen andere informatie nodig heeft, dan de informatie, die al in transacties tussen OTS over hun onderlinge transacties had moeten zijn uitgewisseld. Die informatie is bovendien door twee partijen met tegengestelde belangen (opdrachtgever en opdrachtnemer van de transactie) volgens het vier-ogen-principe getoetst. Het toezicht hoeft daarom helemaal geen extra administratieve lasten te veroorzaken, maar zal zelfs de prestaties in het stelsel kunnen verbeteren als toezichthouders van OTS zouden eisen dat de onder toezicht staande stelselrollen bij de aan te leveren toezichtinformatie ook aantonen, dat deze direct is ontleend aan geleverde prestaties in gerealiseerde transacties.