

**Voordat je als architect aan de slag kunt met een term als Enterprise Service Bus (ESB), wil je graag weten wat dat dan inhoudt. Tegenwoordig hoor je de term ESB regelmatig. Het is te vergelijken met de introductie van Service Oriented Architecture (SOA). Dat was er ook opeens en werd door iedereen geadopteerd. Doordat het door veel collega's wordt gebruikt, worden er evenzoveel definities en meningen over gevormd en uitgestort. Met de term ESB gaat het inmiddels ook die kant op.**

## ESB: een architectuurstijl?

### Discussie over een definitie heeft nog geen consensus opgeleverd

**E**SB. Wat is dat eigenlijk? De eerste pagina's op Google, waar we de oplossing zoeken, verwijzen naar leveranciers als Microsoft, IBM, Sonic, en zo nog een aantal. Deze bedrijven hebben vanzelfsprekend een eigen opvatting over de ESB. Daarnaast is er in menig vakblad over ESB gepubliceerd. Wikipedia, de vrije encyclopedie op internet, geeft aan dat het om een 'softwarearchitectuur-concept' gaat en dat het wordt geïmplementeerd door technologie, die is gerelateerd aan de zogenaamde middleware infrastructuur technologie zoals CORBA (Common Object Request Broker Architecture) en daarop gebaseerde producten. Deze zijn gebaseerd op standaarden die voorzien in diensten voor meer complexe architecturen via een event-gedreven berichten-engine (een Bus).

Een ander beeld kan ontstaan als ESB in verband wordt gebracht met SOA als infrastructuur-instrument. De ESB kent vele invullingen van een puur technische koppel-infrastructuur tot functioneel technische invulling met tal van beheertools. Je praat eigenlijk over traditionele EAI (Enterprise Application Integration), ook wel integratiebroker genoemd, die als fundament dient voor realisatie van een ESB. Wanneer je ESB als robuuste SOA infrastructuur wilt opzetten heb je naast de broker additionele diensten nodig, waarover later meer. De broker kan worden gebruikt om een omgeving te creëren

voor uitwisseling van berichten tussen twee of meerdere applicaties.

EAI pakketten bieden voor het aansluiten van applicaties adapters, zoals BizTalk Server, die het mogelijk maken om functionaliteit van de gebruikte applicaties te ontsluiten voor berichtenverkeer. De communicatie tussen de adapters en de centrale infrastructuur is over het algemeen zo opgezet dat het EAI pakket optimaal te gebruiken is. De integratiebroker biedt, mits deze centraal wordt gestuurd, de volgende voorzieningen om op een adequate manier met berichten om te gaan. De belangrijkste zijn:

**Validatie van berichten:** de structuur van een te ontvangen en/of te verzenden bericht wordt geanalyseerd en gevalideerd tegen de gedefinieerde structuur. Structuur wordt vastgelegd door middel van schema's.

**Transformatie van berichten:** in een ideale wereld wordt er van een berichtdefinitie gebruik gemaakt, maar dit is meestal niet zo en worden voor eenzelfde type object meerdere berichtdefinities gebruikt (schema's). De integratiebroker biedt standaard voorzieningen om berichten om te zetten naar een andere structuur (mapping van het ene schema naar het andere).

**Routering van berichten:** het ontvangen bericht wordt geanalyseerd en op basis daarvan doorgestuurd naar de geabonneerde applicaties. Routering kan plaatsvinden op basis van zoge-

**Steef-Jan Wiggers**  
is Information Architect bij  
Inter Access en Wouter  
Goedvriend is Senior  
Consultant  
bij Seven Stars B.V.

naamde headerinformatie of op berichtinhoud (content based routing).

**Aggregatie van berichten:** het samenvoegen van meerdere berichten tot een nieuw bericht (aggregator pattern). Naast dit pattern zijn met de integratiebroker verscheidene patterns te implementeren, zoals opsplitsen van berichten (splitter pattern), verzamelen van berichten en versturen naar meerdere ontvangers (gather-scatter pattern).

**Betrouwbaarheid en beveiliging:** technisch/functionele diensten zoals gegarandeerde berichtaflevering, beveiligd transport, garantie van afzender en van oorspronkelijkheid van het bericht.

**Adapters en transport:** koppeling en communicatie naar applicaties en systemen.

### Additionele diensten

Bovenstaande voorzieningen zijn allemaal terug te vinden binnen BizTalk Server en kunnen als koppel-infrastructuur worden ingezet. Om te kunnen komen tot een robuuste SOA infrastructuur oftewel ESB heb je zoals eerder aangegeven additionele diensten nodig. De meest prominente daarvan zijn:

**Registry en repository:** een 'centrale database' waarin o.a. de servicedefinities worden beheerd. Een voorbeeld hiervan is UDDI.

**Beveiliging:** een centraal gestuurde beveiligings-service, die toegang tot de services regelt of de uitgewisselde informatie bijvoorbeeld via versleuteling beveiligt. Ook het beschermen tegen hoge (piek) belasting van het aanbiedende systeem is onderdeel van beveiliging.

**Monitoring:** een standaard service, die verschillende vormen van informatie levert over gebruik van de overige services. Monitoring kan zo ver gaan als het leveren van een volledige audit trail voor juridische bewijslast tot informatie over het voldoen van een service SLA of de werking van volledige ESB infrastructuur.

**Orchestratie (samenstelling):** de mogelijkheid om bestaande diensten te combineren tot een nieuwe, gecombineerde dienst. Een voorbeeld is het aanbieden van locatie-gebaseerde informatie, waar de service die de locatie bepaalt, gekoppeld wordt aan een service die informatie over de locatie oplevert. Door combinatie van twee 'eenvoudige' services ontstaat een nieuwe service/nieuw product.

**BAM (Business Activity Monitoring):** van processen en/of samengestelde services kunnen door ESB infrastructuur de KPI's gemeten worden. BAM is de service die dit realiseert en biedt bijvoorbeeld de mogelijkheid om via een cockpit de resultaten te analyseren en op basis van deze analyse het proces aan te passen.

Enkele van de bovenstaande diensten zijn eveneens terug vinden binnen BizTalk Server, zoals de orchestratie en BAM, welke als dienst kan

worden geconfigureerd. Andere diensten zijn toe te voegen met behulp van de Microsoft ESB Guidance package.

### ESB als term

Als je de term ESB op internet uitzet, krijg je een verscheidenheid aan meningen en opvattingen. Vreemd is dat er geen eenduidige lijn in de meningen en opvattingen is te herkennen. Natuurlijk zijn er gelijkenissen, maar er is geen consensus. Eigenlijk vraagt de term ESB om een definitie als je ziet hoe vaak de term wordt gebruikt en hoeveel verschillende opvattingen bestaan.

Het is niet eenvoudig om de definitie te vinden, laat staan te destilleren, uit alle opvattingen over de term. We hebben de vraag via LinkedIn in de 'groep' gegooid en er kwamen uiteenlopende reacties. Ook heeft David Chappell een boek 'Enterprise Service Bus' in 2004 gepubliceerd. Een aantal commentaren op het boek geven aan dat het toen nog om een verwachtingsvolle maar speculatieve technologie gaat. Anderen hebben de term overgenomen, bedrijven hebben er een draai aan gegeven en techneuten zijn het gaan implementeren. Je kunt je voorstellen dat het een eigen leven is gaan leiden. De review commentaren zijn ook niet mals op dit boek. Een van de reviewers op Amazon geeft, in niet mis te verstane woorden, aan dat het boek een 'sales pitch' is met de algemene gedachte dat de ESB dé oplossing is voor alle EAI problemen. Een andere reviewer geeft aan dat het de eerste interessante definitie van een ESB is en dat je, als EAI specialist, het boek gelezen moet hebben.

### Vendor Implementatie

Ook een aantal bedrijven heeft zich gestort op de term ESB. Hiermee proberen deze een invulling te geven aan de term. Microsoft heeft het Microsoft Guidance Package gelanceerd. Dit wordt aangeboden via de Microsoft Patterns & Practices. IBM is gekomen met een ESB implementatie in WebSphere. Oracle kent Oracle Fusion. Er zijn vast andere vendors, die soortgelijke implementaties aanbieden. De positionering is, bij sommige vendors impliciet, als middle ware. Hier komt de term Message Oriented Middleware (MOM) ook naar voren. Niet alle vendors noemen het zo, maar uit de verklaring van hun implementaties is deze term wel te destilleren.

### Algemene opvattingen

Bij het onderzoek naar de term ESB hebben we de volgende vraag voorgelegd aan de community op LinkedIn:

*"Is Enterprise Service Bus (ESB) a well known (super)pattern and if so in which scenario's does one apply it?"*

**Er zijn  
gelijkenissen  
maar er  
is geen  
consensus**

Enkele quotes van antwoorden op deze vraag:

*"My understanding of the concept of an ESB (Enterprise Service Bus) is that it is a way to bring all of the technologies within a company, entity, or enterprise together. Depending on the magnitude of what that could encompass, it \*could\* mean a simple product that 'maps' everything together. Usually though, it is a way - and therefore, an architectural pattern as you stated - to bring all of the different softwares, databases, and knowledge within an entity together to allow mutual communication..."*

*"...ESB by itself doesn't provide any business value as explained very nicely by Bobby Woolf, it is a capability you must have to implement an SOA strategy. Especially when services you are enabling are over many disparate legacy systems..."*

*"The most important concept is that an ESB is a architecture pattern (mostly related with SOA technologies) and not a tool"*

*"As a bus, the ESB is primarily a routing pattern, although, most of the market is selling it as a messaging exchange pattern. The concept of the bus is an abstraction for invocation of services based on either content or endpoint semantics. The value is mediation of service invocation interfaces from the business process, thus simplifying their creation..."*

De algemene perceptie die hier naar voren komt, is dat het om een architectuur pattern gaat, dat sterk gerelateerd is aan SOA. Als je het heel technisch bekijkt kun je het zien als een Message/Routing pattern. De twee smaken lopen, qua implementatieniveau, ver uit elkaar. Messaging en Routing patterns (Gregor Hohpe / Bobby Woolf [EIP], Martin Fowler [Po EAA] en anderen) liggen dicht op de infrastructuur en technische componenten, terwijl SOA gaat over de functionele onderkenning van service gebieden. Natuurlijk resulteert een SOA weer in een, al dan niet technische, implementatie zodat het functionele gebied benut kan worden. Een niet onbelangrijke opmerking – en door ons gedeeld – is dat een ESB is geen bedrijfsdoel op zich is. Je kunt de techniek(en) inzetten om bedrijfsdoelstellingen te realiseren.

Terugkomend op het vraagstuk van de definitie. Geen van de LinkedIn collega's geeft een eigen mening en visie op de ESB. Geen van allen geven ze aan dat er een algemeen aanvaarde definitie is van de ESB. Er zijn genoeg patterns herkend, erkend en bij de grote groep bekend, maar van de ESB is nog geen definitie te vinden. We hebben een aantal kenmerken/eigenschappen opgeschreven. Er zijn er vast veel meer te bedenken. Als je ze hebt of er zijn erbij die er niet thuis horen, dan kun je gerust met ons in discussie gaan.

## Waarom ESB?

Om ESB in te richten of implementeren moet er een geldige business case of aanleiding zijn. Wat zijn de drijfveren voor een organisatie om de eerste stappen te zetten richting ESB.

Opstellen van een business case voor ESB is moeilijk. Voornamelijk omdat de voordelen niet direct zichtbaar zijn en de kosten voor implementatie ervan niet misselijk. Als bedrijf zul je goed moeten nadenken of je een ESB-implementatie wilt in je organisatie. Nodige zaken moeten worden ingeregeld voordat ermee kan worden gestart. We praten dan over inzicht in de vraag welke diensten worden aangeboden aan de bus.

Waarom zou je als bedrijf de ESB binnen je organisatie willen inzetten? Motieven kunnen zijn:

- Hogere mate van hergebruik;
- Lagere operationele kosten;
- Kunnen inspelen op dynamische business veranderingen;
- Analyse mogelijkheden op business en exception niveau;
- Gedreven door business requirements.

Bovenstaande argumenten geven aan welk nut een bedrijf van een ESB kan hebben bij een succesvolle implementatie.

De ESB-markt kent tal van leveranciers en oplossingen. De kosten van aanschaf variëren sterk en hangen veelal af van de extra diensten, die ESB biedt. Verder spelen zaken als robuustheid en bewezen technologie een grote rol bij vaststelling van het prijskaartje. Traditionele EAI leveranciers, die hun integratiesoftware uitbreiden met SOA services en zo een ESB leveren, bevinden zich aan de bovenzijde van de markt. Het voordeel van deze leveranciers is dat ze op het vlak van infrastructuur een stabiele, goed beheer- en beheersbare omgeving kunnen bieden. Een aantal leveranciers richt zich volledig op de ESB markt en heeft het voordeel geen 'legacy' met zich mee te hoeven nemen. Nadeel voor hen is wel dat ze een minder bewezen infrastructuur kunnen bieden en mogelijk geen goed uitgewerkt software release-plan leveren. Prijstechnisch kunnen het goede alternatieven zijn. Naast de genoemde leveranciers zijn er ook open source leveranciers, die oplossingen bieden puur gericht op SOA en ESB standaarden. Veelal goede initiatieven, waarvan een aantal ook support modellen heeft. Nieuwe speler in de markt – voorzover deze het al niet is – is Microsoft met haar EAI pakket BizTalk Server, dat met de ESB guidance ook tot een ESB implementatie kan leiden. Vanwege de flexibiliteit, de implementatiemogelijkheden en onze achtergrond gaan we dieper in op de Microsoft ESB Guidance.

## Microsoft ESB Guidance

Microsoft ESB Guidance stelt architectuurrichtlijnen, patronen, praktijken en een set BizTalk

Server en .NET componenten ter beschikking om de ontwikkeling van een Enterprise Service Bus (ESB) op het Microsoft-platform te realiseren. Daarnaast biedt het mogelijkheden voor een organisatie om de bestaande messaging- en integratieoplossing uit te breiden. Onderstaande figuur geeft een weergave van de enterprise service bus implementatie, die Microsoft voor ogen heeft.

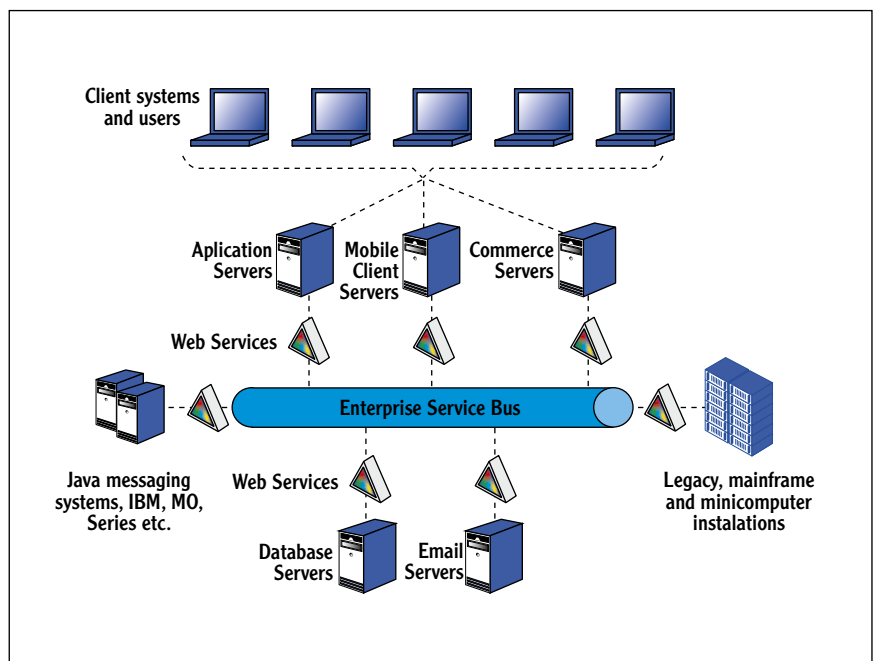
De bus levert een structuur waarbij allerlei diensten kunnen worden aangesloten. Dit zijn de web services in het bovenstaande figuur. Deze services staan in verbinding met database servers, email servers, Java Messaging, Mainframe, applicatieservers enzovoort. In feite stellen services je in staat gegevens beschikbaar te stellen of te consumeren. BizTalk ESB Guidance geeft je de mogelijkheid om een BizTalk Server zodanig op te tuigen dat deze als een Enterprise Service bus kan acteren. Interoperabiliteit is mogelijk. JMS Web Services en Host Integration Server 2006 stellen je in staat om – in combinatie met de BizTalk Adapters – gegevens te onttrekken uit Legacy Systemen zoals IBM Mainframe. Met Enterprise Adapters van BizTalk kun je ook SAP, JD Edwards en Oracle eBusiness suite koppelen. BizTalk is het middelpunt van een ESB zoals Microsoft dat ziet. Daarnaast biedt BizTalk out-of-the-box een groot aantal functionaliteiten, welke ook als kenmerken van een ESB kunnen worden gezien. We praten dan ook over transformaties, routeren, hosten van een proces, orchestratie en BAM.

### Eigenschappen

Met de ESB Guidance ben je in staat een volledige implementatie te verrichten, maar welke eigenschappen bevat deze dan. De belangrijkste eigenschappen van de ESB Guidance zijn:

- Dynamic Service Invocation;
- Unified Exception Management;
- ESB Management Portal;
- Itinerary Processing;
- SOA Governance Integration.

Dynamic Service Invocation is een eigenschap van ESB Guidance package, waarbij een adapterframework wordt geleverd dat je in staat stelt eenvoudig zogenaamde endpoints op te nemen, transformatie requirements te implementeren of verschillende configuraties te leveren aan orchestraties. Hiervoor worden WSMEX, UDDI, BRE en XPATH geleverd. Smart Caching wordt erbij geleverd voor verbetering van performance. Voor uniform exception management is er een framework met de functionaliteit van 'BizTalk Failed Message Routing' voor orchestraties. Met ESB Management Portal kunnen fouten, opgetreden binnen ESB, zichtbaar worden gemaakt. Deze kunnen worden aangepast en



opnieuw aangeboden aan de ESB, auditing kan worden verricht, alerts en abonnementen daarop zijn mogelijk en ten slotte kan analyse worden uitgevoerd op opgetreden fouten. Wat Itinerary processing wordt genoemd kan worden beschouwd als de mogelijkheid BizTalk Services zoals orchestraties, send poorten, transformaties, enzovoort samen te stellen op het moment van aanbieden van een bericht. Tenslotte maakt SOA Governance integratie dat de ESB kan samenwerken met derdepartij-pakketten als AmberPoint en SOA Software. Hierbij kunnen zogenaamde management agents worden aangebracht in pipeline componenten.

### Positionering vanuit Microsoft

Microsoft heeft de productierijpe ESB Guidance als pakket in november 2007 vrijgegeven. Zij positioneert dit pakket als een verzameling van, zoals eerder genoemd, architectuurrichtlijnen, waarmee een ESB implementatie kan worden verricht. Naast deze mogelijkheid kun je meer met het package dan alleen een volledige ESB implementeren. Je kunt onderdelen gewoon in je bestaande omgeving gebruiken zoals exception management zonder een volledige ESB implementatie te doen. Exception Management is bijzonder handig binnen een BizTalk omgeving. Wanneer berichten niet kunnen worden verstuurd of aankomen, kan zogenaamde 'Fail Message Routing' worden toegepast. 'Messaging' oplossingen zijn bijzonder handig, maar wat als nu een fout optreedt binnen een orchestratie. Wat doe je dan? Het exception management onderdeel van het pakket biedt daar een raamwerk voor; sturing voor juiste foutafhandeling binnen je orchestratie kan hiermee worden geregeld.

Figuur 1. Microsoft visie op Enterprise Service Bus.

## Wat kun je ermee?

Terugkomend op het exception management onderdeel heb je het ESB-exception management framework in handen. Dit bestaat uit drie hoofdcomponenten:

- ESB Failed Orchestration Exception Routing Mechanism;
- ESB Fault Processor Pipeline;
- ESB Management Portal and Fault Message Viewer.

Het eerste component stelt je in staat fouten af te handelen op een gelijke manier zoals bij eerder genoemde 'Failed Message Routing'. De pipeline zorgt voor het converteren van message-routing- en orchestrationexceptions naar de ESB-standaard, BAM-tracking en conversie van het bericht conform het SQL-adapterschema. De pipeline geeft het bericht door aan de datasource-adapter, bijvoorbeeld een SQL-adapter die het foutbericht opslaat in een datasource, bijvoorbeeld SQL Server. Het laatste component bestaat uit een management portaal, waarin onder andere een dashboard wordt getoond met allerlei foutstatistieken, waarvan je de foutdetails van elk afzonderlijk bericht kunt bekijken en mogelijk repareren om vervolgens weer aan de ESB aan te bieden.

Het bovenstaande beschrijft een mogelijkheid, die een van de onderdelen van de ESB guidance biedt. Zo zijn er meer onderdelen, die afzonderlijk kunnen worden gebruikt. Deze zijn:

- Namespace Normalization;
- Transformation Agent;
- Resolver WebService.

Met het laatste ben je in staat een Web service aan te roepen zonder door de ESB te gaan. De aanroep naar de Web Service kan 'one-way' of 'request-response' zijn. Directe communicatie is mogelijk tussen aanroepende client en de web service, maar voordat aanroep wordt uitgevoerd zal de endpoint van de service bekend moeten zijn (zie figuur 2). Binnen de documentatie van de ESB Guidance Package zijn naast dit voorbeeld, meerdere voorbeelden te vinden wat de mogelijkheden zijn met onderdelen van de Package.

Voor een volledige implementatie van een ESB gebruik je ook de onderdelen:

- Core Services;
- Generic Delivery Agent;
- Web Services.

De ESB met behulp van Guidance Package levert je het onderstaande figuur waarbij alle componenten en services zichtbaar zijn en BizTalk Server de kern vormt. Het levert een loskoppelbare berichtenuitwisseling-omgeving waar je als organisatie eenvoudig op berichten gebaseerde applicaties kunt maken. Er zijn componenten te

onderkennen bij het uitrollen van een ESB met behulp van BizTalk in combinatie met de guidance. Deze zijn:

- Web services;
- Itinerary services;
- Itinerary on-ramps;
- On-ramps;
- Off-ramps;
- Exception Management Framework;
- ESB Management Portal;

Met al de bovenstaande componenten heb je in feite en bus tot je beschikking waarmee je allerlei diensten met elkaar kunt koppelen en maak je integratie tussen allerlei systemen mogelijk (figuur 1). Het onderstaande figuur geeft je een beeld van de samenhang van alle componenten.

Beschrijving van alle componenten staan in de documentatie die bij de guidance wordt geleverd. In de referenties onderaan dit artikel is meer informatie te vinden.

## Conclusie

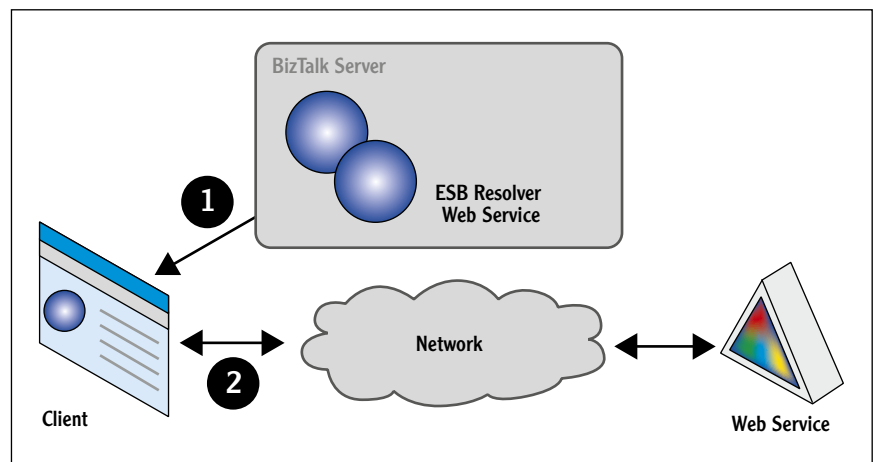
Over het algemeen geven de meeste leveranciers van ESB componenten en oplossingen allemaal eenzelfde overzicht. Microsoft staat in ons geval dicht bij de technologie, waar wij dagelijks mee te maken hebben. Dit is dan ook de reden waarom we de ESB Guidance Package van Microsoft als uitgangspunt en technologisch element hebben gebruikt om eens dieper in het ESB verhaal te duiken.

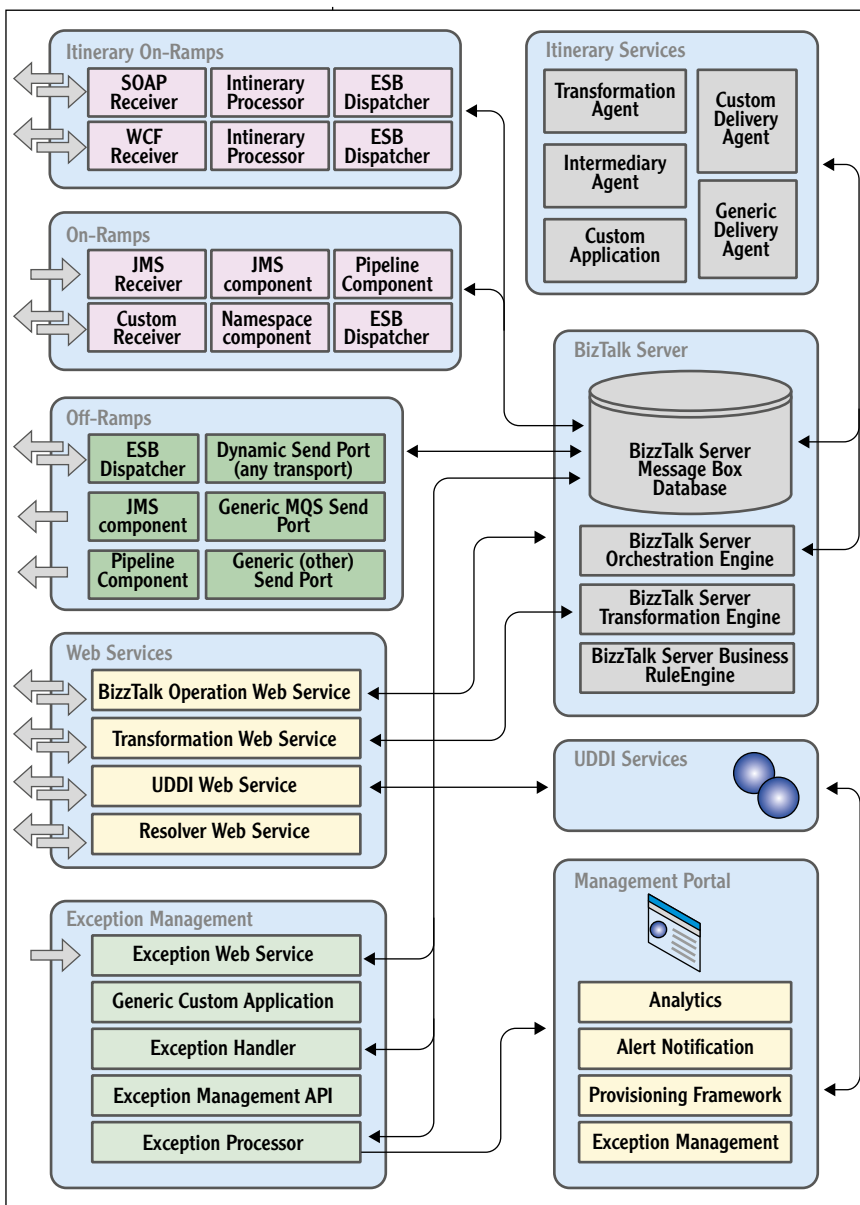
Op basis van de business case en de technische mogelijkheden onderkennen we o.a. de volgende voor- en nadelen voor de ESB.

Voordelen van een ESB zijn:

- In- en externe koppelingen naar diverse systemen;
- Ontsluiting van legacy systemen;
- Schaalbaarheid van service;
- Meer configuratie dan code.

Figuur 2. Aanroep van service via ESB ResolverWeb Service





Figuur 3. BizTalk en ESB Guidance componenten.

Nadelen van een ESB zijn:

- Complexiteit;
- Moeilijk te bepalen RIO;
- Volwassen governance model nodig zoals bijvoorbeeld ITIL met een goed gedefinieerde bedrijfsbrede strategie;
- Andere skillset benodigd;
- Vendor afhankelijkheid.

Naast voor- en nadelen van de ESB onderkennen we ook, net als onze collega's op LinkedIn, dat ESB toch vooral als patroon moet worden gezien.

### Platform

Heb je met de ESB Guidance uiteindelijk een industrie-geaccepteerde architectuurstijl of -patroon geïmplementeerd, of is het toch eigen software van Microsoft, toepasbaar binnen haar wereld, waarbij je de mogelijkheid hebt om te koppelen met andere omgevingen. In het package zelf zit-

ten veel bruikbare componenten, die je eenvoudig kunt toepassen voor je eigen oplossingen. Wat dat betreft biedt het package duidelijk meerwaarde. Het Exception Management Framework blijkt erg waardevol omdat foutafhandeling op een uniforme manier kan worden uitgevoerd.

Je kunt het dus ook beschouwen als uitbreiding (extensie) op BizTalk Server Product om volwassen oplossingen te realiseren. Met andere woorden: een verrijking van het product. BizTalk Server biedt zelf al de nodige mogelijkheden voor realiseren van integratieoplossingen en systeemproceessuring. Met de guidance erbij evolueert BizTalk meer naar een Platform, waarmee diverse architectuurstijlen worden ondersteund zoals EAI, SOA, ESB en Event Driven Architecture. «

### Referenties:

1. Wat is nu eigenlijk een ESB? – Maart 2006 – link: <http://www.techworld.nl/article/1586/wat-is-dat-nou-eigenlijk-een-esbi.html>
2. Enterprise Service Bus gesneden koek of abracadabra – Januari 2005 - <http://www.computable.nl/artikel.jsp?id=1427669>
3. Wikipedia - [http://en.wikipedia.org/wiki/Enterprise\\_service\\_bus](http://en.wikipedia.org/wiki/Enterprise_service_bus)
4. Get on the Bus - [http://www.iprofs.nl/article/ESB-Get\\_On\\_The\\_Bus-v1\\_0.pdf](http://www.iprofs.nl/article/ESB-Get_On_The_Bus-v1_0.pdf)
5. Service Oriëntatie - <http://nl.wikipedia.org/wiki/Service-oriëntatie>
6. Ontwerpen van een Enterprise Service Bus - [http://www.infosupport.nl/Ontwerpen\\_van\\_een\\_ESB.Pdf](http://www.infosupport.nl/Ontwerpen_van_een_ESB.Pdf)
7. Microsoft on the Enterprise Service Bus - <http://msdn2.microsoft.com/en-us/library/aa475433.aspx>
8. Enterprise Service Bus (ESB) – Guidance – <http://www.microsoft.com/biztalk/solutions/soa/esb.mspx>
10. ESB Guidance-website: <http://www.codeplex.com/esb>
11. Microsoft patterns & practices: <http://msdn.microsoft.com/practices>
12. Microsoft ESB-website: <http://www.microsoft.com/biztalk/solutions/soa/esb.mspx>