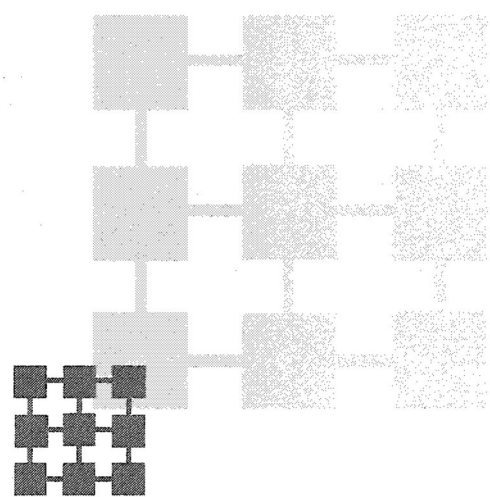


Scriptie

"Referentiearchitectuur"



Scriptie

"Referentiearchitectuur"

Studentnummer: 1142946	versie: Definitief	Gehelming n.v.t.
Naam afstudeerder: Han van 't Zand	Eerste examiner Dhr. G. Ovink	datum 25-05-2005

© 2005

Niets uit dit drukwerk mag worden veeveelvoudigd **en/of** openbaar gemaakt door **middel** van druk. **fotokopie, microfilm** of op welke andere **wijze** dan ook zonder voorafgaande toestemming van de auteur, noch mag hetzonder een dergelijke **toestemming** worden gebruikt voor **enig** ander werk dan waarvoor het **is** vervaardigd.

VOORWOORD

Deze scriptie is het resultaat van mijn afstudeerstage in het kader van mijn opleiding "Information Engineering" aan de Hogeschool van Utrecht. De afstudeeropdracht heeft plaatsgevonden van 14 februari tot 1 juli 2005 bij ingenieurs- en adviesbureau **Witteveen+Bos** te Deventer.

Graag wil ik mijn bedrijfsbegeleider ir. J.G.J. Nijhuis bedanken voor de begeleiding en support die hij mij heeft gegeven tijdens mijn werkzaamheden binnen het bedrijf. Tevens wil ik de overige medewerkers van **Witteveen+Bos** bedanken voor de prettige samenwerking. Daarnaast gaat mijn dank uit naar mijn begeleidend docent van de Hogeschool van Utrecht, dhr. G. Ovink, en naar alle andere mensen die op een bepaalde manier bijgedragen hebben aan deze scriptie.

Deventer, 25 mei 2005
Han van 't Zand

SAMENVATTING

Bij lokale overheden die actief zijn in de watersector bestaat zowel een externe als een interne informatievraag die voortkomt uit de planning, controle en uitvoering van de bedrijfsprocessen. Deze informatievraag kan worden beantwoord door het implementeren van één of meer informatiesystemen. Een kenmerk van deze informatiesystemen is dat zij gegevens combineren uit verschillende aanleverende informatiesystemen (horizontale integratie) en dat zij gegevens aggregeren uit onderliggende systemen (verticale integratie).

Witteveen+Bos adviseert lokale overheden op het gebied van verticale en horizontale integratie. Daarbij is de behoefte ontstaan aan een instrument waarmee de adviseurs ondersteund worden bij het opstellen van de **ICT-architectuur** van een dergelijk informatiesysteem. Dit instrument wordt een referentiearchitectuur genoemd. Deze referentiearchitectuur geeft voor verschillende situaties aan welke **ICT-componenten** en welke interfaces ingezet dienen te worden

Om de referentiearchitectuur te kunnen positioneren is er gebruik gemaakt van het informatiemanagementnegenvlak. Dit leerde ons dat de **informatiearchitectuur**, ICT-strategie en de **ICT-uitvoering** de belangrijkste omgevingsparameters zijn om mogelijke situaties te beschrijven voor de referentiearchitectuur.

De informatiearchitectuur bestaat uit de vraag en aanbod van informatie en de ontwerpkeuzes van de functies. De ICT-architectuur legt vast welke ICT-componenten en interfaces een invulling kunnen geven aan de informatiearchitectuur. De kern van dit onderzoek ligt bij de verschillende interfaces die er voor zorgen dat de verschillende ICT-componenten onderling en met de buitenwereld verbonden kunnen worden.

De ontworpen informatiearchitectuur en de ICT-architectuur zullen als basis dienen voor de verdere **uitwerking** van deze referentiearchitectuur. Tevens kan dit document gebruikt worden als basis voor verder onderzoek. Gedurende de literatuurstudie die tijdens dit onderzoek **heeft** plaatsgevonden, is gebleken dat het werken met (referentie)architecturen een gebied is waar nog veel te ontdekken valt.

INHOUDSOPGAVE

VOORWOORD	3
SAMENVATTING	4
1. INLEIDING	
2. OPDRACHTOMSCHRIJVING	8
2.1. Opdracht	8
2.2. Kader	8
3. PLAN VAN AANPAK	9
3.1. Inleiding	9
3.2. Aanpak en fasering	9
3.3. Resultaten	11
3.4. Acceptatiecriteria	11
4. INFORMATIEMANAGEMENT EN ARCHITECTUREN	12
4.1. Inleiding	12
4.2. Informatiemanagement	12
4.3. Architecturen	13
4.4. Referentiearchitectuur	14
5. INFORMATIEARCHITECTUUR	15
5.1. Inleiding	15
5.2. Informatiemodel	15
5.3. Informatievraag	16
5.3.1. Interne informatievraag	17
5.3.2. Externe informatievraag	18
5.4. Informatieaanbod	19
5.4.1. Dynamische gegevens	19
5.4.2. Basisgegevens	20
5.5. Functiebeschrijvingen	21
6. ICT-ARCHITECTUUR	23
6.1. Inleiding	23
6.2. Gegevensarchitectuur	24
6.3. Interfaces	25
6.3.1. Mogelijke invullingen	25
7. REFERENTIEARCHITECTUUR	27
7.1. Inleiding	27
7.2. Conceptmodel	27

8. CONCLUSIESEN AANBEVELINGEN	28
8.1. Conclusies	28
8.2. Aanbevelingen	28
9. EVALUATIE	29
9.1. Evaluatie van het onderzoek	29
9.2. Voorstellen voor verder onderzoek	29
9.3. Leerpunten	29
INDICES	30
GERAADPLEEGDE LITERATUUR	31

BIJLAGE I: ORGANISATIESTRUCTUUR

BIJLAGE II: RAMING

BIJLAGE III: KOSTEN EN BATEN

BIJLAGE IV INTERFACE ARCHITECTUUR

BIJLAGE V MIDDLEWARE

BIJLAGE VI WEBSERVICES

1. INLEIDING

Witteveen+Bos is een onafhankelijk ingenieurbureau dat gespecialiseerd is in adviesopdrachten binnen de bouwsector, de nationale infrastructuur, milieu en water. Het bedrijf kenmerkt zich door een multidisciplinaire werkaanpak. Medewerkers uit de verschillende sectoren werken samen aan oplossingen op verschillende vraagstukken. Ook wordt er regelmatig samengewerkt met andere bureaus uit **binnen-** en buitenland. Opdrachtgevers zijn onder andere de verschillende overheden, het bedrijfsleven en de verschillende industrieën.

De 700 medewerkers werkzaam binnen **Witteveen+Bos** werken binnen zogeheten **Product-Markt- Combinaties (PMC's)**. Elke PMC richt zich op een bepaald segment van de markt. De integrale aanpak van projecten wordt gewaarborgd door een nauwe samenwerking tussen deze PMC's.

In de sector 'water' valt de PMC Informatie Technologie. Binnen deze PMC heb ik de mogelijkheid gekregen mijn afstudeeropdracht uit te voeren (zie bijlage I voor een organogram). De PMC Informatie Technologie geeft advies op het gebied van procesautomatisering, geografische informatiesystemen en informatiemanagement. In dit laatste gebied moet vooral gedacht worden aan beleidsadvies over informatievoorziening, **kosten/baten** analyses, informatieanalyses, functioneel **ontwerpen**, uitvoeringsbegeleiding en het opstellen van informatie en ICT-architecturen.

Binnen de watersector hebben de lokale overheden zowel een interne als een externe vraag naar informatie. Deze informatie betreft zowel de planning, de controle als **de** uitvoering van bedrijfsprocessen. Door het invoeren van een nieuw informatiesysteem kan deze informatievraag beantwoord worden. Dit informatiesysteem moet de gegevens combineren uit verschillende aanleverende informatiesystemen (horizontale integratie), maar ook gegevens uit de onderliggende systemen zichtbaar maken (verticale integratie).

Witteveen+Bos adviseert de lokale overheden op het gebied van horizontale en verticale integratie. De adviseurs missen hierbij een instrument waarmee zij ondersteund worden om de **ICT-architectuur** van een dergelijk informatiesysteem op te stellen. Dit instrument wordt de referentiearchitectuur genoemd. Deze referentiearchitectuur geeft voor verschillende situaties aan welke **ICT-componenten** en welke interfaces ingezet dienen te worden. De opdracht bestaat dan ook uit het ontwikkelen van een instrument om deze referentiearchitectuur mogelijk te maken.

2. OPDRACHTOMSCHRIJVING

2.1. Opdracht

De opdracht bestaat uit het opstellen van een referentiearchitectuur voor horizontale- en verticale integratie van gegevens voor watergerelateerde informatiesystemen. Deze referentiearchitectuur moet een beslismodel bevatten waarmee kan worden bekeken wanneer welke **ict-component** het beste ingezet kan worden. De nadruk zal hierbij liggen op de interfaces, welke ervoor zorgen dat de verschillende componenten zowel intern als met de buitenwereld gekoppeld kunnen worden.

2.2. Kader

Voor dit onderzoek is het vooral van belang de inzet van moderne architecturen zoals de Enterprise Service Bus (ESB) en de Service Oriented Architecture (SOA) te evalueren. De nadruk zal hierbij liggen op de verschillende interfaces die gebruikt kunnen worden. Zoals in de inleiding reeds is vermeld dient de referentiearchitectuur gericht te worden op de watersector. Gezien de grote hoeveelheid aan bedrijfsprocessen en informatiegebieden wordt de referentiearchitectuur verder ingekaderd tot zogenaamde **BedrijfsGegevensSystemen** (BGS). Deze bedrijfsgegevenssystemen bevatten onder andere procesinformatie afkomstig uit de procesautomatisering (verticale integratie) en uit Laboratorium **InformatieSystemen** (LIS) en OnderhoudsBeheersSystemen (OBS) voor zowel operationeel, tactisch als strategisch niveau.

Het is de bedoeling dat het ontwikkelde model wordt getoetst in een aantal praktijksituaties. Dit zal plaatsvinden binnen projecten die op dit moment door **Witteveen+Bos** worden uitgevoerd. Deze toetsing bestaat uit het ontwikkelen van een specifieke ICT-architectuur, op basis van de referentiearchitectuur. Deze wordt, wanneer de tijd het toelaat, in pilotuitvoering gerealiseerd. Een aantal parameters, zoals bijvoorbeeld functionaliteit, performance en kosten, zullen dan getoetst worden aan de hand van dit model. Andere parameters zoals bijvoorbeeld de onderhoudbaarheid zullen worden getoetst op basis van expert judgement.

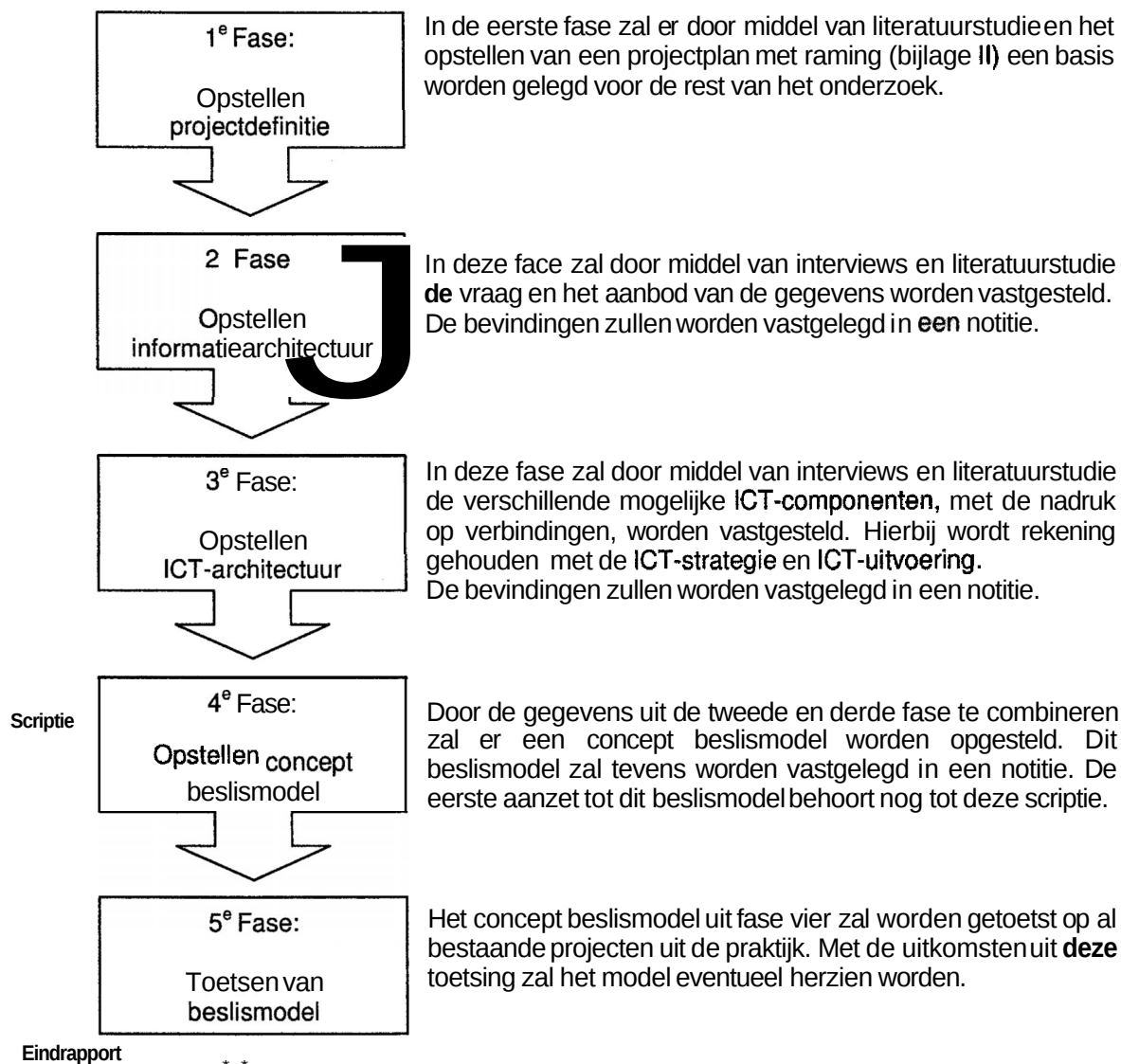
3. PLANVANAANPAK

3.1 Inleiding

In dit hoofdstuk wordt het plan van aanpak in grote lijnen besproken. Paragraaf 3.2 geeft een schematische weergave, met uitleg over dit plan, met de bijbehorende fasering. In paragraaf 3.3 wordt vervolgens ingegaan op de resultaten van dit onderzoek. Tenslotte komen in paragraaf 3.4 de acceptatiecriteria aan bod.

3.2 Aanpak en fasering

Het onderzoek zal worden opgedeeld in de volgende vijf fasen:



Deze aanpak leidt tot de volgende activiteiten:

Fase 1: projectdefinitie

De projectdefinitie betreft een validatie van de definitie zoals in dit projectplan is beschreven. Tevens zal er in deze eerste fase een literatuurstudie worden gedaan om een basis te creëren voor het vervolg van het onderzoek.

Fase 2: het opstellen van de informatiearchitectuur

Bij het opstellen van de informatiearchitectuur zal er worden gekeken naar het soort **informatievragen** die zich kunnen voordoen. Hierbij moet gedacht worden aan de volgende vragen:

- met wat voor soort informatie wordt er gewerkt?
- welke functionaliteit **moet** de informatie bevatten (**ad hoc** of standaardrapportage)?
- op welk niveau is de informatie nodig (planning, controle of uitvoering)?
- wat is de gewenste performance?
- hoe vaak moeten gegevens aangeleverd worden (frequentie)?
- wat zijn de bewaartermijnen van de informatie?

De benodigde informatie zal worden verkregen door middel van literatuurstudie en **interviewsessies** met betrokkenen (ir. J.G.J. Nijhuis en ing. H.W. Jansen).

Tevens zal er in deze fase gekeken worden naar het aanbod van informatie. Deze informatie zal worden verkregen door een literatuurstudie en interviews.

Zowel de informatie met betrekking tot het aanbod als de vraag zal worden vastgelegd in een notitie.

Fase 3: het opstellen van de ICT-architectuur

Hier zal worden gekeken naar de verschillende soorten componenten en verbindingen die mogelijk zijn binnen de ICT-architectuur. Tevens zal er worden gekeken naar de uitgangspunten met betrekking tot ICT binnen de watersector. Benodigde informatie zal vergaard worden door middel van literatuurstudie en interviewsessies (met ir. J.G.J. Nijhuis, ing. H.P. van Essen). Bovenstaande bevindingen zullen in een notitie worden vastgelegd.

Fase 4: opstellen van referentiearchitectuur

Op basis van de informatiearchitectuur en de **ICT-architectuur** zal er een referentiearchitectuur worden opgesteld. Deze scriptie zal afgerond worden met een eerste aanzet tot het concept van deze referentiearchitectuur. Dit referentiemodel zal worden getoetst en eventueel worden aangepast in de vijfde fase van het onderzoek dat voor **Witteveen+Bos** wordt uitgevoerd. De uitkomsten van deze fase zullen in een notitie worden opgenomen.

Fase 5: toetsing en rapportage

De opgestelde referentiearchitectuur zal worden getoetst in een aantal praktijksituaties. Deze toetsing zal bestaan uit het ontwikkelen van een specifieke ICT-architectuur met behulp van het opgestelde referentiemodel. Wanneer de tijd het toelaat, zal deze ICT-architectuur in een pilot uitvoering worden gerealiseerd. Aan de hand van deze pilot kunnen de volgende punten worden getoetst:

- functionaliteit;
- performance;
- kosten.

Eventuele andere punten zoals de hoeveelheid onderhoud kunnen worden getoetst op basis van expert judgement.

Na aanleiding van deze toetsing kan het concept model nog worden aangepast om vervolgens definitief te worden vastgesteld.

Alle bevindingen en conclusies zoals beschreven in de notities zullen worden vastgelegd in een eindrapportage.

3.3. Resultaten

Het resultaat van het project wordt als volgt omschreven:

"Een instrument waarmee adviseurs ondersteund worden bij het opstellen van een ICT-architectuur voor horkontale- en verticale integratie systemen"

In de onderstaande tabel wordt per fase aangegeven welke producten opgeleverd worden.

Tabel 1: Producten per fase

Face	product
Projectdefinitie	projectplan
Informatiearchitectuur	notitie informatiearchitectuur
ICT-architectuur	notitie ICT-architectuur
Referentiearchitectuur	scriptie t.b.v. opleiding
	notitie referentiearchitectuur
Toetsing	(evt.) pilot ICT-architectuur
Rapportage	eindrapport t.b.v. Witteveen+Bos

3.4. Acceptatiecriteria

De acceptatiecriteria voor het totale project zijn **als** volgt gedefinieerd:

Het projectresultaat moet een model opleveren dat::

- door adviseurs gebruikt kan worden ter ondersteuning bij het opstellen van een ICT-architectuur.

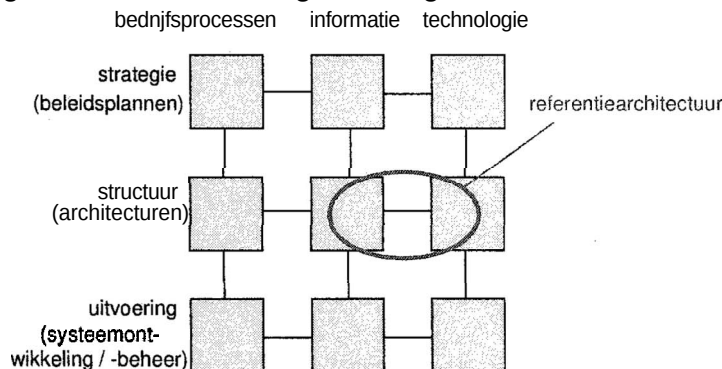
Door adviseurs het model te laten toepassen zal worden gekeken of er aan dit criterium wordt voldaan.

4. INFORMATIEMANAGEMENT EN ARCHITECTUREN

4.1. Inleiding

Om de referentiearchitectuur te kunnen positioneren maken we gebruik van het informatiemanagement negenvlak van de Universiteit van Amsterdam¹. Door de referentiearchitectuur in het negenvlak te plaatsen zien we wat de belangrijkste omgevingsparameters zijn: de informatiearchitectuur en de ICT-architectuur.

Figuur 1: Informatiemanagement negenvlak en referentiearchitectuur



4.2. Informatiemanagement

Situaties waarin informatiemanagement een rol speelt zijn de volgende:

- de situatie waarin informatie en de kwaliteit van informatie belangrijker wordt. Hierbij kan het zowel gaan om interne als externe oorzaken;
- de situatie waarin de omvang en daarmee de kosten van de informatievoorziening sterk toenemen. Hierbij gaat het niet alleen om investeringskosten, maar ook om beheerkosten.

In dergelijke situaties heeft het management veelal behoefte aan een stuurmiddel voor de begeleiding en vernieuwing van de informatievoorziening. Dit middel is informatiemanagement. Dit is in feite een beleids- en planvormingsproces met betrekking tot informatievoorziening. Hierbij gaat het erom informatie te managen als bedrijfsmiddel en dit af te stemmen en afgestemd te houden op de bedrijfsprocessen.

Veel voorkomende misverstanden over informatiemanagement zijn de volgende:

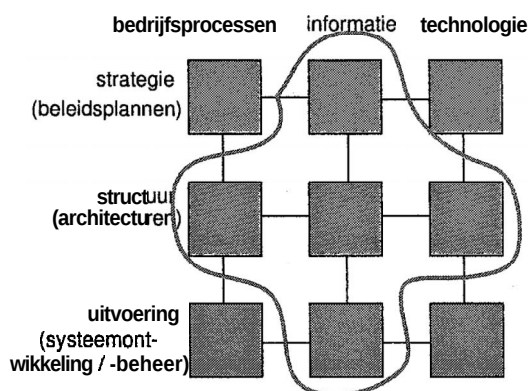
- 'informatiemanagement is duur'. Voor de korte termijn klopt dit, omdat een beleids- en planvormingsproces wordt opgestart wat extra kosten met zich meebrengt. Voor de lange termijn weegt dat echter duidelijk op tegen de extra reparatiekosten die gebrek aan planning met zich meebrengen;
- 'informatiemanagement is alleen iets voor zeer grote organisaties'. Dit is een typische misvatting. Iedere hedendaagse organisatie zal informatie en informatiesystemen moeten managen. Voor kleinere organisaties zou dat eenvoudiger kunnen zijn als er vooraf goed over werd nagedacht. Dat laatste ontbreekt echter vaak;
- 'informatiemanagement impliceert maatwerk in plaats van standaardpakketten'. Ook dit is onjuist. Informatiemanagement maakt het juist mogelijk om voor de toekomst die situaties te benoemen waarin standaardpakketten goed kunnen worden toegepast. Zowel voor maatwerk als voor standaardpakketten geldt dat deze prima kunnen worden toegepast, mits de voor- en nadelen hiervan onderkend en geaccepteerd zijn door het management.

¹ Maes, R. (1999) *A Generic Framework for Information Management*, Amsterdam: Universiteit van Amsterdam.

In onderstaande figuur is een overzicht weergegeven van de aandachtsgebieden van informatiemanagement. Dit zijn de vlakken binnen het rood omcirkelde deel.

Het blauwe vlak hierbinnen vormt in feite het informatiebeleid. Dit is een concrete en consistente visie van het management van een organisatie ten aanzien van de hoofdlijnen bij de organisatie van de informatievoorziening. Dit zal afgestemd moeten zijn op het lange-termijn organisatieplan. Het bevat doelstellingen, uitgangspunten en richtlijnen die voor de informatievoorziening in acht moeten worden genomen. Het management neemt deze beleidsbeslissingen en draagt deze uit naar de organisatie.

Figuur 2: Overzicht informatiemanagement



De rode vlakken vormen in de kern het informatieplan. Dit is een uitwerking van het informatiebeleid in een actieprogramma van uit te voeren activiteiten voor de komende jaren. Het gaat hierbij niet alleen om projecten voor nieuwe systemen, maar ook om betrieefactiviteiten.

4.3. Architecturen

Architecturen, de groene vlakken in figuur 2, vormen een belangrijke schakel tussen informatiebeleid, uitvoering en beheer. Het genootschap voor informatiearchitecten² heeft de architectuur als volgt gedefinieerd: *“Een architectuur is een samenhangende visie van een organisatie op haar informatievoorziening. Een informatiearchitectuur komt tot stand door een gezamenlijk proces van beeldvorming van de onderhandeling tussen alle betrokkenen. In een informatiearchitectuur komen de elementen van de informatievoorziening en hun samenhang tot uitdrukking, alsmede hun aansluiting op de bedrijfsarchitectuur en de ICT-architecturen het waarom hiervan. Inherent aan een informatiearchitectuur zijn keuzes op het gebied van informatiefunctionaliteit en structuren. Deze keuzes worden vastgelegd in de vorm van principes, standaarden en modellen. Daarmee is een informatiearchitectuur het bestemmingsplan voor de vernieuwing van de informatievoorziening van een organisatie.”*

De architecturen zijn in de praktijk vaak onderdeel van het informatiebeleid. Ze geven nadrukkelijk de gekozen gewenste situatie weer. Architecturen geven o.a. richting door ontwerpkeuzes die zijn vastgelegd. Architecturen dwingen af dat gekeken wordt naar de relaties tussen informatiesystemen. Architecturen geven aan welke keuzes zijn gemaakt, alsmede welke nog niet zijn gemaakt! Daarmee zijn architecturen een belangrijk stuurmiddel binnen informatiemanagement.

Op het gebied van informatievoorziening zijn een tweetal architecturen van belang, te weten:

- informatiearchitectuur;
- ICT-architectuur.

² GIA Genootschap voor Informatie Architecten 1998.

informatiearchitectuur

Een informatiearchitectuur voor een **organisatie(onderdeel)** legt het volgende vast:

- informatievraag (behoefte);
- informatieaanbod;
- ontwerpkeuzestav. functies.

De informatievoorziening voor belanghebbenden bij deze informatie wordt verzorgd door een verzameling van verschillende functies³:

- presentatie van informatie;
- distributie van informatie;
- bewerkingen van gegevens tot informatie;
- opslag van gegevens en informatie;
- validatie van gegevens;
- extractie van gegevens uit aanleverende gegevensbronnen;
- beheersfunctie.

Er moet voor gezorgd worden dat de juiste informatie voor de organisatie wordt geleverd en de functies optimaal op elkaar afgestemd zijn. De functies worden doorgaans binnen een organisatie op meerdere wijzen ingevuld. Bijvoorbeeld, sommige gegevens worden in een Word-document opgeslagen terwijl andere in een **Oracle** database worden opgeslagen (bijvoorbeeld namen en adressen voor een mailinglijst en namen en adressen uit het GBA). Beide functies voorzien in een andere informatiebehoefte, maar een nadere beschouwing kan leren dat zowel de kwaliteit als de efficiency kan verbeteren **als** dit wordt geïntegreerd.

Ontwerpkeuzes worden vastgelegd als ze betrekking hebben op meerdere functies of extra aandacht behoeven.

ICT-architectuur

De ICT-architectuur legt vast welke ICT-componenten een invulling geven aan de informatiearchitectuur. Zo kan een functie uit de informatiearchitectuur worden vervuld door één of meer samenwerkende ICT-componenten en kunnen omgekeerd meerdere functies door één ICT-component worden afgehandeld. Bijvoorbeeld, een standaardpakket op het gebied van waterzuivering vervult delen van meerdere functies binnen de waterzuivering. Het standaardpakket maakt gebruik van de functie opslaan van het databasesysteem Oracle. Dit databasesysteem wordt vervolgens ook weer gebruikt voor veel andere functies. Zo ontstaat al snel een wirwar van verschillende ICT-componenten. Aangezien de hoge prijs van licenties en de vereiste kennis en capaciteit bij de beheersorganisatie is het een goed idee om **de** aanschaf van **ICT-componenten** te sturen.

De functie van de **ICT-architectuur** is het structureren van deze ICT-componenten zodat een aangeschafte component aan de eisen van de functie voldoet, kan communiceren met andere ICT-componenten en flexibel genoeg is om toekomstige functies ook te kunnen ondersteunen.

4.4. Referentiearchitectuur

De referentiearchitectuur zal bestaan uit een overzicht van mogelijke interfaces en een beslissingsmodel wanneer welk type het best ingezet kan worden. Parameters in dit model zijn standaard softwarekwaliteitsparameters zoals functionaliteit, performance, robuustheid, onderhoudbaarheid en licentiekosten.

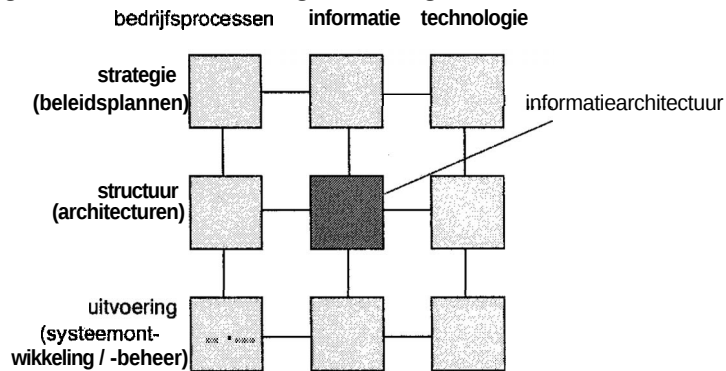
³ Deze functies worden in **5.5** nader toegelicht.

5. INFORMATIEARCHITECTUUR

5.1. Inleiding

In dit hoofdstuk zal de informatiearchitectuur in hoofdlijnen worden, uitgezet. Het gaat hierbij om een inventarisatie van de vraag en aanbod van gegevens en de ontwerpkeuzes voor de functies van een informatiesysteem.

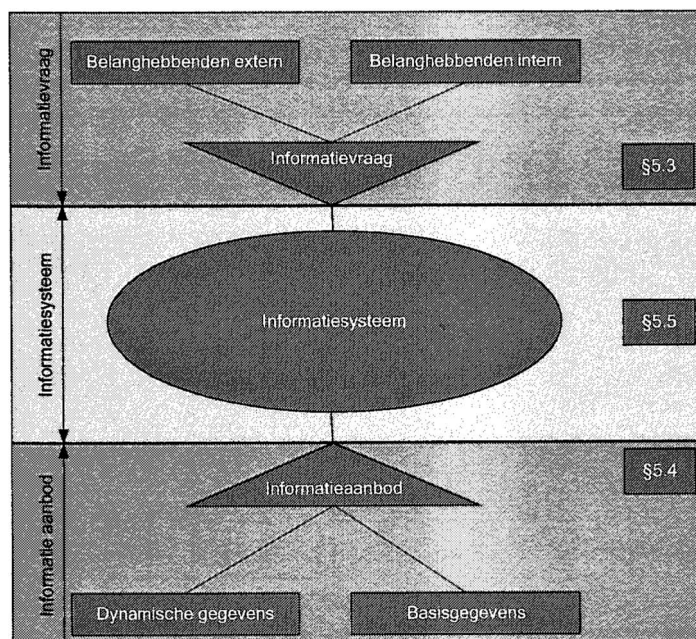
Figuur 3: Informatiemanagementnegenvlak en informatiearchitectuur



5.2. Informatiemodel

De informatiearchitectuur bestaat uit een beschrijving van de informatievraag, het informatieaanbod en de ontwerpkeuzes voor de functies van een informatiesysteem (zie 94.3). De informatievraag specificeert waar het informatiesysteem aan moet voldoen. **Het** informatieaanbod beschrijft of de benodigde gegevens hier ook voor geleverd kunnen worden en hoe deze dan geleverd moeten worden. Door de informatievraag en het aanbod tegen elkaar weg te zetten volgen de ontwerpkeuzes voor de informatiesysteemfuncties (zie figuur 4).

Figuur 4: Informatievraag en aanbod



5.3. Informatievraag

De informatievraag specificeert de eisen en wensen die door belanghebbenden aan het informatiesysteemgesteld worden.

De vraag valt op te splitsen in zowel een interne informatievraag als een externe informatievraag.

Zoals reeds uit de definitie (94.3) is gebleken zijn de keuzes op het gebied van informatiefunctie en structuren inherent aan een informatiearchitectuur. Deze keuzes worden vastgelegd in de vorm van principes, standaarden en modellen.

Bij het opstellen van de informatievragen dient al rekening gehouden te worden met verschillende ontwerpaspecten. Deze aspecten worden vastgelegd in het volgende model:

- **Functionaliteit:**
Hoe wordt de informatievraag gepresenteerd en welke gebruikersinteractie is voorzien. Bijvoorbeeld, een maandelijkse standaardrapportage stelt andere eisen aan een informatiesysteem dan ad-hoc analyse zoals bijvoorbeeld het analyseren van bepaalde trends;
- **Granulariteit:**
Granulariteit, oftewel fijnmazigheid, geeft aan met welk detail gegevens opgeslagen moeten worden. Voorbeelden hiervan zijn de opslag van meetgegevens op seconde-, minuut-, uur- of dagniveau. Het vastleggen van gegevens per regio, gebied, installatie, installatieonderdeel of nog gedetailleerder;
- **Actualiteit:**
Hoe actueel moeten de gegevens zijn? Voor sommige toepassingen is het niet erg dat gegevens een week achterlopen op de huidige situatie. Voor anderen moeten real-time de gegevens beschikbaar zijn;
- **Tijdsduur opslag:**
Hoe lang moeten gegevens worden opgeslagen;
- **Performance:**
Wat voor een performance is gewenst? Bijvoorbeeld, ad-hoc analyses vragen een snellere responstijd dan standaardrapportage en uitwisselbare bestanden;
- **Kwaliteit**
Voor sommige gegevens is het niet erg als er een paar waarden missen. Voor andere toepassingen kan dit fataal zijn. Hetzelfde geldt voor de nauwkeurigheid van de informatie;
- **Reproduceerbaarheid:**
Dit is de vraag in hoeverre het duidelijk en herleidbaar is hoe aan bepaalde informatie is gekomen;
- **Flexibiliteit**
Is het systeem makkelijk aan te passen op wisselende vraag en aanbod.

5.3.1. Interne informatievraag

De mogelijke vormen van interne informatievraag zijn onder te verdelen in de volgende drie groepen:

- Strategische informatievraag (directie, beleidsmedewerkers);
- Tactische informatievraag (management);
- Operationele informatievraag (werkvloer).

Tabel 2: Ontwerpaspectenmodel interne informatievraag

	Functionaliteit	Granulariteit	Actualiteit	Tijdsduur opslag	Performance	Kwaliteit	Reproduceerbaarheid	Flexibiliteit
Strategisch	Standaard rapportage	Laag	Near-time	>2 jaar	Dag	Hoog	Hoog	Hoog
Tactisch	Standaard rapportage	Gemiddeld	Near-time	<2 jaar	Minuut	Hoog	Hoog	Gemiddeld
	Adhoc analyse							
Operationeel	Standaard rapportage	Hoog	Real-time	<3 mnd	Seconden	Gemiddeld	Laag	Gemiddeld
	Adhoc analyse							

Strategische informatievraag

Op dit niveau wordt er gebruik gemaakt van standaardrapportage t.b.v. jaarplannen. Er zal op dit niveau hoofdzakelijk gewerkt worden met geaggregeerde data op jaar niveau. De informatie die hiervoor nodig is bestaat uit:

Dynamische gegevens (§5.4.1):

- hoeveelheid water;
- afgezet slib;
- verbruikt aantal KWh per jaar per regio;
- storingen;
- totaal aantal uren onderhoud;
- aantal klachten;
- aantal opgeloste klachten;
- gemiddelde meetwaarden.

Basis gegevens (§5.4.2):

- installatiegegevens;
- regio's.

Gegevens op dit niveau zullen voor langere tijd opgeslagen moeten worden (>2jaar)

Beleidsmedewerkers zullen t.b.v. analyses gegevens in dimensies (OLAP⁴) willen plaatsen om zo bijvoorbeeld het stroomverbruik van een bepaald gebied voor een bepaalde tijd in kaart te kunnen brengen.

Tactische informatievraag

Op dit niveau zullen er hoofdzakelijk gegevens gevraagd worden ten behoeve van optimalisatie van processen en procesonderdelen. Er wordt hier gewerkt met zowel standaard als ad hoc rapportages. Gegevens dienen hier meer geaggregeerd te zijn dan bij de operationele informatie, maar minder geaggregeerd dan de strategische informatie. Tevens zal er op dit niveau ingezoomd moeten kunnen worden om zo te kunnen zien of bepaalde afwijkingen worden veroorzaakt door een bepaalde Regionale Water Zuiverings Installatie (RWZI).

Benodigde gegevens:

Dynamische gegevens (§5.4.1):

- hoeveelheid water;
- afgezet slib;
- verbruikt aantal KWh;

⁴ OnLine Analytical Processing. Techniek die het mogelijk maakt snel door grote hoeveelheden multidimensionale gegevens te navigeren voor analyse.

- trend;
- storingen;
- totaal aantal uren onderhoud;
- aantal klachten;
- aantal opgeloste klachten;
- gemiddelde meetwaarden.

Basis gegevens (§5.4.2):

- installatie gegevens;
- regio's.

Operationele informatievraag

Op dit niveau is er sprake van procesbesturing en bewaking. Hier wordt gewerkt met hoogfrequente datastromen. Door middel van deze real-time data wordt er direct gestuurd op het primaire proces. De informatievraag is hier dus hoog frequent, de opgevraagde gegevens zijn niet geaggregeerd.

Benodigde gegevens:

Dynamische gegevens (§5.4.1):

- hoeveelheid water;
- afgezet slib;
- verbruikt aantal KWh;
- storingen;
- trends;
- events;
- onderhoudsplannen;
- soort klacht;
- meetwaarden.

Basis gegevens (§5.4.2):

- installatie gegevens.

5.3.2. Externe informatievraag

Externe informatievragen kunnen opgevraagd worden door onder andere:

- Wet Verontreiniging Oppervlaktewater (WVO);
- Benchmark;
- Centraal Bureau voor de Statistiek (CBS);
- Burgers.

Tabel 3: Ontwerpaspectenmodel externe informatievraag

	Functionaliteit	Granulariteit	Actualiteit	Tijdsduur opslag	Performance	Kwaliteit	Reproduceerbaarheid	Flexibiliteit
WVO	Standaard rapportage	Laag	Near-time	>3jaar	Hoog	Hoog	Hoog	Hoog
Benchmark	Standaard rapportage	Gemiddeld	Near-time	>1jaar	Gemiddeld	Hoog	Gemiddeld	Gemiddeld
CBS	Standaard rapportage	Hoog	Near-time	>1jaar	Laag	Gemiddeld	Laag	Laag
Burgers	Standaard rapportage	Hoog	Near-time	1jaar	Laag	Laag	Laag	Gemiddeld

Wet Verontreiniging Oppervlaktewater

In 1970 is de Wet Verontreiniging Oppervlaktewateren (WVO) van kracht gegaan. Het doel is de vervuiling van het oppervlaktewater tegen te gaan.

Het WVO zal door middel van gemeten waarden uit het proces controleren of de waterzuivering voldoet aan de gestelde normen. Het WVO maakt hiervoor gebruik van de volgende gegevens:

- Basisgegevens (§5.4.2);
- Dynamische gegevens (§5.4.1);
- Meetgegevens (intern);
- Meetgegevens (extern).

De berekeningen die WVO met de verschillende gegevens uitvoert zijn complex en aan verandering onderhevig.

Benchmark

In 1999 hebben de waterschappen⁵ het initiatief genomen om hun kosten en prestaties volledig transparant te maken, enerzijds om verantwoording af te leggen en **anderzijds** om te leren en te verbeteren.

De bedrijfsvergelijking zal met een lage frequenties (jaarlijks) moeten worden opgeleverd en zal bestaan uit standaardrapportage. De gegevens zullen moeten worden geaggregeerd uit de procesgegevens. De informatie zal bestaan uit dynamische gegevens (§5.4.1) Verder zullen er gegevens uit verschillende informatiebronnen moeten worden verkregen (klachtensysteem, Proces Automatisering, etc.)

Centraal Bureau voor de Statistiek

Het CBS heeft als doel onafhankelijk gegevens te verzamelen en deze om te zetten in statische informatie ten behoeve van praktijk, beleid en wetenschap. Ook in dit geval zal het gaan om geaggregeerde dynamische gegevens. (bron: www.cbs.nl)

Burgers

Ook burgers zullen mogelijk inzicht willen krijgen in bepaalde gegevens. Hierbij dient gedacht te worden aan:

- Hoe vaak is er overgestort op de sloot achter mijn huis?;
- Hoeveelheid geproduceerd afvalwater.

De burger zal met zijn bestaande middelen toegang willen krijgen tot de gewenste informatie. Deze zal snel en makkelijk toegankelijk moeten zijn.

Gegevens zullen geaggregeerd worden aangeleverd en de burger moet bijvoorbeeld door middel van het invoeren van zijn of haar postcode de betreffende informatie op kunnen vragen.

5.4. Informatieaanbod

Bij informatieaanbod gaat het om de vraag welke huidige, geplande en toekomstige automatiseringsfuncties en gegevensbronnen de benodigde informatie leveren.

De gegevens zijn op te delen in twee soorten gegevens:

- basisgegevens;
- dynamische gegevens.

5.4.1. Dynamische gegevens

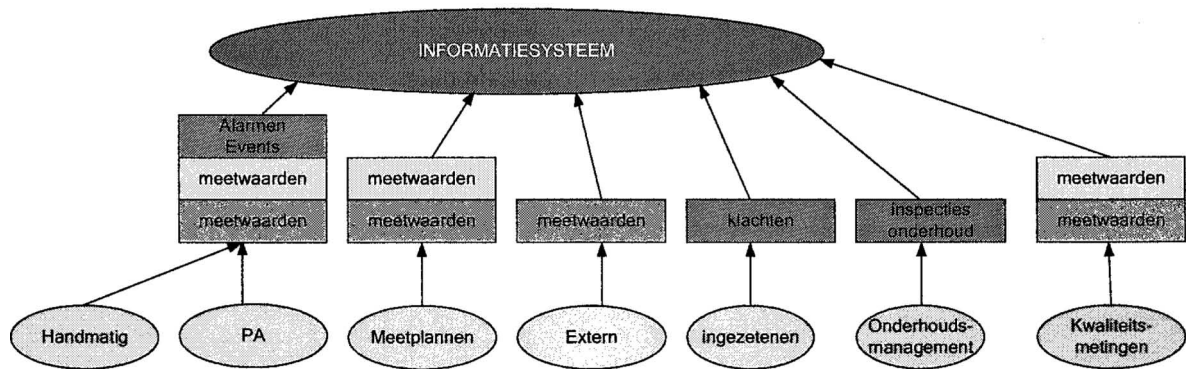
Dynamische gegevens zijn gegevens die met een bepaalde regelmaat worden aangeleverd. Deze gegevens worden door verschillende informatiesystemen beschikbaar gesteld.

Voorbeelden van deze gegevens zijn:

- Procesgegevens;
- Meetgegevens;
- Klachten;
- Onderhoudsgegevens;
- Projectmatige metingen;
- Handmatige invoer;
- Externe (**meet**)gegevens.

⁵ Rapport: Bedrijfsvergelijking Zuiveringsbeheer, unie van waterschappen, 2002.

Figuur 5: Voorbeeld informatieaanbod dynamische gegevens



In bovenstaande afbeelding staan voorbeelden van verschillende gegevensbronnen voor dynamische gegevens. De grijze ovals beschrijven de gegevensbronnen. Onderscheid wordt gemaakt in drie soorten dynamische gegevens:

- tijdreeksen (oranje rechthoek):
 - gegevens die met hoge frequentie beschikbaar zijn (< 1 uur);
- dagwaarden (gele rechthoek):
 - gegevens die met een lagere frequentie beschikbaar zijn (> 1 uur);
- gebeurtenissen (groene rechthoek):
 - gegevens die alleen beschikbaar zijn ten gevolge van een bepaalde gebeurtenis.

5.4.2. Basisgegevens

Basisgegevens zijn de gegevens die de dynamische gegevens beschrijven en weinig veranderen. Voorbeelden hiervan zijn:

- installatiegegevens;
- coderingen;
- regio's.

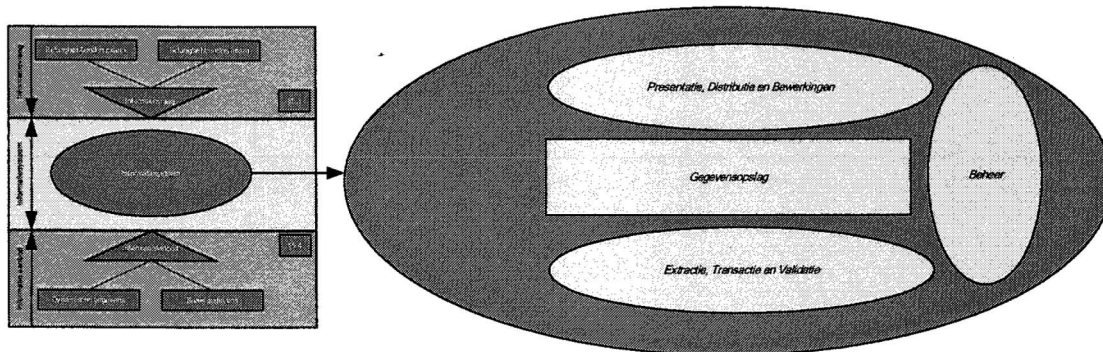
Een bestaand risico is dat dezelfde basisgegevens op meerdere locaties worden opgeslagen de zogenaamde eilandarchitectuur. Dit heeft een lagere kwaliteit van gegevens die moeilijk te beheren zijn tot gevolg. Een voorbeeld hiervan is dat dezelfde basisgegevens worden gebruikt binnen de procesautomatisering t.b.v. het meten en monitoren als wel bij het onderhoudsbeheer.

5.5. Functiebeschrijvingen

De verschillende functies van het informatiesysteem bestaan uit:

- de presentatie van informatie;
- de distributie van informatie;
- het bewerken van gegevens tot informatie;
- de opslag van gegevens en informatie;
- validatie van gegevens;
- extractie van gegevens uit aanleverende gegevensbronnen;
- het beheren van de bovengenoemde functies.

Figuur 6: Functies van een informatiesysteem



Extraheren, Bewerken en valideren.

Het eerste onderdeel van een informatiesysteem zijn de functies extraheren, bewerken en valideren.:

ETL-functie:

Extractie, Transactie en Laden (ETL). Deze functie zorgt ervoor dat brongegevens uit de aanleverende systemen worden onttrokken en op een juiste manier in een datawarehouse worden geplaatst.

Validatie:

Met deze functie dienen de gegevens te worden bewaakt. De functionaliteit van deze tool is:

- Signaleren van technische onvolkomenheden in de uitwisseling van **de** data;
- Signaleren van mogelijke problemen met de kwaliteit van de data zoals ontbrekende gegevens en waarschijnlijke fouten;
- Kwaliteitstempel op **de** data;
- Handmatige correctie van gegevens via standaardtools;
- Automatische correctie van gegevens.

Gegevensopslag

Deze functie zorgt voor de opslag van:

- Basisgegevens;
- Dynamische gegevens:
 - Meetwaarden;
 - Daggegevens;
 - etc.
- Geometrie.

Bewerkingen, presentatie en distributie

Het derde onderdeel van een informatiesysteem zijn de functies bewerkingen, presentatie en distributie:

Bewerkingen:

Functie om bewerkingen zoals berekeningen mee uit te voeren;

Presentatie:

Functie om de resultaten voor belanghebbenden te presenteren;

Distributie:

Functie om de resultaten bij de juiste personen te krijgen.

Beheer

De beheersfuncties zorgen ervoor dat een informatiesysteem bij wijzigingen in de functies, data, of organisatie correct blijft werken. Daarnaast worden beheersfuncties **gebruikt** om fouten te kunnen herstellen.

De beheerfunctie bevat **de** volgende functionaliteit:

- autorisatie;
- gegevensbeheer;
- beheer rapportages, analyses en uitwisselbestanden;
- back-up en **restore**.

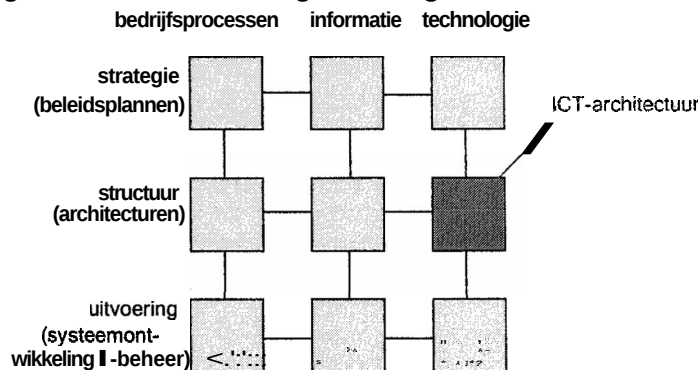
6. ICT-ARCHITECTUUR

6.1. Inleiding

Het doel van dit hoofdstuk is het opzetten van een ICT-architectuur voor een bedrijfsgegevenssysteem. Deze architectuur geeft richting aan de in te zetten ICT-componenten⁶ en hun interfaces⁷ om de verschillende informatievragen te kunnen beantwoorden.

Om de ICT-architectuur te positioneren maken we wederom gebruik van het informatiemanagement negenvlak.

Figuur 7: Informatiemanagement negenvlak en ICT-architectuur



Met het bovenstaande model is goed te zien dat de ICT-architectuur nauwe relaties heeft met de volgende onderdelen:

- Informatiearchitectuur (95.3): de informatiearchitectuur specificeert welke informatie de bedrijfsprocessen **nodig** hebben. De **ICT-architectuur** geeft de richting aan met welke **ICT-technologie** de informatiearchitectuur kan worden gerealiseerd;
- ICT-strategie: welke uitgangspunten worden binnen de watersector gehanteerd op ICT-gebied? Een voorbeeld hiervan is dat de meeste bedrijven binnen de **waterzuiveringssector** niet of nauwelijks zelf software ontwikkelen;
- ICT-uitvoering: Er moet rekening gehouden worden met de ontwikkelaspecten en de beheeraspecten.

De ICT-architectuur legt vast **welke** ICT-componenten een invulling geven aan de informatiearchitectuur. Zo kan een functie uit de informatiearchitectuur worden vervuld door één of meer samenwerkende ICT-componenten en kunnen omgekeerd meerdere functies door één ICT-component worden afgehandeld.

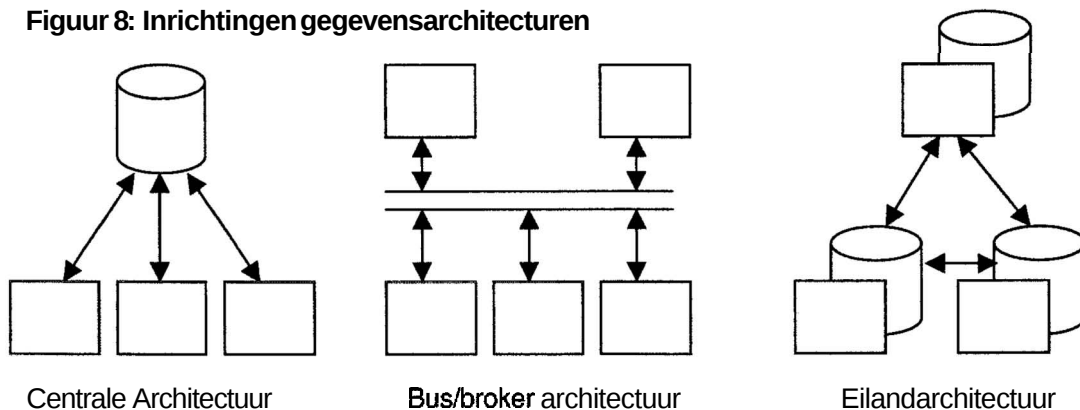
⁶ Een ICT-component is een software module, applicatie of pakket die een functie zelfstandig kan uitvoeren en kan communiceren met andere ICT-componenten.

⁷ Een interface zorgt voor de communicatie tussen de verschillende componenten.

6.2. Gegevensarchitectuur

Niet iedere functionele component moet in alle gevallen worden opgenomen. Zo kan in een busarchitectuur de gegevensopslagfunctionaliteit worden overgelaten aan de systemen die gegevens aanleveren. Ook kan **één** functionele component door meerdere software componenten worden uitgevoerd. Eveneens kunnen meerdere componenten door één softwarecomponent worden uitgevoerd. In de volgende paragraaf zal worden geconvergeerd naar een aantal veel gebruikte configuraties om vervolgens de aandacht te kunnen leggen op de kern van dit onderzoek "de interfaces"

Figuur 8: Inrichtingen gegevensarchitecturen



Voor- en nadelen van de verschillende gegevensarchitecturen:

- centrale architectuur:
Gegevens worden centraal beheerd in een basisregistratie. De kwaliteit is hier beter beheersbaar en het beheer is efficiënter. Vaak is dit een ideaalbeeld, omdat het niet altijd technisch mogelijk is iedere applicatie gebruik te laten maken van de basisregistratie. Ook moet rekening worden gehouden met een extra inspanning aan het begin;
- **Bus/broker** architectuur:
Een bus / broker architectuur zorgt ervoor dat de informatiesystemen, door middel van een centrale **gegevensbus/broker**, op elkaar worden aangesloten. De centrale **bus/broker** fungeert als een dirigent bij een orkest en zorgt voor orkestratie van de gegevensstromen en taken als 'routing' (wie is aan de beurt), 'event handling', 'transformatie' (vertaalslag) en 'mapping';
- eilandarchitectuur:
Het op een ad-hoc wijze applicaties verkrijgen en inzetten om informatie vragen te beantwoorden resulteert in een wirwar van applicaties en verbindingen met als gevolg dat de kwaliteit van de gegevens laag is en het beheer inefficiënt. Een groot risico is dat dezelfde basisgegevens op meerdere plaatsen worden bijgehouden. Het gevaar zit er dan in dat wanneer onderhoudsbeheer bijvoorbeeld een bepaalde installatie vervangt voor een andere dat wel doorvoert in het **OnderhoudsBeheerSysteem** (OBS) maar dat het op **de** andere plaatsen niet wordt doorgevoerd.

6.3. Interfaces

Interfaces kunnen worden beschouwd als de koppelstukken tussen de componenten waarmee gegevens zijn uit te wisselen.

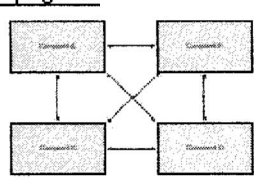
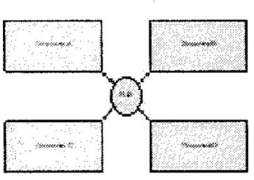
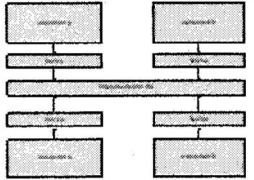
Er zijn meerdere mogelijkheden om de verschillende componenten te koppelen.

We maken hierbij onderscheid uit de volgende 3 inrichtingen (zie figuur 8 en bijlage IV):

- Eiland architectuur;
- Broker architectuur (centrale HUB);
- Bus architectuur (centrale BUS).

In de onderstaande figuur worden de verschillende architecturen tegen elkaar weggezet.

Tabel 4: Vergelijkingstabel

	Spaghetti	Broker	Bus
			
Invulling	Uitwisselbestand	Middleware	Webservices
Performance	Laag	Hoog	Laag
Flexibiliteit	Gemiddeld	Laag	Hoog
Realtime	Ja	Nee (asynchroon)	Ja
Coupled		Tight	Loosely
Onderhoudbaarheid	Moeilijk	Lastig	Simpeel

6.3.1. Mogelijke invullingen

Handmatig

Niet in alle gevallen is het nodig om applicaties te koppelen, in sommige gevallen wegen de kosten van een geautomatiseerd systeem niet op tegen voordelen die men er van heeft. In dit geval vindt er geen automatische koppeling plaats maar kunnen gegevens met de hand overgezet worden van systeem A naar systeem B.

Uitwisselbestanden

In het geval van uitwisselbestanden worden gegevens van systeem A naar systeem B overgezet. Hierbij maken gestandaardiseerde tekstformats platform onafhankelijke uitwisseling van informatie mogelijk. Bij voorkeur wordt er voor gestandaardiseerde tekstdocumenten gebruik gemaakt van een taal waarvan de structuur van een document wordt vastgelegd. De opmaak kan dan aangebracht worden met behulp van een taal die beschrijft hoe de gestructureerde tekst eruit moet zien.

Voorbeelden van gestandaardiseerde tekstdocumenten zijn:

- **ASCII:**
American Standard Code for Information Interchange. Deze zeven bits gecodeerde tekenset voor informatie uitwisseling vormt de basis voor de meeste computercodesets voor de pc. De standaard voor ASCII is vastgelegd in ISO8859;
- **XML:**
eXtensible Markup Language is een standaard voor het definiëren van formele markup-talen voor de representatie van gestructureerde gegevens in de vorm van platte tekst. Deze representatie is zowel machineleesbaar als leesbaar voor de mens. Met andere woorden: XML is een bepaalde manier om gegevens gestructureerd vast te leggen (bijvoorbeeld in een bestand, maar ook voor bijvoorbeeld doorsturen van informatie over het internet). Het is

ontworpen om zowel door een programma als door een mens leesbaar te zijn. XML is een vereenvoudigde vorm van SGML, Standard Generalized **Markup** Language, een heel complexe standaard die gebruikt werd om ingewikkelde documenten vorm te geven. Een eerder vereenvoudigde versie van SGML is HTML **HyperText Markup** Language. HTML heeft voor een doorbraak in **SGML-achtig** vormgegeven tekst gezorgd, maar gegevens die op een HTML-pagina staan zijn voor computers niet als zodanig te herkennen. XML zorgt voor die herkenbaarheid van gegevens.

De tekstdocumenten kunnen door middel van **FTP** worden overgezet van systeem A naar Systeem B.

- **FTP:**

File Transfer Protocol is een **cliënt/server** protocol dat gebruikers toelaat bestanden van één computer naar een andere over te zetten over **TCP/IP**.

Functionaliteit uitwisseling

Middleware

Middleware kan gezien worden als het koppelingselement tussen architectuur en functionaliteit. Het **levert** een uniforme interface tot ongelijke systemen en een gemeenschappelijk transport van data, zonder dat hierbij **een één op één** koppeling (synchroon) tussen deze systemen tot stand wordt gebracht. Hierdoor wordt het eenvoudiger om applicaties en services te laten samenwerken zonder rekening te houden met afhankelijkheden tussen communicatieprotocollen, besturingssystemen en hardware platforms.

We onderscheidende volgende types middleware (zie bijlage V):

- Middleware op basis van berichten (**Message** Oriented Middleware - MOM);
- Externe Procedure-aanroepen (Remote Procedure **Calls** - RPC);
- Object Request Broker (ORB):
 - Microsoft Component Object Model (COM);
 - Common Object Request **Broker** Architecture (CORBA);
- Database middleware: Remote Data Access (RDA).

Webservices (zie bijlage VI):

Bovengenoemde middleware platformen kennen vier belangrijke mankementen⁸:

- de interfacetalen zijn niet uitwisselbaar en niet universeel geaccepteerd;
- de informatiebussen zijn niet altijd uitwisselbaar en niet universeel;
- de interfaces van CORBA en **JAVA/RMI** zijn in de praktijk statisch terwijl **DCOM** een zeer ingewikkeld model gebruikt voor dynamische interfaces;
- het model voor de interfaces en het programmeermodel rijt bij deze drie platformen sterk gekoppeld.

Deze vier mankementen worden opgelost met behulp van webservices. Deze gebruiken XML als basis voor: de interfaces (Web Services Description Language, WSDL), de communicatie (**Simple** Object Access Protocol, Soap) en voor het beschrijven, ontdekken en integreren van services (Universal Description, **Discovery** & Integration, UDDI). Dit zijn allemaal open standaarden, en worden geaccepteerd en ondersteund door in ieder geval de belangrijkste spelers in de industrie (Microsoft, SUN en OMG) en gebruiken het **inter-** en of **intranet** als informatiebus.

Middleware is op dit moment óf documentgeoriënteerd óf proceduregeoriënteerd en de integratie is gericht op respectievelijk documenten of procedures en methoden. Het voordeel van webservices is dat deze beide vormen ondersteunen.

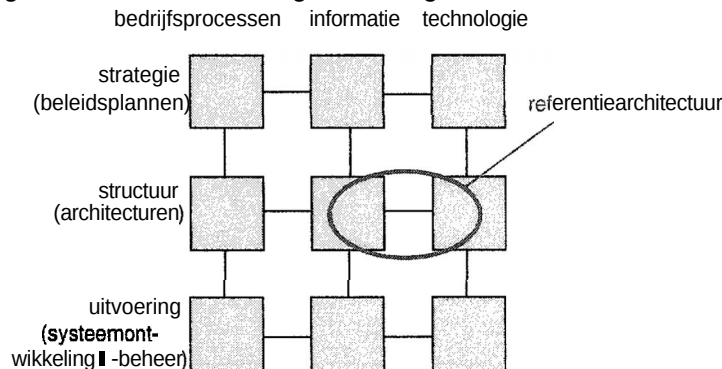
⁸ O'Toole, A., Web Service Oriented Architecture: the Best Solution for Business Integration, Cape Clear Software, 2003.

7. REFERENTIEARCHITECTUUR

7.1. Inleiding

De referentiearchitectuur zal gaan dienen ter ondersteuning bij het opstellen van de optimale ICT-architectuur voor de bedrijfsgegevenssystemen. Om een beeld te geven welke omgevingsparameters van belang zijn om mogelijke situaties te beschrijven binnen de referentiearchitectuur, plaatsen we deze binnen het informatiemanagement negenvlak.

Figuur 9: Informatiemanagement negenvlak en referentiearchitectuur



De volgende **omgevingsparameters** zijn van belang om mogelijke situaties te beschrijven in de referentiearchitectuur:

- Informatiearchitectuur (Hstk 5): welk soort informatievragen kunnen zich voordoen. Informatievragen bestaan onder andere uit de verschillende soorten informatie, de functionaliteit (**adhoc** of standaardrapportage), het niveau (planning, controle of uitvoering), de gewenste performance, de aanleverfrequentie en de bewaartermijnen;
- **ICT-strategie**: welke uitgangspunten worden binnen de watersector gehanteerd op ICT-gebied. Een voorbeeld hiervan is dat de meeste waterbedrijven niet of nauwelijks zelf nog software ontwikkelen;
- **ICT-uitvoering**: er moet rekening gehouden worden met zowel de ontwikkelaspecten als met de beheeraspecten.

7.2. Conceptmodel

De referentiearchitectuur zal bestaan uit een overzicht van mogelijke interfaces en een beslissingsmodel wanneer welk type interface het best ingezet kan worden. Parameters in dit model zullen zijn:

- wel of geen standaardrapportage;
- wel of geen ad-hoc rapportage (analyse);
- opslag van data;
 - Meetwaarden;
 - Daggegevens;
 - Geometrie;
- Etc.

Tevens zullen standaard softwarekwaliteitsparameters zoals functionaliteit, performance, robuustheid, onderhoudbaarheid en licentiekosten worden meegenomen bij de keuze voor de optimale ICT-architectuur bij elke betreffende situatie.

Bovenstaand model zal in de laatste fase van dit onderzoek worden uitgewerkt en is dus niet meegenomen in deze scriptie. Het model zal tijdens de presentatie worden **geïntroduceerd** en nader worden toegelicht.

8. CONCLUSIES EN AANBEVELINGEN

8.1. Conclusies

Op dit moment zijn alle voorbereidingen getroffen om de referentiearchitectuur te realiseren. In het laatste stadium van dit onderzoek zal de referentiearchitectuur zijn daadwerkelijke vorm krijgen. Op basis van de vastgestelde parameters zal bepaald worden welke ICT-architectuur in welke situatie het meest optimaal is.

Doordat er nog weinig wetenschappelijk gefundeerd onderzoek is gedaan naar de verschillende interfaces is er weinig bekend over het daadwerkelijke nut van de verschillende interfaces, maar er wordt wel veel over geschreven. Vaak is er nog niet bekend of deze methoden daadwerkelijk een aanvulling zijn, of dat er sprake is van een 'hype'.

Dit is een document geworden dat als een basis kan dienen voor verder onderzoek. Onderzoek zou kunnen bestaan uit het praktisch testen van de referentiearchitectuur, maar ook uit verdere theoretische verdieping van dit onderwerp.

8.2. Aanbevelingen

Deze kunnen op dit moment nog niet gegeven worden, daar de inhoudelijke versie van de referentiearchitectuur in het laatste stadium van dit onderzoek aan bod komt.

9. EVALUATIE

9.1 Evaluatie van het onderzoek

Het werken met architecturen was voor mij aan het begin van dit onderzoek nog onbekend. Door het lezen van verschillende **websites** en boeken, maar ook door het voeren van interviews met ervaringsdeskundigen, werd dit stap voor stap duidelijk. Tijdens het werken met deze architecturen, werd al gauw duidelijk dat dit een erg brede manier van werken is. Er komt veel informatie op je af, waardoor het moeite kost om je tot de kern van het onderzoek te beperken.

Tevens heeft het geruime tijd geduurd voordat ik bekend was met de watersector. Om hiermee vertrouwd te raken, heb ik me verdiept in de aanwezige literatuur en een aantal malen een gesprek gehad met adviseurs binnen **Witteveen+Bos**. Aangezien dit een voornamelijk theoretische verdieping is geweest, was het moeilijk om me een voorstelling te kunnen maken van de praktische uitvoering.

Doordat ik in het beginstadium van dit onderzoek nog onbekend was met bovenstaande zaken, heb ik de mogelijkheid gehad om mij hierin te verdiepen. Dit heeft ervoor gezorgd dat mijn kennis met betrekking tot architecturen, maar ook mijn kennis met betrekking tot de watersector, aanzienlijk is toegenomen. Doordat het duidelijk is hoe ik de verschillende architecturen vorm moet geven, vormt dit een solide basis voor het verder ontwikkelen van de referentiearchitectuur. Ik verwacht dan ook dat dit een instrument wordt waar de verschillende adviseurs gebruik van kunnen gaan maken in de verschillende situaties.

9.2 Voorstellen voor verder onderzoek

Allereerst wordt dit onderzoek nog afgerond door de uitgewerkte referentiearchitectuur toe te voegen. Tijdens de presentatie van deze scriptie zal hier inhoudelijk op worden ingegaan. Er zal een besismodel gepresenteerd worden waarin duidelijk wordt wanneer welke architecturen het best toegepast kunnen worden.

Het gebied waarin dit onderzoek is uitgevoerd, is erg aan veranderingen onderhevig. Het is dan ook aan te bevelen om dit onderzoek continue uit te bereiden. Er zal steeds aandacht besteed moeten worden aan welke interfacemethoden er op dat moment gebruikt kunnen worden.

Dit onderzoek kan gerealiseerd worden door het regelmatig bijhouden van de vakliteratuur. Wanneer er nieuwe methoden op de markt komen, zal er een afweging gemaakt moeten worden over het nut van deze methode. Wanneer er blijkt dat een nieuwe methode geschikt zou kunnen zijn, is het aan te bevelen dit eerst nader te onderzoeken.

9.3 Leerpunten

Op dit moment worden de verschillende architecturen voornamelijk in hoofdlijnen besproken. Een volgende stap in mijn persoonlijke ontwikkeling is het uitwerken van de verschillende architecturen. In verder onderzoek zou ik mij graag meer specifiek willen gaan richten op de details van de verschillende aanwezige architecturen.

Tevens heb ik gemerkt dat het verdiepen in een onderwerp erg interessant is, maar ook erg intensief kan zijn. Een goede tijdsplanning (zie bijlage 2) heeft er echter voor gezorgd dat dit toch binnen de gestelde tijd gelukt is. In een volgend onderzoek wil ik deze planning dan ook weer gebruiken, zodat de gestelde tijd ook optimaal benut wordt.

INDICES

Figuurindex

Figuur 1: Informatiemanagementnegenvlak en referentiearchitectuur.....	12
Figuur 2: Overzicht informatiemanagement.....	13
Figuur 3: Informatiemanagementnegenvlak en informatiearchitectuur	15
Figuur 4: Informatievraag en aanbod.....	15
Figuur 5: Voorbeeld informatieaanbod dynamische gegevens	20
Figuur 6: Functies van een informatiesysteem.....	21
Figuur 7: Informatiemanagementnegenvlak en ICT-architectuur	23
Figuur 8: Inrichtingen gegevensarchitecturen.....	24
Figuur 9: Informatiemanagementnegenvlak en referentiearchitectuur.....	27
Figuur 10: Organogram.....	33
Figuur 11: Eiland architectuur.....	37
Figuur 12: Broker architectuur.....	37
Figuur 13: Busarchitectuur.....	38
Figuur 14: SOA	38
Figuur 15: SOA + ESB	39
Figuur 16: Webservicesmodel	43

Tabelindex

Tabel 1: Producten per fase	11
Tabel 2: Ontwerpaspectenmodelinterne informatievraag.....	17
Tabel 3: Ontwerpaspectenmodelexterne informatievraag.....	18
Tabel 4: Vergelijkingstabel.....	2 5

GERAADPLEEGDE LITERATUUR

Boeken:

- Bots, R.T.M. & Jansen, W. (2001). Organisatie en Informatie, Groningen: Wolters-Noordhoff.
- Maes, R. (1999). *A Generic Framework for Information Management*, Amsterdam: Universiteit van Amsterdam.
- Maes, R. (1999). *Reconsidering Information Management Through A generic Framework*, Amsterdam: Universiteit van Amsterdam.
- Maes, R. (2003). Informatiemanagement in kaart gebracht, Amsterdam: Universiteit van Amsterdam.
- Truijens, O. (2000). Het strategie Alignmentmodel als hulpmiddel bij IT-strategieadvies, Amsterdam: Universiteit van Amsterdam.

Artikelen:

- Bijl, D. (2003) Van webservices naar SOA, *Microsoft*, .net magazine, 3.
- Elswijk van, M. (2003) Aspecten van een integratiestrategie, *Informatie*, 11.
- Florijn, G. (1997) Software Engineering en Software Architectuur het gebruik van architecturen in het ontwikkelproces, SERC, Universiteit Utrecht.
- Schekkerman, J. (2000) Ketenintegratie en de rol van architectuur, Cap Gemini Ernst & Young.
- Vandenbulcke, J. (2002) Fysieke integratie met webservices, *Informatie*, 12.

Websites:

- <http://java.sun.com/products/jdh/rmi> (RMI)
- <http://msdn.microsoft.com/architecture> (architecturen)
- <http://www.cbs.nl> (centraal bureau voor de statistiek)
- <http://www.degama.com> {Service Oriented Architecture}
- <http://www.gia.nl> (genootschap van informatiearchitecten)
- <http://www.middleware.org> (CORBA, DCOM)
- <http://www.milieuhelp.nl/water> (Wet Verontreiniging Oppervlaktewater)
- <http://www.w3c.org> (XML, SOAP, UDDI, WSDL)

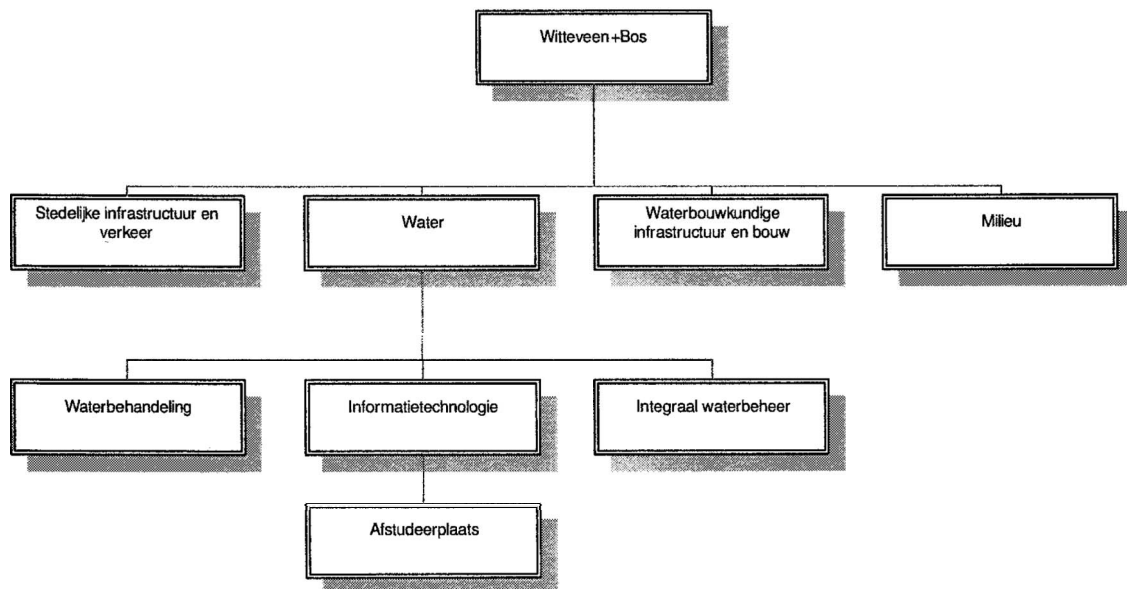
Bijlagen:

BIJLAGE I: ORGANISATIESTRUCTUUR

Organisatieomschrijving

Witteveen+Bos is een ingenieurbureau dat advies geeft in de sectoren bouw, infrastructuur, milieu en water. In deze laatst genoemde sector valt de afdeling Informatietechnologie.

Figuur 10: Organogram



Betrokkenen:

Opdrachtgever:
Witteveen+Bos
Van Twickelostraat 2
7411SC Deventer

Bedrijfsbegeleider:
ir. JGJ Nijhuis
J.Nijhuis@witbo.nl
0570-697327

Docentbegeleider:
Dhr. G. Ovink
g.ovink@fnt.hvu.nl
033-4228941

Opdrachtnemer:
Han van 't Zand
han@van-t-zand.nl
0570-614940

BIJLAGE II: RAMING

Referentiearchitectuur horizontale- en verticale integratie'						
Fase/activiteit	Inzet (mandagen)					
	Totaal	Projectleider H. van 't Zand	Totaal PL	Projectbegeleider ir. J.G.J. Nijhuis	Interview ing. H.P. van Essen	Interview ing. H.W. Jansen
						Totaal Overig
Fase 1: Projectdefinitie	11,25		10,50			0,75
Inlezen		4,00				
Concept Plan van Aanpak opstellen		4,00				
Leestijd Plan van Aanpak				0,25		
Bespreken Plan van Aanpak		0,50		0,50		
Aanpassen Plan van Aanpak		2,00				
Fase 2: Opstellen informatiearchitectuur	19,13		17,25			1,88
Literatuur onderzoek (info vraag)		2,00				
Notitie opstellen		2,00				
Leestijd notitie				0,25		
Bespreken notitie		0,25		0,25		
Interviews voorbereiden		1,00				
Interview met ir. J.G.J. Nijhuis		0,25		0,25		
Interview met ing. H.W. Jansen		0,25			0,13	
Interview extern		1,00				
Notitie aanpassen (Literatuur+Interviews)		3,00				
Leestijd notitie				0,25		
Bespreken notitie		0,25		0,25		
Aanpassen van notitie		1,00				
Literatuur onderzoek (info aanbod)		2,00				
(evt.) Interview		1,00				
Notitie opstellen		2,00				
Leestijd notitie				0,25		
Bespreken notitie		0,25		0,25		
Aanpassen notitie		1,00				
Fase 3: Opstellen ICT-architectuur	26,13		24,50			1,63
Literatuur onderzoek (componenten verbindingen)		15,00				
Notitie opstellen		2,00				
Leestijd notitie				0,50		
Bespreken notitie		0,25		0,25		
Interviews voorbereiden		1,00				
Interview met ir. J.G.J. Nijhuis		0,50		0,25		
Interview met ing. H.P. van Essen		0,50			0,13	
(evt.) Interviews		1,00				
Notitie aanpassen (Literatuur+Interviews)		2,00				
Leestijd notitie				0,25		
Bespreken notitie		0,25		0,25		
Aanpassen van notitie		2,00				
Fase 4: Opstellen referentiearchitectuur	5,75		5,00			0,75
Opstellen conceptmodel		3,00				
Leestijd model				0,25		

Doornemen model		1,00		0,50			
Aanpassen model		1,00					
Fase 5: Toetsing / rapportage	14,50		12,00				2,50
Model toetsen		2,00		0,50			
Uitkomst toetsing analyseren		1,00		1,00			
evt. Model aanpassen		1,00					
Eindrapport		8,00					
Leestijd eindrapport				0,50			
Doornemen eindrapport		0,50		0,50			
Aanpassen eindrapport		2,00					
EXTRA	19,50		18,25				1,25
2 wekelijkse bespreking schoolbegeleider		7,00					
Scriptie (wk21)		5,00					
Beoordeling ir. J.G.J. Nijhuis (wk21)		0,25		0,25			
Afstudeerzitting (wk24/wk25)		1,00		1,00			
Uitlooptijd		5,00					
Subtotaal dagen	96,26		87,50	8,50	0,13	0,13	8,76
kosten			7.490,00	8.934,95	95,98	189,51	9.220,43
Repro & Secretariaat (1 % en 3 %)			299,60				
Subtotaal			7.789,60	8934,95	95,98	189,51	9.220,43
Onvoorzien (10 %)			778,96	893,49	9,60	18,95	922,04
Totaal in EUR			8.568,56	9.828,44	105,58	208,46	10.142,48
Totaal kosten in EUR exclusief omzetbelasting	18.711,04						
Vakantiedagen (Pasen/Hemelvaart/Pinksteren/etc.)		5,00					
Tentamens (28-02 & 14-03)		2,00					
Tarief in EUR H. van 't Zand	85,60						
Tarief in EUR ir. J.G.J. Nijhuis	1.051,17						
Tarief in EUR ing. H.P. van Essen	738,30						
Tarief in EUR ing. H.W. Jansen	1.457,77						
Tarieven zijn dagtarieven (=uurtarief x 8 x 1,07)							

BIJLAGE III: KOSTEN EN BATEN

Baten

De baten van het project worden als volgt benoemd:

- een besparing van tijd voor de ingenieurs die zijn belast met het opstellen van een ICT-architectuur

Kosten

De kosten van het totale project zijn in onderstaande tabel opgenomen. (zie bijlage II voor specificatie van de kosten per fase)

Totale kosten van het project

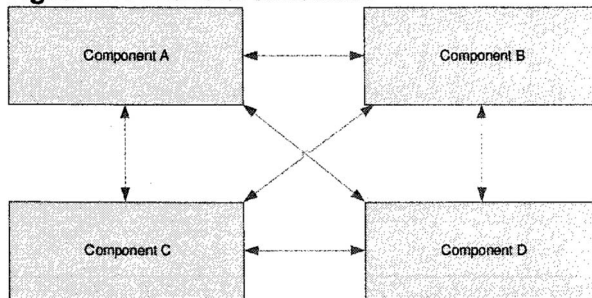
<i>'Referentiearchitectuur horizontale en verticale integratie'</i>			
Fase/activiteit			
		Totaal (dagen)	Kosten(E)
Fase 1: Projectdefinitie		11,25	2187
Fase 2: Opstellen informatiearchitectuur		19,13	3718
Fase 3: Opstellen ICT-architectuur		26,13	5079
Fase 4: Opstellen referentiearchitectuur		5,75	1118
Fase 5: Toetsing / Rapportage		14,5	2819
Extra		19,5	3790
Totaal		96,26	18711

BIJLAGE IV INTERFACE ARCHITECTUUR

Eiland architectuur

Traditioneel worden interfaces tussen componenten op basis van maatwerk ontwikkeld. Vaak met behulp van bestandsoverdrachten of specifiek ontwikkelde databasekoppelingen. **Iedere** interface naar een nieuw systeem moet opnieuw ontwikkeld worden. Er ontstaat zo al snel de zogeheten eiland integratie, een complex geheel van moeilijk te beheren informatiestromen.

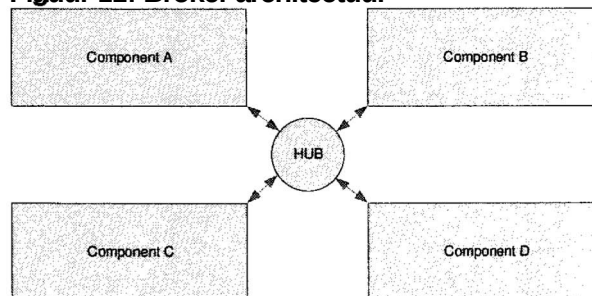
Figuur 11: Eiland architectuur



Broker architectuur

Bij een broker architectuur wordt een groot deel van de complexiteit van de applicatie integratie opgelost met een standaard integratieproduct en standaard integratieconcepten. Elke applicatie kent nog maar één koppeling. Gegevens worden uitgewisseld door middel van bijvoorbeeld XML berichten. De XML definitie is zeer flexibel waardoor verschillende berichtenstromen mogelijk zijn over deze koppeling. Alle berichten gaan langs een centrale "HUB" (zie figuur 2). De "HUB" verzorgt verdere routing van het bericht naar de juiste ontvanger (de "spokes"). Ook eventuele definitieverschillen tussen de applicaties worden opgelost doordat het standaard integratieproduct regels voor vertaling van gegevens kent.

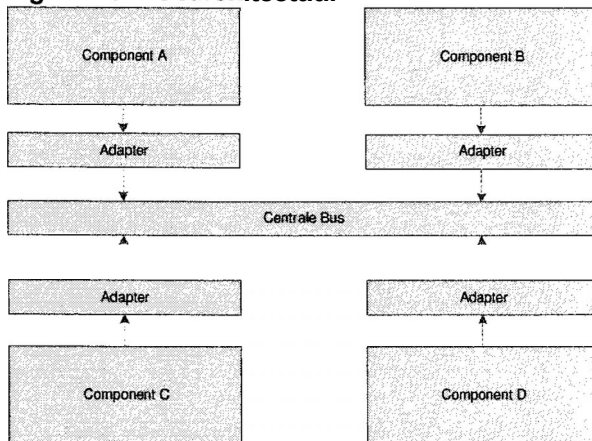
Figuur 12: Broker architectuur



Bus architectuur

Bij integratiestandaarden zien we meestal een bepaalde "busstructuur" in de communicatie, met een gemeenschappelijke protocol (berichtenformaat) waar iedereen op moet inhaken. Eventuele vertalingen van het eigen protocol naar het gemeenschappelijke busprotocol gebeurt dan vaak door een decentrale vertaalcomponent (adapter) een voordeel hiervan is dat de centrale bus relatief eenvoudig blijft, het nadeel is dat de onderhoudslast bij protocol wijzigingen geheel gedecentraliseerd is: iedereen moet zijn eigen vertaalcomponenten aanpassen.

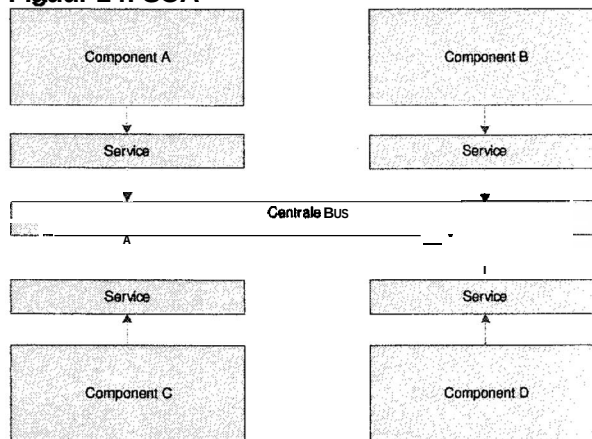
Figuur 13: Busarchitectuur



SOA

Met de komst van webservices is het SOA (Service Oriented architectuur) tijdperk aangebroken. SOA is een standaardisering van interfaces doormiddel van services. Het kenmerk van een soa is dat alle verschillende systemen en applicaties op een eenduidige manier benadert kunnen worden. Karakteristiek voor een SOA is het gebruik van internationale standaarden en een zeer "loosely" coupled manier van integreren. Er wordt gebruik gemaakt van moderne technologieën en standaarden zoals XML, inter- intranet en Business Proces Execution Language (BPEL).

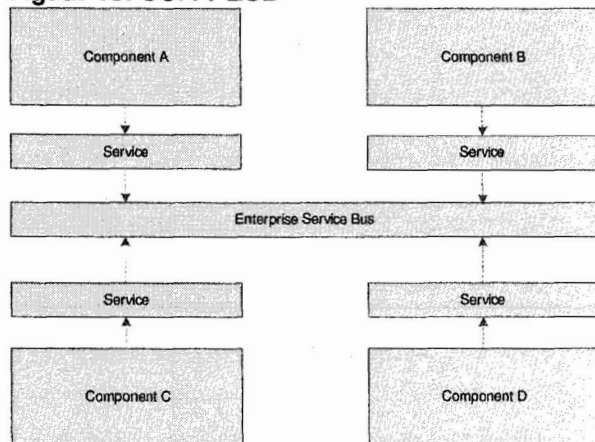
Figuur 14: SOA



ESB

De laatste ontwikkelingen op het gebied van busarchitecturen is de zogenaamde **ESB** (Enterprise Service Bus). Na het definiëren van de gewenste architectuur (in dit geval SOA), volgt de implementatie ervan. Dit gebeurt middels een 'Enterprise Service Bus' (**ESB**). Dit is de implementatievorm van een SOA. Het omvat aspecten van een ontwikkelomgeving, maar ook zaken als procesbesturing, adapters en datatransformatie.

Figuur 15: SOA + ESB



BIJLAGE V MIDDLEWARE

Middleware op basis van berichten (MOM)

Door middel van Message-Oriented Middleware (MOM) kunnen programma's op verschillende systemen berichten met elkaar uitwisselen. Daarbij kunnen ze wachten op antwoord (synchrone communicatie), of gewoon verder functioneren in afwachting van andere berichten (asynchrone communicatie). De berichten die bij een bepaald subsysteem aankomen, blijven opgeslagen in een **wachtrij (queue)** alvorens te worden verwerkt. MOM is een geschikte oplossing voor **peer-to-peer** communicatie, maar ook voor **cliënt/server-systemen** die gebruik maken van gedistribueerde processen.

Externe procedure-aanroepen (RPC)

Een variant van het basismodel voor het doorgeven van berichten is de externe procedureaanroep (Remote Procedure **Call**, RPC). De externe procedureaanroep is een techniek waarbij twee programma's op verschillende machines communiceren via een algemene syntax en semantiek voor procedure-aanroepen en returns. Een 'cliënt' proces roept een functie aan bij de externe server en het proces dat de '**call**' vervaardigd wacht totdat het de resultaten krijgt. Het aangeroepen programma en het aanroepende programma **gedragen** zich beide alsof het partnerprogramma wordt uitgevoerd op dezelfde machine. Dat wil zeggen: de procedureaanroep wordt gebruikt voor de toegang tot externe diensten.

Het programma dat de service aanvraagt kan dit doen zonder de netwerk details te kennen. De code roept een procedure op die zich bevindt op een extern systeem en de resultaten worden teruggegeven. RPC gebruikt het **cliënt/server** model. Het programma dat de aanvraag verstuurt is de cliënt en het programma dat de service verleent is de server. Zoals een gewone **locale** procedureaanroep, is RPC een synchrone (gelijktijdige) operatie die vereist dat de aanvrager geschorst wordt totdat het resultaten van de externe procedure teruggegeven zijn. Het gebruik van **lightweight processes** of **threads** die dezelfde adresruimte delen laten echter toe dat meerdere RPC's gelijktijdig kunnen uitgevoerd worden.

RPC's werken goed voor kleine, eenvoudige applicaties waar de communicatie hoofdzakelijk **point-to-point** is (eerder dan **one system to many**). RPC's zijn niet goed schaalbaar tot grote, bedrijfskritische applicaties door hun synchrone aard. Bovendien **laten** RPC's vele cruciale details over aan de programmeur. Voorbeelden hiervan zijn het afhandelen van **netwerk-** of systeemfouten, de ondersteuning van meerdere connecties, buffering en flow controle, en de synchronisatie tussen processen.

Object request broker

Toen objectgeoriënteerde technologie belangrijker werd bij het besturingssysteemontwerp zijn ook cliënt/server-ontwerpers deze benadering gaan toepassen. Hierbij verzenden **cliënten** en servers berichten heen en weer tussen objecten. Bij objectgeoriënteerde systemen wordt de scheidingslijn tussen data en programma's gesloopt: een **object** bevat zowel informatie als bewerkingen (insiders spreken van **state** en **behaviour**). De **objectcommunicatie** kan afhankelijk zijn van een onderliggende berichten- of RPC-structuur of rechtstreeks boven op de objectgeoriënteerde mogelijkheden van het besturingssysteem worden ontwikkeld. Een cliënt die een dienst nodig heeft, verzendt een verzoek naar een object request broker (ORB), een 'makelaar in objectverzoeken', die werkt als een overzicht van alle externe diensten die beschikbaar zijn in het netwerk. De broker roept het juiste object aan en **geeft** daaraan alle relevante gegevens door. Vervolgens behandelt het externe object het **verzoek** en verzendt het een antwoord naar de broker, die het antwoord doorzendt naar de cliënt. De ondersteuning van ORB in een netwerk laat de cliënt toe om een service aan te vragen zonder **te** weten waar de server zich in het gedistribueerde netwerk bevindt, **en/of** hoe de interface tot die server er uit ziet. Componenten kunnen elkaar herkennen en interface informatie uitwisselen tijdens hun uitvoering. Het succes van de objectgeoriënteerde benadering is afhankelijk van de standaardisatie van het **objectmechanisme**. Helaas bestaan er op dit gebied enkele concurrerende ontwerpen. Een daarvan is het **Component Object Model (COM)** van Microsoft. Een concurrerende benadering, die werd ontwikkeld door de Object Management Group, is de **Common Object Request Broker**

Architecture (CORBA). Java's Remote Method Invocation (RMI) kan eveneens beschouwd worden als een ORB. RMI is echter hoofdzakelijk nuttig bij het vereenvoudigen van de communicatie tussen Java programma's en houdt zich niet bezig met andere programmeertalen, terwijl DCOM en CORBA wel andere talen ondersteunen. RMI is een manier waarop Java-objecten (software componenten) elkaar kunnen aanroepen. Hiermee is het mogelijk Java-objecten die zich op verschillende computers bevinden te laten samenwerken. RMI werkt alleen met softwarecomponenten die in Java geschreven zijn (om ook niet-Java programma's te kunnen aanroepen moet RMI over **IIOP** (zie *Corba*) worden gebruikt). Omdat Java geschikt is voor vrijwel alle hardwareplatforms kan er met behulp van RMI een gedistribueerd systeem over meerdere hardwareplatforms worden gerealiseerd.

Component Object Model (COM)

Het **Microsoft** Component Object Model (COM) wordt gekenmerkt door het gebruik van binaire interfaces waardoor dit model zich grotendeels taalafhankelijk mag noemen. Het is een component model dat de meeste **OO** (Object Oriented) principes goed ondersteunt, behalve overerving. Opvallend is het gebrek aan netwerk transparantie: er is **locatie** transparantie voor componenten onderling op een enkele computer op voorwaarde dat het **Microsoft Windows** platform gebruikt wordt. Tussen verschillende computers onderling is er, ondanks het **RPC** protocol waarop COM communicatie steunt, geen netwerk communicatie voor **COM** mogelijk. Hieraan probeert men, zonder veel succes, een mouw aan te passen door middel van een uitbreiding van COM: **Distributed COM** (DCOM). Communicatie tussen verschillende componenten die op verschillende computers verblijven via een netwerk verbinding is nu wel mogelijk, maar van transparantie is nog altijd geen sprake.

Common Object Request Broker Architecture (CORBA)

CORBA (Common Object Request Broker Architecture) is een architectuur en specificatie voor het maken, distribueren en beheren van gedistribueerde objecten. **CORBA** is opgesteld door de Object Management Group (OMG), een samenwerkingsverband van meer dan 700 ICT leveranciers. Zowel de International Organization for Standardization (**ISO**) als **X/Open** hebben **CORBA** uitgeroepen tot de standaard architectuur voor gedistribueerde objecten. Het grote verschil met COM is dat **CORBA** een specificatie is en geen implementatie. **CORBA** gebruikt een Interface Definition Language (IDL) waarmee het de interface van componenten specificeert. Door middel van deze interface definitie kan men elke taal gebruiken voor de implementatie waarvoor de desbetreffende **CORBA** Object Request Broker (ORB) een '*mapping*' voorziet. **CORBA** is dus grotendeels taalafhankelijk.

Database middleware

Database middleware is een set van technologieën en standaards die het opvragen van data uit verschillende types van databases, op verschillende platformen **en/of** in meerdere locaties, eenvoudiger maakt. Het levert de connectie tussen een applicatie en een database. Database middleware implementeert een **RDA** (Remote data Access) protocol voor het zenden van *data manipulation language statements* naar de juiste Database server. Nadat de Database server de aanvraag verwerkt heeft wordt het resultaat teruggezonden door RDA naar het verzoekende proces. Het RDA concept maakt het mogelijk applicaties te maken die beschikbaar gesteld kunnen worden over het Internet aan een cliënt Database Middleware wordt gekenmerkt door het heen en weer vervoeren van **SQL** en database gegevens tussen de applicatie en de database. Bovendien helpt het bij het garanderen van de beveiliging en het vermijdt dat applicaties direct moeten handelen met de database server. Database middleware is beschreven door verscheidene connectivity standaards zoals *Open DataBase Connectivity (ODBC)*, *Java DataBase Connectivity (JDBC)* en productspecifieke standaards.

Open DataBase Connectivity (ODBC)

ODBC is een verzameling van database access middleware drivers die programma's toegang verleent tot de data in een database. ODBC definieert enkel de cliënt zijde van de database connectiviteit. **Voor** de server zijde **veronderstelt** ODBC de aanwezigheid van een bedrijfseigen ODBC-driver (bv: **SQL*Net** voor Oracle). De databaseleverancier moet hiervoor dus een extra stuk software ontwikkelen (de ODBC-driver) waarmee de gegevens extern benaderd kunnen worden. ODBC drivers transformeren ODBC **calls** naar leverancierspecifieke access requests en responses. Dit **heeft** zowel **voor-** als nadelen: ODBC biedt een standaard interface, maar zonder rekening te houden met de eigenschappen van de bedrijfseigen middleware. Hierdoor consumeert ODBC relatief **veel** geheugen en vertraagt het de toegang tot de data in een database. De laatste variant hierop is OLEDB deze doet hetzelfde maar is nieuwer, flexibeler en sneller. Nadeel t.o.v. ODBC is dat deze minder databases ondersteund.

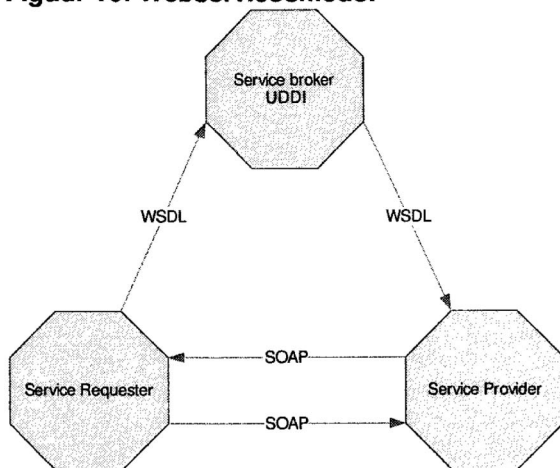
JAVA DataBase Connectivity (JDBC)

JDBC is een verzameling van database access middleware drivers die Java **applets, servlets of** andere Java applicaties toegang verleent tot data van meerdere database omgevingen zoals onder meer Oracle en Sybase. De werking van JDBC is gelijk aan die van ODBC.

BIJLAGE VI WEBSERVICES

Webservices zijn te definiëren als een mogelijkheid voor applicaties om met elkaar te communiceren over intra- internet. Er ontstaat dus een mogelijkheid voor een koppeling of integratie van twee, mogelijk qua architectuur totaal verschillende, systemen zonder hoge kosten. Webservices zijn te vinden als communicatiemiddel tussen bedrijf en **klant/afnemer**, maar ook tussen verschillende applicaties binnen een bedrijf. Drie technieken staan hierbij centraal: Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) en **Universal** Description, Discovery, and Integration (UDDI). Al deze technieken zijn gebaseerd op XML, en zijn open standaarden die gedragen worden door een groot aantal bedrijven, waaronder IBM en Microsoft. UDDI is een directory protocol waarin bedrijven op zoek kunnen naar webservices die worden aangeboden door andere bedrijven. Als er een geschikte webservice is gevonden kan de beschrijving van deze webservice via een WSDL document. Hierin staat beschreven waar de webservice zich bevindt en hoe de berichten eruit zien die naar de **webservice** moeten worden gestuurd en die van de webservice terugkomen. De berichtuitwisseling tussen de twee applicaties wordt gedaan door berichten met behulp van het SOAP protocol. Uitwisseling van SOAP berichten loopt in de regel via het standaard Internet protocol '**http**', maar SOAP berichten zijn ook aan andere transport protocollen te koppelen.

Figuur 16: Webservicesmodel



Een van de problemen met webservices is de veiligheid. Een recente ontwikkeling op dit gebied is de standaard **Ws-Security**, voor het veilig versturen van SOAP berichten. Voor authenticiteit van berichten is er op dit moment echter nog geen eenduidige oplossing. Verschillende bedrijven komen met een eigen oplossing. Standaardisatie lijkt dan ook de grootste bedreiging voor het succes van webservices. Als organisaties verschillende, tegengestelde belangen hebben duurt het ontwikkelen van standaarden erg lang. Maar ook de patenten die verschillende grote organisaties over delen van de technologie aanvragen, komt de ontwikkeling van webservices niet ten goede. Recente ontwikkelingen geven echter aan dat bedrijven bereidt zijn hun patenten zonder royalty's voor de standaarden aan te bieden, mits alle bedrijven dat doen. **Het** succes voor webservices in de toekomst hangt hier ook sterk vanaf. Bedrijven gaan in **de** economisch moeilijkere tijden steeds meer kijken of de ontwikkelingskosten opwegen tegen de economische voordelen. Dit lijkt zeker het geval, op korte termijn vooral binnen bedrijven. De langere termijn voor webservices is zeer onduidelijk, maar ziet er **volgens** analisten zeer gunstig uit.