

Opdrachtgever
Rijkswaterstaat RIZA
Contactpersoon
Dhr. J. Rienks
Vertis / CSO adviesbureau
Contactpersonen
Dhr. R. Beikes (Vertis) Dhr. J. Rijnbeek (CSO)



m i l i e u
r u i m t e
w a t e r

Software architectuur WABinfo

- onderdeel van het Functioneel Ontwerp WABinfo -

Opdrachtgever

Rijkswaterstaat RIZA
Postbus 17
8200 AA Lelystad
Telefoon: 0320 – 298411
Fax: 0320 – 249218
Contactpersoon
Dhr. J. Rienks

Vertis bv / CSO adviesbureau

Contactpersonen
Dhr. R. Beikes (Vertis)
Telefoon: 0598 – 666028
e-mail: beikesr@vertis.nl

Dhr. J. Rijnbeek (CSO)
Telefoon: 030 – 6594 391
e-mail: j.rijnbeek@cs0.nl

Projectcode Vertis bv / CSO

RDIJRPV01 / 04.W011.00

Datum

November 2004

Auteurs

J. Clevering, M. de Kwant, G. Walrecht, R. Alders
(Vertis)

Status

versie 1.3

INHOUDSOPGAVE

1	INLEIDING.....	4
1.1	INTRODUCTIE.....	4
1.2	VOORWAARDEN ARCHITECTUUR	4
1.3	LEESWIJZER.....	4
2	ARCHITECTUUR WABINFO	5
2.1	ALGEMEEN	5
2.2	ADVIES TECHNOLOGIE KEUZE WABINFO	5
2.2.1	TECHNOLOGIE GEGEVENSLAAG	5
2.2.2	TECHNOLOGIE APPLICATIE- EN PRESENTATIELAAG.....	6
2.2.3	ADVIES	7
2.3	HERGEBRUIK ARCHITECTUREN	7
2.4	HET MVC MODEL.....	8
2.5	ALGEMENE ARCHITECTUUR BINNEN DE WABINFO.....	8
3	EXTERNE DATABRONNEN	10
3.1	ARCHITECTUUR	10
3.2	KOPPELING WADI.....	11
3.2.1	MODELLERING PARTIJ MET MEETGEGEVENS	11
3.2.2	RELATIE PARTIJ MET MEETGEGEVENS	11
3.2.3	AANMAKEN VAN PARTIJ EN MEETGEGEVENS	11
3.2.4	UPDATE VAN PARTIJ EN MEETGEGEVENS	11
3.2.5	VERWIJDEREN VAN PARTIJ EN MEETGEGEVENS.....	12
3.2.6	PERFORMANCE EN BESCHIKBAARHEID.....	12
3.3	KOPPELING GEOSERVICES	14
3.3.1	GEO-RAADPLEGEN	14
3.3.2	GEOMUTEREN	15
3.4	DATABASE BEHEER.....	16
3.4.1	ORACLE ENTERPRISE MANAGER	16
3.4.2	PATROL EXPRESS	16
4	BEVEILIGING	17
4.1	HORIZONTALE AUTORISATIE.....	17
4.1.1	AUTORISATIE IN GEGEVENSLAAG VERSUS APPLICATIELAAG.....	17
4.1.2	AFSCHERMEN GEGEVENS AFHANKELIJK VAN ROL.....	17
4.1.3	GEGEVENSBEVEILIGING WADI IN RELATIE TOT WABINFO	18
4.2	VERTICALE AUTORISATIE	18
4.3	ADVIES	18
5	CONCLUSIE.....	19
6	LITERATUUR.....	20

1 INLEIDING

1.1 INTRODUCTIE

In dit document wordt het technisch raamwerk voor WABInfo beschreven, is onderdeel van het Functioneel Ontwerp WABInfo en in opdracht van het Rijksinstituut voor Intergraal Zoetwaterbeheer en Afvalwaterbehandeling (RIZA) opgesteld door de projectcombinatie Vertis/CSO. Het informatiesysteem WABInfo richt zich op het verbeteren van de informatiehuishouding rond waterbodemerelateerde processen bij Rijkswaterstaat.

Bij de beschrijving van het technisch raamwerk van WABInfo wordt rekening gehouden met de visie ten aanzien van de architectuur die de opdrachtgever voor ogen heeft. Deze visie staat beschreven in het document *KANS 2: Vooronderzoek UCN, SRN en GAIN, 16 april 2004, Concept versie 0.3*. Daarnaast is rekening gehouden met de wensen van de opdrachtgever om expliciet de werking van WABInfo in relatie tot WADI en Geoservices te omschrijven.

1.2 VOORWAARDEN ARCHITECTUUR

De opdrachtgever heeft een aantal voorwaarden gesteld aan de architectuur en toepassing van WABInfo. Deze voorwaarden zijn bepalend voor de invulling van de architectuur en de toe te passen technologieën. Hieronder worden alle voorwaarden benoemd:

- WABInfo moet beschikbaar zijn via het Internet (Webbased)
- WABInfo moet in staat zijn om data op te halen uit andere applicaties m.b.v. zogeheten Webservices. (vlgs architectuur KANS 2)
- Zoveel mogelijk gebruik maken van open standaarden (XML, J2EE, SOAP, Wsdl , UDDI)
- Mogelijkheid om meerdere user interfaces te definiëren op dezelfde businesslogica.

1.3 LEESWIJZER

Onderhavig document is als volgt opgebouwd: de architectuur van WABInfo is gelijk aan WADI wat betreft de toepassing van 3-tier, presentatie-, applicatie- en gegevenslaag. De technologiekeuze om invulling te geven aan deze architectuur is door de opdrachtgever nog niet onderstreept. Om toch een concrete architectuurschets bij de opdrachtgever neer te leggen wordt in hoofdstuk 2 daarom begonnen met een advies ten aanzien van de technologiekeuze voor WABInfo. De technologieën die worden die in dit hoofdstuk worden aangedragen zijn daarna in de navolgende hoofdstukken waar nodig verder uitgediept.

De opdrachtgever heeft aangegeven behoefte te hebben aan een nauwgezette beschrijving van de werking van Geoservices en WADI in relatie tot WABInfo. Hoofdstuk 3 Externe databronnen behandelt beide koppelingen uitvoerig.

De beveiliging van WABInfo in relatie tot WADI is een punt van discussie. Hoe autorisatie voor WABInfo is opgebouwd en wat er in relatie met WADI mogelijkwijs moet gebeuren staat in hoofdstuk 4 beschreven.

Hoofdstuk 5 behandelt mogelijke beheertools voor een Oracle database.

Tot slot wordt in het laatste hoofdstuk 6 een korte samenvatting weergegeven met conclusies.

2 Architectuur WABInfo

2.1 ALGEMEEN

Voordat gestart kan worden met de technische realisatie van WABInfo is het belangrijk te weten volgens welke architectuur en met welke technologie de realisatie plaats gaat vinden. Architectuur en technologie zijn onlosmakelijk met elkaar verbonden: het architectuurplaatje bepaalt welke technologieën worden toegepast en omgekeerd.

WABInfo hangt nauw samen met WADI en zal evenals WADI toegankelijk zijn via het internet en ook integratie functionaliteit beschikbaar stellen in de vorm van webservices.

De architectuur zoals geschetst volgens het AGI in het document KANS 2 impliceert dat gegevensuitwisseling tussen applicatiesystemen binnen RWS plaats gaat vinden met webservices. Voordeel hiervan is onder meer dat onafhankelijkheid wordt gecreëerd tussen afzonderlijke applicatiesystemen ten aanzien van platvormen, databases en invulling van de ontwikkeltechnologie. Vraagtekens worden nog gezet bij de haalbare performance die deze architectuur biedt bij onder meer de gegevensuitwisseling tussen WABInfo en WADI.

2.2 ADVIES TECHNOLOGIE KEUZE WABINFO

De opdrachtgever heeft aangegeven dat WABInfo zoveel mogelijk aan moet sluiten op de architectuur waarop WADI is gebaseerd. WADI is een 3-tier oplossing, bestaande uit een gegevens-, applicatielogic- en presentatielaag. Deze lagen worden technisch gevormd door een Oracle 9i Database voor de gegevensopslag, de applicatielogica bevindt zich op een 9iAS applicatieserver waarbij de nodige softwarecomponenten zijn ontwikkeld in J2EE m.b.v. Oracle Jdeveloper. WADI heeft nog niet echt een presentatielaag. Inmiddels is voor WADI een demo ontwikkeld in J2EE.

Vraag is of WABInfo in technische zin aan moet haken op WADI. Een duidelijke keuze met welke technologieën de 3-tier architectuur wordt ingevuld is nog niet gemaakt.

Hieronder volgt een toelichting van een aantal producten die breed gangbaar zijn in de markt om een 3-tier architectuur te kunnen realiseren. De producten worden opgesplitst in producten die de gegevenslaag en producten die de applicatie- en presentatielagen kunnen realiseren.

2.2.1 TECHNOLOGIE GEGEVENS LAAG

In de markt zijn meerdere technologieën voor handen om de gegevens-, applicatie- en presentatielaag te realiseren.

Voor de gegevensopslag zijn op de markt allerhande databases te krijgen. Hieronder staan de vier meest gangbare databases:

- Acces
- MySQL
- SQL Server
- Oracle

Acces database

Zie voor meer informatie: <http://office.microsoft.com/assistance/topcategory.aspx?TopLevelCat=CH79001800>

- Eenvoudige filebased database, geen modern en professioneel RDMS systeem
- Microsoft product, functioneert alleen op Microsoft platformen
- Biedt weinig functionaliteit ten aanzien van performance tuning
- Niet schaalbaar, niet geschikt voor grootschalig gebruik.
- Ondersteunt geen geografie
- Goedkoop

MySQL

Zie voor specificaties: <http://www.mysql.com/products/>

- Modern en professioneel RDMS database systeem in wording
- Platform onafhankelijk

- Open source database
- Biedt goede integratiemogelijkheden met open standaarden als XML en JAVA
- Biedt binnen de database geen standaard voorgedefinieerde functionaliteit aan voor applicatieontwikkeling. Zo beschikt het over een beperkte range aan SQL features. Procedures onderbrengen in de database is niet mogelijk waardoor naar alternatieve oplossingen gegrepen moet worden die doorgaans arbeidsintensief zijn om te realiseren. Dit leidt doorgaans tot relatief veel programmeerwerk.
- Ondersteunt geen geografie
- Relatief goedkoop ten opzichte van onderstaande twee databases

SQL Server

Zie voor specificaties: <http://www.microsoft.com/sql/evaluation/overview/default.asp>

- Modern en professioneel RDMS database systeem
- Microsoft product, functioneert alleen op Microsoft platvormen
- Biedt goede integratiemogelijkheden met Microsoftproducten als .NET
- Ondersteunt de open standaard XML
- Biedt voor applicatieontwikkeling binnen de database een beperkte hoeveelheid standaard voorgedefinieerde functionaliteit aan. Beschikt over een beperkte range aan SQL features. Procedures onderbrengen in de database is niet mogelijk waardoor naar alternatieve oplossingen gegrepen moet worden die doorgaans arbeidsintensief zijn om te realiseren. Dit leidt doorgaans tot relatief veel programmeerwerk.
- Ondersteunt geen geografie
- Beperkt schaalbaar, alleen hardwarematig middels CPU's en schijfruimte
- Redelijke prijsstelling.

Oracle 10g (9i) database

Zie voor details: <http://otn.oracle.com/products/database/oracle10g/index.html>

- Modern en professioneel RDMS database systeem
- Platform onafhankelijk, presteert uitstekend op Linux, Unix en Microsoft systemen.
- Database zelf is geen open source.
- Biedt goede integratiemogelijkheden met open standaarden als XML en JAVA
- Biedt een erg breed scala aan standaard functionaliteit in de database die applicatieontwikkeling versnellen, brede range aan SQL features en voorgedefinieerde procedures. Bestaande standaard functionaliteit drukt het hoeveelheid programmeerwerk en daarmee de ontwikkelkosten.
- Ondersteunt geografie (Oracle spatial)
- Database is volledig geïntegreerd met de Oracle 10g applicatieserver.
- Erg goed softwarematig schaalbaar (RAC e.d.)
- Duur, maar de door grote hoeveelheid standaard functionaliteit en draaiende onder een Linux configuratie worden deze kosten gedrukt.

Oracle is het summum van RDBMS systemen die er in de markt te krijgen is. Ten aanzien van WABinfo bieden het geografische aspect in de database, brede scala aan standaard functionaliteit en schaalbaarheid een groot voordeel. Nadeel zijn de hoge licentiekosten en de afhankelijkheid van Oracle.

De hoge licentiekosten worden deels gedrukt door de voordelen die tijdens applicatieontwikkeling en beheer en onderhoud worden behaald als ook het kunnen gebruiken van Linux platvormen waarvoor de exploitatiekosten relatief laag zijn.

De afhankelijkheid m.b.t. de database voor gegevensopslag wordt bepaald door de strategie die andere databaseleveranciers voor ogen hebben. Afhankelijkheid is in die zin betrekkelijk en mag geen doorslag geven in de keuze. Want, er is geen database leverancier die migratie mogelijkheden biedt om zijn eigen database te migreren naar die van de concurrent. Omgekeerd wel, Oracle bijvoorbeeld biedt standaard migratiesoftware om bijvoorbeeld Access of SQL server databases over te zetten naar een Oracle database.

2.2.2 TECHNOLOGIE APPLICATIE- EN PRESENTATIELAAG

Hieronder worden de drie meest voor de hand liggende omgevingen opgesomd die momenteel op de markt zijn om applicatie- en presentatielagen te realiseren:

- J2EE(Sun, IBM, Oracle...)
- .NET(Microsoft)
- Oracle Webforms.

J2EE

- Open Source
- Platform onafhankelijk. Draait op Unix, Windows, Linux en mac os.
- Breed gedragen door allerlei softwareleveranciers
- Standaard objecten om webservices te consumeren en te publiceren
- Goede integratiemogelijkheden met databases (Oracle, MySQL enz.)
- Proven technologie, lang op de markt, hierdoor zijn er vele design patterns beschikbaar

.NET

- Draait maar op één platform, namelijk Microsoft Windows.
- Excellente integratiemogelijkheden met Microsoft producten zoals Office
- Mogelijkheden om webservices te consumeren en publiceren
- Integratiemogelijkheden met databases (SQL sever, Oracle enz.)
- Het framework .NET is nog niet zo lang op de markt maar is breed inzetbaar
- Wordt alleen door een Microsoft applicatie server ondersteund.

Oracle Webforms

- 4gl programmeertaal. De ontwikkeltijd is relatief kort vergeleken met de vorige twee (3gl) opties door gebruik CASE omgeving en generatie van code.
- Toegankelijk voor bijna ??
- Geen standaard mogelijkheden voor consumptie van webservices die gebruik maken van soap
- Minder transparant dan voorgaande ontwikkelomgevingen.
- Wordt alleen ondersteund door de Oracle application server.

2.2.3 ADVIES

Gezien de hierboven genoemde kenmerken van de technologieën en de voorwaarden zoals in hoofdstuk 1.2 gestipuleerd, is Oracle de geschikte database technologie om de gegevenslaag te ondersteunen. Daarnaast lijken zowel J2EE als .NET geschikte technologieën om te kunnen gebruiken voor het realiseren van de applicatie- en presentatielaag.

Echter met WADI is de Oracle database J2EE weg ingeslagen in combinatie met Linux als operating system. Linux is relatief goedkoop evenals de hardware waarop het functioneert. De keuze om de gegevens-, applicatie- als presentatielaag van WABinfo op MySQL en .NET technologie te baseren brengt in relatie met WADI een paar min punten naar voren:

1. een verhoogd risico op technische/compatibiliteits problemen;
2. bij het toepassen van meer verschillende technologieën wordt ontwikkelen, integratie, onderhoud en beheer meer complex en daarmee kostbaarder.

Gezien de weg die RIKZ is ingeslagen met WADI, lijkt het voor de hand te liggen met WABinfo, niet alleen dezelfde architectuur te volgen, maar ook toegepaste technologieën Linux, Oracle 10gAS met Oracle database en J2EE.

In de navolgende paragrafen wordt de architectuur die te gebruiken is voor WABinfo met gebruikmaking van J2EE nader beschreven.

2.3 HERGEBRUIK ARCHITECTUREN

Tegenwoordig worden niet alleen objecten binnen OO programmeertalen hergebruikt. Ook wordt steeds meer gezocht naar standaard architectuur oplossingen. Wanneer de doelstellingen en eisen van een applicatie duidelijk zijn, is het mogelijk om op zoek te gaan naar zogenoemde ontwerppatronen ("design patterns") die voldoen aan de gestelde voorwaarden. De ontwerppatronen zijn ontstaan uit vele aangepaste ontwerpen en veranderde coderingen. Dit komt doordat ontwerpers naar betere mogelijkheden voor hergebruik en meer flexibiliteit in hun software hebben gestreefd.

Ontwerppatronen geven deze oplossingen weer in een beknopte en makkelijke toepasbare vorm. Ze vereisen geen ongebruikelijke taalkenmerken of verbazingwekkende programmeertrucs. Een van de meeste bekende ontwerppatronen is het MVC model. Door gebruik te maken een dergelijke architectuur wordt een applicatie transparant, flexibel en eenvoudig uitbreidbaar.

2.4 HET MVC MODEL

Het MVC model is een breed gedragen software pattern welke veel gebruikt wordt in combinatie met een J2EE framework, .NET en andere ontwikkelplatformen. Hierover meer verderop in dit document.

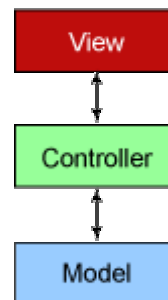
Op hoofdlijnen richt de architectuur zich op:

- strikte scheiding tussen de “kern” applicatie functionaliteit en de user-interface, zodat de user-interface gemakkelijk aangepast kan worden zonder effecten op de kern functionaliteit
- ondersteuning voor meerdere user-interfaces voor één applicatie, waarbij de verschillende interfaces onderling, en ten opzichte van de applicatie, consistent zijn.

Binnen de MVC architectuur worden drie rollen onderscheiden, te weten:

- het **model**, de kern applicatiefunctionaliteit (gegevens, operaties)
- de **view**, een presentatie van (bepaalde gegevens uit) het model
- de **controller** de verwerking van input voor een bepaalde view

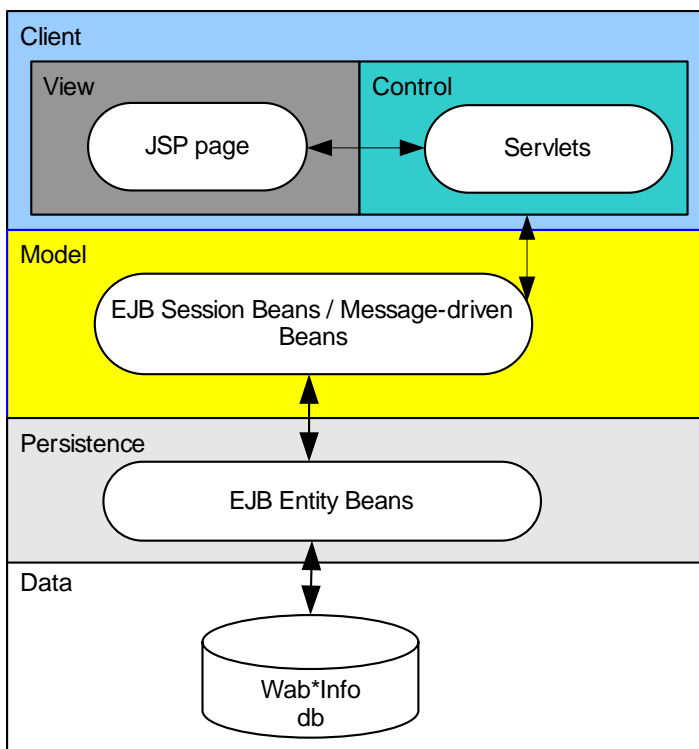
Een view en controller vormen samen de “user interface” voor een model.



2.5 ALGEMENE ARCHITECTUUR BINNEN DE WABINFO

Hieronder wordt de uitwerking van WABinfo getoond volgens een MVC architectuur binnen een J2EE framework.

Wab*Info (MVC model) binnen een j2ee framework



Afbeelding 1 WABinfo (MVC model) binnen een J2EE framework

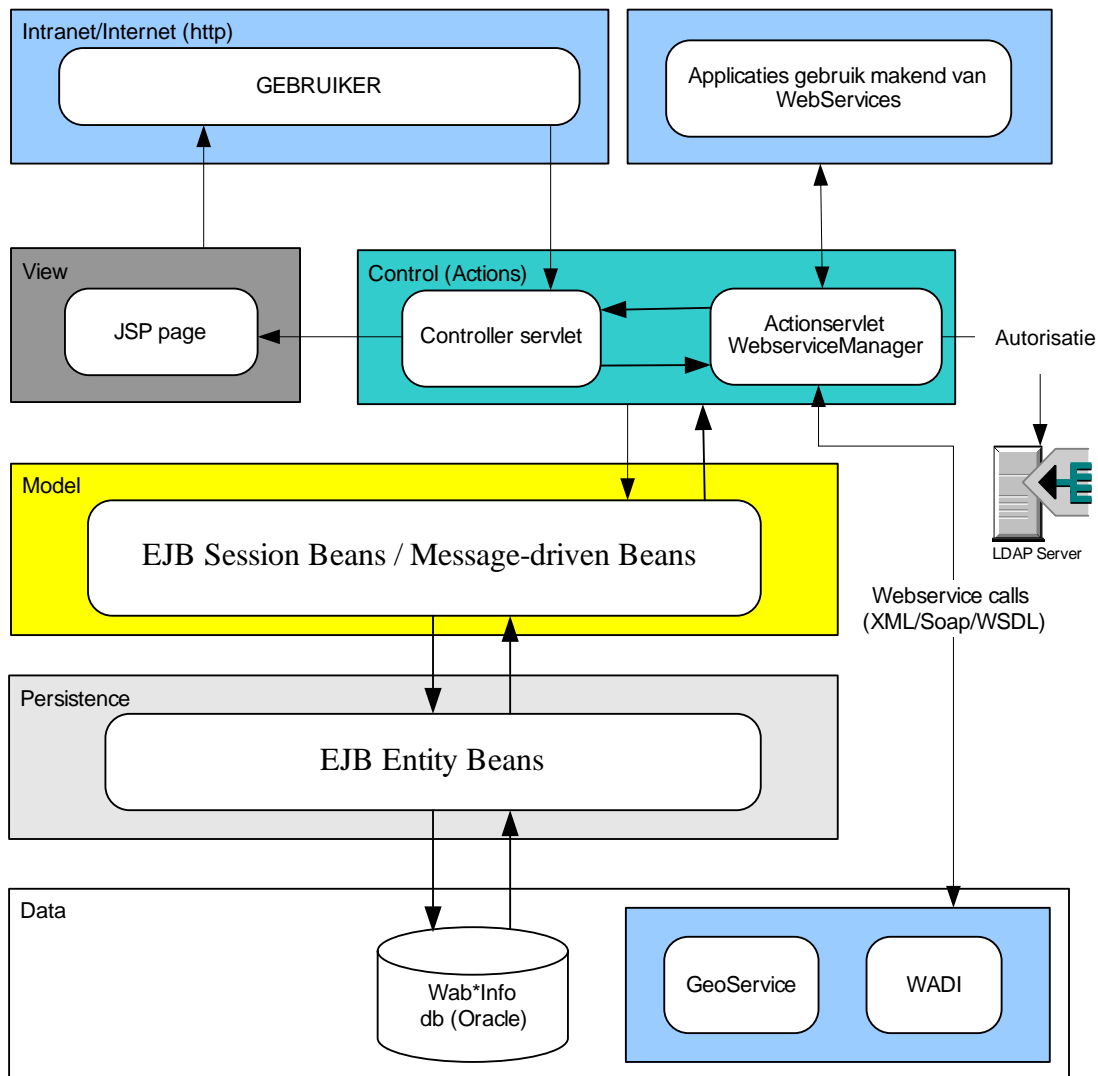
De view wordt gerealiseerd met behulp van bijvoorbeeld een jsp pagina en de servlets zullen gaan fungeren als controller. Het is duidelijk dat er een strikte scheiding is tussen de user-interface en de applicatie. Het model weet niet welke user-interfaces er bestaan en hoe deze werken. De views weten welk model ze presenteren en vragen, wanneer dit nodig is, gegevens op uit het model en tonen deze. Op dezelfde wijze zal de controller bepaalde input-

acties op het model (en de view) uitvoeren die kunnen leiden tot veranderingen van gegevens cq. toestanden in de WABinfo applicatie.

In het bovenstaande overzicht mist nog de interactie met WADI en GeoServices. Binnen de WABinfo applicatie moeten zowel meetgegevens uit WADI en een grafische weergave worden getoond m.b.v. GeoServices. De WABinfo applicatie zal gaan beschikken over meerdere databronnen, waarbij alle externe databronnen benaderd zullen worden via webservices. De client (view) zal echter niet zien dat de daadwerkelijke gegevens uit verschillende databronnen afkomstig zijn.

Nu de controllerlaag duidelijk gescheiden is van de view (presentatielaag), kan in deze laag ook bepaald worden of het een zoekvraag betreft voor WABinfo's eigen model, dat informatie opgehaald moet worden uit WADI of dat een GeoServices geraadpleegd moet worden omdat het geo-informatie betreft. Het ligt dus voor de hand om de controller laag uit te breiden met een webservice controller. Deze webservice controller moet er voor zorgen dat externe webservices benaderd kunnen worden, data verzameld wordt en dat deze data vervolgens in de view gepresenteerd kan worden.

Hieronder zijn alle verschillende onderdelen van de applicatie weergegeven, inclusief de controller welke alle via de webservice binnengekomen aanvragen afhandelt.

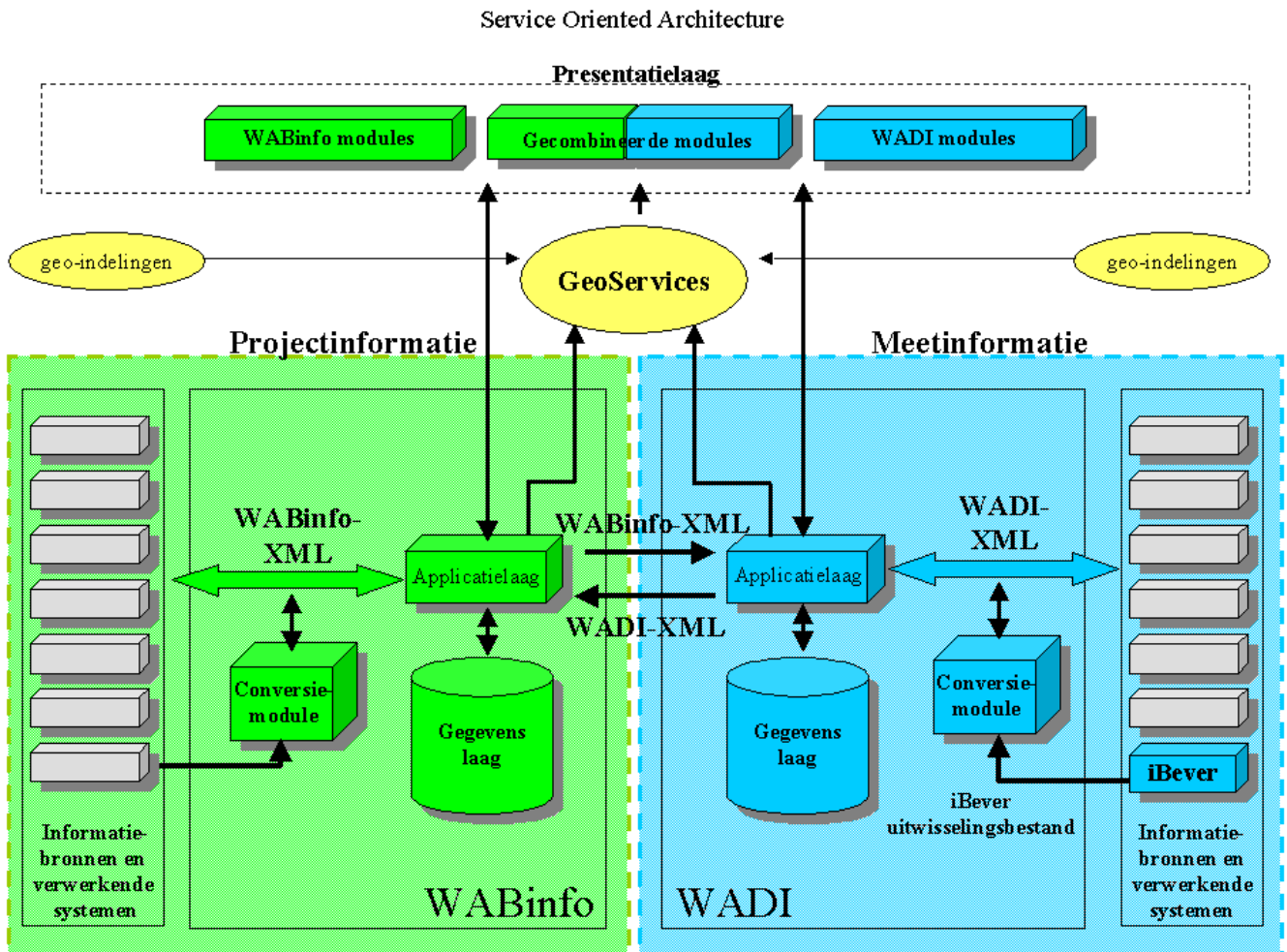


Afbeelding 2 WABinfo architectuur in J2EE framework, gebruik makend van webservices

3 Externe databronnen

3.1 ARCHITECTUUR

In het schema hieronder staat WABinfo weergegeven in relatie tot WADI en Geoservices.
 WABinfo wisselt projectinformatie uit met externe informatiebronnen en verwerkende systemen. WADI doet hetzelfde maar dan voor meetinformatie. De externe informatiebronnen en verwerkende systemen hoeven voor WABinfo en WADI niet perse, in de praktijk zal dat ook niet zo zijn, hetzelfde te zijn. Communicatie tussen WABinfo en externe omgevingen loopt tussen een webservice die het WABinfo XML format ondersteund, voor WADI loopt communicatie tussen externe omgevingen met een webservice die het WADI XML format ondersteund.
 Voor de koppeling van WABinfo met WADI – zie voor meer detailinformatie hierover paragraaf 3.2 - wordt er vanuit gegaan dat partijgegevens worden ondergebracht in WADI en dat meetinformatie vanuit de WABinfo beheerschermen opgevaardeerd kunnen worden. Voor deze koppeling zijn minimaal twee webservices nodig tussen WABinfo en WADI, namelijk de WADI-XML webservice die partij en meetgegevens doorgeeft aan WABinfo en de WABinfo-XML webservice die eventuele wijzigingen aan partij en meetgegevens terug stuurt naar WADI.
 De applicatielagen bevragen/sturen GeoServices via de daarvoor beschikbaar gestelde geo-webservices, en geoservices genereert het plaatje volgens de gevraagde context en stuurt deze door naar de presentatielaag. De presentatielaag triggert de applicatielagen. De applicatielagen triggeren de gewenste aansturing en applicatieloga. De presentatielaag kan bestaan uit combinaties van formulieren (stukken functionaliteit geclusterd in een module) Welke formulieren tot WADI of WABinfo behoren, wordt geregeld door autorisaties. In principe zijn zonder autorisaties alle formulieren voor iedereen toegankelijk of andersom, het is maar hoe het geregeld wordt.



3.2 KOPPELING WADI

Naar het zich nu laat aanzien zullen meetgegevens ten bate van WABinfo worden ondergebracht in de database van WADI. Alle projectgegevens worden ondergebracht in WABinfo. Deze verdeling is logisch, WABinfo projecten, WADI meetgegevens. Partijgegevens is een discussiepunt. Naar het zich nu laat aanzien wordt het opgenomen binnen WADI. Een vraag tijdens een overleg was of een partij opgenomen kan worden in de gegevensverzameling MONSTEROBJECT. Na de toelichting hieronder wordt duidelijk dat het antwoord hierop is dat een partij niet als subtype opgenomen kan worden in de WADI gegevensverzameling MONSTEROBJECT en een eigen gegevensverzameling voor partijen blijft bestaan met daarnaast een koppeltabel. Deze laatste, de koppeltabel, zal de fysieke koppeling zijn tussen de projectgegevens van WABinfo en de meetgegevens in WADI.

Hieronder wordt de “onderwater” relatie cq. koppeling met WADI beschreven met WABinfo. Daarbij wordt ondermeer aandacht gegeven aan de status van meetgegevens in WADI bij gebruik in WABinfo.

3.2.1 MODELLERING PARTIJ MET MEETGEGEVENS

WABinfo moet het mogelijk maken om een partij met meetgegevens te kopiëren. Dit omdat door voortschrijdende inzichten tijdens een project een partij qua inhoud herzien kan worden. Wenselijk is daarbij de historie van de partij met bijbehorende meetgegevens vast te houden. Ook is het wenselijk om de oorspronkelijk mee gekopieerde meetgegevens te verwijderen en nieuwe meetgegevens uit WADI te relateren. Binnen WADI is het niet wenselijk dezelfde meetgegevens meerdere malen op te slaan. Reden hiervoor is redundante opslag en omvang van data.

Oplossing om zowel aan te sluiten op de wensen van WABinfo als voor WADI is om partij en meetgegevens met behulp van een koppeltabel aan elkaar te relateren. Dit maakt dat een partij gevormd kan worden door één of meer meetgegevens en een meetgegeven voor kan komen bij één of meerdere partijen.

Zoals het gegevensmodel van WADI is vormgegeven en rekening houdende met de hierboven genoemde wensen past een partij niet in de gegevensverzameling MONSTEROBJECT van WADI.

3.2.2 RELATIE PARTIJ MET MEETGEGEVENS

WABinfo moet de mogelijkheid hebben om haar projectgegevens te relateren (koppelen) aan een partij en een partij geeft aan welke meetgegevens de bron zijn van deze partij.

Om dit mogelijk te maken zal een verwijzing naar de gekoppelde meetgegevens gelegd worden. De WABinfo gebruiker beschikt over een beheerscherm die dat mogelijk maakt. Deze koppeling komt technisch tot stand door de unieke sleutel van de gegevensverzameling MONSTEROBJECT in combinatie met de unieke sleutel van een partij in een koppeltabel vast te leggen.

3.2.3 AANMAKEN VAN PARTIJ EN MEETGEGEVENS

Partijgegevens worden door de eindgebruiker aangemaakt in WABinfo m.b.v. van een beheerscherm.

Meetgegevens zijn afkomstig van bronsystemen en komen terecht in WADI door middel van het importeren van bestanden met meetgegevens. De WABinfo eindgebruiker zal zelf via een beheerscherm in WABinfo de door hem/haar geïmporteerde meetgegevens relateren aan de juiste partijen.

3.2.4 UPDATE VAN PARTIJ EN MEETGEGEVENS

Partijgegevens kunnen door de WABinfo gebruiker met een beheerscherm aangepast worden.

Meetgegevens die in het kader van WABinfo in WADI worden geïmporteerd worden vanuit WABinfo niet bewerkt. Er wordt vanuit gegaan dat deze meetgegevens correct zijn. Mochten wijzigingen noodzakelijk zijn aan een meetgegeven dan zal dit in de bronsystemen moeten plaats vinden. Concreet betekent dit dat een verzameling meetgegevens wordt aangepast (expert judgement evt.) gevalideerd in het bronsysteem, van daaruit geëxporteerd als een bestand (formaat in SIKB, iBever WADI XML of WABinfo XML) en middels triggering vanuit WABinfo wordt dit bestand in WADI ingelezen.

Bij hernieuwd automatisch inlezen van een verzameling meetgegevens, de unieke coderingen van de meetgegevens van het bestand bestaan reeds in WADI, worden de betreffende meetgegevens in WADI volledig overschreven. Bestaande relaties blijven bestaan.

3.2.5 VERWIJDEREN VAN PARTIJ EN MEETGEGEVENS

WADI is een archief systeem. Dat houdt in dat gegevens definitief bevroren zijn. Mutaties op deze gegevens zijn niet meer wenselijk.

WABInfo zal haar eindgebruikers de mogelijkheid bieden via de presentatielaag meetgegevens te laten importeren/exporteren. De status van geïmporteerde meetgegevens hoeven in het licht van WABInfo niet definitief te zijn, oftewel de noodzaak van archiveren is nog niet daar. De meetgegevens zouden in principe zelfs weer verwijderd kunnen worden uit WADI.

Binnen WABInfo is het mogelijk om aan een project partijen te relateren en aan partijen weer meetgegevens. Voor te stellen is dat een partij wordt verwijderd waaraan ook meetgegevens zijn gerelateerd.

Vraag die dan rijst: mag WABInfo de aan partijen gerelateerde meetgegevens uit WADI verwijderen?

Logisch lijkt van niet, want een partij ontstaat door de resultaten van metingen. Onderzoekresultaten resulteren in een partij. Uit deze volgorde in werken volgt dan direct de vraag: als een MONSTEROBJECT wordt verwijderd uit WADI en die gerelateerd is aan een partij, moet dan de partij worden verwijderd uit WABInfo?

3.2.6 PERFORMANCE EN BESCHIKBAARHEID

De gegevensuitwisseling tussen WABInfo en WADI zal volgens de geschetste architectuur volgens KANS2 (Service Oriented Architecture) met webservices gerealiseerd kunnen worden. Deze manier van werken kan mogelijk twee problemen op leveren. Namelijk :

1. de beschikbaarheid van WABInfo, deze is sterk afhankelijk van de beschikbaarheid van WADI
2. de performance ten aanzien van gegevensuitwisseling via webservices.

Ad 1.

Beschikbaarheid van WADI is met name met de huidige stand van de technologie in technisch opzicht geen probleem, het is volledig op te lossen. Beschikbaarheid in een bepaalde mate garanderen is tegenwoordig een hardwarematig investeringsvraagstuk.

Ad 2.

Het performance-vraagstuk is een heikel punt. Het aantal bevestigingen en hoeveelheid data die opgevraagd worden uit WADI zal gaan oplopen. Het is de bedoeling dat de informatie-uitwisseling tussen WADI en WABInfo real-time is. Zorg is of het gebruik van webservices voldoende snelheid geeft tussen WABInfo en WADI., De reden hiervoor is dat een webservice een extra interface vormt dat indirect van buiten de databaseschil gegevensuitwisseling laat plaatsvinden tussen twee applicaties. Resultaat is dat een webserviceconstructie meer bewerkingen nodig heeft om gegevens van plaats a naar b te krijgen dan wanneer gegevens worden uitgewisseld binnen de schil van de database (interne database transactie middels databaselink of meerdere schema' s binnen één en dezelfde database is aanzienlijk sneller). Daarnaast is de huidige WADI webservices generiek uitgevoerd. Dit bevordert de performance niet.

Een elementaire vraag waarop het antwoord de toekomstige architectuur van de applicatiesystemen van RWS richting geeft, is de vraag of de gegevensuitwisseling tussen RWS systemen met in dit geval tussen WABInfo en WADI, indirect middels webservices zou moeten gaan plaats vinden of direct binnen de schil van de database. Wat zijn eigenlijk de voordelen van een webservice architectuur in dit kader. Een aardig document waarin het principe van een webservice en de toepassing hiervan staat beschreven is te vinden via <http://www.serc.nl/lac/docs/Papers/Artikel%20webservices%20-%20CGEY.PDF>

Een groot voordeel van een webservice is met name het creëren van onafhankelijkheid op allerlei vlakken waardoor minder inspanningen verricht hoeven worden om informatie te kunnen uitwisselen tussen informatie-eigenaren en informatie-vragers. De investeringsstap naar samenwerking cq. integratie (=richten op de core business) wordt sneller gemaakt. De onafhankelijkheid van een webservice is op drie niveaus te onderscheiden:

- *Platform en database onafhankelijkheid;* Door gebruik te maken van een mechanisme als de webservice dat breed gedragen wordt (=standaard) door technologie-leveranciers, wordt het mogelijk om applicatiesystemen die op verschillende platformen functioneren direct met elkaar te laten communiceren. Daarnaast maakt de database ook niet echt meer uit.
- *Leveranciersonafhankelijkheid;* Als gevolg van platform en databaseonafhankelijkheid hoeft men zich in huis niet meer te conformeren aan een technologie van een paar leveranciers, bijvoorbeeld alle applicaties onderbrengen op Oracle databases of servers met alleen Windos als operating system, ontstaat een mate van leveranciersonafhankelijkheid.

- *Applicatieonafhankelijkheid*; een webservice kan gezien worden als een zelfstandige toepassing waarbinnen allerhande bewerkingen geautomatiseerd kunnen worden. Integratie oftewel gegevensuitwisseling tussen applicaties vindt met gebruikmaking van een webservice op een eenduidige, centrale en intelligente wijze plaats.

Ten aanzien van dit laatste punt valt in het kader van performance versus onafhankelijkheid meer te zeggen.

Indien niet voor een onafhankelijke webservice methode tussen WABinfo en WADI wordt gekozen omdat ingestoken wordt op optimale performance, is het aan te raden beide systemen op basis van dezelfde databasetechnologie (Oracle) te laten uitvoeren waarbij beide applicaties middels een database link de gegevens met elkaar gaan uitwisselen. Database links zijn een veilige en makkelijk te implementeren en te onderhouden Oracle technologie.

WADI en WABinfo zijn twee applicatiesystemen die elk een bedrijfsbehoefte binnen RWS vertegenwoordigen. Vraag is wie de eigenaren van beide applicaties worden. Een en dezelfde (beheer)organisatie of elk systeem een eigen (beheer)organisatie? Indien er twee (beheer)organisaties zijn draagt een webservice architectuur bij in eenduidigheid in afspraken (SLA) en onafhankelijkheid ten aanzien van doorvoeren van wijzigingen. Het B2B principe wordt uitgedragen.

Duidelijk is dat kiezen voor onafhankelijkheid met gebruik van webservices bij real-time transacties ten kosten gaat van de transactiesnelheid (performance verlies) . Indien het RIZA voor een Oracle database kiest is ten aanzien van WADI mogelijk om een snellere real-time transactie oplossing te kiezen via database links. Bijkomend voordeel van database links is dat het ontwikkelen van een interface makkelijker is te realiseren. Het RIZA zal zelf moeten beslissen hoe de real-time uitwisseling tussen WABinfo en WADI moet gaan plaats vinden.

Om de mogelijke performance problemen ten aanzien van gebruikte webservices tussen WADI en WABinfo te reduceren, kan bijvoorbeeld gekozen worden voor een soort cache data laag aan de WABinfo kant. In deze cache worden alle resultaten van reeds eerder gestelde zoekvragen een x aantal minuten opgeslagen. Op deze manier wordt voorkomen dat er overbodige zoekvragen worden gesteld aan WADI. Naast de cache oplossing zou ook de huidige set van WADI webservices uitgebreid kunnen worden met webservices welke meer specifiek toegespitst zijn op de wensen van WABinfo. Deze webservices zouden bv. alleen gegevens kunnen retourneren die voor WABinfo interessant zijn. Een nog verdergaande oplossing is om alle door WABinfo gewenste gegevens uit WADI op te halen en lokaal in een voor WABinfo geschikt formaat op te slaan en deze gegevens periodiek te verversen.

Performance is trouwens een ingewikkeld vraagstuk. Er zijn een aantal factoren die hierbij een rol spelen. Twee daarvan zijn de machine capaciteit van zowel WABinfo en WADI database server en applicatieserver en de architecturen van de informatiesystemen . n tier omgeving is bij gelijke omgevingsfactoren minder performant dan n-1 tier. Andere aspecten zijn bandbreedte van het netwerk, fysieke afstand van de database en applicatieservers ten opzichte van elkaar etc.

Tegenwoordig is een groot deel van slechte performance relatief eenvoudig en goedkoop te bedwingen door schaalbare hardware en software configuraties. Een architectuur als het MVC model en de beschrijvingen in KANS 2 is een voorbeeld van een schaalbare software architectuur. Niet alle componenten hoeven op dezelfde server te staan om het systeem naar behoren te laten werken, het kan wel als het moet (mits de OS systemen niet conflicteren).

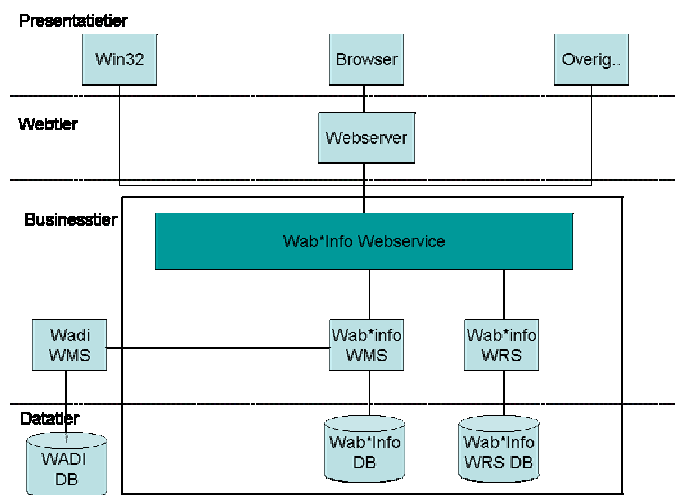
3.3 KOPPELING GEOSERVICES

Met betrekking tot geometrie zijn er twee hoofdoelen te definiëren:

1. Het presenteren van geo-informatie (raadplegen);
2. Het grafisch aanpassen/toevoegen van geo-informatie (muteren).

3.3.1 GEO-RAADPLEGEN

Voor het presenteren van geo-informatie zou de architectuur er als volgt uit kunnen te zien (basis model):



Afbeelding 3 Architectuur voor het raadplegen van geografische gegevens

De architectuur wijkt in wezen niet af van de door de AGI voorgeschreven architectuur. Hieronder wordt stapsgewijs een toelichting gegeven op de lagen van de architectuur.

Presentatielaag

Dit is het gedeelte die de gebruiker op zijn of haar scherm ziet. Dit kan een desktop-applicatie zijn, maar ook een webapplicatie die door middel van de browser getoond wordt. Dit zijn zgn. 'thin clients' waarbij de logica van de applicaties op de businessier is geïmplementeerd. Het realiseren van nieuwe applicaties is hierdoor betrekkelijk eenvoudig omdat de logica al vastgelegd op de businessier.

Webtier

De webtier wordt alleen gebruikt in het geval van een webapplicatie (browser).

Businessier

Deze tier is de kern van de architectuur. Hier wordt de logica grotendeels vastgelegd. De businessier bestaat uit de volgende onderdelen:

- WABinfo Webservice
- WABinfo WRS
- WABinfo WMS

WABInfo webservice

De WABInfo service is een eenduidige interface met de 'buitenwereld' zodat er voor WABInfo functionaliteit slechts één service aangesproken hoeft te worden. De WABInfo service spreekt de geo-services aan zoals vastgelegd conform het opengis consortium. In dit figuur zijn dat de Web registry service (WRS) en de Web Mapping Service (WMS). Buiten beschouwing is gelaten volgens welke technologie de requests worden verstuurd en afgehandeld (SOAP, CGI, etc). De WABInfo applicatie wordt gebruikt voor de directe interactie met eindgebruikers, de webservice wordt echter alleen gebruikt door (andere) applicaties. Het aanspreken van de GeoServices zal plaats vinden via de "ActionServlet webservicemanager" zoals aangegeven in afbeelding 2.

Web Registry Service

Dit is een register waarin mapping services geregistreerd worden. Dit moet gezien worden als een soort gouden gids voor mapping services. Deze wordt door de WABinfo service geraadpleegd om vast te kunnen stellen welke kaarten beschikbaar zijn. Over WRS zijn diverse inzichtelijke artikelen geschreven:

- GeoServices: RWS realiseert centrale InternetGis Infrastructuur. AGI*Geonieuws 2004-1, pag. 26-31
- Rapport Technisch ontwerp GeoServices. Geodan september 2003. Kenmerk ITP03215-TO

Web Mapping Service

De webmapping service produceert het uiteindelijk kaartbeeld in de vorm van een afbeelding (png, jpg, etc). Het betreft dus een visuele representatie van de het kaartbeeld zoals die op dat moment opgevraagd wordt en dus niet de data zelf. Het kaartbeeld bestaat dan wel uit verschillende lagen. Wanneer een laag uitgezet wordt, wordt een nieuw kaartbeeld gegenereerd.

Het is binnen een WMS Service mogelijk meerdere kaarten samen te voegen door met andere WMS Services te 'cascaden'. Hierbij wordt data van verschillende WMS Services samengevoegd tot één kaart(afbeelding). Nadere specificaties zijn te vinden in het document Web Map Service Implementation Specification en het document Web Map Service Interface, beide uitgegeven door het OpenGis Consortium. In Afbeelding 3 wordt 'cascading' uitgevoerd tussen de WABinfo WMS en het WADI WMS, waardoor kaartlagen samengevoegd worden vanuit verschillende systemen.

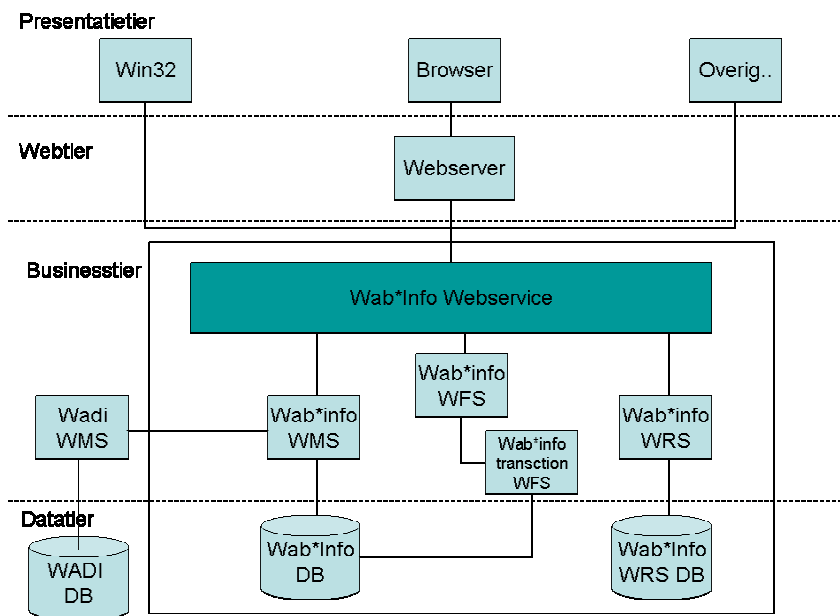
Het advies bij de bovengenoemde services (WRS en WMS) is om de richtlijnen van het OpenGis Consortium te volgen, zodat het mogelijk is dat andere applicaties waarbij ook geo-data gepresenteerd gaat worden gebruik kunnen maken van de WABinfo WMS.

Datatieer

Hiermee worden de verschillende geo-gegevensverzamelingen bedoeld, zoals de WADI, WABinfo (spatial)databases.

3.3.2 GEOMUTEREN

Voor Editten van geometrie komt de architectuur er als volgt uit te zien



Afbeelding 4 architectuur voor het muteren van geografische gegevens

Voor het Editten van geometrie worden de volgende services toegevoegd:

- Web Feature Service (WFS);
- Web Feature Transaction Service.

Web Feature Service (WFS)

In tegenstelling tot de WMS geeft de WFS geen plaatjes door, maar data in de vorm van GML. Deze webservice is readonly en is bedoeld voor het ophalen en doorgeven van data en kan geen mutaties uitvoeren. Zie voor uitgebreide informatie: *Web Feature Service Implementation Specification* uitgegeven door het Open Gis Consortium.

Voorbeeld

Bij het muteren van een perceel is niet alleen een visuele representatie van het perceel nodig, maar ook informatie die aangeeft hoe het perceel is opgebouwd (de polygoon). De Web Feature Service levert dit aan in de vorm van GML.

Transaction WFS

Dit is een extensie op de WFS die ervoor zorgt dat een geometrie gemuteerd kan worden. Deze verzorgt de volgende transactietypen:

- create
- update
- delete

Bij het muteren via de transaction WFS is het mogelijk één of meerdere objecten uit de gegevensverzameling te locken zodat deze niet gemuteerd kunnen door andere gebruikers tijdens het uitvoeren van de transactie.

De transacties worden doorgegeven via GML.

Vervolg Voorbeeld

De client heeft een visuele representatie (via WMS) en de topografische informatie (GML via WFS) van een perceel. De Transaction WFS heeft het perceel gelockt in de gegevensverzameling. De client voert de mutaties uit op het GML. Het GML wordt teruggestuurd naar de transaction WFS, welke de mutatie doorvoert in de gegevensverzameling. Wanneer de transactie compleet is, wordt het object weer vrijgegeven (unlocked).

Binnen RWS (AGI) is nog geen WFS en transactie WFS aanwezig. Indien deze gerealiseerd gaan worden voor WABinfo verdient het sterk de aanbeveling dit conform OpenGis uit te voeren zodat deze services (her)gebruikt kunnen worden voor vervolg projecten.

3.4 DATABASE BEHEER

3.4.1 ORACLE ENTERPRISE MANAGER

Het RIZA heeft aangegeven dat de gegevens van WABinfo worden opgeslagen in een Oracle database. Standaard wordt bij de Oracle Standard Edition database van al de zogeheten Enterprise manager meegeleverd. Dit is een windows georiënteerde toepassing die het mogelijk maakt om visueel een of meerdere Oracle databases te beheren. Sterk punt van deze toepassing is de eenvoud. Namelijk alles om een Oracle database goed te beheren is aanwezig en wordt grafisch en op een logische manier gepresenteerd. SQL kennis is met de Oracle Enterprise manager niet nodig.

3.4.2 PATROL EXPRESS

Voor performance, beschikbaarheid van servers, applicaties en netwerk componenten is een monitoring toepassing aan te bevelen. PATROL Express van BMC Software is een bekende toepassing voor het monitoren van de IT infrastructuur.

Zie voor meer informatie over het product PATROL Express de website van de firma BMC Software

http://www.bmc.com/products/proddocview/0,2832,19052_19429_2064403_9548,00.html

4 Beveiliging

De beveiliging van WABInfo is op te delen in twee elementen. Namelijk horizontale en verticale autorisatie. Verticale autorisatie regelt welke toepassing cq. functionaliteit een gebruiker al dan niet kan benaderen. Horizontale autorisatie heeft betrekking op gegevensbeveiliging en regelt welke gegevens de gebruiker kan opvragen en muteren.

Het security mechanisme voor WABInfo is aanzienlijk uitgebreider dan die WADI momenteel hanteert.

4.1 HORIZONTALAUTE AUTORISATIE

Binnen WABInfo is het wenselijk dat de applicatiebeheerder en projectleiders middels een autorisatiemechanisme gegevens kunnen beveiligen. Zo zal bijvoorbeeld een projectleider willen aangeven welke collega's mutaties mogen verrichten op zijn projectgegevens, en een applicatiebeheerder wil aangeven welke projecten zichtbaar zijn voor een bepaalde instantie. Om deze reden zijn in het logisch model van WABInfo, zie hiervoor het document *Logisch gegevensmodel WABInfo*, logische gegevensverzamelingen als AutorisatieOrganisatie, Rol en Privilege weergegeven die functionaliteit ten aanzien van gegevensbeveiliging ondersteunen.

Binnen WABInfo hebben de applicatiebeheerder en de projectleider de mogelijkheid gegevensbeveiliging met daarvoor bestemde beheerschermen te onderhouden. De technische realisatie van de horizontale autorisatie komt tot stand door een relatie te leggen tussen de gebruiker die zich geïdentificeerd heeft aan WABInfo middels een inlog procedure (al dan niet automatisch) en de rol met bijbehorende privileges (zie het logisch gegevensmodel) die de applicatiebeheerder vooraf aan de gebruiker heeft toegekend. Voor elke bevraging of mutatie die de gebruiker uitvoert wordt de rol die de gebruiker heeft met bijbehorende privileges gevalideerd.

Of een gebruiker al dan niet een gegeven mag raadplegen en/of muteren (=aanmaken, overschrijven of verwijderen) wordt bepaald aan de hand van het autorisatiemechanisme zoals gemodelleerd in het logisch gegevensmodel van WABInfo. Een gebruiker heeft een rol en een rol heeft privileges. Deze privileges geven aan wat voor rechten een gebruiker heeft ten aanzien van o.a. raadplegen en muteren. Dit beveiligingsmechanisme is binnen de gegevenslaag (Oracle database) als ook binnen de applicatielaag (J2EE componenten) te realiseren.

4.1.1 AUTORISATIE IN GEGEVENSLAAG VERSUS APPLICATIELAAG

Horizontale autorisatie is voor WABInfo op twee plaatsen mogelijk, namelijk binnen de gegevenslaag of de applicatielaag. Waar de horizontale autorisatie wordt ondergebracht is een vraagstuk waar de meningen over verschillen.

Horizontale autorisatie binnen de gegevenslaag: als de autorisatie op de gegevenslaag wordt afgehandeld, wordt de beveiliging volgens de technische mogelijkheden die de database biedt uitgevoerd. Alle beveiligingslogica is ondergebracht binnen de database. Voordeel hiervan is dat de beveiliging bij de bron - de tabellen - is aangebracht en daarmee centraal en eenduidig te onderhouden is. Het maakt niet uit van waar af de gegevensbronnen benaderd worden, de beveiliging functioneert altijd. Gegarandeerde veiligheid.

Horizontale autorisatie binnen de applicatielaag: Als de autorisatie wordt ondergebracht binnen de applicatielaag zijn de programmeer en beheer aspecten complexer en daarmee kostbaarder. Voor elk stuk functionaliteit dient namelijk het beveiligingsmechanisme geprogrammeerd te worden. Risico is dat voor stukken functionaliteit het beveiligingsmechanisme wordt vergeten en daarmee een security leak ontstaat.

4.1.2 AFSCHERMEN GEGEVENS AFHANKELIJK VAN ROL

Een technologie voor afscherming van opgevraagde WABInfo gegevens middels zoekvragen, zou gerealiseerd kunnen worden door gebruikmaking van de *Virtual Private Database (VPD)* techniek. Deze techniek is Oracle specifiek en realiseert *rowlevel security*. De werking hiervan is globaal als volgt voor te stellen: tijdens het uitvoeren van een zoekvraag op een tabel in de database wordt voorafgaand aan het uitvoeren van deze zoekvraag een *context* vastgelegd in de database. Context kan bijvoorbeeld de accountnaam van de gebruiker zijn. Voor het bepalen welke tabelregels een gebruiker mag raadplegen uit de betreffende tabel wordt vooraf automatisch de zogeheten *policy* vastgesteld door het doen van aanroep naar een procedure in de database met de context als parameter. Deze procedure stelt volgens de logica die voor WABInfo is uitgewerkt in het logisch model, het zogenoemde *predicaat* vast. Een predicaat is een SQL voorwaarde die wordt geplakt aan de uit te voeren zoekvraag. Kort gezegd wordt door het VPD voorafgaand aan het uitvoeren van een zoekvraag een extra zoekvoorwaarde gegenereerd die wordt toegevoegd aan de zoekvraag.

Zie voor meer informatie ten aanzien van de voordelen <http://www.securityfocus.com/infocus/1743>

4.1.3 GEGEVENSBEVEILIGING WADI IN RELATIE TOT WABINFO

De gegevensbeveiliging in WADI is erg eenvoudig en niet bepaald veilig te noemen. De raadpleegbeveiliging is geregeld met zogeheten *certificaten*. Een certificaat is een bestandje dat wordt uitgedeeld aan een gebruiker die dit bestandje vervolgens op zijn/haar lokale PC o.i.d. plaatst. Het werken met certificaten is een vertrouwenskwestie. Doorsturen van een certificaatbestandje naar collega's is snel gedaan.

Op dit moment kent WADI twee gebruikersgroepen die alleen gegevens kunnen raadplegen:

- Publieke gebruikers: De gebruikers hoeven zich niet bekend te maken bij WADI en hebben toegang tot alle publieke meetgegevens.
- Geautoriseerde gebruikers: Deze gebruikers beschikken een over een certificaat waardoor de gebruiker toegang krijgt tot alle meetgegevens. Zowel publiek als niet publieke gegevens.

De presentatielaag van WABinfo gaat gegevens uit WADI tonen met gebruikmaking van een WADI webservice. *Vraag is nog of de gegevens waartoe de WABinfo gebruikers toegang hebben vallen onder publieke meetgegevens of niet publieke meetgegevens.* Indien de gegevens vallen onder niet publieke meetgegevens zullen de WABinfo gebruikers een certificaat nodig hebben.

WADI gebruikers kunnen niet muteren met uitzondering van gegevensbeheerders. Deze gegevensbeheerders zijn daartoe geregistreerd in een Lightweight Directory Access Protocol server LDAP waardoor deze toegang krijgen tot WADI mutatie functionaliteit.

WABinfo gebruikers dienen mutaties te verrichten in WADI voor bijvoorbeeld het importeren van iBever bestanden. WABinfo gebruikers zullen dus evenals WADI gegevensbeheerders geregistreerd moeten worden in de LDAP die gebruikt wordt voor WADI.

4.2 VERTICALE AUTORISATIE

In de situatie voor WABinfo en het geschetste J2EE model is het gebruik van een Lightweight Directory Access Protocol server (LDAP) een geschikte methode om verticale autorisatie te realiseren. Een LDAP is een methodiek die ook recht doet aan de denkwijze van KANS. Het betreft hier een open standaard.

Zie voor meer informatie omtrent de werking en mogelijkheden van LDAP <http://www.gracion.com/server/whatldap.html>

4.3 ADVIES

Daar de technische oplossing voor de gegevenslaag naar alle waarschijnlijkheid een Oracle database zal zijn is het om beheer(s)matige en beveiligingsredenen aan te bevelen horizontale autorisatie binnen de database te regelen. Gegevensbeveiliging centraal aan de bron. Kans op security leaks minimaal en implementatiewerkzaamheden rond beveiliging zijn eenduidig en eenvoudig.

Ten aanzien van de verticale autorisatie is het gebruik van LDAP een prima keus. WABinfo gebruikers worden centraal geregistreerd, krijgen een rol waarbij de toegang tot WABinfo en WADI functionaliteit is geregeld.

Het gebruik van certificaten voor WABinfo is af te raden gezien beheersmatige en beveiligingsredenen.

Hoe ziet het autorisatiemechanisme voor WABinfo er nu organisatorisch uit? Heel simpel en eigenlijk bijna identiek aan de werkwijze bij "oude" Client Server systemen.

Een gebruiker wenst toegang tot WABinfo, doet hiertoe twee aanvragen aan:

1. *Systeembeheerder*; deze registreert het account van de gebruiker in LDAP en relateert daarbinnen de nodige WABinfo rol en WADI rol aan dit account.
Stuurt eventueel de gebruiker een certificaat toe m.b.t. WADI
2. *WABinfo applicatiebeheerder*; deze registreert hetzelfde account als in LDAP geregistreerd van de gebruiker en relateert de nodige applicatierol t.b.v. horizontale autorisatie binnen WABinfo.

5 CONCLUSIE

De architectuur zoals in dit document beschreven, voldoet aan de punten zoals weergegeven in paragraaf 1.2. Zolang de applicatie wordt opgebouwd met duidelijke scheidingen tussen de client en de applicatielaag, is het goed mogelijk om de applicatie uit te breiden met toekomstige functionaliteiten eventueel gebaseerd op webservices. De controller van de client zal de communicatie moeten regelen naar een webservice en naar de WABinfo gegevens.

In de WABinfo database worden hoofdzakelijk gegevens rondom projecten vastgelegd. Samen met de externe database van WADI en met GeoServices vormt dit de WABinfo applicatie. Door de gecreëerde scheiding tussen de gegevens en de uiteindelijke client is het ook mogelijk een andere client te bouwen gebaseerd op het WABinfo model waarin alleen projectgegevens getoond worden of waarin projectgegevens getoond worden in combinatie met andere gegevens naast de meetgegevens uit WADI. Kortom de geschetste architectuur biedt veel mogelijkheden en is dermate flexibel dat het goed aansluit op de voorwaarden en op de visie zoals deze beschreven is binnen KANS 2.

De beschreven architectuur waarin GeoServices via WABinfo webservices benadert worden, wijkt in wezen niet af van de door AGI voorgeschreven architectuur. Hierin mist alleen nog de functionaliteit om geometrie te muteren. Een extensie van de open GIS standaard WFS (Web Feature Service) biedt mogelijkheden hiervoor. Een implementatie hiervan lijkt mogelijk binnen de beschreven architectuur. Er moet hiervoor een WFS-component ingezet worden met een extensie t.b.v. transacties in de gegevensbron waar de geografische gegevens opgeslagen zijn.

De architectuur zoals geschetst volgens het AGI in het document KANS2 impliceert dat gegevensuitwisseling tussen applicatiesystemen binnen RWS plaats gaat vinden met webservices. Voordeel hiervan is onder meer dat onafhankelijkheid wordt gecreëerd tussen afzonderlijke applicatiesystemen ten aanzien van platvormen, databases en invulling van de ontwikkeltechnologie. Vraagtekens worden nog gezet bij de haalbare performance die deze architectuur biedt bij real-time gegevensuitwisseling tussen WABinfo en WADI. Indien het RIZA evenals RIKZ kiest voor een Oracle database is het raadzaam dat het RIZA en RIKZ in overleg met het AGI bepalen hoe de gegevensuitwisseling zal moeten gaan plaats vinden. Met behulp van webservices of direct bij de bron in de database middels database links.

Zoals de lezer kan opmaken uit dit document staan er nog een aantal vragen open. Het is aan het RIZA om in overleg met o.a. RIKZ en het AGI antwoorden op deze vragen te formuleren. Na het maken van een afgewogen keuze van o.a. database, wijze van real-time transacties tussen WADI en WABinfo en de wijze van autorisatie kan onder meer met gebruikmaking van de beschreven kennis in dit document een concrete invulling aan de bouw van het WABinfo gegeven worden.

6 Literatuur

1. Functioneel ontwerp Geoservices, Geodan IT b.v. in opdracht van Meetkundige Dienst, 29 september 2003, Concept versie 0.2
2. Technisch ontwerp Geoservices, Geodan IT b.v. in opdracht van Meetkundige Dienst, 26 september 2003, Concept versie 0.2
3. Offerte WADI
4. Offerte WABinfo
5. KANS 2: Vooronderzoek UCN, SRN en GAIN, 16 april 2004, Concept versie 0.3