

# Kavelruilmonitor Bergeijk

## Inzicht in het ruilproces

Versie

1.0

Auteur

Joost Venema





Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
1 van 40

## Kavelruilmonitor Bergeijk

Inzicht in het ruilproces

### Opdrachtgevers

Hogeschool Utrecht

Kadaster

### Status

Definitief

<b>Datum:</b>	1 juni 2011
<b>Student:</b>	Joost Venema
<b>Studentnummer:</b>	1555053
<b>Opleiding:</b>	Geodesie / Geo-Informatica
<b>Begeleiding Kadaster:</b>	Ron Nijhuis
<b>Begeleiding Hogeschool Utrecht:</b>	Frans de Vroege (1 <sup>e</sup> begeleider) Bernard Raterman (2 <sup>e</sup> begeleider)



Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
2 van 40

## Voorwoord

In het kader van mijn studie Geodesie / Geo-Informatica heb ik een softwareproduct ontwikkeld dat ondersteuning biedt aan het kavelruilproces in de gemeente Bergeijk. Dit rapport gaat in op de aanleiding, achtergrond en werking van dat product: De Kavelruilmonitor.

Mijn persoonlijke doel van deze eindopdracht was om een tastbaar product af te leveren dat een goede kans zou maken om ingezet te worden binnen de organisatie. Je hoort regelmatig van het 'in de kast belanden' van afstudeeronderzoeken. Dat beeld spreekt mij niet zo aan. Dat dit een arbeidsintensieve opgave zou worden, was mij vooraf al duidelijk. Zo ben ik erg lang bezig geweest met het uitlijnen van de invoervelden van het mutatieformulier. Het zijn vaak de kleine dingen die veel tijd in beslag nemen...

De laatste maand van het onderzoek stond geheel in het teken van het realiseren van dit verslag. Welnu, na intensief met het schrijven aan de gang te zijn geweest, ligt er nu een compleet eindrapport.

Tot slot wil ik iedereen binnen het Kadaster bedanken die mij heeft geholpen in dit onderzoek: Ron Nijhuis, Frank van den Bogaert, Willem Lepelaars en Willem Witlox<sup>1</sup>. Daarnaast mijn dank aan het Kadaster zelf, voor het mogen volgen van deze opleiding en de zeer goede studiefaciliteiten die mij zijn geboden.

Meppel, Juni 2011

---

<sup>1</sup> Citaat van Willem na het lezen van mijn conceptscriptie:

*"blad 12, 3.2 Kavelruil in Bergeijk.....Bergeijk ligt in de provincie Noord Brabant,...en.....nie in Limburg, nondeju !"*



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

3 van 40

## Samenvatting

In de gemeente Bergeijk is een kavelruilproject gestart met als doel het realiseren van de Ecologische Hoofdstructuur (EHS) en landbouwstructuurverbetering (LSV). Dienst Landelijk Gebied (DLG) had hierin vanaf de start van het project de coördinerende rol, maar heeft dit in de loop van het traject door capaciteitsproblemen overgedragen aan het Kadaster.

De afdeling Ruimte & Advies (R&A) neemt binnen het Kadaster deze taak op zich. R&A had behoefte aan ondersteunend kaartmateriaal en grafieken om zichzelf en de samenwerkende partijen inzicht te verschaffen in het ruilproces. De afdeling GIS-maatwerk & Advies (GMA) heeft als pilotproject handmatig een aantal kaarten vervaardigd op basis van de administratie van R&A. Hoewel de kaarten er goed uitzagen en bruikbaar waren, was het proces omslachtig. Van daaruit kwam de vraag naar een geautomatiseerd, slim systeem: de Kavelruilmonitor.

De doelstelling van dit onderzoek kan als volgt worden geformuleerd:

*Het realiseren van een oplossing die voorziet in het koppelen van administratieve, financiële en geografische gegevens met betrekking tot een kavelruilproces, zodanig dat deze op overzichtelijke wijze beschikbaar zijn voor medewerkers van het Kadaster in een kartografische- en statistische vorm.*

Als eerste stap zijn de wensen en eisen geïnterviewd en is van daaruit gestart met de ontwikkeling van het daadwerkelijke product. De ontwikkeling is via een iteratieve softwareontwikkelmethode ontwikkeld, genaamd: Agile.

Het product bestaat uit een aaneenschakeling van verscheidene softwarepakketten, waarvan het merendeel open source software is. De applicatie is vanaf elke Kadaster-werkplek te benaderen in de webbrowser. Het uiteindelijke product heeft de volgende functionaliteiten:

- Mutatiebeheer
- Interactieve kaart en statistiek
- Automatische synchronisatie met de kadastrale registraties

De registratie van de financiële gegevens is wegens tijdgebrek niet gerealiseerd. De verwachting is dat dit met een kort aanvullend onderzoek goed te realiseren is.

De Kavelruilmonitor is verder volledig functioneel, echter draait deze nog op een ontwikkelomgeving. Een aanbeveling is daarom om verder onderzoek te doen naar de implementatie binnen de Kadaster ICT-infrastructuur. Daarnaast is een onderzoek naar de beveiliging aan te bevelen.



Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
4 van 40

## Inhoudsopgave

<b>1</b>	<b>Inleiding .....</b>	<b>6</b>
1.1	Achtergrond.....	6
1.2	Doel- en vraagstelling .....	6
1.3	Afbakening .....	6
1.4	Leeswijzer .....	7
<b>2</b>	<b>Het Kadaster .....</b>	<b>8</b>
2.1	Organisatie.....	8
2.2	Afdeling GIS Maatwerk & Advies (GMA).....	9
2.3	Afdeling Ruimte & Advies (R&A) .....	10
<b>3</b>	<b>Kavelruil.....</b>	<b>12</b>
3.1	Kavelruil en het Kadaster .....	12
3.2	Kavelruil in Bergeijk .....	12
<b>4</b>	<b>Product: databronnen en software.....</b>	<b>14</b>
4.1	Ontwikkelmethode .....	14
4.2	Databronnen .....	14
4.3	GMA Software-referentiearchitectuur.....	16
4.4	Software .....	17
<b>5</b>	<b>Product: werking van de Kavelruilmonitor .....</b>	<b>22</b>
5.1	Inventarisatie wensen en eisen .....	22
5.2	Deelsystemen .....	23
5.3	Architectuur Kavelruilmonitor.....	23
5.4	Software in detail .....	25
5.4.1	Kern .....	25
5.4.2	Synchronisatie.....	27
5.4.3	Mutatiebeheer .....	30
5.4.4	Presentatie.....	31
5.5	Historie: de temporele component .....	33
5.6	Ontwikkelsoftware .....	34
<b>6</b>	<b>Lopende ontwikkelingen en de toekomst .....</b>	<b>35</b>
<b>7</b>	<b>Conclusies en aanbevelingen .....</b>	<b>36</b>
7.1	Conclusies .....	36
7.2	Aanbevelingen.....	36
<b>8</b>	<b>Bronnen .....</b>	<b>38</b>



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

5 van 40

**Begrippenlijst .....39**

**Bijlagenoverzicht.....40**



## 1 Inleiding

### 1.1 Achtergrond

Het Kadaster is regelmatig betrokken bij kavelruilen. Sinds 2009 heeft het Kadaster bij het kavelruilproject Bergeijk de coördinerende rol. De afdeling Ruimte & Advies (R&A) neemt binnen het Kadaster deze rol op zich. Binnen R&A bestaat de behoefte aan een kaart/systeem dat de actuele stand van zaken kan visualiseren. Elementen die hierin betrokken moeten worden zijn: administratie, percelen en de financiële situatie.

In de vestiging in Eindhoven is al enige ervaring opgedaan. In een soort van pilot-fase zijn er handmatig al enkele kaarten vervaardigd voor het doelgebied de gemeente Bergeijk. Hoewel deze kaarten er goed uit zien en als nuttig worden ervaren, is het vervaardigen ervan een omslachtig proces. Daarom is de vraag ontstaan naar de ontwikkeling van een slim systeem: de Kavelruilmonitor

### 1.2 Doel- en vraagstelling

Het doel van dit onderzoek is het realiseren van een oplossing die voorziet in het koppelen van administratieve, financiële en geografische gegevens met betrekking tot een kavelruilproces, zodanig dat deze op overzichtelijke wijze beschikbaar zijn voor medewerkers van het Kadaster in een kartografische- en statistische vorm.

De centrale vraag in dit onderzoek luidt:

*Is het mogelijk om een systeem te realiseren waarin een kavelruiladministratie wordt gekoppeld aan de geografische gegevens van het Kadaster, zodanig dat deze goed toegankelijk en overzichtelijk is?*

Naast deze vraagstelling zijn er drie deelvragen geformuleerd:

- *Kan het systeem gebruik maken van een temporele component?*
- *Is er een oplossing mogelijk om mutaties te signaleren?*
- *Is het systeem inzetbaar voor andere doelgebieden dan de gemeente Bergeijk?*

### 1.3 Afbakening

Het onderzoek richt zich op het realiseren van een softwareproduct. De afbakening zal daarom de eerste productieversie van het product omvatten (1.0). Het onderzoek zal zich beperken tot het kavelruilproject Bergeijk. Het product zal ontwikkeld worden en gaan draaien in een ontwikkelomgeving. De implementatie van het product binnen de ICT-infrastructuur wordt buiten beschouwing gelaten.



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

7 van 40

## 1.4 Leeswijzer

Vooropgesteld dient te worden dat er van de lezer verwacht wordt dat hij of zij enige basiskennis bezit van geo-informatica. In dit verslag wordt bijvoorbeeld geen definitie gegeven van een GIS. In het volgende hoofdstuk wordt ingegaan op het Kadaster. In hoofdstuk 3 wordt uitgelegd wat kavelruil is en wat daarin de taak van het Kadaster is. Hoofdstuk 4 en 5 richten zich op het product. In hoofdstuk 4 worden de verschillende softwarepakketten toegelicht en worden de databronnen binnen het Kadaster beschreven. Hoofdstuk 5 gaat in op de technische werking van de Kavelruilmonitor. Lopende ontwikkelingen en een doorkijk naar de toekomst komen in hoofdstuk 6 aan bod. Ten slotte staan de conclusies en aanbevelingen in hoofdstuk 7.

In dit rapport komt een grote verscheidenheid aan termen en afkortingen voor. Een overzicht hiervan is terug te vinden aan het eind van dit rapport, in hoofdstuk 9.





## 2 Het Kadaster

Het Kadaster is een zelfstandig bestuursorgaan (ZBO) dat haar primaire bestaansrecht ontleent aan de uitvoering van een aantal wettelijke taken. De kerntaken van het Kadaster worden uitgevoerd in het belang van de rechtszekerheid en de geo-informatievoorziening die genoemd worden in de Kadasterwet en de Landinrichtingswetgeving, dan wel taken die daaruit voortvloeien. Naast deze wetten vormt de Organisatiewet Kadaster het kader voor de juridische en bestuurlijke status van het Kadaster.

De Kadasterwet, die ten dele is gebaseerd op bepalingen uit het Burgerlijk Wetboek, vormt de wettelijke grondslag voor de activiteiten op de terreinen openbare registers, Basisregistratie Kadaster, Basisregistratie Topografie en geometrie. Deze wet verzekert een betrouwbare input en verwerking, zodat volledige en actuele vastgoed informatie, informatie over schepen en luchtvaartuigen en geo-informatie permanent beschikbaar is.

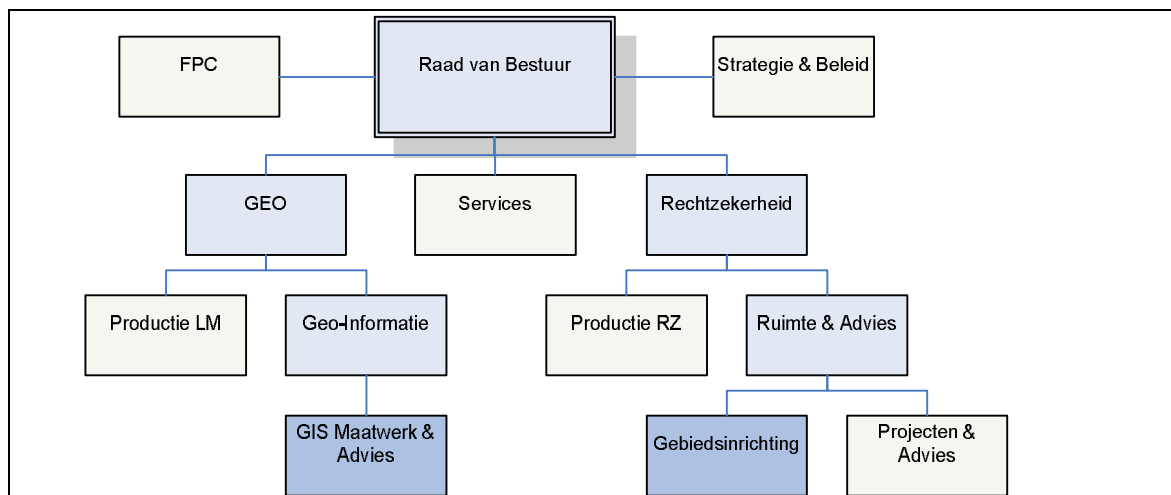
De Kadasterwet kent aan het Kadaster de volgende taken toe:

- Het houden van de openbare registers en het bijhouden van de Basisregistratie Kadaster en de kadastrale kaart
- De registraties van schepen en luchtvaartuigen
- Het in stand houden van een net van coördinaatpunten (Rijksdriehoeksmeting)
- Het registreren en het kartografisch weergeven van geografische gegevens
- Het verstrekken van informatie op basis van deze registraties

De Landinrichtingswet geeft de wettelijke basis voor de ondersteuning bij de landinrichting. Deze ondersteuning heeft als belangrijkste accenten het dienen van de rechtszekerheid van de rechthebbenden en de toedelingstaak. Naast de Landinrichtingswet zijn er de Herinrichtingwet Oost-Groningen en de Gronings Drentse Veenkoloniën, de Reconstructiewet Midden Delfland, evenals de Reconstructiewet concentratiegebieden. De ondersteuning die het Kadaster verleent op basis van deze wetten wijkt niet wezenlijk af van de ondersteuning op basis van de Landinrichtingswet. (Kadaster, 2011)

### 2.1 Organisatie

In figuur 1 is de organisatie van het Kadaster schematisch weergegeven. Voor de overzichtelijkheid zijn minder relevante afdelingen uit het schema weggelaten.



**Figuur 1: Organigram Kadaster organisatie**

De uitvoering van de primaire processen van het Kadaster is ondergebracht in twee directies, Rechtszekerheid en GEO. De directie Rechtszekerheid is verantwoordelijk voor het juridisch-registratieve deel van de Basisregistratie Kadaster en voor de landinrichting. De directie GEO is verantwoordelijk voor het landmeetkundige deel van de Basisregistratie Kadaster, voor de Basisregistratie Topografie en voor de Rijksdriehoeksmeting. Ondersteunende taken zijn ondergebracht in de directie Services. De dagelijkse leiding is in handen van de Raad van Bestuur, die daarin wordt ondersteund door een directie Strategie & Beleid en een directie Financiën, Planning en Control (FPC). (Kadaster, 2011)

Omdat de afdelingen GIS Maatwerk & Advies en Ruimte & Advies centraal staan in dit onderzoek, zal hieronder op deze afdelingen worden ingezoomd.

## 2.2 Afdeling GIS Maatwerk & Advies (GMA)

GMA is het GIS competence center voor het Kadaster. GMA maakt maatwerkproducten op het gebied van vastgoed-, hypotheek- en geo-informatie en moderne GIS technologieën. Dit kunnen speciale topografische kaarten zijn, maar bijvoorbeeld ook een eigenarenkaart op basis van kadastrale percelen al dan niet gekoppeld aan gegevens van de klant. De afdeling is in vier taakvelden actief:

- **Nationaal civiel:** In dit taakveld is GMA een samenwerkingspartner voor externe burgers en bedrijven en interne collega's. De ontsluiting van informatie geschiedt onder andere door een internetportaal.
- **Nationaal militair:** Van het Nederlandse grondgebied worden geo-informatie en kaarten geproduceerd in opdracht van de Dienst Geografie Koninklijke Landmacht, dit hangt samen met de internationale NATO verplichting voor het genereren van nationaal militaire kaartseries. Dit is in feite het werkterrein van de afdeling Kartografie. Eventueel afwijkende klantvragen worden door GMA opgepakt.
- **Internationaal civiel:** Ook op internationaal niveau is GMA actief. Er wordt vooral met Europese partners samengewerkt voor het ontwikkelen van een internationale ruimtelijke informatie



infrastructuur. Daarvoor is het belangrijk dat de data die daarin is opgenomen wordt geharmoniseerd. Zo is GMA producent van Europese maatwerk producten onder de paraplu van Eurogeographics. Tevens biedt GMA ondersteuning aan Kadaster International.

- **Internationaal militair:** Door de participatie in internationale operaties is het voor de Nederlandse Defensie van belang mee te groeien in de ontwikkelingen van moderne geo-informatiesystemen in de wereld. In opdracht van de Dienst Koninklijke Luchtmacht werkt GMA mee aan internationale productie programma's. (Kadaster, 2011)

### 2.3 Afdeling Ruimte & Advies (R&A)

R&A is binnen het Kadaster een afdeling die zich voornamelijk richt op de samenwerking met externe partijen, zoals de overheid en het bedrijfsleven. R&A kan de klant adviseren bij de duurzame inrichting van stedelijk en landelijk gebied en bij 'grote' infrastructurele vraagstukken. De afdeling is opgedeeld in een vijftal teams, waarvan er vier vallen onder 'gebiedsinrichting' en één, landelijk opererend team onder 'Projecten & Advies'.

#### *Gebiedsinrichting:*

De op het "planmatig ruilen" gebaseerde gebiedsinrichtingswerkzaamheden (incl. aanvullende diensten, zoals haalbaarheidsonderzoeken, begeleiding vrijwillige kavelruil, etc.) zijn in de sector gebiedsinrichting gepositioneerd en worden in districten uitgevoerd.

#### *Projecten & Advies:*

Projecten & Advies houdt zich bezig met transitieprocessen van een oude naar een nieuwe rechtstoestand, waarbij het ruilmechanisme niet aan de orde is, maar waarbij door de inbreng van de Kadastrale expertise de rechtszekerheid van het maatschappelijk verkeer in onroerend goed eveneens wordt bevorderd.

R&A levert dan ten behoeve van in- en externe projecten inbreng in de vorm van procesadvies, -begeleiding en -uitvoering.

De afdeling richt zich op drie aandachtsgebieden:

- **Inrichtingsactiviteiten in het landelijk gebied:** De betrokkenheid van het Kadaster bij de landinrichting vloeit voort uit de bepaling in diverse wetten dat het Kadaster bijstand verleent aan Gedeputeerde Staten c.q. de commissie die het project uitvoert. Naast de wettelijke taken voert Ruimte en Advies aanvullende diensten uit, zoals het uitvoeren van haalbaarheidsonderzoeken, het uitvoeren van diverse analyses, het adviseren over grondposities en gerichte grondverwerving en het begeleiden van planmatige en vrijwillige kavelruil. In 2007 is de Wet Inrichting Landelijk Gebied (WILG) van kracht geworden. Via deze wet worden de provincies zelf in staat gesteld om de inrichting van het landelijk gebied te realiseren.
- **Verbetering van de eigen kadastrale registratie:** Diverse ontwikkelingen verplichten het Kadaster ertoe om de kadastrale registraties op een verantwoord niveau te brengen. De belangrijkste ontwikkelingen zijn: het mogelijk toekennen van de derdenbescherming aan de kadastrale registratie en de toekenning van het predicaat basisregistratie aan de kadastrale registratie.



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

11 van 40

- **Activiteiten in het kader van ‘nieuwe business voor externe klanten’:** Verschillende activiteiten worden er verricht in de domeinen:

- Infrastructuur
- Natuur en landschap
- Wonen
- Werken
- Water en klimaat
- Openbare orde en veiligheid
- Duurzaamheid

Enkele voorbeelden zijn:

- Bijdragen aan de realisering van locaties voor waterberging, hogere en bredere waterkeringen, duurzame drinkwatervoorziening en van maatregelen tegen verdroging van landbouw- en natuurgebieden
- Informatiefundament bij herstructurering van bestaande bedrijventerreinen
- Objectief inzicht in de verschillende mogelijkheden voor uitbreiding van bovengrondse en ondergrondse infrastructuur (Kadaster, 2011)



### 3 Kavelruil

Kavelruil is een proces waarbij grondeigenaren kavels met elkaar ruilen. Het doel van dit proces is om een meerwaarde te creëren voor alle betrokken partijen. Er zijn verschillende aanleidingen mogelijk om een kavelruilproces te starten:

- **Landbouwstructuurverbetering:** Dit kan van toepassing zijn door het verbeteren van:
  - De bereikbaarheid van percelen
  - Vergroting van bedrijfskavels
  - De waterbeheersing
  - De ligging van percelen van dezelfde eigenaar
- **Ecologische Hoofdstructuur (EHS):** De EHS is een aaneengesloten netwerk van bestaand en nog te ontwikkelen natuurgebied. Het doel van de EHS is het versterken van de biodiversiteit in Nederland. Aangezien de gronden die zijn aangemerkt als EHS niet allemaal in eigendom van de overheden zijn, wordt er vaak een kavelruilproject opgezet om deze gronden naar de overheden te ruilen.

De opzet van een kavelruil kan vrijwillig of wettelijk zijn. Bij vrijwillige kavelruil kan dit:

- Via een losse kavelruil (vaak kleinschalig)
- Op ondersteunende basis
- Op planmatige basis

Van wettelijke kavelruil is sprake als dit wordt uitgevoerd in het kader van de WILG.

#### 3.1 Kavelruil en het Kadaster

Doorgaans heeft een partij als DLG het projectleiderschap in een kavelruilproject. Het Kadaster wordt vaak ingehuurd om te ondersteunen in het project. Deze ondersteuning omvat onder andere de volgende werkzaamheden:

- Voorlichting aan streek
- Wenszitting (inclusief versturen kaartmateriaal)
- Opstellen ruilplannen
- Uitwerken en bedenken van ruilvoorstellen
- Splitsen, vervaardigen van nieuwe perceelnummers

#### 3.2 Kavelruil in Bergeijk

Bergeijk is een gemeente onderin de provincie Noord-Brabant. Het heeft een totale oppervlakte van 103km<sup>2</sup> waarvan een groot deel bestaat uit buitengebied. De versnippering in dit buitengebied is groot. Door een instrument als kavelruil in te zetten kan het herstructureren van dit landschap voordelen hebben voor niet alleen de boeren, denk aan besparing op logistiek, maar ook voor de inwoners van Bergeijk, door het creëren van ruimte voor recreatie, wonen en natuur.

Door de omvang van het project wordt kavelruil Bergeijk op een vrijwillige, planmatige basis uitgevoerd. Normaliter speelt DLG dan hierin een coördinerende rol. Echter door een capaciteitsprobleem bij DLG is overeengekomen dat het Kadaster deze rol op zich neemt.

In de coördinerende rol worden er frequent gesprekken en overleggen gevoerd. R&A mist hierbij ondersteunend materiaal in de zin van een kaart en grafieken waarin de voortgang inzichtelijk kan worden gemaakt.



**Figuur 2: Ligging Bergeijk**



## 4 Product: databronnen en software

In dit hoofdstuk wordt ingegaan op de gebruikte databronnen en software.

### 4.1 Ontwikkelmethode

Een veelgebruikte softwareontwikkelmethode is de watervalmethode. Deze methode is strikt van aard met veel voorschriften en documentatie. Het leent zich daarom niet zo goed voor projecten waarvan de doelstellingen niet tot in de fijnste details zijn uitgewerkt. In het geval van wijzigingen moet de hele waterval opnieuw worden doorlopen.

Een tegenhanger van de watervalmethode is de iteratieve ontwikkelmethode, bijvoorbeeld Agile. Bij Agile wordt een project in kleine stukken (timeboxes) opgedeeld. De timeboxes worden één voor één ontwikkeld en tussendoor wordt er steeds gekeken of het project bijgestuurd moet worden. Veelvuldig communiceren is hierbij van belang. De methode leent zich goed voor kleinschalige projecten waarbij de projectgroep uit niet veel meer dan 20 personen bestaat.

Voor de Kavelruilmonitor is er voor gekozen om de software te ontwikkelen via de Agile-methode.

### 4.2 Databronnen

Voor het functioneren is de Kavelruilmonitor afhankelijk van verschillende bronnen. Deze bronnen worden hieronder toegelicht:

#### **Databronnen binnen het Kadaster:**

##### **AKR**

AKR staat voor Automatisering Kadastrale Registraties. Het is samen met de hypothecaire registers ondergebracht in het AKR/HYP-systeem. De AKR bevat alle administratieve gegevens behorende bij een perceel. De ICT-dienst AKR/HYP is geïmplementeerd op een IBM/IDMS-platform. De exploitatie van dit platform is uitbesteed aan het bedrijf T-Systems. Rechtstreekse benadering van de deze database is niet mogelijk. Vanwege de gedateerdheid van het systeem is het ook niet wenselijk.

##### **LKI / CIDB**

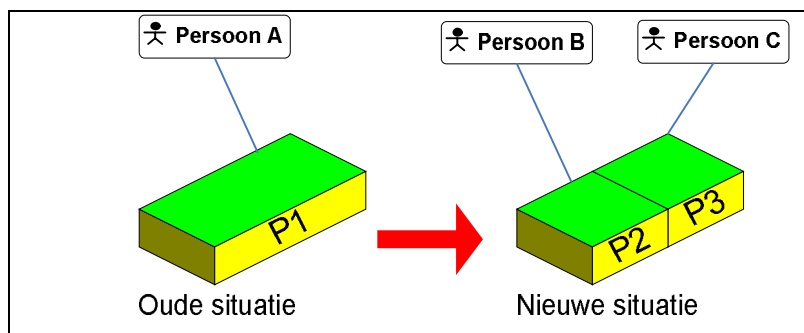
Het Landmeetkundig Kartografisch Informatiesysteem (LKI) is een applicatie voor het inwinnen, verwerken en presenteren van kadastrale en topografische informatie. Het LKI bevat de geometrieën (grenzen) behorende bij een perceel. De gegevens zijn opgeslagen in de CIDB, wat staat voor Cartografische Informatie Database. De CIDB is niet rechtevreeks te benaderen.

### Relatie AKR en LKI

Regelmatig wordt aangenomen dat een perceel in de AKR eenvoudig te koppelen is aan de bijbehorende geometrie in het LKI en vice versa. Dat dit met name door wijzigingen van recente aard niet altijd mogelijk is, zal worden uitgelegd aan de hand van een vereenvoudigd voorbeeld:

Stel dat persoon A eigenaar is van een perceel met het nummer P1. Persoon A verkoopt dit perceel aan twee nieuwe eigenaren, Persoon B en C, welke beiden een helft van het perceel in bezit krijgen. Met andere woorden: het perceel wordt gesplitst. Zodra alle partijen bij de notaris zijn geweest voor het laten passeren van de akte, wordt deze akte ingeschreven bij het Kadaster. In de AKR staat nu dat persoon B eigenaar is van perceel P2 en persoon C eigenaar van perceel P3. In het LKI zijn echter de percelen P2 en P3 nog niet bekend, omdat deze eerst moeten worden ingemeten door de landmeter. Zolang dit niet is gebeurd spreken we van 'deelpercelen'. Pas als de landmeter heeft gemeten en de gegevens heeft bijgewerkt in het LKI kan het perceel gekoppeld worden.

In de praktijk zit er tussen de bijwerking van de AKR en het LKI enkele maanden.



**Figuur 3: Perceelsplitsing**

### MO

MO staat voor Massale Output en is een levering van de AKR aan de klanten van het Kadaster. Kortweg gezegd is het een momentopname van de AKR die wordt weggeschreven naar een Oracle database. De massale output wordt één keer per maand gedraaid.

### ISPS

InformatieService Primaire Processen. Op moment van schrijven is dit systeem nog in ontwikkeling. Het doel van dit systeem is om alle registraties onder te brengen in één database. Tevens wordt deze database dagelijks bijgewerkt, zodat men kan beschikken over actuele informatie. Vooralsnog is alleen de CIDB hierin ondergebracht en werkt de dagelijkse bijwerking niet naar behoren. Wel wordt de ISPS maandelijks opnieuw opgebouwd.



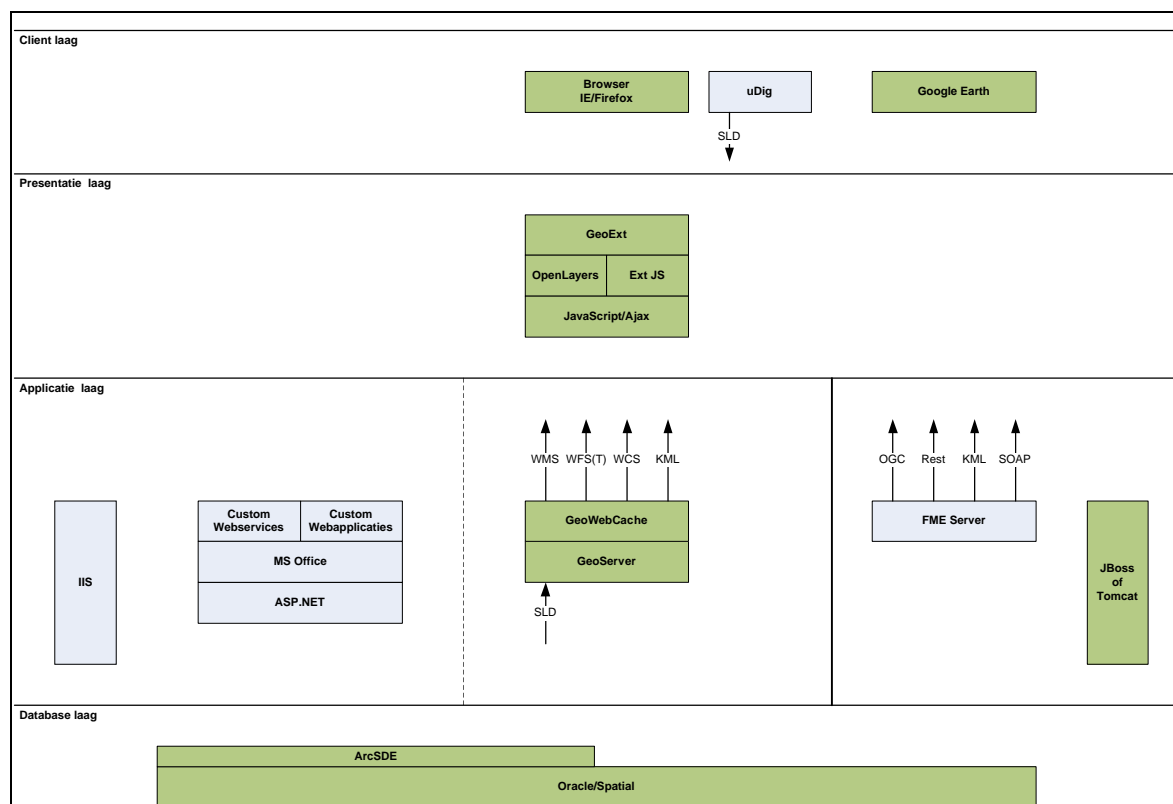
## Externe bronnen:

### EHS

De EHS is beschikbaar als een bestand in vector-formaat (ESRI shape). Het bestand is samengesteld door DLG en bevat polygonen die de begrenzing van de EHS aangeven.

### 4.3 GMA Software-referentiearchitectuur

GMA beschikt over een zogeheten referentiearchitectuur. Deze architectuur bestaat uit een schema dat de software weergeeft waarmee op de afdeling gewerkt wordt. De bedoeling is om binnen de kaders van deze architectuur te werken. Het doel hiervan is dat wildgroei van allerlei verschillende software voorkomen wordt en zo het onderhoud en beheer binnen de perken blijft.



Figuur 4: GMA Software-referentiearchitectuur

#### 4.4 Software

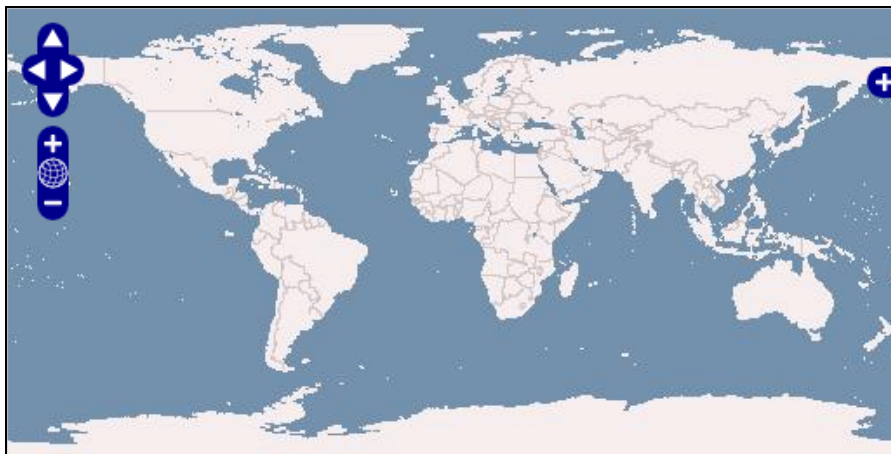
De Kavelruilmonitor maakt gebruik van verschillende onderdelen. Om een beeld te krijgen wat deze software kan en doet, is hieronder een overzicht weergegeven. De beschrijving is kort en bondig gehouden, voor diepgaande details wordt verwezen naar de handleidingen die te vinden zijn in het bronnenoverzicht.

##### Browser IE/Firefox:

De browser of webbrowser is een applicatie voor het weergeven van webpagina's.

##### Openlayers:

Openlayers is een Javascript library die het mogelijk maakt om kaartdata weer te geven in de browser. Met behulp van een API kunnen hiermee applicaties gebouwd worden die veel weg hebben van bijvoorbeeld Google Maps of Microsoft Virtual Earth.



Figuur 5: Visualisatie Openlayers

```
);

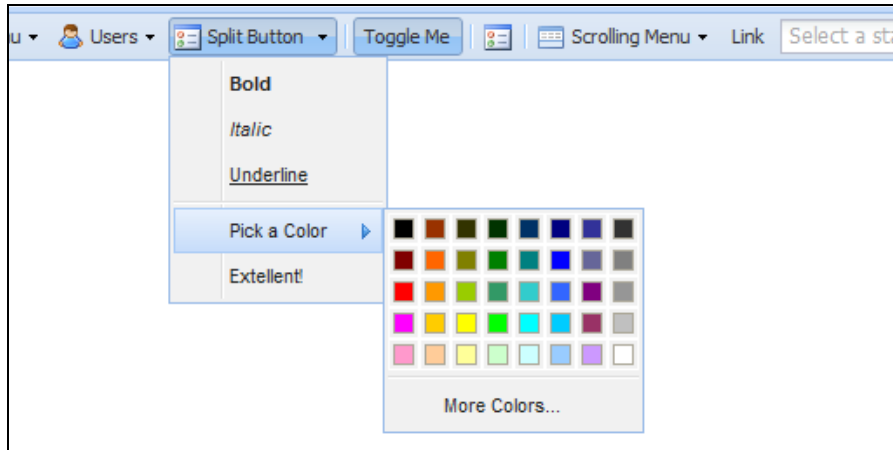
var dm_wms = new OpenLayers.Layer.WMS(
    "Canadian Data",
    "http://www2.dmsolutions.ca/cgi-bin/mswms_gmap",
    {
        layers: "bathymetry,land_fn,park,drain_fn,drainage," +
            "prov_bound,fedlimit,rail,road,popplace",
        transparent: "true",
        format: "image/png"
    },
    {isBaseLayer: false, visibility: false}
);

map.addLayers([ol_wms, jpl_wms, dm_wms]);
map.addControl(new OpenLayers.Control.LayerSwitcher());
map.zoomToMaxExtent();
}
```

Figuur 6: Codefragment Openlayers

### Ext JS:

Ext JS is een Javascript framework voor het ontwikkelen van webapplicaties. Men kan hiermee op een vrij eenvoudige en vlotte manier elementen creëren zoals: vensters, menustructuren, formulieren en grafieken.



Figuur 7: Interface gemaakt met Ext JS

```
items: [{
    text: '<b>Bold</b>', handler: onItemClick
}, {
    text: '<i>Italic</i>', handler: onItemClick
}, {
    text: '<u>Underline</u>', handler: onItemClick
}, '-', {
    text: 'Pick a Color',
    handler: onItemClick,
    menu: {
        showSeparator: false,
        items: [
            Ext.create('Ext.ColorPalette', {
                listeners: {
                    select: function(cp, color){
                        Ext.example.msg('Color Selected', 'You chose {0}');
                    }
                }
            })
        ]
    }
}]
```

Figuur 8: Codefragment Ext JS behorende bij voorgaande figuur

### Python:

Python is een objectgeoriënteerde programmeertaal. De taal lijkt veel op programmeertalen zoals Perl en Java. De taal is door zijn heldere syntax vrij eenvoudig onder de knie te krijgen. Door ondersteuning van zogeheten modules kan extra functionaliteit worden toegevoegd.

```
if len(sys.argv)==1:
    print 'Usage: celsius temp1 temp2 ...'
    sys.exit(0)

# Loop over the arguments
for i in sys.argv[1:]:
    try:
        fahrenheit=float(string.atoi(i))
    except string.atoi_error:
        print repr(i), "not a numeric value"
    else:
        celsius=(fahrenheit-32)*5.0/9.0
```

Figuur 9: Codefragment Python

#### CherryPy:

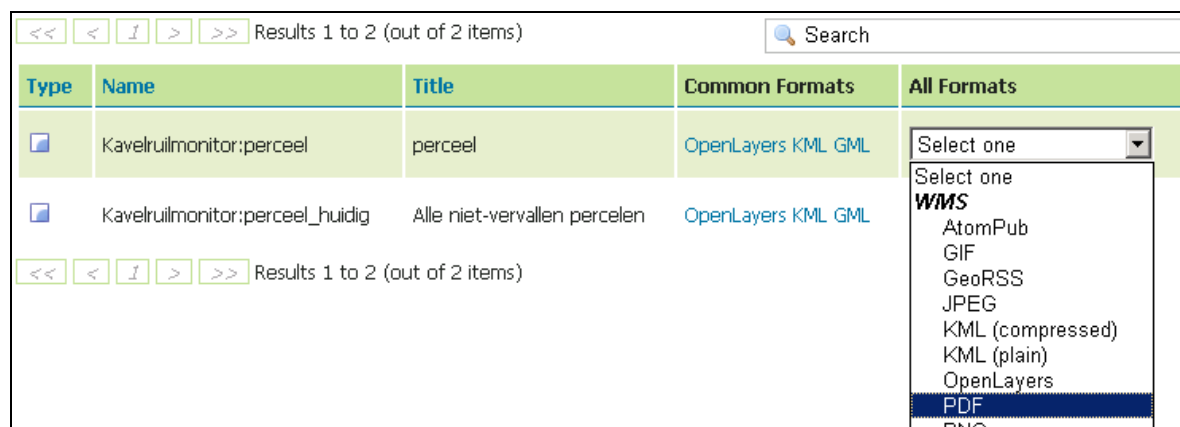
CherryPy is een module voor Python gericht op het ontwikkelen van webapplicaties. Het beschikt over een ingebouwde webserver, maar kan ook draaien achter andere webservers zoals Apache of IIS.

#### Psycopg2:

De module Psycopg2 dient als adapter voor PostgreSQL databases.

#### Geoserver:

Geoserver is een applicatie voor het publiceren en bewerken van ruimtelijke data. Het kan een aantal veelgebruikte bestandsformaten en databases benaderen. Deze data kan vervolgens opgemaakt worden en worden weergegeven als bijvoorbeeld WMS, WFS, KML en PDF.



The screenshot shows the Geoserver administrative interface. At the top, there are navigation buttons and a search bar. Below, a table lists two layers:

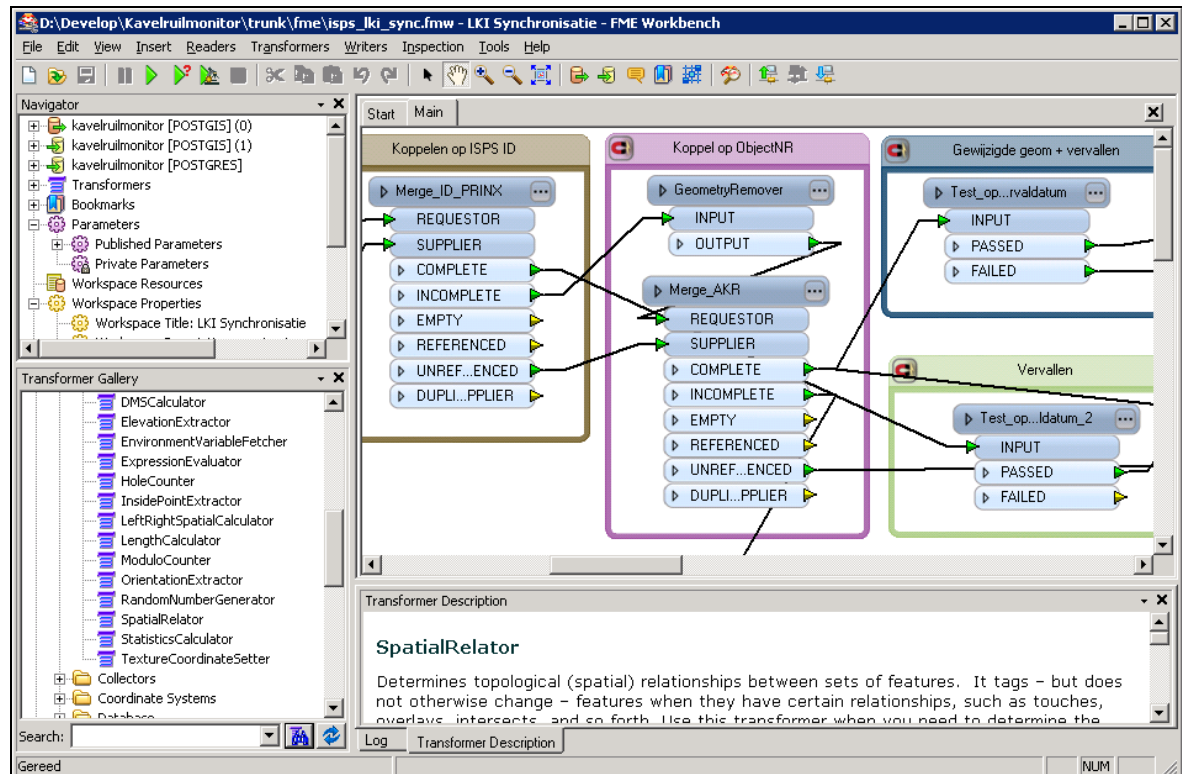
Type	Name	Title	Common Formats	All Formats
	Kavelruilmonitor:perceel	perceel	OpenLayers KML GML	Select one WMS AtomPub GIF GeoRSS JPEG KML (compressed) KML (plain) OpenLayers PDF PNG
	Kavelruilmonitor:perceel_huidig	Alle niet-vervallen percelen	OpenLayers KML GML	

Below the table, there are more navigation buttons and another instance of the 'Results 1 to 2 (out of 2 items)' text.

Figuur 10: Administratie-interface Geoserver

### FME Desktop:

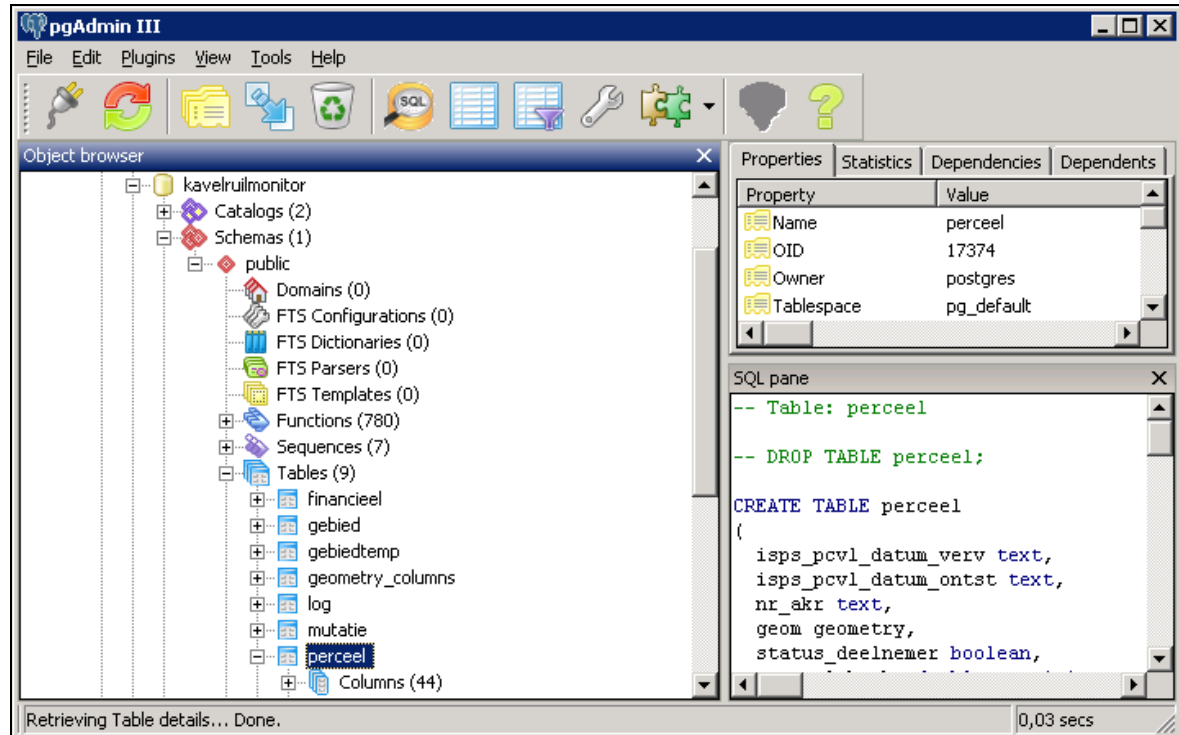
FME staat voor Feature Manipulation Engine. De applicatie valt onder de noemer ETL (Extract, Transform and Load) en kan bijna elk denkbaar ruimtelijk of niet-ruimtelijk bestand lezen en schrijven. Het beschikt over een grote verzameling 'transformers' waarmee de data gemanipuleerd kan worden.



Figuur 11: Gebruikersinterface FME

### PostgreSQL:

PostgreSQL is een objectrelatieel database management systeem (ORDBMS). Het systeem beschikt over een groot scala aan geavanceerde functies en is te vergelijken met Oracle.



**Figuur 12: Administratie-interface voor PostgreSQL (pgAdmin III)**

#### PostGIS:

PostGIS is een add-on voor PostgreSQL waardoor functionaliteit wordt toegevoegd om te werken met geografische objecten.

#### Oracle Spatial:

Oracle Spatial is vergelijkbaar met de combinatie PostgreSQL+PostGIS.



## 5 Product: werking van de Kavelruilmonitor

### 5.1 Inventarisatie wensen en eisen

Omdat er voor een Agile methode is gekozen, zijn de wensen en eisen tot stand gekomen door interactief overleg tussen R&A en GMA. Zou er voor een meer conservatieve ontwikkelmethode als de watervalmethode zijn gekozen, dan zou de voorkeur uitgaan naar een programma van eisen (PvE).

In de basis moet het volgende mogelijk zijn met het systeem:

- Centraal benaderbaar vanaf een standaard Kadaster werkplek
- Gegevens met betrekking tot het kavelruilproces kunnen invoeren (muteren)
- Het tonen van een kaartbeeld
- Het tonen van statistiek
- Het bijhouden van historie
- Mutatiesignalering

Gegevens die aan een perceel gekoppeld of ingevoerd moeten kunnen worden zijn:

- De kadastrale aanduiding van het perceel (objectnummer)
- De eigenaar
- Het oppervlak wat binnen het EHS polygoon valt
- Het ruilnummer
- Indicator of het perceel wel of niet mee doet in de LSV
- Indicator of het perceel getipt is aan BBL of aangekocht door BBL
- Indicator of eigenaar deelnemer is in het project
- Gegevens met betrekking tot historisch, verkennend en aanvullend onderzoek
- Voortgang van de deelnemer in het proces als de stappen: Nog geen gesprek, in gesprek, voorovereenkomst, definitieve voorovereenkomst, wil niet meewerken
- Eventuele opmerkingen

Door het gebruik van bovenstaande gegevens moeten in het kaartbeeld de percelen worden gevisualiseerd op basis van de voortgang in het project. Ook moet zichtbaar zijn welke percelen meedoen in de LSV. De overige percelen moeten een neutrale kleur krijgen. Met een muisklik op het perceel moet het mogelijk zijn om informatie over het perceel weer te geven in (bijvoorbeeld) een pop-up venster.

De statistiek moet bestaan uit een drietal grafieken:

- Taartdiagram van de totale voortgang uitgedrukt in oppervlakte EHS
- Taartdiagram van de LSV ten opzichte van het gewenste doel (1900 Ha) uitgedrukt in de werkelijke oppervlakte

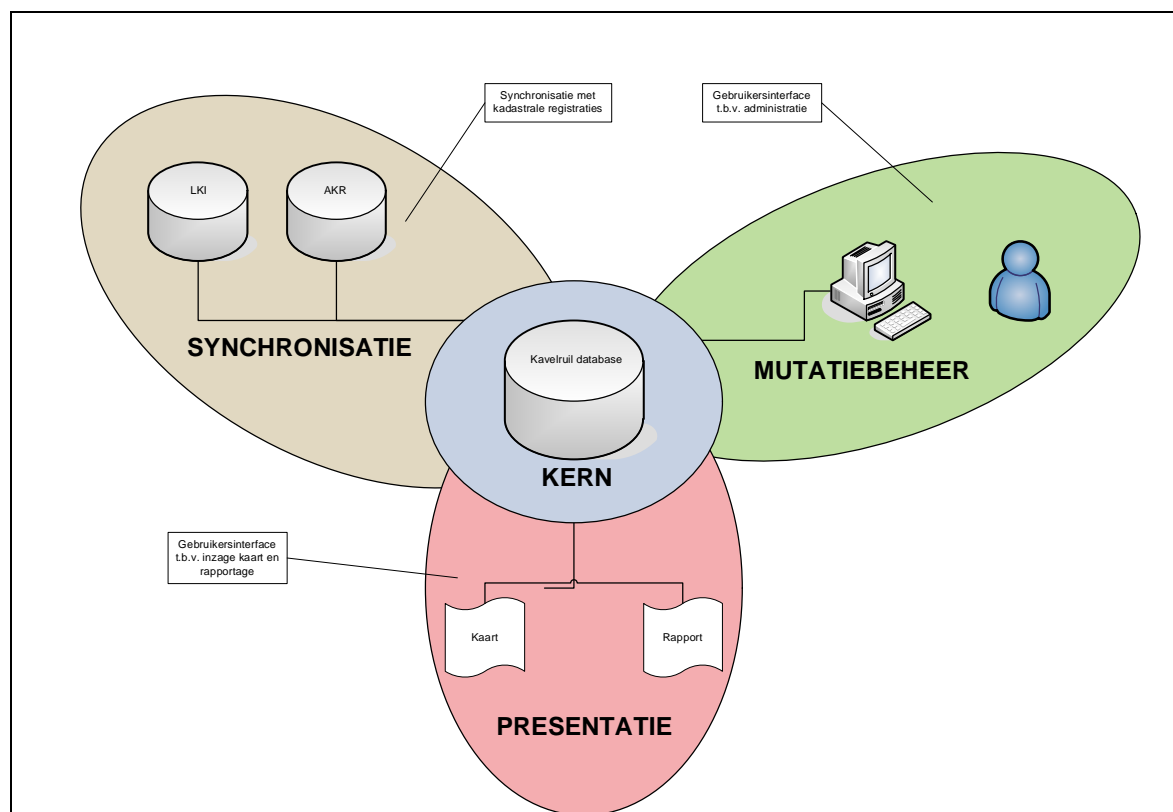
- Lijndiagram met daarin de hoogte van het nog beschikbare budget (financiële situatie)

Om periodieke vergelijkingen van de voortgang van het project te doen is het nodig dat er een vorm van historie wordt bijgehouden.

Tijdens de looptijd van het project bestaat er de mogelijkheid dat percelen van eigenaar veranderen of gesplitst dan wel samengevoegd worden. Het systeem moet daarom in een vorm van mutatiesignalering voorzien, zodat deze percelen gedetecteerd kunnen worden.

## 5.2 Deelsystemen

De applicatie is opgezet rondom vier deelsystemen: Kern, synchronisatie, mutatiebeheer en presentatie. Op deze manier kan er iteratief ontwikkeld worden door één deelsysteem per iteratie te ontwikkelen.



Figuur 13: Deelsystemen Kavelruilmonitor

## 5.3 Architectuur Kavelruilmonitor

De architectuur van de Kavelruilmonitor is niet volledig conform de referentiearchitectuur van GMA (hoofdstuk 4.3) ontwikkeld. Dit heeft te maken met het feit dat er de behoefte bestond om te onderzoeken welke mogelijkheden open source software kan bieden voor dergelijke applicaties. In overleg met de





bedrijfsbegeleider is daarom besloten om de Kavelruilmonitor door middel van open source software te ontwikkelen.

### Open source software

Open source software (OSS) is software waarvan de broncode en de rechten daarop zijn vrijgegeven. Dit in tegenstelling tot commerciële software waarvan de broncode veelal geheim is. Iedereen is vrij om de software te gebruiken, aan te passen of zelfs te verkopen. Wel is men verplicht te allen tijde de broncode te publiceren. Om dit te waarborgen wordt de software vaak uitgebracht onder de General Public License (GPL).

Er zitten zowel voor- als nadelen aan OSS. Hieronder staan de meest voorkomende:

Voordelen:

- Gratis beschikbaar
- Levendige community
- Aan te passen naar eigen wensen

Nadelen:

- Ondersteuning. Het is niet altijd duidelijk waar men terecht kan als de software gebreken vertoont
- Geen garantie

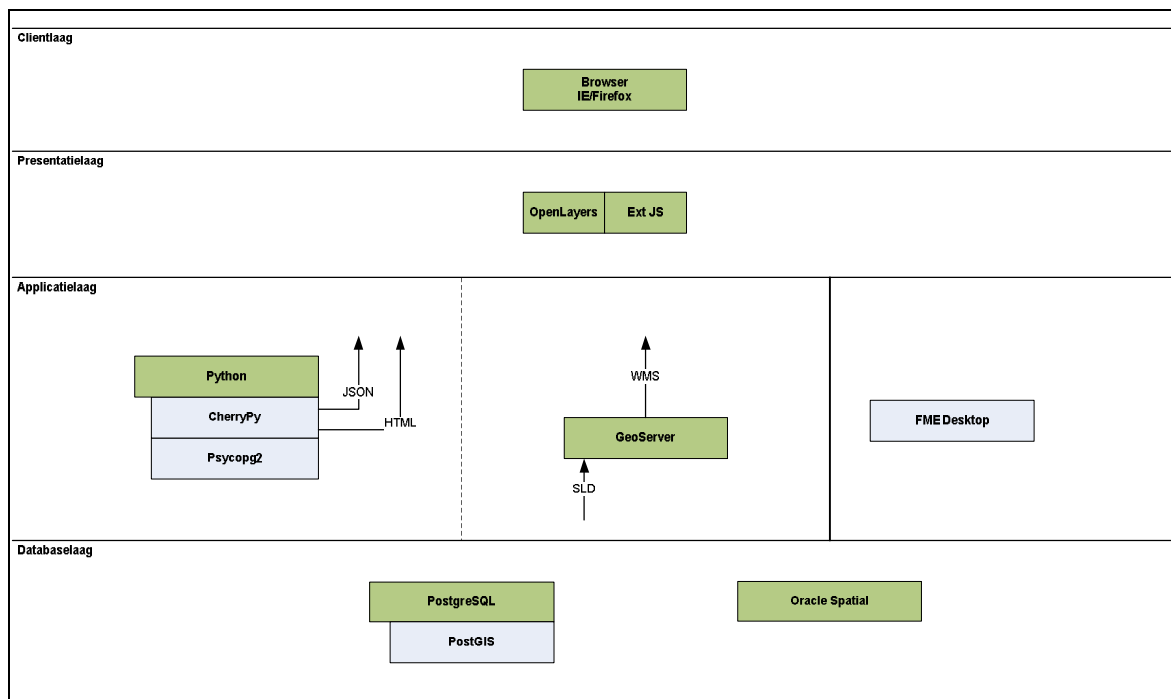
In figuur 14 is de architectuur van de Kavelruilmonitor weergegeven. Zowel FME als Oracle Spatial zijn commerciële pakketten. FME is een dusdanig specialistisch pakket, dat hier geen OSS alternatief voor is. Oracle Spatial is in het schema opgenomen, omdat de kadastrale basisbestanden hierin beschikbaar zijn en deze worden aangesproken om te synchroniseren met de Kavelruildatabase.

### 3-tierarchitectuur

De architectuur bestaat uit meerdere lagen en wordt een '3-tierarchitectuur' genoemd. De indeling van deze 3-tierarchitectuur is als volgt:

- Tier 1: Databaselaag
- Tier 2: Applicatielaag
- Tier 3: Presentatie- en clientlaag

Elke tier kan als onafhankelijk beschouwd worden en daarom kunnen de tiers op verschillende machines, platformen en locaties draaien. Ook kan elke tier vervangen of aangepast worden zonder dat dit gevolgen heeft voor de overige tiers.



**Figuur 14: Architectuur Kavelruilmonitor**

## 5.4 Software in detail

### 5.4.1 Kern

Onder de kern vallen alle onderdelen die het hart van het systeem vormen en aan de 'server-kant'. Er wordt onderscheid gemaakt in twee onderdelen:

- De Kavelruildatabase (databaselaag)
- Een API (applicatielaag)

#### Kavelruildatabase

Voor de opslag van data is gekozen voor een database. Vanwege praktische overwegingen is de database niet verder genormaliseerd dan de eerste normaalvorm. Hierdoor blijft het aantal tabellen beperkt en zijn deze eenvoudig te koppelen aan overeenkomstige velden in de webapplicatie. Tevens komt dit de performance ten goede.

De database bestaat uit vijf tabellen: perceel, gebied, financieel, mutaties en log. Een uitgebreide beschrijving van de attributen is weergegeven in het gegevensmodel (Bijlage I).



De tabel *gebied* is bestemd voor ondersteunende gebieden die worden gebruikt om selecties te maken, te rekenen of om af te beelden als overlay. Voor Bergeijk zijn er twee gebieden in deze tabel opgenomen: Het EHS polygoon en het doelgebied. Het doelgebied is een polygoon dat als selectiegebied fungeert en is gemaakt door R&A.

De tabel *perceel* bevat alle percelen binnen het doelgebied. Deze tabel is de belangrijkste binnen het systeem. Het bevat naast de geometrie alle administratieve gegevens die van belang zijn voor het project.

*Financieel* is een tabel met daarin de financiële situatie van het project. Hierin kunnen de totale uitgaven van het project worden geregistreerd.

De tabellen *mutaties* en *log* kunnen gezien worden als de administratie van het systeem zelf. Hierin kan worden teruggevonden wanneer een bepaald script heeft gedraaid of welke percelen zijn gemuteerd zonder tussenkomst van de gebruiker.

Als databasesoftware is gekozen voor PostgreSQL aangevuld met PostGIS. Deze software is open source, stabiel, heeft omvangrijke functionaliteit en kan goed overweg met ruimtelijke data. Het heeft veel weg van het commerciële pakket Oracle Spatial.

### Application Programming Interface (API)

Een API is een verzameling definities waarmee verschillende soorten software of onderdelen met elkaar kunnen communiceren. Veelal wordt het ingezet om een programma met een hoog abstractieniveau taken te laten uitvoeren door een programma met een lagere abstractie.

In de Kavelruilmonitor wordt een API ingezet om vanuit de webpagina pythonscripts op de server aan te roepen. Het aanroepen kan door middel van de URL in de browser, of via Javascript. Als respons stuurt het script een object in JSON. JSON staat voor JavaScript Object Notation en is een uitwisselingsformaat voor data. Het is een alternatief voor bijvoorbeeld XML, met als voordeel dat het rechtstreeks in Javascript kan worden ingelezen.

```
{
  "data":
    { "nr_akr": "BEZ00G 00015G0000", "naam_gerechtigde": "JANSEN, A.B.C." },
  "success": true
}
```

Figuur 15: Object in JSON



In onderstaande tabel staan de verschillende manieren van aanroepen weergegeven:

Ingang	Verplichte attributen (query)	Respons
zoekperceel/xy	x, y	Administratie behorende bij perceel
zoekperceel/objectnummer	nr_akr	idem
bewerkperceel	id + de te bewerken attributen	succes / geen succes
statistiek/lsv	-	Statistiek op basis van LSV
statistiek/status	-	Statistiek op basis van voortgang in project

Het aanroepen van de API gaat via dit format: `http://hostname/[ingang]/query?[query]`

Stel dat men de administratie van een perceel wil opvragen met de coördinaten: X = 146108 en Y = 366153.

De URL wordt dan: `http://hostname/zoekperceel/xy/query?146108 366153`

#### 5.4.2 Synchronisatie

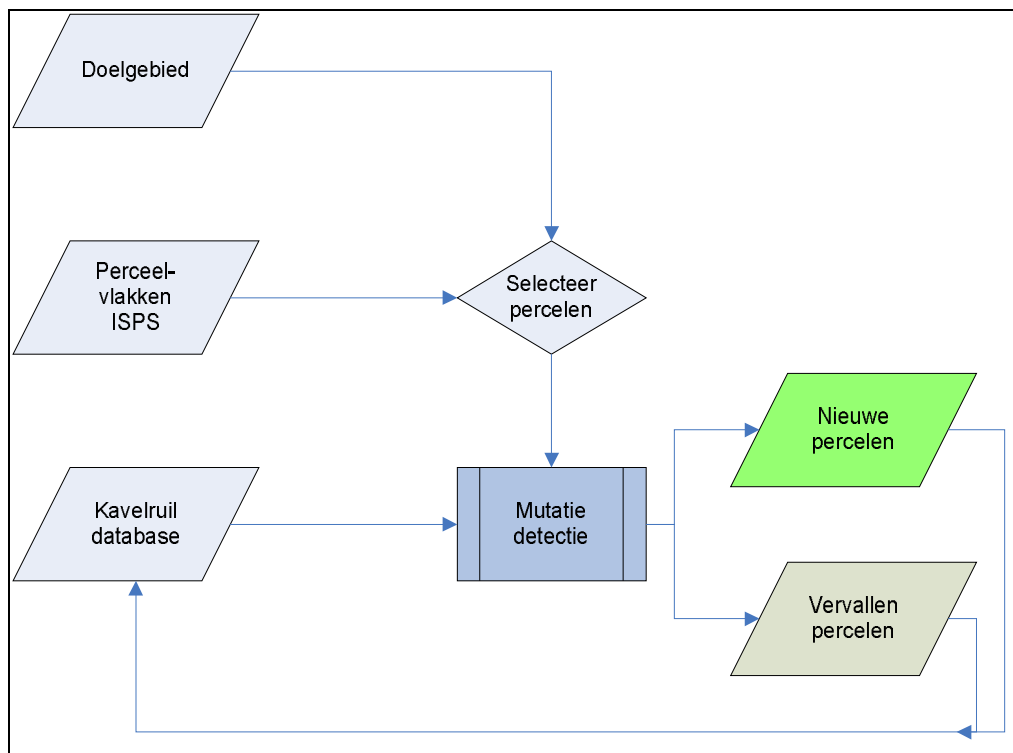
Het onderdeel synchronisatie behelst het actueel houden van de Kavelruildatabase ten opzichte van de kadastrale registraties. Dit is nodig, omdat er tijdens een kavelruilproject percelen gesplitst of samengevoegd worden, of wisselen van eigenaar. Daarnaast dient voor nieuw ontstane percelen de oppervlakte binnen de EHS te worden bepaald. Om dit allemaal te automatiseren zijn er drie 'workbenches' gemaakt met FME.

- Synchroniseren LKI
- Synchroniseren AKR
- Berekenen oppervlakte EHS

##### Synchroniseren LKI

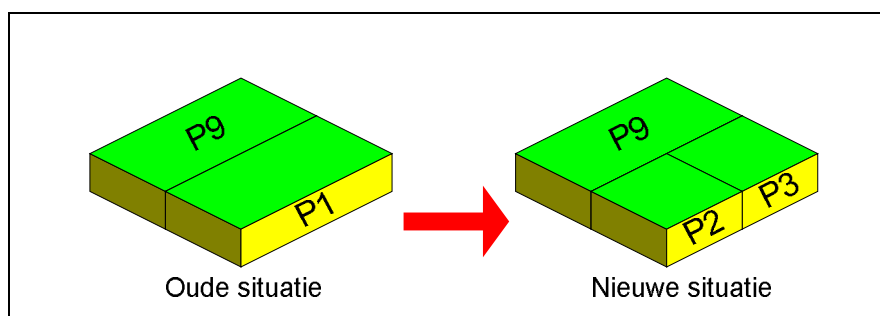
Deze workbench werkt de geometrieën van de Kavelruildatabase bij aan de hand van het LKI percelen in ISPS. De ISPS kent een tabel met daarin perceelvlakken. Deze vlakken zijn automatisch gegenereerd uit de perceelsgrenzen. Elk perceel kent een begintijd en in het geval van een vervallen perceel ook een eindtijd. Door het gebruik van deze attributen is het dus mogelijk om de historie te behouden. Voor de Kavelruildatabase is voor eenzelfde methodiek gekozen.

De vereenvoudigde werking van de workbench is te zien in figuur 16. Een gedetailleerder overzicht is terug te vinden in bijlage III.



**Figuur 16: Bijwerking geometrieën**

Eén bijzonderheid in de mutatiedetectie is een mechanisme om het verschil tussen een reëel vervallen perceel en een perceel dat door technische redenen is vervallen te detecteren. Dit fenomeen zal aan de hand van onderstaand voorbeeld verduidelijkt worden:



**Figuur 17: Mutaties in ISPS**

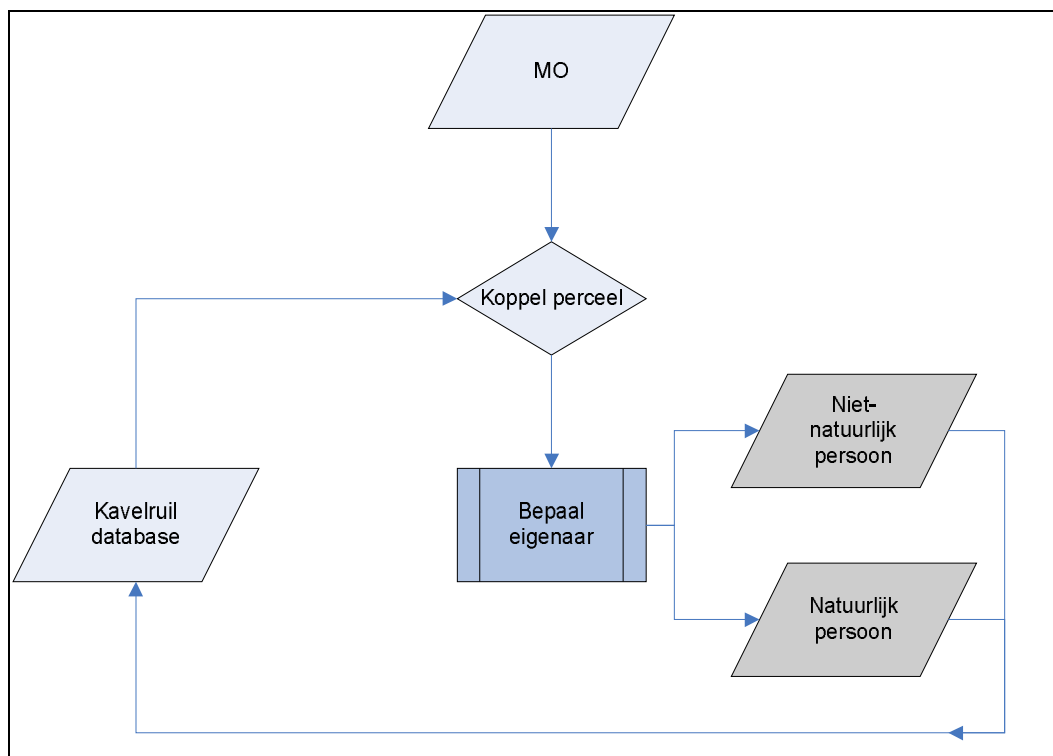
Stel dat perceel P1 gesplitst wordt in de percelen P2 en P3. Het perceel P1 krijgt een eindtijd toegewezen, omdat het niet langer bestaat. Echter perceel P9 krijgt ook een eindtijd, omdat de geometrie is gewijzigd. De wijziging is minimaal: het betreft een extra vertex op het punt dat P2, P3 en P9 met elkaar delen. De wijziging

in P9 heeft geen invloed op de oppervlakte of attributen en daarom wordt in de Kavelruildatabase wél het overeenkomstige perceel bijgewerkt, maar zónder dit als historie vast te leggen.

De synchronisatie met het LKI wordt via een scheduler elke nacht gedraaid.

### Synchroniseren AKR

Daar de AKR nog niet in de ISPS is opgenomen, worden deze gegevens gesynchroniseerd met de MO. De MO wordt maandelijks geüpdatet en daarom draait deze workbench één keer in de maand. Op basis van het objectnummer wordt de Kavelruildatabase gekoppeld aan de MO.



**Figuur 18: Bijwerking eigenaargegevens**

Uit de AKR worden de eigenaargegevens ontleend. Het gaat hierbij om het subjectnummer en de belastingplichtige. Het subjectnummer is een uniek nummer dat gekoppeld is aan een persoon of instantie. Omdat er regelmatig sprake is van meerdere eigenaren per perceel wordt de belastingplichtige eigenaar geselecteerd. De belastingplichtige wordt aan de hand van onderstaand schema bepaald:

- A. één rechthebbende: deze is belastingplichtig
- B. man en vrouw zijn rechthebbenden: de man
- C. partners (gelijk geslacht): de oudste
- D. verschillende rechten: degene die het genot van de zaak heeft
- E. gelijke aandelen: degene die op het perceelsadres woont

- F. *gelijke aandelen anders dan E.: de oudste*  
G. *verschillende rechthebbenden: degene met het grootste aandeel*

Ten slotte wordt er onderscheid gemaakt tussen een natuurlijk persoon (een persoon van vlees en bloed) en een niet natuurlijk persoon (bijvoorbeeld een instantie, gemeente of het rijk).

De workbench is in detail te zien in bijlage IV.

### Berekenen oppervlakte EHS

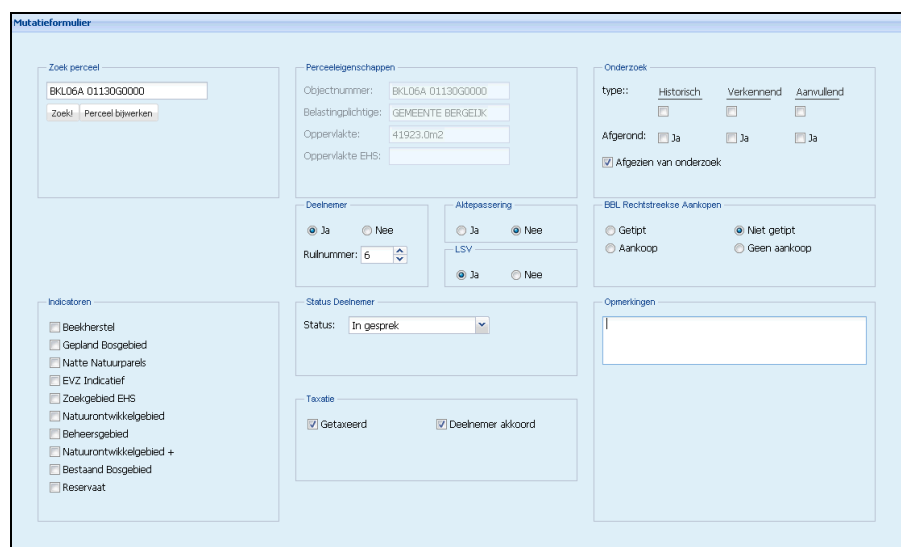
Deze workbench is een vrij eenvoudige: Het confronteert de percelen met het EHS polygoon en berekent de oppervlakten van het gemeenschappelijke deel.

#### 5.4.3 Mutatiebeheer

Het mutatiebeheer is een interface die de gebruiker in staat stelt de eigenschappen van een bepaald perceel te wijzigen. Hiervoor wordt gebruikt gemaakt van Ext JS. De pagina bestaat in feite uit een groot formulier waarin alle administratie van een perceel inzichtelijk is en grotendeels is te wijzigen. Er kan gezocht worden naar een perceel door het objectnummer in te geven. Indien het perceel gevonden wordt, dan wordt deze gelijk getoond. Is er geen overeenkomstig perceel gevonden, dan verschijnt er een foutmelding.

Als de eigenschappen van een perceel zijn aangepast, kan het formulier worden verzonden en volgt er een melding of de bijwerking al dan niet geslaagd is.

Voor het verzenden en ontvangen van het formulier wordt de API aangesproken. De werking hiervan is terug te lezen in hoofdstuk 5.4.1 (kern).



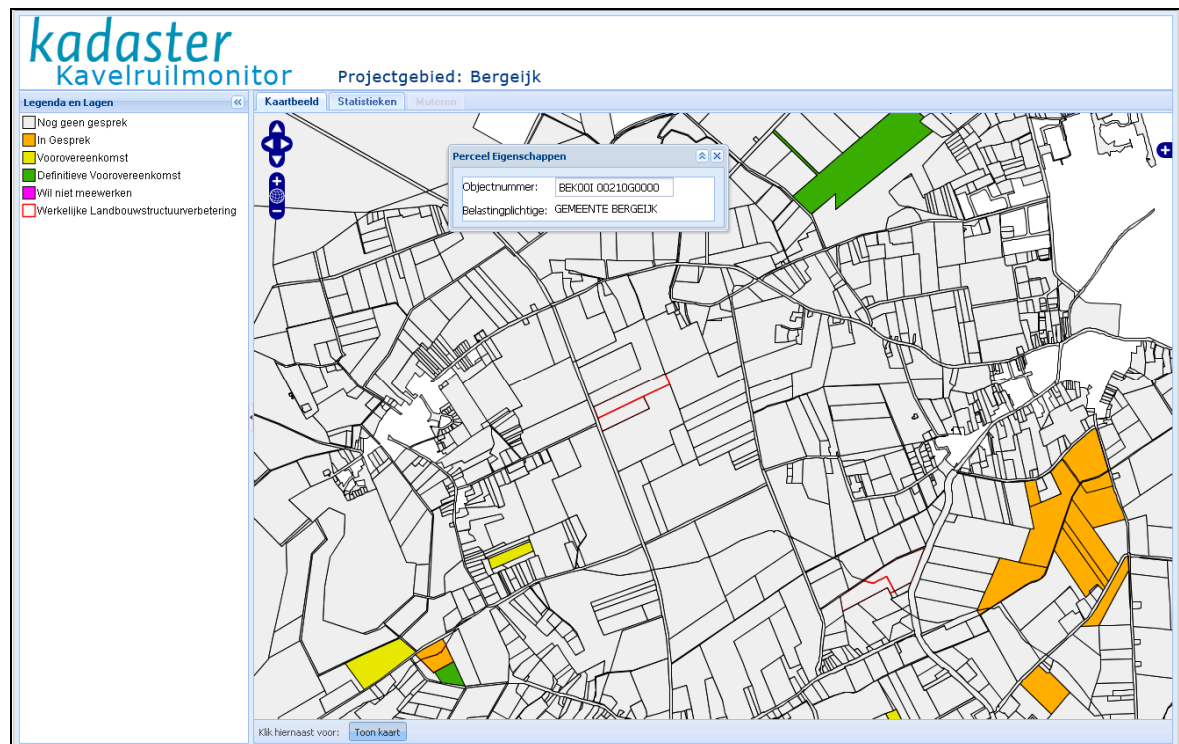
**Figuur 19: Mutatiebeheer**

#### 5.4.4 Presentatie

De presentatie uit zich in een interface waarin zowel het kaartbeeld als de statistiek inzichtelijk gemaakt worden. De pagina beschikt over twee tabbladen: Het kaartbeeld en de statistiek.

##### Tabblad: Kaartbeeld

Het kaartbeeld visualiseert alle niet-vervallen percelen. De percelen zijn geclassificeerd op voortgang in het proces. Door op een perceel te klikken opent er een pop-up met daarin het objectnummer en de belastingplichtige.



**Figuur 20: Kaartbeeld**

Voor de interface is Ext JS ingezet. Het kaartbeeld wordt getoond met behulp van Openlayers. Openlayers maakt het onder andere mogelijk om de kaart te pannen, in- en uit te zoomen en via een API coördinaten te berekenen.

Het daadwerkelijke kaartbeeld, 'het plaatje', wordt door Geoserver gerenderd. Zoals de naam al doet vermoeden is dit een applicatie die draait aan de server-kant. Voor de Kavelruilmonitor wordt Geoserver ingezet als WMS server. Dit houdt in dat de server aan te roepen is via een URL en als antwoord een afbeelding retourneert met daarin het gerenderde kaartbeeld. De stijl of opmaak van dit kaartbeeld wordt bepaald in een Styled Layer Descriptor (SLD). Een SLD is een XML-bestand waarin gedefinieerd wordt hoe de kaartdata gevisualiseerd moet worden.

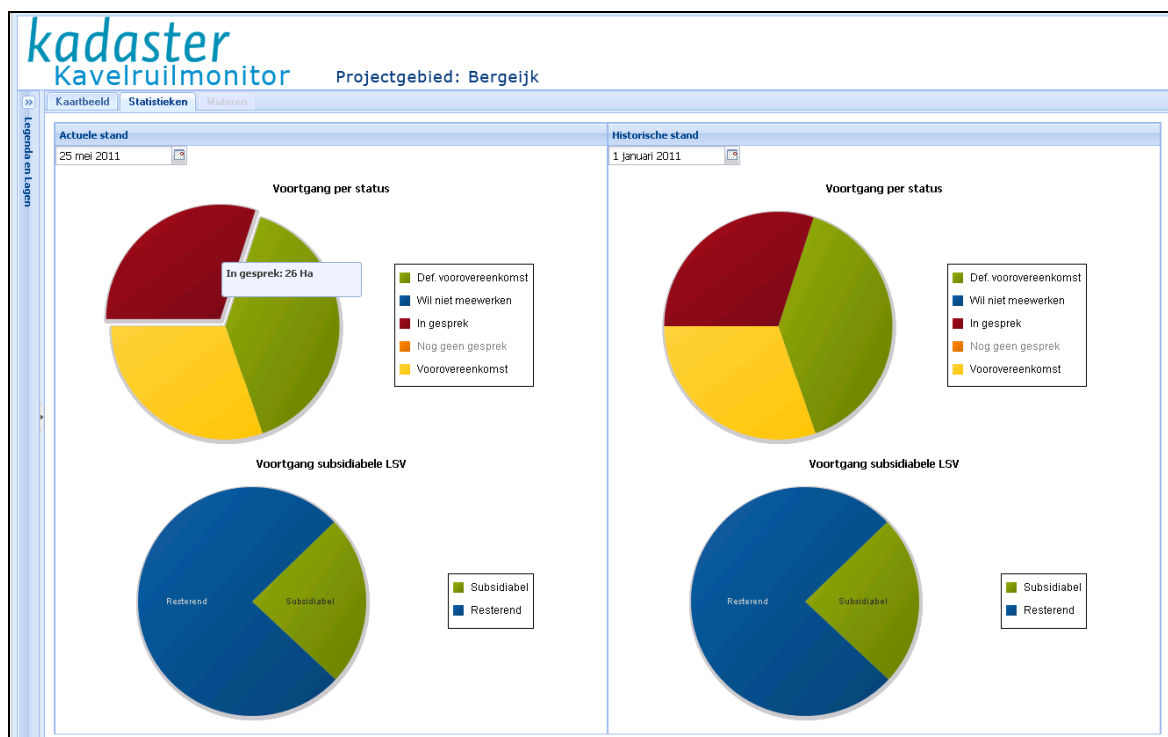


```
<FeatureTypeStyle>
  <Rule>
    <Name>Status</Name>
    <Title>Nog geen gesprek</Title>
    <ogc:Filter>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>voortgang_deelnemer</ogc:PropertyName>
        <ogc:Literal>1</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Filter>
    <PolygonSymbolizer>
      <Fill>
        <CssParameter name="fill">#EDEDDE</CssParameter>
      </Fill>
      <Stroke>
        <CssParameter name="stroke">#000000</CssParameter>
        <CssParameter name="stroke-width">0.5</CssParameter>
      </Stroke>
    </PolygonSymbolizer>
  </Rule>
</FeatureTypeStyle>
```

Figuur 21: Codefragment SLD

### Tabblad: Statistiek

In het tabblad statistiek wordt de voortgang van het project gevisualiseerd door middel van taartdiagrammen. Deze diagrammen zijn opgedeeld in twee kolommen: Eén voor de actuele stand en één voor een historische stand. Bovenaan de kolom van historische stand kan een datum worden ingevoerd waarvan de statistiek getoond moet worden.



Figuur 22: Statistiek



Het bovenste taartdiagram geeft de voortgang weer op basis van de status in het project. De cijfers worden verkregen door de oppervlakte binnen de EHS van alle percelen te aggregeren op het attribuut *status voortgang*. Het onderste taartdiagram geeft de voortgang LSV weer. Het gaat hierbij om subsidiabele LSV. Deze cijfers worden verkregen door de oppervlakte te sommeren van alle percelen van eigenaren waarvan één of meer van hun percelen aangemerkt zijn voor LSV.

De berekeningen vinden plaats op de server via een Pythonscript. Via de API worden de cijfers in JSON naar Ext JS gestuurd. Hieronder staan twee query's afgebeeld waarmee de selecties uit de database worden gemaakt:

```
SELECT voortgang_deelnemer, round(sum(oppervlak_ehs) / 10000) AS  
oppervlak_status FROM perceel WHERE perceel_eindtijd IS NULL GROUP BY  
voortgang_deelnemer
```

**Query: voortgang per status (actuele stand)**

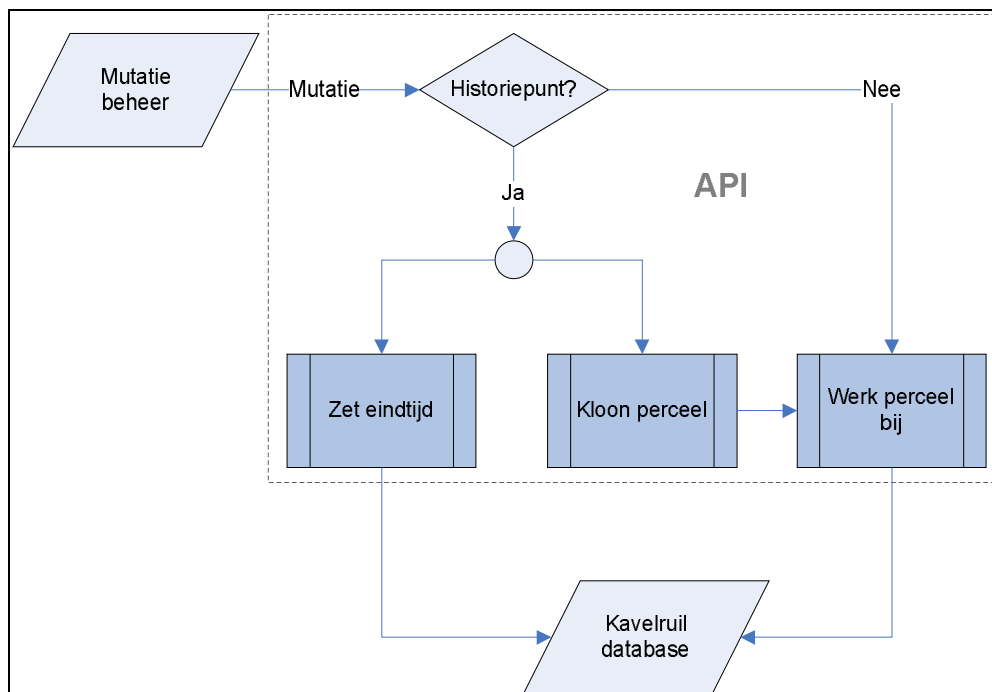
```
SELECT round(sum(ST_Area(geom)) / 10000) AS lsv_subsiadiabel FROM perceel  
WHERE nr_gerechtigde IN(SELECT nr_gerechtigde FROM perceel WHERE  
nnp_gerechtigde = false AND status_lsv = true) AND perceel_eindtijd IS  
NULL
```

**Query: subsidiabele LSV (actuele stand)**

De financiële gegevens zijn nog niet beschikbaar in grafiekvorm. Er zijn hier op het moment van schrijven nog geen gegevens van beschikbaar. Wel is er een tabel aangemaakt waar in de toekomst deze gegevens naar toe geschreven kunnen worden.

## 5.5 Historie: de temporele component

Eén van de gewenste functionaliteiten is het bijhouden van historie. Het moet bijvoorbeeld mogelijk zijn om de statistieken te laten berekenen van een half jaar geleden. Om dit te realiseren heeft elk perceel een begin- en eindtijd. Als een perceel gemuteerd wordt, bepaalt de API of er een attribuut is gewijzigd dat van invloed is op de historie. In het gegevensmodel is te zien welke attributen dit zijn. Een perceel kan ook geen eindtijd hebben, het is dan een actueel perceel. Figuur 23 geeft dit proces schematisch weer.



**Figuur 23: Bijhouding historie**

## 5.6 Ontwikkelsoftware

Voor de ontwikkeling van de Kavelruilmonitor is verscheidene ondersteunende software gebruikt. Voor het schrijven van alle broncode is gebruik gemaakt van Notepad++. Deze applicatie lijkt veel op het standaard kladblok dat in Microsoft Windows wordt meegeleverd, alleen is het gericht op ontwikkelaars. Tevens zit er een heel uitgebreid pakket aan functies bij. De meest handige zijn de regelnummering en de zogenaamde 'syntax-highlighting'.

Voor het beheer van Oracle en PostgreSQL zijn respectievelijk SQL Developer en PgAdmin III gebruikt.



Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
35 van 40

## 6 Lopende ontwikkelingen en de toekomst

### Voorlopige kadastrale grenzen

Het idee achter de voorlopige grenzen is dat voorafgaand aan de aktepassering van een gedeelte van een perceel, de nieuwe kadastrale grenzen al worden gevormd. Dit splitsen kan gedaan worden via de applicatie 'Splits'. Zolang de landmeter de percelen nog niet in heeft gemeten, krijgen deze percelen de status 'voorlopig'. Het grote voordeel van dit systeem is dat er nu niet langer deelpercelen worden gevormd en de AKR en het LKI nagenoeg synchroon gaan lopen.

Voor de Kavelruilmonitor heeft dit als voordeel dat de nieuw gevormde percelen eerder beschikbaar zijn. Daardoor kunnen deze percelen worden meegenomen in de oppervlakteberekeningen en kan er dus een meer actueel beeld gegeven worden.

Op het moment van schrijven staat dit systeem op het punt om ingevoerd te worden, en zullen alle gebruikers gefaseerd worden opgeleid voor de Splitsapplicatie met als doel dat op termijn de deelpercelen volledig worden vervangen door de voorlopige kadastrale grenzen.

### Politiek

Aan het begin van de ontwikkeling had de EHS een prominente rol binnen de Kavelruilmonitor. Dit kwam, omdat het realiseren van de EHS het grootste doel van de kavelruil in Bergeijk was. In het nieuwe regeerakkoord van het kabinet Rutte is er echter flink bezuinigd op de EHS. Het beheer van de EHS wordt wel in stand gehouden, maar voor de aankoop van EHS-gronden gaat de geldkraan dicht. Omdat de financieringen voor de LSV wel in stand blijven, is de focus veel meer daar op komen te liggen.

De architectuur en ontwikkelmethode van de Kavelruilmonitor hebben er voor gezorgd dat een aanpassing naar LSV vrij eenvoudig in te passen was.



## 7 Conclusies en aanbevelingen

### 7.1 Conclusies

Het doel van dit onderzoek is het realiseren van een systeem dat als instrument kan worden gebruikt om ondersteuning te bieden bij het kavelruilproject Bergeijk. Vastgesteld kan worden dat dat doel voor een groot deel is behaald. Enkel de koppeling met de financiële administratie is niet gerealiseerd.

Als eerste stap is er een Kavelruildatabase ontworpen en ingericht in een PostgreSQL omgeving. Deze database is vervolgens geladen met percelen uit het LKI en aangevuld met gegevens uit de AKR. De gegevens in de Kavelruildatabase worden periodiek bijgewerkt door ze te vergelijken met het LKI en de AKR. Op deze manier is de mutatiesignalering geborgd. Om het systeem goed toegankelijk te maken is er gekozen voor een oplossing in de vorm van een webapplicatie. Via één overzichtelijke pagina kunnen de percelen gemuteerd worden. Deze mutaties worden rechtstreeks bijgewerkt in de database. Via een andere pagina wordt kaart getoond en is de statistiek te zien in de vorm van taartdiagrammen. Door aan de percelen een temporele component toe te voegen in de vorm van een begin- en eindtijd is er voorzien in historiebijhouding.

De Kavelruilmonitor kan door zijn modulaire opzet ook ingezet worden voor andere projecten dan kavelruilproject Bergeijk. Voorwaarde is wel dat deze projecten een vergelijkbaar karakter als dat van Bergeijk hebben. Naar verwachting zullen er wel kleine aanpassingen in de broncode nodig, dus zal hiervoor een GIS- of ICT-specialist ingeschakeld moeten worden.

Een aanvullende vraag die tijdens het onderzoek naar voren kwam was, welke mogelijkheden open source software kan bieden binnen een dergelijke applicatie. Gesteld kan worden dat de OSS die binnen de Kavelruilmonitor is gebruikt uitermate goed functioneert. Vrijwel alle pakketten worden al langere tijd ontwikkeld en hebben zeer uitgebreide documentatie en een levendige community. Met name voor Ext JS is er veelvuldig het gebruikersforum geraadpleegd, waar vele problemen mee zijn opgelost. Het verschil ten opzichte van commerciële software is de houding die men als gebruiker aan moet nemen. Met OSS is een actieve houding aan te bevelen: zelf op zoek gaan naar antwoorden. Bij commerciële software wordt sneller de support van een helpdesk ingeschakeld.

### 7.2 Aanbevelingen

#### Ontwikkelen in een team

Voor toekomstige ontwikkeling van de software wordt het werken in een team aanbevolen. Door de modulaire opzet (de vier deelsystemen) van de Kavelruilmonitor is het eenvoudig om met meerdere personen tegelijk te werken. Een minimum van twee personen wordt aangeraden: één persoon ontwikkelt de kern en synchronisatie, de andere persoon ontwikkelt het mutatiebeheer en de presentatie. Daarnaast worden 'Single Points Of Knowledge' geëlimineerd.



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

37 van 40

### **Onderzoek naar implementatie en beveiliging**

De implementatie binnen de ICT-infrastructuur van het Kadaster evenals de beveiliging van de applicatie vallen buiten dit onderzoek. Het is daarom noodzakelijk, dat voor het in productie nemen van de Kavelruilmonitor deze twee onderwerpen worden onderzocht.

### **Registratie van de financiën**

Het is in dit onderzoek niet gelukt om een oplossing te vinden voor de registratie van de financiën. Hier zal dus nog verder onderzoek naar gedaan moeten worden. Echter, een mogelijke oplossing kan zijn dat de verantwoordelijke medewerker maandelijks de totale uitgaven invult op een speciale pagina van de Kavelruilmonitor.

### **Code optimalisatie**

Hoewel de performance van de Kavelruilmonitor goed is, valt er aan de broncode nog het één en ander te verbeteren. Nu zitten er nog een aantal 'harde' verwijzingen in naar de database en tabellen. Door deze te parametriseren wordt de applicatie flexibeler en is daardoor eenvoudiger om te bouwen naar andere projecten. Ook is de manier waarop de query's naar de database worden samengesteld gedaan op een methode die star van aard is en een mogelijk veiligheidsrisico met zich meebrengt.

### **FME Server**

In de huidige situatie wordt er gebruik gemaakt van FME Desktop voor de synchronisatie van de database. Deze processen draaien op de server en daarom zou een server-versie van FME wenselijk zijn. FME Server is ontworpen om serverprocessen te draaien en is daarom robuuster van aard. Bijkomend voordeel is de mogelijkheid om workbenches te 'schedulen' en het versturen van een e-mail bij eventuele fouten.



Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
38 van 40

## 8 Bronnen

Onderstaande websites zijn geraadpleegd in de periode januari 2011 – juni 2011

[www.kavelruilbergeijk.nl](http://www.kavelruilbergeijk.nl)  
[www.kadaster.nl](http://www.kadaster.nl)  
[www.wikipedia.org](http://www.wikipedia.org)  
[www.openlayers.org](http://www.openlayers.org)  
[www.sencha.com](http://www.sencha.com) (Ext JS)  
[www.geoserver.org](http://www.geoserver.org)  
[www.python.org](http://www.python.org)  
[www.cherrypy.org](http://www.cherrypy.org)  
[www.initd.org](http://www.initd.org) (Pyscopg2)  
[www.safe.com](http://www.safe.com) (FME)  
[www.postgresql.org](http://www.postgresql.org)  
<http://postgis.refractory.net>  
[www.oracle.com](http://www.oracle.com)

Intranet Kadaster

### Boeken en vakbladen:

Lutz, Mark (2009). *Learning Python, Fourth Edition*. Sebastopol: O'Reilly Media, Inc.

Westra, Erik (2010). *Python Geospatial Development*. Birmingham: Packt Publishing.

Nederhoed, Peter (2010). *Helder Rapporteren*. Houten: Bohn Stafleu van Loghum.

Osch, Berry van & Sterenberg, Meindert (Geo-Info 2011-5). *Kadaster vervroegt perceelvorming bij grondtransacties*.



Datum

1 juni 2011

Titel

Kavelruilmonitor Bergeijk

Versie

1.0

Blad

39 van 40

## Begrippenlijst

In dit rapport komen een flinke verzameling begrippen en afkortingen voor, die niet gelijk voor zichzelf spreken. Hieronder staan deze opgesomd, op alfabetische volgorde:

AKR	Automatisering Kadastrale Registraties
API	Application Programming Interface
BBL	Bureau Beheer Landbouwgronden
CIDB	Cartografische Informatie Database
DLG	Dienst Landelijk Gebied
EHS	Ecologische Hoofdstructuur
ETL	Extract, Transform and Load
FME	Feature Manipulation Engine
GPL	General Public License
ISPS	Informatiesysteem Primaire Services
JSON	Javascript Standard Object Notation
LKI	Landmeetkundig Kartografisch Informatiesysteem
LSV	Landbouwstructuurverbetering
MO	Massale Output
ORDBMS	Object Relational Database Management System
OSS	Open Source Software
SLD	Styled Layer Descriptor
SQL	Structured Query Language
URL	Uniform Resource Locator
WILG	Wet Inrichting Landelijk Gebied
XML	eXtensible Markup Language
ZBO	Zelfstandig Bestuursorgaan





Datum  
1 juni 2011

Titel  
Kavelruilmonitor Bergeijk

Versie  
1.0

Blad  
40 van 40

## Bijlagenoverzicht

- I. Gegevensmodel
- II. FME workbench: Synchroniseren LKI
- III. FME workbench: Synchroniseren AKR
- IV. FME workbench: Berekenen oppervlakte EHS
- V. Python broncode
- VI. Javascript broncode

Bijlage I:  
**Gegevensmodel**

Tabel: Perceel

Attribuut	Type	Waarden	Opmerkingen
geom	geometry	Geometrie van het type polygoon. Precisie: 3 decimalen. SRS ID: EPSG:28992	
id (pk)	serial	Unieke sleutel. Wordt automatisch in oplopende volgorde aangemaakt door de database.	
nr_akr	text	Het objectnummer van het perceel. Format: <b>XXX01 012345G0000</b> . Gelijk aan het objectnummer in AKR.	
perceel_begintijd	date	De systeemtijd waarop het perceel wordt aangemaakt. Format: <b>YYYY-MM-DD</b>	
perceel_eindtijd	date	De systeemtijd waarop het perceel vervalt. Format: <b>YYYY-MM-DD</b>	
perceel_eindtijd_info	text	Informatie over de reden waarom het perceel is vervallen.	
nnp_gerechtigde	boolean	<b>true</b> : Perceel behoort toe aan een niet-natuurlijk persoon. <b>false</b> : Perceel behoort toe aan een natuurlijk persoon.	
nr_gerechtigde	bigint	Het subjectnummer van de belastingplichtige op het perceel. Gelijk aan het subjectnummer in AKR.	
naam_gerechtigde	text	De naam behorende bij nr_gerechtigde.	
isps_mi_prinx (fk)	integer	Gelijk aan het attribuut 'MI_PRINX' in de ISPS-database.	
status_deelnemer	boolean	<b>true</b> : Perceel doet mee in het project. <b>false</b> : Perceel doet niet mee in het project.	Attribuut is trigger voor historiepunt.
status_lsv	boolean	<b>true</b> : Perceel doet mee in de Landbouwstructuurverbetering. <b>false</b> : Perceel doet niet mee in de Landbouwstructuurverbetering.	
oppervlak_ehs	double	Het oppervlak van het perceel dat overlapt met het EHS-Polygoon. Eenheid in vierkante meters.	
ruilnummer	smallint	Het nummer van de 'ruil' waar het perceel in mee doet.	
bbl_getipt	boolean	<b>true</b> : Bureau Beheer Landbouwgronden is getipt. <b>false</b> : Bureau Beheer Landbouwgronden is niet getipt.	
bbl_aankoop	boolean	<b>true</b> : Perceel is aangekocht door BBL. <b>false</b> : Perceel is niet aangekocht door BBL.	
status_taxatie	boolean	<b>true</b> : Perceel is getaxeerd. <b>false</b> : Perceel is (nog) niet getaxeerd.	
deelnr_taxatie	boolean	<b>true</b> : Deelnemer is akkoord met taxatie. <b>false</b> : Deelnemer is niet akkoord met taxatie.	

status_gesprek	boolean	<b>true:</b> In gesprek met deelnemer. <b>false:</b> Niet in gesprek met deelnemer.	
status_aktepassering	boolean	<b>true:</b> Akte is gepasseerd. <b>false:</b> Akte is niet gepasseerd.	Attribuut is trigger voor historiepunt.
status_hist_onderzoek	boolean	<b>true:</b> Historisch onderzoek is gestart. <b>false:</b> Historisch onderzoek is niet gestart.	
result_hist_onderzoek	boolean	<b>true:</b> Historisch onderzoek is afgerond. <b>false:</b> Historisch onderzoek is afgerond.	
status_verk_onderzoek	boolean	<b>true:</b> Verkennend onderzoek is gestart. <b>false:</b> Verkennend onderzoek is niet gestart.	
result_verk_onderzoek	boolean	<b>true:</b> Verkennend onderzoek is afgerond. <b>false:</b> Verkennend onderzoek is niet afgerond..	
status_aanv_onderzoek	boolean	<b>true:</b> Aanvullend onderzoek is gestart. <b>false:</b> Aanvullend onderzoek is niet gestart.	
result_aanv_onderzoek	boolean	<b>true:</b> Aanvullend onderzoek is afgerond. <b>false:</b> Aanvullend onderzoek is niet afgerond.	
onderzoek_nee	boolean	<b>true:</b> Deelnemer heeft afgezien van onderzoek. <b>false:</b> Deelnemer heeft niet afgezien van onderzoek.	
ind_beekherstel	boolean	<b>true:</b> Perceel valt onder het beekherstel. <b>false:</b> perceel valt niet onder het beekherstel.	
ind_natuurparels	boolean	<b>true:</b> Perceel valt onder natte natuurparels, <b>false:</b> perceel valt niet onder natte natuurparels.	
ind_zoekgebied_ehs	boolean	<b>true:</b> Perceel valt onder het zoekgebied EHS. <b>false:</b> perceel valt niet onder het zoekgebied EHS.	
ind_beheersgebied	boolean	<b>true:</b> Perceel valt onder het beheersgebied. <b>false:</b> perceel valt niet onder het beheersgebied.	
ind_bestaand_bos_natuur	boolean	<b>true:</b> Perceel valt onder bestaand bos- of natuurgebied. <b>false:</b> perceel valt niet onder bestaand bos- of natuurgebied.	
ind_gepland_bos_natuur	boolean	<b>true:</b> Perceel valt onder gepland bos- of natuurgebied. <b>false:</b> perceel valt niet onder gepland bos- of natuurgebied..	
ind_evz_indicatief	boolean	<b>true:</b> Perceel valt onder het EVZ indicatieve gebied. <b>false:</b> perceel valt niet onder het EVZ indicatieve gebied.	
ind_natuurontw_gebied	boolean	<b>true:</b> Perceel valt onder natuurontwikkelingsgebied. <b>false:</b> perceel valt niet onder natuurontwikkelingsgebied.	
ind_natuurontw_gebied_plus	boolean	<b>true:</b> Perceel valt onder natuurontwikkelingsgebied+.	

		<b>false:</b> perceel valt niet onder natuurontwikkelingsgebied+.	
ind_reservaatgebied	boolean	<b>true:</b> Perceel valt onder reservaatgebied. <b>false:</b> perceel valt niet onder reservaatgebied.	
voortgang_deelnemer	integer	<p><b>1: Nog geen gesprek</b> Er heeft nog geen gesprek met deelnemer plaatsgevonden.</p> <p><b>2: In gesprek</b> Gesprek heeft met deelnemer plaatsgevonden.</p> <p><b>3: Voorovereenkomst</b> Voorovereenkomst is getekend.</p> <p><b>4: Definitieve voorovereenkomst</b> Definitieve voorovereenkomst is getekend.</p> <p><b>5: Wil niet meewerken</b> Deelnemer verleent (nog) geen medewerking.</p>	Attribuut is trigger voor historiepunt.
opmerkingen	text	Opmerkingen met betrekking tot het perceel of deelnemer.	

**Tabel: Gebied**

Attribuut	Type	Waarden	Opmerkingen
geom	geometry	Geometrie van het type polygoon. Multi-part toegestaan. Precisie: 3 decimalen. SRS ID: EPSG:28992	
id	serial	Unieke sleutel. Wordt automatisch in oplopende volgorde aangemaakt door de database.	
naam	text	Naam van het gebied	
info	test	Informatie over het gebied	

**Tabel: Financieel**

Attribuut	Type	Waarden	Opmerkingen
id	serial	Unieke sleutel. Wordt automatisch in oplopende volgorde aangemaakt door de database.	
datum	date	Datum die overeenstemt met de hoogte van het saldo.	
saldo	numeric	Het saldo in Euro's.	
opmerking	text	Eventuele opmerkingen	

**Tabel: Log**

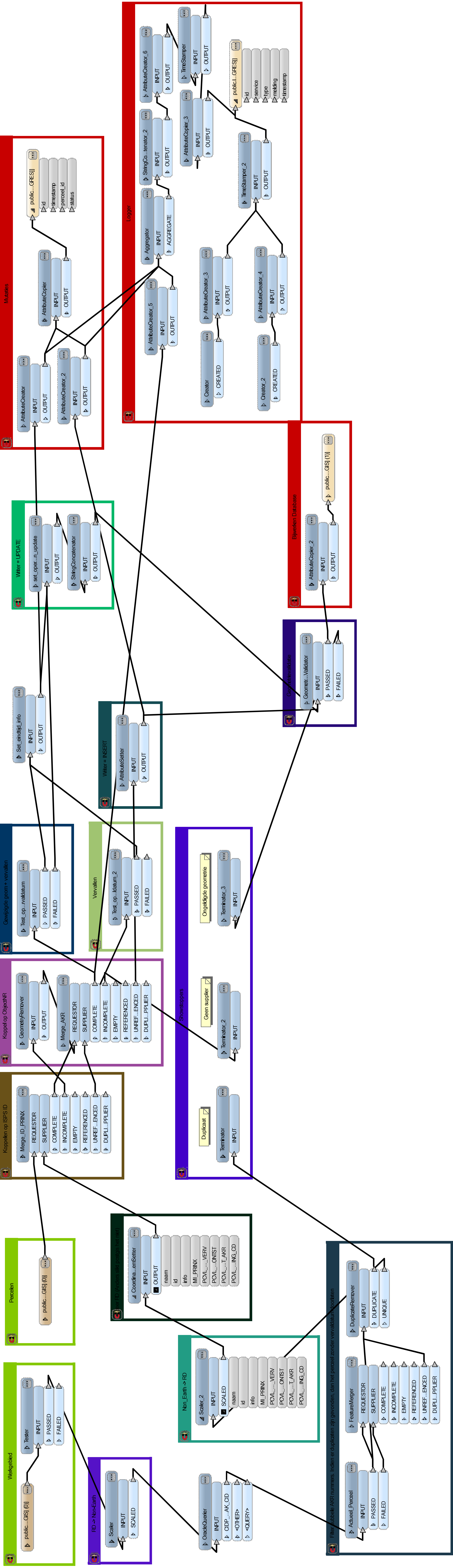
Attribuut	Type	Waarden	Opmerkingen
service	text	Naam van de service waar de melding betrekking op heeft.	
type	text	Type van de melding: <b>error, warning of message</b>	
melding	text	Omschrijving van de melding	
timestamp	timestamp	Timestamp waarop de melding zich voordoet. Format: <b>YYYY-MM-DD HH:MM:SS</b>	

**Tabel: Mutatie**

Attribuut	Type	Waarden	Opmerkingen
id (pk)	serial	Unieke sleutel. Wordt automatisch in oplopende volgorde aangemaakt door de database.	
perceel_id (fk)	integer	Overeenkomstig met de primaire sleutel uit de tabel: perceel.	
status	text	“nieuw”, “vervallen” of “gewijzigd”. Optioneel tussen rechte haakjes de reden van wijziging of vervallen.	
timestamp	timestamp	Timestamp waarop de mutatie is gesignaleerd. Format: <b>YYYY-MM-DD HH:MM:SS</b>	

## Bijlage II:

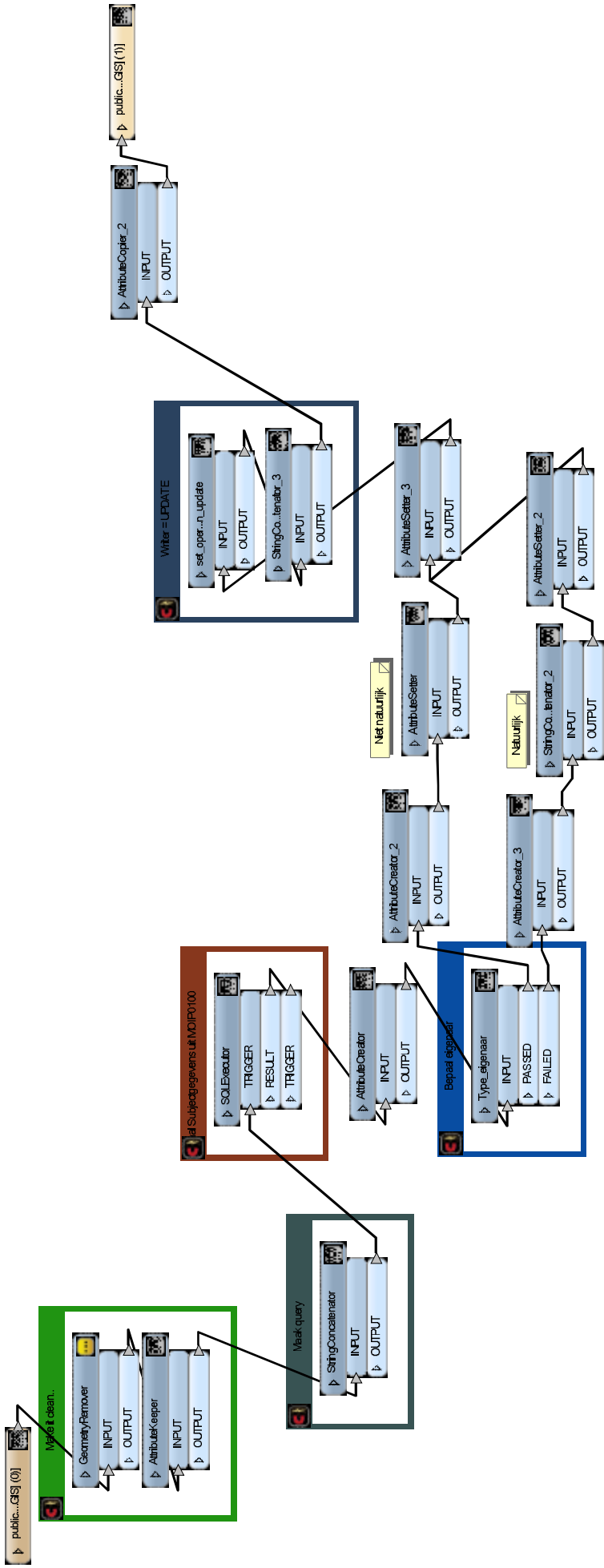
### **FME Workbench: Synchroniseren LKI**





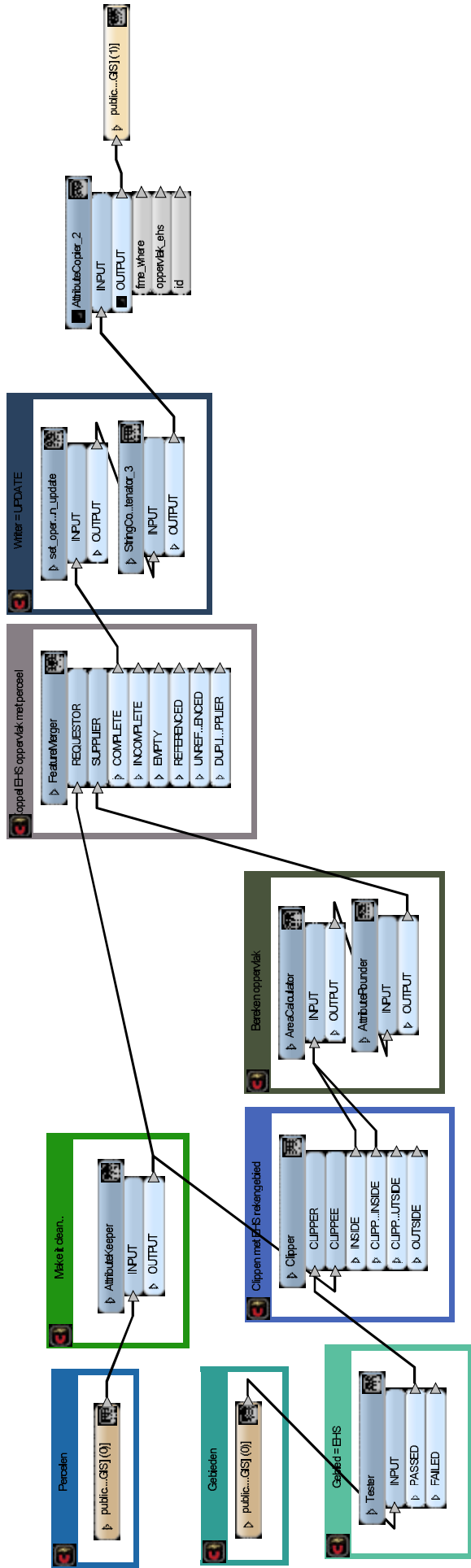
## Bijlage III:

### **FME Workbench: Synchroniseren AKR**



## Bijlage IV:

### **FME Workbench: Berekenen oppervlakte EHS**



## Bijlage V:

### Python Broncode

```

import cherrypy
import psycopg2
from Cheetah.Template import Template
from functools import wraps
import json
import datetime

def jsonify(func):
    '''JSON decorator for CherryPy'''
    @wraps(func)
    def wrapper(*args, **kw):
        value = func(*args, **kw)
        cherrypy.response.headers["Content-Type"] = "application/json"
        return json.dumps(value)

    return wrapper

class KrmApp:
    def index(self):
        t = Template(file="index.tmpl"
            , searchList = {
                "title" : "Kavelruilmonitor Webapplicatie"
                , "cssFiles" : ["ext4/resources/css/ext-all.css"]
                , "jsFiles" : ["ext4/bootstrap.js",
"openlayers/lib/OpenLayers.js", "js/krmapp.js"]
                , "content" : "Kavelruilmonitor webapplicatie"
            }
        )
        return str(t)

    def muteren(self):
        t = Template(file="muteren.tmpl"
            , searchList = {
                "title" : "Kavelruilmonitor Webapplicatie | Muteren"
                , "cssFiles" : ["ext4/resources/css/ext-all.css"]
                , "jsFiles" : ["ext4/bootstrap.js", "js/muteren.js"]
                , "content" : "Kavelruilmonitor webapplicatie"
            }
        )
        return str(t)

    @jsonify
    def zoekperceel(self, *args, **kw):
        try: # maak connectie met Postgres
            conn = psycopg2.connect("host=localhost port=5432
dbname=kavelruilmonitor user=postgres password=postgres")
            cur = conn.cursor()
            # open cursor
        except:
            print "connectie mislukt" #TODO naar .log schrijven
        if 'objectnummer' in args:
            cur.execute("SELECT nr_akr, \
                naam_gerechtigde, \
                round(ST_Area(geom)) AS oppervlakte, \
                oppervlak_ehs, \
                status_deelnemer, \

```

```

voortgang_deelnemer, \
ruilnummer, \
bbl_getipt, \
bbl_aankoop, \
status_taxatie, \
deelnr_taxatie, \
status_aktepassering, \
ST_XMin(geom) AS xmin, \
ST_YMin(geom) AS ymin, \
ST_XMax(geom) AS xmax, \
ST_YMax(geom) AS ymax, \
status_lsv, \
status_hist_onderzoek, \
status_verk_onderzoek, \
status_aanv_onderzoek, \
result_hist_onderzoek, \
result_verk_onderzoek, \
result_aanv_onderzoek, \
onderzoek_nee, \
opmerkingen, \
id, \
ind_beekherstel, \
ind_natuurparels, \
ind_zoekgebied_ehs, \
ind_beheersgebied, \
ind_bestaad_bos_natuur, \
ind_gepland_bos_natuur, \
ind_evz_indicatief, \
ind_natuurontw_gebied, \
ind_natuurontw_gebied_plus, \
ind_reservaatgebied \
FROM perceel WHERE nr_akr = ' ' + kw['query'] + ' ' AND
perceel_eindtijd IS NULL") ## TODO

result = cur.fetchone()
conn.close()

#Alle booleans omzetten naar string (dit kan beter in een loop
vooraf --> TODO)
data = {"success": bool('True'), "data": {
    "objectnummer": result[0],
    "naam_gerechtigde": result[1],
    "oppervlakte": str(result[2]) + 'm2',
    "oppervlakte_ehs": result[3],
    "status_deelnemer": str(result[4]),
    "voortgang_deelnemer": result[5],
    "ruilnummer": result[6],
    "bbl_getipt": str(result[7]),
    "bbl_aankoop": str(result[8]),
    "status_taxatie": str(result[9]),
    "deelnr_taxatie": str(result[10]),
    "status_aktepassering": str(result[11]),
    "xmin": result[12],
    "ymin": result[13],
    "xmax": result[14],
    "ymax": result[15],
    "status_lsv": str(result[16]),

```

```

        "status_hist_onderzoek": str(result[17]),
        "status_verk_onderzoek": str(result[18]),
        "status_aanv_onderzoek": str(result[19]),
        "result_hist_onderzoek": str(result[20]),
        "result_verk_onderzoek": str(result[21]),
        "result_aanv_onderzoek": str(result[22]),
        "onderzoek_nee": str(result[23]),
        "opmerkingen": result[24],
        "id": result[25],
        "ind_beekherstel": result[26],
        "ind_natuurparels": result[27],
        "ind_zoekgebied_ehs": result[28],
        "ind_beheersgebied": result[29],
        "ind_bestaand_bos_natuur": result[30],
        "ind_gepland_bos_natuur": result[31],
        "ind_evz_indicatief": result[32],
        "ind_natuurontw_gebied": result[33],
        "ind_natuurontw_gebied_plus": result[34],
        "ind_reservaatgebied": result[35]}}

    return data
    ## Query o.b.v. coördinaten
elif 'xy' in args:
    cur.execute("SELECT nr_akr, \
naam_gerechtigde \
FROM perceel_huidig \
WHERE ST_Intersects(ST_PointFromText('SRID=28992;POINT(" +
kw['query'] + ")'), geom);")

    result = cur.fetchone()
    conn.close()

    data = {"success": bool('True'), "data": {
        "nr_akr": result[0],
        "naam_gerechtigde": result[1]}}

    return data

else:
    ## Voor toekomstig gebruik
    if "start" in kw:
        offset = kw['start']
    else:
        offset = str(0)
    cur.execute("SELECT count(*) FROM perceel WHERE nr_akr LIKE '" +
args[0] + "%'")
    results = cur.fetchone()

    cur.execute("SELECT nr_akr, nr_gerechtigde, ST_Area(geom) AS
oppervlakte FROM perceel WHERE nr_akr LIKE '" + args[0] + "%' OFFSET "
+ offset + "LIMIT 50")

    result = cur.fetchall()
    conn.close()
    rows = []
    for r in result:

```



```

        rows.append(dict([('Objectnummer', r[0]), ('Gerechtigde',
r[1]), ('Oppervlak', r[2])]))
        data = {"success": 'true', "results": 2600, "rows": rows}
        return data

@jsonify
def bewerkperceel(self, *args, **kw):

    try: # maak connectie met Postgres
        conn = psycopg2.connect("host=localhost port=5432
dbname=kavelruilmonitor user=postgres password=postgres")
        cur = conn.cursor()
        # open cursor

    except:
        print "connectie mislukt" #TODO naar .log schrijven
        cur.execute("SELECT id, \
            status_deelnemer, \
            voortgang_deelnemer, \
            status_aktepassering, \
            perceel_begintijd, \
            geom, \
            nr_akr, \
            naam_gerechtigde, \
            nr_gerechtigde, \
            isps_mi_prinx, \
            nnp_gerechtigde \
            FROM perceel where perceel_eindtijd IS NULL AND id = " + kw['id'])
        result = cur.fetchone()

        # Status veranderd welke belangrijk is voor historie, en meer dan
        1 dag verschil tussen records? --> vul eindtijd in en creëer nieuw
        object met eindtijd=beginntijd
        if (eval(kw['status_deelnemer']) != bool(result[1]) or
eval(kw['voortgang_deelnemer']) != result[2] or
eval(kw['status_aktepassering']) != bool(result[3])) and
datetime.date.today() > result[4]:
            data = {"success": bool('true'), "data": kw}
            geom = result[5]
            timestampdb = result[4]
            timestamp = datetime.date.today()
            id = kw.pop('id') ## ID separeren van de waarden, om update te
voorkomen
            for k, v in kw.items(): ## Niks = NULL
                if v == '':
                    x = {k: 'Null'}
                    kw.update(x)
                if k == 'opmerkingen':
                    x = {k: "" + v + ""}
                    kw.update(x)
            keys = values = []
            keys = ",".join(["%s" % (k) for k, v in kw.items()])
            values = ",".join(["%s" % (v) for k, v in kw.items()])

            updatequery = "UPDATE perceel SET perceel_eindtijd_info = 'Status
gewijzigd via muteerpagina', perceel_eindtijd = '" + str(timestamp) +
" ' WHERE id = " + id + ";"

```

```

        insertquery = "INSERT INTO perceel (geom, perceel_begintijd,
nr_akr, naam_gerechtigde, nr_gerechtigde, isps_mi_prinx,
nnp_gerechtigde, " + keys + ") VALUES ('" + geom + "','" +
str(timestamp) + "','" + result[6] + "','" + result[7] + "','" +
str(result[8]) + "','" + str(result[9]) + "','" + str(result[10]) + "','" +
values + "');" #TODO parameteriseren!
        cur.execute(updatequery)
        cur.execute(insertquery)
        conn.commit()
        conn.close()
        # Geen historie nodig: update het perceel
    else:
        data = {"success": bool('true'), "data": kw}
        updatefields = []
        id = kw.pop('id') ## ID separeren van de waarden, om update te
voorkomen.
        for k, v in kw.items(): ## Niks = NULL
            if v == '':
                x = {k: 'Null'}
                kw.update(x)
            if k == 'opmerkingen':
                x = {k: "" + v + ""}
                kw.update(x)
        updatefields = ",".join(["%s=%s" % (k, v) for k, v in
kw.items()])
        updatequery = "UPDATE perceel SET " + updatefields + " WHERE id =
" + id + ";"
        cur.execute(updatequery)
        conn.commit()
        conn.close()

    return data

@jsonify
def statistiek(self, *args, **kw):
    try: # maak connectie met Postgres
        conn = psycopg2.connect("host=localhost port=5432
dbname=kavelruilmonitor user=postgres password=postgres")
        cur = conn.cursor()
        # open cursor
    except:
        print "connectie mislukt" #TODO naar .log schrijven
    if 'lsv' in args:
        cur.execute("select round(sum(ST_Area(geom)) / 10000) AS
lsv_subsidiabel from perceel where nr_gerechtigde IN(SELECT
nr_gerechtigde from perceel where nnp_gerechtigde = false and
status_lsv = true) AND perceel_eindtijd IS NULL")
        results = cur.fetchone()
        conn.close()
        lsv_resterend = 1900 - results[0]

        data = {"success": bool('True'), "data": [{"status":
'Subsidiabel', "oppervlak": int(results[0])}, {"status": 'Resterend',
"oppervlak": int(lsv_resterend)}]}

    return data
    elif 'status' in args:

```

```

        cur.execute("select voortgang_deelnemer, round(sum(oppervlak_ehs)
/ 10000) AS oppervlak_status from perceel where perceel_eindtijd IS
NULL group by voortgang_deelnemer")
        results = cur.fetchall()
        conn.close()
        rows = []
        for r in results:
            if r[0] == 1:
                status = 'Nog geen gesprek'
            elif r[0] == 2:
                status = 'In gesprek'
            elif r[0] == 3:
                status = 'Voorovereenkomst'
            elif r[0] == 4:
                status = 'Def. voorovereenkomst'
            elif r[0] == 5:
                status = 'Wil niet meewerken'

            rows.append(dict([('status', status), ('oppervlak', r[1])]))
        #data = {"success": 'true', "results": 2600, "rows": rows}
        data = {"success": bool('True'), "data": rows}

    return data

```

```

index.exposed = zoekperceel.exposed = muteren.exposed =
beworkperceel.exposed = statistiek.exposed = True

```

```

cherryconfig = ('cherry.py.conf')
cherry.py.quickstart(KrmApp(), config = cherryconfig)

```

Bijlage VI:  
**Javascript Broncode**

## KrmApp.js

```
Ext.onReady(function(){

    //Nederlandse benamingen voor het datumveld
    Ext.Date.monthNames = [
        'januari',
        'februari',
        'maart',
        'april',
        'mei',
        'juni',
        'juli',
        'augustus',
        'september',
        'oktober',
        'november',
        'december'
    ];

    function zoekPerceelXy(x,y){
        PerceelWinForm.getForm().load({
            url: 'http://spc-venemjx-1:9001/zoekperceel/xy',
            params: {
                query: (x + ' ' + y)
            },

            failure: function(form, action) {
                Ext.Msg.alert("Fout", "Perceel niet gevonden!");
            },
            waitMsg: 'Haal gegevens op...'
        });
    }

    var bounds = new OpenLayers.Bounds(
        142636.002, 363569.57,
        160788.68, 377848.226
    );

    var map = new OpenLayers.Map({
        projection: 'EPSG:28992',
        units: 'm',
        maxExtent: bounds,
        maxResolution: 70.90889843749994

    });

    var percelenhuidig = new OpenLayers.Layer.WMS(
        "Percelen Bergeijk",
        "http://spc-venemjx-1:9002/geoserver/wms",
        {
            layers: 'Kavelruilmonitor:perceel_huidig',
            srs: 'EPSG:28992',
            format: 'image/png',
            transparent: true
        }
    );
});
```

```

        }, {
            singleTile: true,
            isBaseLayer: true
        }
    ];

    var luchtfoto = new OpenLayers.Layer.MapServer("Luchtfoto",
        "http://luchtfoto.services.gbo-
provincies.nl/mapserv.cgi?",
        {
            map: "nl.map",
            map_imagetype: "image/jpeg",
            layers: "nl_luchtfoto",
            srs: "EPSG:28992"

        }, {
            //isBaseLayer: false,
            singleTile: true,
            visibility: false,
            buffer: 0
        }
    );

    map.addLayers([percelenhuidig]);
    map.addControl( new OpenLayers.Control.LayerSwitcher() );
    map.addControl( new OpenLayers.Control.WMSGetFeatureInfo() );

    // Click event aanmaken...
    OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
        defaultHandlerOptions: {
            'single': true,
            'double': false,
            'pixelTolerance': 0,
            'stopSingle': false,
            'stopDouble': false
        },

        initialize: function(options) {
            this.handlerOptions = OpenLayers.Util.extend(
                {}, this.defaultHandlerOptions
            );
            OpenLayers.Control.prototype.initialize.apply(
                this, arguments
            );
            this.handler = new OpenLayers.Handler.Click(
                this, {
                    'click': this.trigger
                }, this.handlerOptions
            );
        },

        trigger: function(e) {
            var lonlat = map.getLonLatFromViewPortPx(e.xy);
            Ext.getCmp('PerceelWin').show();
        }
    });

```

```

        zoekPerceelXy(lonlat.lon, lonlat.lat);
    }

});

    var click = new OpenLayers.Control.Click();
    map.addControl(click);
    click.activate();

var openlayersmap = Ext.container.Container({
    region: 'center',
    html: '<div id="map"></div>', //Openlayers hack
    dockedItems: [{
        xtype: 'toolbar',
        dock: 'bottom',
        items: ['Klik hiernaast voor:',
            ' ', {
                xtype: 'button',
                itemId: 'toggleCw',
                text: 'Toon kaart',
                enableToggle: true,
                toggleHandler: function() {
                    //cw.setVisible(!cw.isVisible());

                    map.zoomToExtent(bounds);
                    map.render('map');

                }, {
                    xtype: 'button',
                    itemId: 'toggleLufo',
                    text: 'Luchtfoto',
                    enableToggle: true,
                    toggleHandler: function() {
                        luchtfoto.setVisibility(true);

                    }
                }
            ]
        }
    ], {
        }
    ]
});

var cw;

var PerceelWinForm = new Ext.form.FormPanel({
    padding: 10,
    items: [{
        xtype: 'textfield',
        fieldLabel: 'Objectnummer',
        name: 'nr_akr',
        width: 240
    }, {
        xtype: 'displayfield',
        fieldLabel: 'Belastingplichtige',
        name: 'naam_gerechtigde',
        width: 300
    }
    ]
});

```

```

var PerceelWin = new Ext.Window({
    width: 320,
    height: 100,
    x: 35,
    y: 33,
    id: 'PerceelWin',
    title: 'Perceel Eigenschappen',
    collapsible: true,
    collapsed: false,
    closable: true,
    expandOnShow: false,
    items: [PerceelWinForm]
});

Ext.regModel('ChartItems', {
    fields: [
        {name: 'status', type: 'string'},
        {name: 'oppervlak', type: 'int'}
    ]
});

var lsvStore = new Ext.data.Store({
    model: 'ChartItems',
    proxy: {
        type: 'ajax',
        url : 'http://spc-venemjx-1:9001/statistiek/lsv',
        reader: {
            type: 'json',
            root: 'data'
        }
    },
    autoLoad: true
});

var statusStore = new Ext.data.Store({
    model: 'ChartItems',
    proxy: {
        type: 'ajax',
        url : 'http://spc-venemjx-1:9001/statistiek/status',
        reader: {
            type: 'json',
            root: 'data'
        }
    },
    autoLoad: true
});

var chartStatusHuidig = new Ext.chart.Chart({
    id: 'chartStatusHuidig',
    height: 300,
    anchor: '90%',
    animate: true,
    store: statusStore,
    shadow: true,
    legend: {
        position: 'right'
    }
});

```



```

    },
    insetPadding: 20,
    theme: 'Base:gradients',
    series: [{
        type: 'pie',
        field: 'oppervlak',
        showInLegend: true,
        highlight: {
            segment: {
                margin: 10
            }
        },
        label: {
            field: 'status',
            display: 'none',
            contrast: true,
            font: '10px Arial'
        },
        tips: {
            trackMouse: true,
            width: 160,
            height: 40,
            renderer: function(statusStore,
item) {

                this.setTitle(statusStore.get('status') + ': ' +
statusStore.get('oppervlak') + ' Ha');
            }
        }
    }]

});

var chartStatusHist = new Ext.chart.Chart({
    id: 'chartStatusHist',
    height: 300,
    anchor: '90%',
    animate: true,
    store: statusStore,
    shadow: true,
    legend: {
        position: 'right'
    },
    insetPadding: 20,
    theme: 'Base:gradients',
    series: [{
        type: 'pie',
        field: 'oppervlak',
        showInLegend: true,
        highlight: {
            segment: {
                margin: 10
            }
        },
        label: {
            field: 'status',

```

```

        display: 'none',
        contrast: true,
        font: '10px Arial'
    },
    tips: {
        trackMouse: true,
        width: 160,
        height: 40,
        renderer: function(statusStore,
item) {

        this.setTitle(statusStore.get('status') + ': ' +
statusStore.get('oppervlak') + ' Ha');
    }
    }
}

});

var chartLsvHuidig = new Ext.chart.Chart({
    id: 'chartLsvHuidig',
    height: 300,
    anchor: '90%',
    animate: true,
    store: lsvStore,
    shadow: true,
    legend: {
        position: 'right'
    },
    insetPadding: 20,
    theme: 'Base:gradients',
    series: [{
        type: 'pie',
        field: 'oppervlak',
        showInLegend: true,
        highlight: {
            segment: {
                margin: 10
            }
        },
        label: {
            field: 'status',
            display: 'rotate',
            contrast: true,
            font: '10px Arial'
        },
        tips: {
            trackMouse: true,
            width: 160,
            height: 40,
            renderer: function(lsvStore, item) {

                this.setTitle(lsvStore.get('status')
+ ': ' + lsvStore.get('oppervlak') + ' Ha');
            }
        }
    }

```

```

    }1

    });
    var chartLsvHist = new Ext.chart.Chart({
        id: 'chartLsvHist',
        height: 300,
        anchor: '90%',
        animate: true,
        store: lsvStore,
        shadow: true,
        legend: {
            position: 'right'
        },
        insetPadding: 20,
        theme: 'Base:gradients',
        series: [{
            type: 'pie',
            field: 'oppervlak',
            showInLegend: true,
            highlight: {
                segment: {
                    margin: 10
                }
            },
            label: {
                field: 'status',
                display: 'rotate',
                contrast: true,
                font: '10px Arial'
            },
            tips: {
                trackMouse: true,
                width: 140,
                height: 28,
                renderer: function(lsvStore, item) {
                    this.setTitle(lsvStore.get('status')
+ ': ' + lsvStore.get('oppervlak') + ' Ha');
                }
            }
        }
    ]

    });
    //var chartFinanHuidig
    //var chartFinanHist
    var statsPanel = Ext.create('Ext.panel.Panel', {
        layout: 'column',

        items: [{
            xtype: 'container',
            layout: 'anchor',
            columnWidth: .5,
            items: [{xtype: 'panel',
                layout: 'anchor',
                title: 'Actuele stand',
                items: [{xtype: 'datefield', grow: true, value:
new Date(), format: 'j F Y'}],

```

```

        {xtype: 'container', html:
'<br><center><b>Voortgang per status</b></center>'},
        chartStatusHuidig,
        {xtype: 'container', html:
'<center><b>Voortgang subsidiabele LSV</b></center>'},
        chartLsvHuidig] }}

    },{
        xtype: 'container',
        layout: 'anchor',
        columnWidth: .5,
        items: [{xtype: 'panel',
            layout: 'anchor',
            title: 'Historische stand',
            items: [{xtype: 'datefield', grow: true, value:
new Date('1/1/2011'), format: 'j F Y'},
                {xtype: 'container', html:
'<br><center><b>Voortgang per status</b></center>'},
                chartStatusHist,
                {xtype: 'container', html:
'<center><b>Voortgang subsidiabele LSV</b></center>'},
                chartLsvHist] }}

        ]}
    });

Ext.create('Ext.Viewport', {
    layout: {
        type: 'border',
        padding: 5
    },
    defaults: {
        split: true
    },
    items: [{
        region: 'north',
        collapsible: false,
        //title: 'North',
        split: false,
        height: 85,
        html: ''
    },{
        region: 'west',
        id: 'legendaPanel',
        collapsible: true,
        floatable: true,
        title: 'Legenda en Lagen',
        split: true,
        width: '20%',
        html: ''
    },{
        region: 'center',
        layout: 'border',
        border: true,
        xtype: 'tabpanel',

```

```

activeTab: 0,
defaults :{
    //bodyPadding: 10
},
items: [{
    title: 'Kaartbeeld',
    html: '<div id="map"></div>', //Openlayers hack
    dockedItems: [{
        xtype: 'toolbar',
        dock: 'bottom',
        items: ['Klik hiernaast voor:',
            ' ', {
                xtype: 'button',
                itemId: 'toggleCw',
                text: 'Toon kaart',
                enableToggle: true,
                toggleHandler: function() {
                    //cw.setVisible(!cw.isVisible());

                    map.zoomToExtent(bounds);
                    map.render('map');
                }
            }
        ]
    }],
    }, {
        title: 'Statistieken',
        padding: 10,
        items: [statsPanel]
    }, {
        title: 'Muteren',
        disabled: true
    }
    ]
    }];
});

```

## Muteren.js

```
Ext.onReady(function() {

    Ext.regModel('VoortgangDeelnr', {
        fields: [
            {type: 'int', name: 'code'},
            {type: 'string', name: 'status'}
        ]
    });

    // data voor Voortgang status
    var voortgangcodes = [
        {"code": "1", "status": "Nog geen gesprek"},
        {"code": "2", "status": "In gesprek"},
        {"code": "3", "status": "Voorovereenkomst"},
        {"code": "4", "status": "Definitieve voorovereenkomst"},
        {"code": "5", "status": "Wil niet meewerken"}
    ]

    // Datastore voor Voortgang Deelnemer combobox
    var deelnrstore = new Ext.data.Store({
        model: 'VoortgangDeelnr',
        data: voortgangcodes
    });

    // Funtie om een Query te doen naar de server
    function zoekPerceel(){
        form.getForm().load({
            url: 'http://spc-venemjx-
1:9001/zoekperceel/objectnummer',
            params: {
                query: (Ext.getCmp('zkveld1').getValue())
            },

            failure: function(form, action) {
                Ext.Msg.alert("Fout", "Perceel niet gevonden!");
            }
        });
    }

    var fieldset1 = {
        xtype: 'fieldset',
        title: 'Zoek perceel',
        width: 300,
        height: 170,
        x: 30,
        y: 30,
        items: [{
            xtype: 'textfield',
            submitValue: false,
            hideLabel: true,
            name: 'zoekveld1',
```

```

        width: 200,
        id: 'zkveld1',
        emptyText: 'Vul objectnummer in...'
    }, {
        xtype: 'button',
        text: 'Zoek!',
        handler: zoekPerceel
    }, {
        xtype: 'button',
        text: 'Perceel bijwerken',
        handler: function() {
var form = this.up('form').getForm();
        form.submit({
            success: function(form, action) {
                Ext.Msg.alert('Info', 'Perceel bijgewerkt!');
            },
            failure: function(form, action) {
                Ext.Msg.alert('Fout', 'Bewerken mislukt!');
            },
            waitMsg: 'verzenden...',
            timeout: 10
        });
    }
}
}

```

```

var imghtml = ''

```

```

var fieldset2 = {
    // Perceeleigenschappen
    xtype: 'fieldset',
    title: 'Perceeleigenschappen',
    collapsible: false,
    x: 350,
    y: 30,
    width: 350,
    height: 170,
    defaultType: 'textfield',
    items: [{
        name: 'objectnummer',
        disabled: true,
        fieldLabel: 'Objectnummer'
    }, {
        name: 'naam_gerechtigde',
        disabled: true,
        fieldLabel: 'Belastingplichtige'
    }, {
        name: 'oppervlakte',
        disabled: true,
        fieldLabel: 'Oppervlakte'
    }, {

```

```

        name: 'oppervlakte_ehs',
        disabled: true,
        fieldLabel: 'Oppervlakte EHS'
    }
}

var fieldset3 = {
    // Deelnemer
    xtype: 'fieldset',
    title: 'Deelnemer',
    layout: 'anchor',
    defaults: {
        anchor: '100%'
    },
    x: 350,
    y: 200,
    width: 165,
    height: 110,
    items: [{
        xtype: 'radiogroup',
        hideLabel: true,
        items: [
            {boxLabel: 'Ja', name: 'status_deelnemer',
inputValue: 'True'},
            {boxLabel: 'Nee', name:
'status_deelnemer', inputValue: 'False'}
        ],
        }, {
        xtype: 'numberfield',
        name: 'ruilnummer',
        fieldLabel: 'Ruilnummer',
        labelWidth: 65,
        labelAlign: 'left',
        anchor: '90%'
    }
    ]
}

var fieldset4 = {
    // Taxatie
    xtype: 'fieldset',
    title: 'Taxatie',
    layout: 'anchor',
    defaults: {
        anchor: '100%'
    },
    x: 350,
    y: 440,
    width: 350,
    height: 110,
    items: [{
        xtype: 'checkboxgroup',
        hideLabel: true,
        columns: 2,
        items: [
            {boxLabel: 'Getaxeerd', name: 'status_taxatie',
inputValue: 'True', uncheckedValue: 'False'},

```



```

        {boxLabel: 'Deelnemer akkoord', name:
'deelnr_taxatie', inputValue: 'True', uncheckedValue: 'False'}
    ]
    }]
}

var fieldset6 = {
    // Indicatoren
    xtype:'fieldset',
    title: 'Indicatoren',
    layout: 'anchor',
    defaults: {
        anchor: '100%'
    },
    x: 30,
    y: 320,
    width: 300,
    height: 280,
    items: [{
        xtype: 'checkboxgroup',
        hideLabel: true,
        columns: 1,
        items: [
            {boxLabel: 'Beekherstel', name:
'ind_beekeerstel', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Gepland Bosgebied', name:
'ind_gepland_bos_natuur', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Natte Natuurparels', name:
'ind_natuurparels', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'EVZ Indicatief', name:
'ind_evz_indicatief', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Zoekgebied EHS', name:
'ind_zoekgebied_ehs', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Natuurontwikkelgebied', name:
'ind_natuurontw_gebied', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Beheersgebied', name:
'ind_beheersgebied', inputValue: 'True', uncheckedValue: 'False'},
            {boxLabel: 'Natuurontwikkelgebied +', name:
'ind_natuurontw_gebied_plus', inputValue: 'True', uncheckedValue:
'False'},
            {boxLabel: 'Bestaand Bosgebied', name:
'ind_bestaand_bos_natuur', inputValue: 'True', uncheckedValue:
'False'},
            {boxLabel: 'Reservaat', name:
'ind_reservaatgebied', inputValue: 'True', uncheckedValue: 'False'}
        ]
    }]
}

var fieldset7 = {
    // VoortgangDeelnemer
    xtype:'fieldset',
    title: 'Status Deelnemer',
    x: 350,
    y: 320,
    width: 350,
    height: 100,

```

```

        items :[
            new Ext.form.ComboBox({
                fieldLabel: 'Status',
                name: 'voortgang_deelnemer',
                valueField: 'code',
                displayField: 'status',
                width: 230,
                labelWidth: 50,
                store: deelnrstore,
                queryMode: 'local',
                typeAhead: true
            })
        ]
    }

    var fieldset8 = {
        // BBL
        xtype:'fieldset',
        title: 'BBL Rechtstreekse Aankopen',
        layout: 'anchor',
        defaults: {
            anchor: '100%'
        },
        x: 720,
        y: 200,
        width: 350,
        height: 110,
        items: [{
            xtype: 'radiogroup',
            hideLabel: true,
            columns: 2,
            items :[
                {boxLabel: 'Getipt', name: 'bbl_getipt',
inputValue: 'True'},
                {boxLabel: 'Niet getipt', name: 'bbl_getipt',
inputValue: 'False'},
                {boxLabel: 'Aankoop', name: 'bbl_aankoop',
inputValue: 'True'},
                {boxLabel: 'Geen aankoop', name: 'bbl_aankoop',
inputValue: 'False'}
            ]
        }]
    }

    var fieldset10 = {
        // Aktepassering
        xtype:'fieldset',
        title: 'Aktepassering',
        layout: 'anchor',
        defaults: {
            anchor: '100%'
        },
        x: 535,
        y: 200,
        width: 165,
        height: 55,
        items: [{

```

```

        xtype: 'radiogroup',
        hideLabel: true,
        columns: 2,
        items :[
            {boxLabel: 'Ja', name: 'status_aktepassering',
inputValue: 'True'},
            {boxLabel: 'Nee', name: 'status_aktepassering',
inputValue: 'False'}
        ]
    }

    var fieldset11 = {
        // Onderzoek
        xtype:'fieldset',
        title: 'Onderzoek',
        layout: 'anchor',
        defaults: {
            labelWidth: 60,
            anchor: '100%'
        },
        x: 720,
        y: 30,
        width: 350,
        height: 170,
        items: [{
            xtype: 'checkboxgroup',
            //vertical: true,
            columns: 3,
            fieldLabel: 'type:',
            items :[
                {xtype: 'checkbox', html: 'Historisch', cls:'x-
form-check-group-label'},
                {xtype: 'checkbox', html: 'Verkennend', cls:'x-
form-check-group-label'},
                {xtype: 'checkbox', html: 'Aanvullend', cls:'x-
form-check-group-label'},
                {xtype: 'checkbox', name:
'status_hist_onderzoek', inputValue: 'True', uncheckedValue: 'False'},
                {xtype: 'checkbox', name:
'status_verk_onderzoek', inputValue: 'True', uncheckedValue: 'False'},
                {xtype: 'checkbox', name:
'status_aanv_onderzoek', inputValue: 'True', uncheckedValue: 'False'}
            ]
        },{
            xtype: 'checkboxgroup',
            vertical: true,
            columns: 3,
            fieldLabel: 'Afgerond',
            items: [
                {boxLabel: 'Ja', name: 'result_hist_onderzoek',
inputValue: 'True', uncheckedValue: 'False'},
                {boxLabel: 'Ja', name: 'result_verk_onderzoek',
inputValue: 'True', uncheckedValue: 'False'},
                {boxLabel: 'Ja', name: 'result_aanv_onderzoek',
inputValue: 'True', uncheckedValue: 'False'}
            ]
        }
    ]
}

```

```

        }, {
            xtype: 'checkbox',
            name: 'onderzoek_nee',
            boxLabel: 'Afgezien van onderzoek',
            inputValue: 'True',
            uncheckedValue: 'False'
        }
    ]
}

var fieldset12 = {
    // Opmerkingen TODO: Hoogte instellen...
    xtype: 'fieldset',
    title: 'Opmerkingen',
    layout: 'anchor',
    defaults: {
        anchor: '100%'
    },
    x: 720,
    y: 320,
    width: 350,
    height: 280,
    items: [{
        xtype: 'textareafield',
        name: 'opmerkingen',
        hideLabel: true,
        grow: true,
        height: '100%'
    }]
}

var fieldset13 = {
    // LSV
    xtype: 'fieldset',
    title: 'LSV',
    layout: 'anchor',
    defaults: {
        anchor: '100%'
    },
    x: 535,
    y: 255,
    width: 165,
    height: 55,
    items: [{
        xtype: 'radiogroup',
        hideLabel: true,
        columns: 2,
        items :[
            {boxLabel: 'Ja', name: 'status_lsv', inputValue:
'True'},
            {boxLabel: 'Nee', name: 'status_lsv',
inputValue: 'False'}
        ]
    }]
}

var recordid = {
    xtype: 'hiddenfield',

```

```

        name: 'id'
    }

    var form = new Ext.form.FormPanel({
        title: 'Mutatieformulier',
        labelWidth: 75, // label settings here cascade unless overridden
        url: 'http://spc-venemjx-1:9001/bewerkperceel/',
        frame: true,
        width: 1120,
        height: 670,
        renderTo: document.body,
        clearInvalid: Ext.emptyFn, // Hack om foutmelding te voorkomen:
        "a.ClearInvalid is not a function"
        layout: 'absolute', // Laten we het eerst simpel houden...
        items: [
            fieldset1,
            fieldset2,
            fieldset3,
            fieldset4,
            fieldset6,
            fieldset7,
            fieldset8,
            fieldset10,
            fieldset11,
            fieldset12,
            fieldset13,
            recordid
        ]
    });
});

```