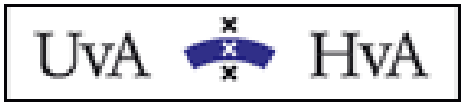
Geodan Holding b.v.
President Kennedylaan 1
1079 MB Amsterdam (NL)
Tel. +31 (0)20 - 5711 311
Fax +31 (0)20 - 5711 333
E-mail info@geodan.nl
Website www.geodan.nl

FleetOnline

Afstudeerscriptie Master Software Engineering

In opdracht van
Geodan Mobile Solutions

Naam Student: Erik-Jan Kok
Datum: 21 juni 2004
Opleidingen: Universiteit van Amsterdam
In samenwerking met: Hogeschool van Amsterdam
Vrije Universiteit
Afstudeerdocent: Alban Ponse
Stagebegeleider: Barend Gehrels



vrije Universiteit amsterdam

Samenvatting

Dankzij de mogelijkheden van positiebepaling wordt een mobiele toestel een gericht zoekinstrument. De gebruiker heeft toegang tot praktische informatie op basis van de plaats waar hij/zij zich bevindt. Diensten ontwikkeld op basis van de LBS-technologie worden de lokatiegebonden diensten genoemd. De geografische positie op wordt gebruikt om interessante diensten aan te bieden. Vodafoon biedt een dienst aan die het mogelijk maakt om mensen via hun mobiele telefoon te lokaliseren. Hierbij wordt gebruik gemaakt van de coördinaten van de zendmast waar de telefoon is aangemeld. Tijdens dit project worden de mogelijkheden onderzocht om deze dienst op een mobiele telefoon beschikbaar te maken. Het hele traject zal onderzocht worden. Van het vaststellen van de functionaliteit tot het ontwikkelen van een prototype.

Inhoudsopgave

1 Achtergrond en onderzoeksvraag

1.1 Wat is Geodan

Geodan Mobile Solutions, opgericht in 1999, is uitgegroeid tot een leverancier van Location-Based Services aan de zakelijke en consumenten markt. Location-Based Services zijn diensten die gebaseerd zijn op de positie van een mobiele gebruiker en diens omgeving. Daarbij kan gedacht worden aan diensten als routenavigatie en intelligente zoeksystemen (bijvoorbeeld de dichtstbijzijnde vrije parkeerplaats). De verwachting is dat Location-Based Services vanaf 2007 de meest gebruikte mobiele diensten gaan worden. [5]

Bij het benaderen van mobiele gebruikers is het van belang dat er aan twee fundamentele behoeften wordt voldaan: het overal en altijd aanbieden van informatie ***en*** het leveren van gerichte en relevante informatie. De mobiele gebruiker moet overal over die informatie kunnen beschikken die precies op zijn omstandigheden is toegesneden (de plaats waar hij zich bevindt, zijn activiteit, het tijdstip en zijn voorkeuren). **Locatie** en **personificatie** vormen daarom de sleutelbegrippen in dit proces. Geodan Mobile Solutions combineert deze twee sleutelbegrippen in de locatiegebonden diensten die zij de mobiele gebruikers biedt.

Geodan Mobile Solutions, onderdeel van de Geodan-groep

Geodan Mobile Solutions kan zich, als onderdeel van de Geodan-groep, beroepen op:

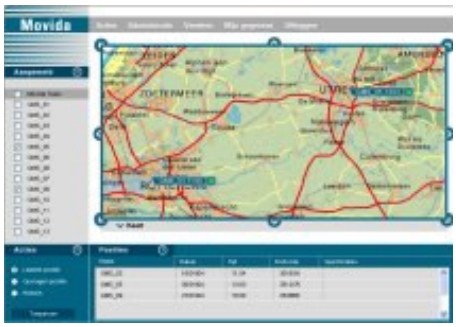
- Meer dan 15 jaar ervaring in locatietechnologie;
- Kennis van de geografische informatiemarkt (Geodan heeft ruim 300 klanten in Europa);
- Ervaring in het bieden van toepassingen met toegevoegde waarde binnen de zakelijke markt;
- Kennis en ervaring in het toepassen van OpenGIS-standaarden.

Door locatiegebonden informatie te integreren met e-commerce en ICT, speelt Geodan Mobile Solutions een rol in de snelgroeiende, internationale markt voor mobiele communicatie en Location-Based Services.

1.2 Wat is Movida

Movida: een complete toolbox voor mobiele locatiediensten, lokaliseert de positie van mobiele telefoons voor telecom-operators, biedt nauwkeurige GPS-locaties en introduceert vernieuwende Wi-Fi en RFID-positionering binnen gebouwen. Mobiele professionals kunnen terwijl ze onderweg zijn gebruik maken van Movida om op basis van hun locatie toegang te hebben tot informatie. Organisaties kunnen het gebruik van hun mobiele resources optimaliseren en hun efficiency verhogen. Het bedrijfsleven kan individuele goederen van begin tot eind volgen, van productielijn tot plaatsing in het schap.

Dankzij de toenemende beschikbaarheid van locatietechnologie biedt Movida één enkel toegangspunt voor locatiediensten binnen en buiten gebouwen. Movida is beschikbaar als:



- Location Center: een geïntegreerde oplossing voor het lokaliseren van gebruikers of objecten op basis van de Movida Toolbox;
- Toolbox: een geïntegreerd portfolio van modules voor weergave op kaarten, het zoeken van locaties, verzenden van berichten, routering en geocodering;
- Location Server: één toegangspunt naar de locatie van meerdere telecom-operators, GPS, Wi-Fi en andere locatiemethoden.

Privacymanagement is één van de centrale elementen van de Movida-architectuur. Deze omvat eenduidige opt-in en opt-out mechanismen voor persoonlijke locatie, de controle over de persoonlijke informatie die door het platform wordt verwerkt evenals maatregelen voor nakoming van de strikte privacybepalingen van de Europese regelgeving.

1.3 De opdracht

Vodafoon biedt een dienst aan die het mogelijk maakt om mensen via hun mobiele telefoon te lokaliseren. Hierbij wordt gebruik gemaakt van de coördinaten van de zendmast waar de telefoon is aangemeld. De nieuwste implementatie van deze dienst heet FleetOnline. Naast internetapplicaties wil Geodan ook de mobiele markt betreden. Tijdens deze stage wordt er geprobeerd om deze dienst ook aan te bieden via een mobiele applicatie. De functionaliteit van FleetOnline wordt als uitgangspunt genomen. Er wordt gekeken welke onderdelen er voor mobiele implementatie van toepassing zijn.
Begane grond

Het eindresultaat wordt een prototype FleetOnline met de volgende kenmerken:

- Inloggen op de FleetOnline server
- Opvragen van gebruikers
- Opvragen van de locatie van deze gebruikers
- Tonen van een kaart met de posities

De volgende onderdelen moeten gedurende de stage onderzocht worden:

- de ontwikkelomgeving voor Java MicroEdition J2ME [6,7,8,9,10];
- het besturingssysteem van het mobiele apparaat;
- hoe wordt een applicatie ontwikkeld met een beperkte User-Interface; [13]
- de beste architectuur voor de applicatie; [12]
    - welke onderdelen komen op de server;
    - welke onderdelen komen op de client;
    - hoe verloopt de client-server communicatie;
    - welke beperkingen brengt GPRS (General Packet Radio Service) met zich mee; De datasnelheid over het huidige GSM-netwerk via GPRS bedraagt maximaal 115 Kb/s. Een standaard internetverbinding is tegenwoordig 786 Kb/s. [27]

2 Plan van aanpak

In het plan van aanpak wordt er uitgewerkt hoe de resultaten van dit project tot stand komen. Het eindresultaat is een prototype.Er wordt rekening gehouden met de korte tijdsbestek (afstudeerstage van drie maanden) van dit project. De functionaliteit die uiteindelijk wordt geïmplementeerd hangt af van de voortgang. Er wordt ook rekening gehouden met het feit dat er bij aanvang van het project weinig ervaring is met de gebruikte technieken.

2.1 Analyse van FleetOnline

Om een goed beeld te krijgen van het project wordt FleetOnline als eerste geanalyseerd. Er wordt gekeken naar de functionaliteit, lay-out, navigatiestructuur. Binnen Geodan was er voor dit doel geen geschikte documentatie van FleetOnline aanwezig. Na het in kaart brengen van deze onderdelen wordt er bepaald welke functionaliteit er in de mobiele applicatie terug komt. FleetOnline is ontwikkeld als internetapplicatie. Het is bedoeld om deze applicatie op een normale PC te draaien. De lay-out en benodigde dataverkeer zijn hierop afgestemd. Er moet dus bepaald worden welke onderdelen geschikt zijn om te implementeren in een mobiele applicatie. De criteria voor de besluitvorming zijn:

- Valt het onder kernfunctionaliteit of extra functionaliteit;
- Hoe vaak wordt de functie gebruikt;
- Is de benodigde capaciteit op het mobiele apparaat aanwezig;
    - Schermindeling / grootte;
    - Hoeveelheid dataverkeer;
- Benodigde dataverkeer;

Het resultaat van deze stap is een overzicht van alle deelsystemen inclusief subsystemen en de functionaliteit.

2.2 De navigatiestructuur

De functionaliteit voor de mobiele applicatie staat nu vast. De volgende stap is het bepalen van de navigatiestructuur. Tijdens het ontwerp moet er rekening worden gehouden met de beperkte navigatie mogelijkheden op een mobiel apparaat. Per scherm zijn er maar twee knoppen beschikbaar. Een knop heeft de standaard functionaliteit “terug naar het vorige scherm”. Aan de andere knop kan men een of meer functies toekennen. Wanneer een knop met meerdere functies wordt geselecteerd krijgt de gebruiker een submenu te zien waar hij de juiste functie uit kan kiezen. Denk hier bij aan een SMS bericht. Bij een knop staat er “back” en bij de andere knop staat “options”. Wanneer de “options” wordt gekozen komt de gebruiker in een submenu met de opties “read”, “delete”, “reply”. Dit is wel een extra stap die genomen moet worden. Het toekennen van meerdere functies aan een knop moet vermeden worden. Het resultaat van deze stap is een flow-diagram.

2.3 Implementatie op de hardware

Er is nog niet bekend op welke hardware deze applicatie beschikbaar moet zijn. De hardware waarop de applicatie op draait wordt gedurende dit project worden bepaald. De opzet is om een applicatie te maken die meerdere machines kan draaien. De onderzoekscriteria zijn:

- compatibiliteit;
- performance;
- aanwezigheid van internetconnectie via GPRS/WAP/UMTS;
- user-Interface mogelijkheden;
- besturingssysteem;
    - pocket PC (Microsoft);
    - palm OS 5.0 (PalmOne) ;
    - symbian 6.1;

2.4 Klassediagram

Naast de navigatie structuur is er ook een Applicatie structuur. Dit wordt gedaan aan de hand van een klassediagram. Hierbij ligt de focus op de statische structuur van het software systeem. Gedragsaspecten zoals navigatiestructuur e.d. worden buitenbeschouwing gelaten. Het basisconcept wordt m.b.v. het klassediagram beschreven. Hierbij zullen objectgeoriënteerde begrippen als klasse, object, overerving, enz terugkomen Dit proces geschiedt volgens de stappen die in hoofdstuk 5 van het boek praktisch UML [12] zijn beschreven.

2.5 Client-Server communicatie

De data moet via een internetconnectie van een server worden opgehaald. Het gaat hier om de kaartdata en tekstberichten van berichten en gebruikers. Er worden beslissingen genomen het gebied van netwerkprotocol en de manier waarop tekstberichten over het netwerk worden verstuurd. De volgende onderzoekscriteria worden gebruikt bij het nemen van deze beslissing.

- wordt het ondersteund in J2ME;
- bandbreedteconsumptie;
- processorbelasting;
- veiligheid;
- onderhoudbaarheid van de code.;
- ervaring binnen Geodan;

2.6 Ontwikkelomgeving

Dit is het eerste project binnen Geodan waar J2ME [referenties naar J2ME boeken] wordt gebruikt. Het is niet bekend welke ontwikkelomgeving het best geschikt is voor dit platform. Aan het begin van het project wordt er onderzocht welke mogelijkheden er zijn. De onderzoekscriteria voor de geschiktheid zijn:

- ondersteuning van meerdere besturingssystemen, zodat de implementatie op de hardware los staat van de ontwikkelomgeving;
- integratie van verschillende emulatoren en externe apparaten;
- een goede debugomgeving;
- stabiel programma, dus geen bèta programma's;

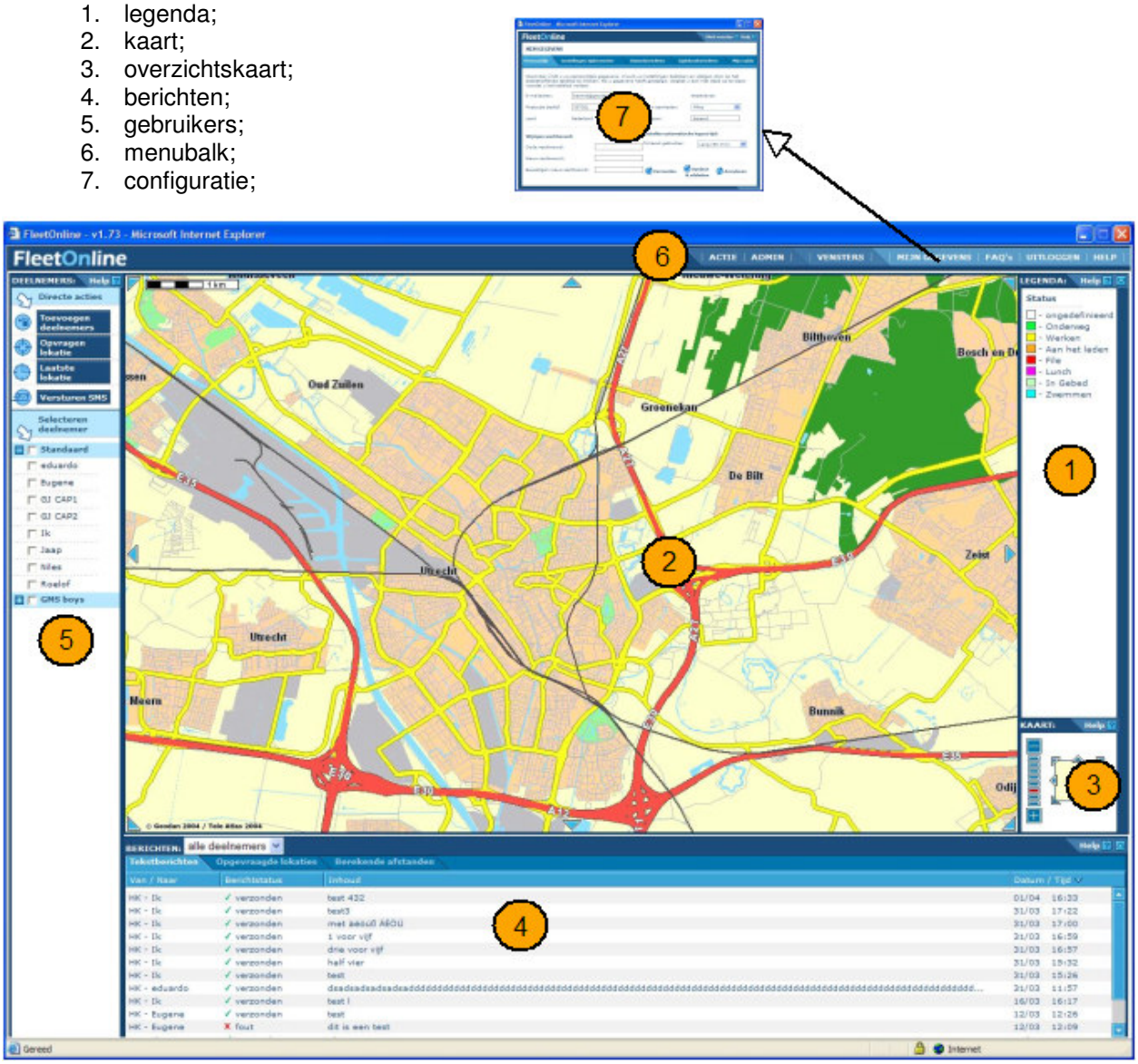
3 Uitvoering

Om dit project tot een goed einde te brengen moeten technieken zoals XML, Java MicroEdition en Servlets beheerst worden. Bij aanvang van dit project had ik een beperkte kennis van deze technieken. De eerste vijf weken van dit project heb ik twee dagen per week besteed aan het inlezen en inwerken in deze technieken. Stap 2.6 en 2.3 uit het plan van aanpak zijn gelijktijdig met 2.1 en 2.2 en 2.4 uitgevoerd. Op deze manier is er genoeg kennis opgedaan over de technieken zodat deze toegepast kunnen worden op het moment dat het nodig is.

3.1 Analyse huidige systeem.

In de analyse is er naar voren gekomen dat FleetOnline uit zeven deelsystemen / componenten bestaat.

1. legenda;
2. kaart;
3. overzichtskaart;
4. berichten;
5. gebruikers;
6. menubalk;
7. configuratie;



Ieder deelsysteem bestaat uit een aantal subsystemen. Na deze onderverdeling wordt er bepaald welke functionaliteit er in de mobiele applicatie terug komt.

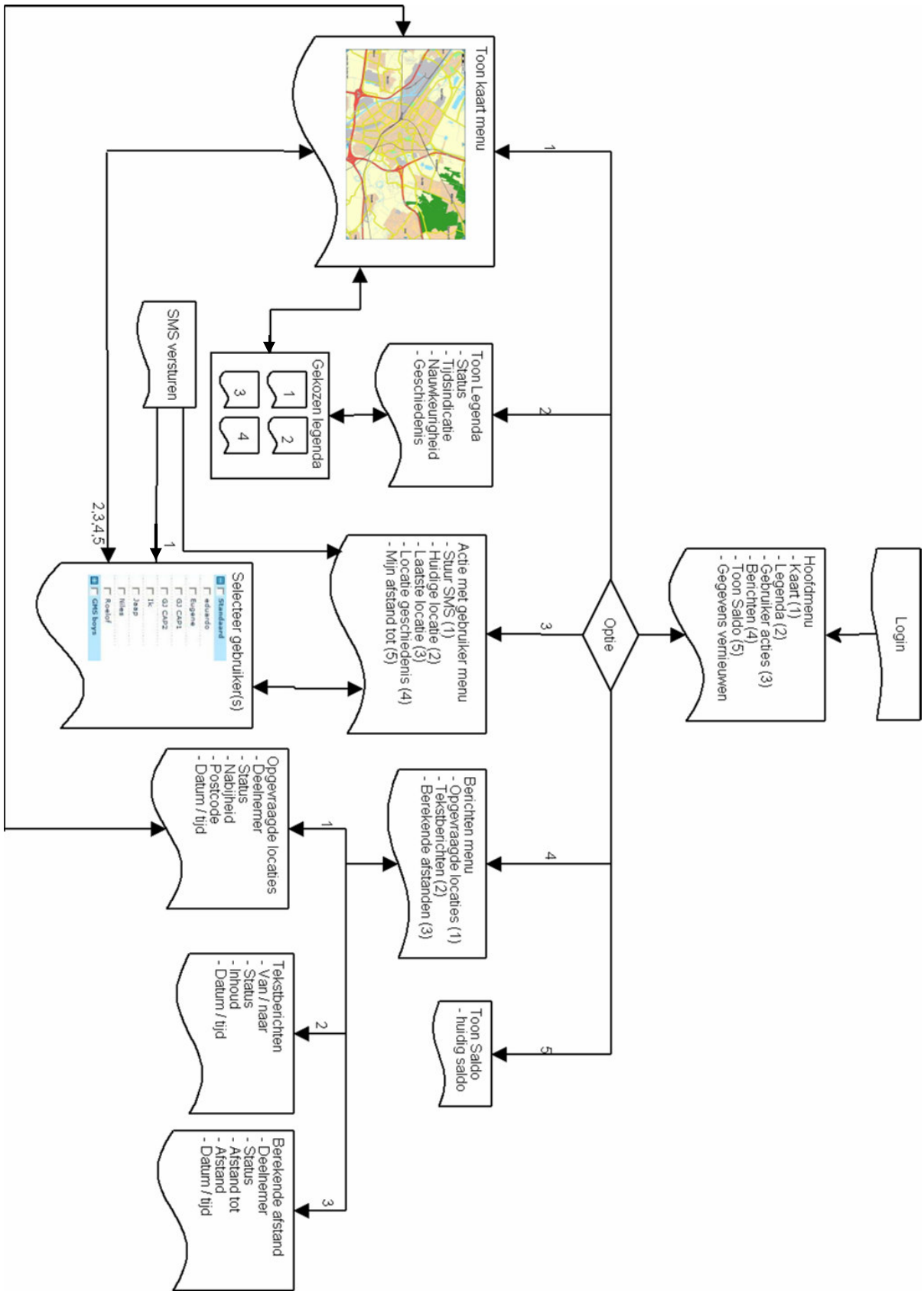
Een aantal voorbeelden van functionaliteit die wel en niet terug komen zijn:

- Bij een symbolen op een kaart hoort een legenda. Dus ook bij de mobiele applicatie zal er voor iedere aanduiding op de kaart een item in de legenda komen.
- De overzichtskaart wordt weggelaten. Hier wordt aangegeven op welk deel van de kaart de gebruiker is ingezoomd. (afbeelding 1) Dit venster staat altijd naast de originele kaart. Bij een mobiele telefoon is er geen ruimte voor deze kaart en het zou extra data verkeer genereren.
- Er zijn drie soorten berichten die uitgewisseld kunnen worden. Deze worden allemaal in de mobiele applicatie meegenomen.
    - SMS-berichten, uitwisseling van informatie
    - Locatieberichten, Een persoon bevind zich op een bepaalde plaats op een bepaalde tijd.
    - Afstandsbericht. Welke afstand zit er tussen persoon A en persoon B.
- Algemene instellingen zoals n.a.w. gegevens, status gegevens worden in de praktijk weinig gebruikt. Deze worden niet meegenomen.

Afbeelding 1

3.2 Navigatie structuur

Tijdens deze stap word de functionaliteit in een navigatiestructuur gegoten [13]. Een knop heeft standaard de functionaliteit “terug naar vorig scherm” De andere knop heeft wisselende functionaliteit. Gedurende het project was dit diagram aan verandering onderhevig. In eerste instantie was er een navigatie mogelijk tussen “Toon Legenda” en “Toon kaart menu”. In de praktijk bleek dat een gebruiker pas de kaart wil zien nadat een legenda is geselecteerd. Nu staat er een pijl van “Gekozen legenda” naar “Toon kaart menu”. In afbeelding 2 is de uiteindelijke navigatiestructuur te zien.

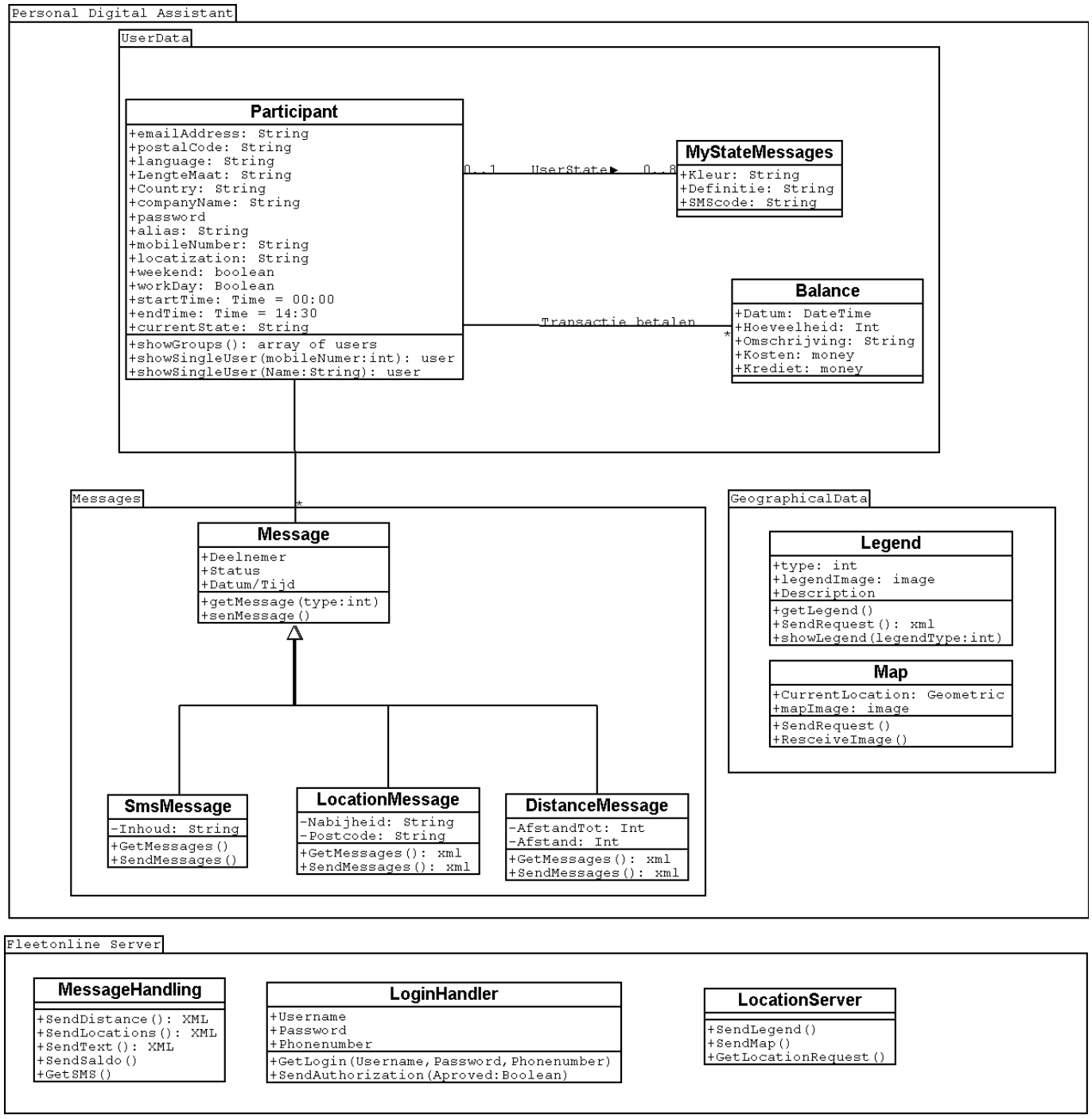

Toon kaart menu
SMS versturen
Gekozen legenda
1
2
3
4
Toon Legenda
- Status
- Tijdsindicatie
- Nauwkeurigheid
- Geschiedenis
2,3,4,5
1
Selecteer gebruiker(s)
Actie met gebruikermenu
- Stuur SMS (1)
- Huidige locatie (2)
- Laatste locatie (3)
- Locatie geschiedenis (4)
- Mijn afstand tot (5)
3
Optie
Hoofdmenu
- Kaart (1)
- Legenda (2)
- Gebruiker acties (3)
- Berichten (4)
- Toon Saldo (5)
- Gegevens vernieuwen
Login
Opgevraagde locaties
- Deelnemer
- Status
- Nabijheid
- Postcode
- Datum / tijd
1
Berichten menu
- Opgevraagde locaties (1)
- Tekstberichten (2)
- Berekende afstanden (3)
4
Tekstberichten
- Van / naar
- Status
- Inhoud
- Datum / tijd
2
Toon Saldo
- huidig saldo
5
Berekende afstand
- Deelnemer
- Status
- Afstand tot
- Afstand
- Datum / tijd
3

3.3 Klassediagram

De methode uit het boek Praktisch UML [12] is gebruikt om dit klassediagram te maken. Deze methode bestaat uit negen stappen

1. Identificatie van alle mogelijke kandidaat-klassen
2. Selecteren van klassen uit de lijst van kandidaten
3. Maak een modeldictionary
4. Identificeer associaties
5. Identificeer attributen
6. Identificeer operaties
7. Generaliseer m.b.v. overerving
8. OCL constraints maken
9. Groepeer klassen in packages

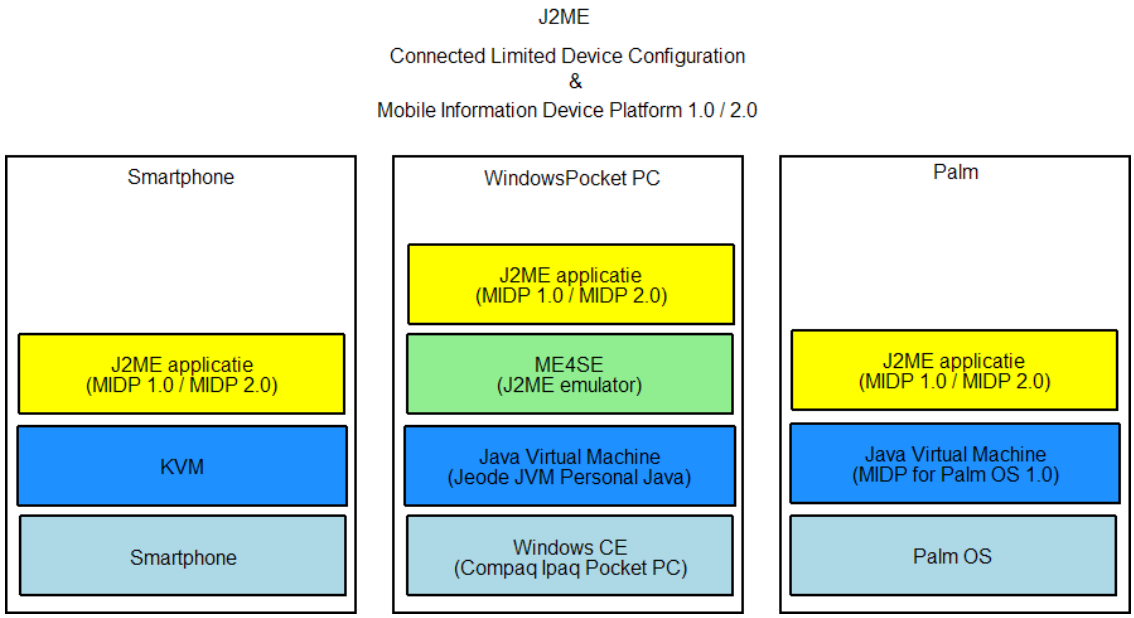
Het eindresultaat is een klassediagram. Dit diagram is gebruikt als blauwdruk voor de statische structuur van de applicatie.


Personal Digital Assistant
UserData
Participant
+emailAddress: String
+postalCode: String
+language: String
+LengteMaat: String
+Country: String
+companyName: String
+password
+alias: String
+mobileNumber: String
+locatization: String
+weekend: boolean
+workDay: Boolean
+startTime: Time = 00:00
+endTime: Time = 14:30
+currentState: String
+showGroups(): array of users
+showSingleUser(mobileNumer:int): user
+showSingleUser(Name:String): user
0..1
UserState
0..8
MyStateMessages
+Kleur: String
+Definitie: String
+SMScode: String
Transactie betalen
*
Balance
+Datum: DateTime
+Hoeveelheid: Int
+Omschrijving: String
+Kosten: money
+Krediet: money
Messages
*
Message
+Deelnemer
+Status
+Datum/Tijd
+getMessage(type:int)
+senMessage()
SmsMessage
-Inhoud: String
+GetMessages()
+SendMessages()
LocationMessage
-Nabijheid: String
-Postcode: String
+GetMessages(): xml
+SendMessages(): xml
DistanceMessage
-AfstandTot: Int
-Afstand: Int
+GetMessages(): xml
+SendMessages(): xml
GeographicalData
Legend
+type: int
+legendImage: image
+Description
+getLegend()
+SendRequest(): xml
+showLegend(legendType:int)
Map
+CurrentLocation: Geometric
+mapImage: image
+SendRequest()
+ResceiveImage()
Fleetonline Server
MessageHandling
+SendDistance(): XML
+SendLocations(): XML
+SendText(): XML
+SendSaldo()
+GetSMS()
LoginHandler
+Username
+Password
+Phonenumber
+GetLogin(Username, Password, Phonenumber)
+SendAuthorization(Aproved:Boolean)
LocationServer
+SendLegend()
+SendMap()
+GetLocationRequest()

3.4 Deployment op het apparaat

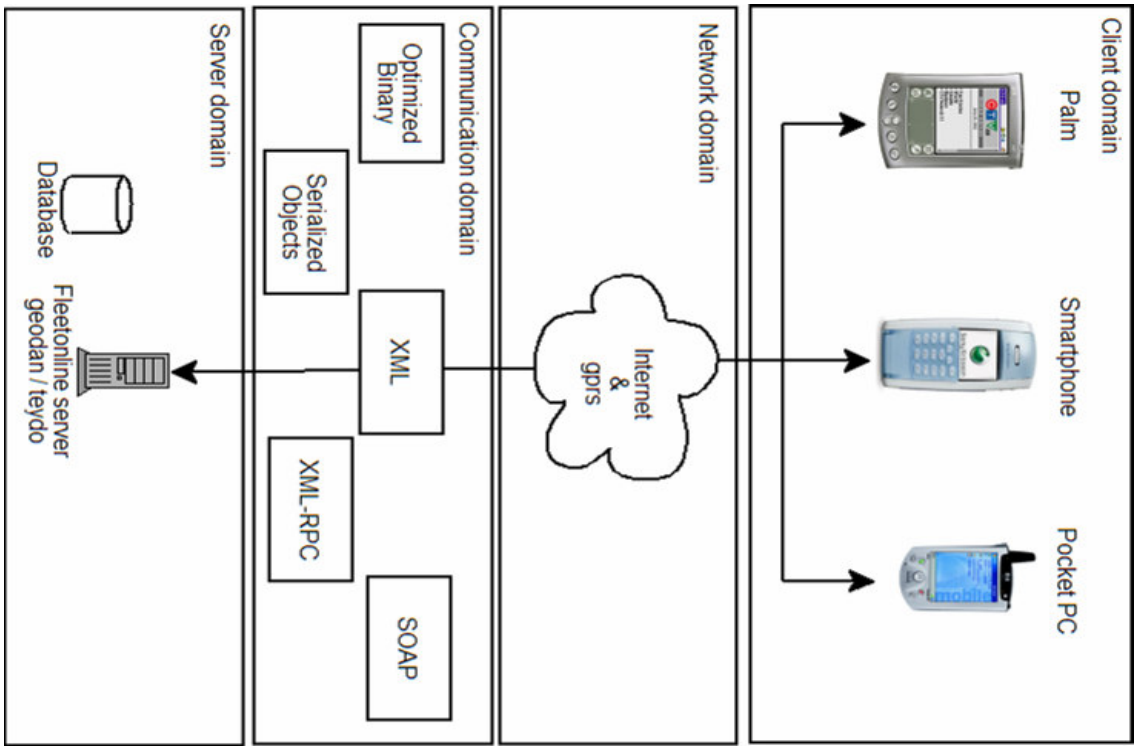
J2ME ondersteuning is niet standaard op ieder apparaat. Vooral bij Microsoft producten is de ondersteuning van Java niet optimaal. Er is op drie verschillende platformen onderzocht welke software er nodig is om de Java virtual Machine voor J2ME te installeren.

- **Smartphone**
  De smartphones biedt standaard ondersteuning voor J2ME. Het worden ook wel Java enabled mobile devices genoemd.
- **Windows CE (Microsoft Pocket PC 2003)**
  Om J2ME mogelijk te maken op Windows CE moeten er een J2SE Virtual Machine worden geïnstalleerd. Na vergelijking [28, 29] is de J2SE VM Jeode uitgekozen. Ten eerste omdat Geodan een ander project met deze JVM succesvol heeft gebruikt.Daarop moet de emulator ME4SE worden geïnstalleerd. [30]
- **Palm OS 5.0**
  Een standaard J2ME virtual Machine die voor alle Palm OS versies J2ME mogelijk maakt. [31]



3.5 Client-server communicatie

De tekst- en kaart informatie op de server moet beschikbaar worden voor de mobiele applicatie. Het onderzoek naar de beste methode is tweeledig. Aan de ene kant is er het transportmiddel / netwerkprotocol en aan de andere kant is er de opmaak van het bericht.



3.5.1 Netwerk protocol

3.5.1.1 HTTP connectie

HTTP is een vraag-antwoord protocol. De parameters van de vraag dienen als eerst te worden verzonden waarna een actie volgt. Dit type connectie heeft die stadia waar in het kan verkeren.

- setup;
  Het instellen van de parameters.
- connected;
  Het parameters zijn verstuurd en het antwoord komt binnen.
- closed;
  Het antwoord is ontvangen, dit kan het verwachte resultaat zijn of een foutmelding, en de verbinding wordt afgesloten

3.5.1.2 HTTPS connectie

HTTPS heft dezelfde functionaliteit als het HTTP protocol, maar bied meer privacy.
De verbinding is beveiligd zodat niemand mee kan kijken wat er tussen de client en server verzonden wordt. Om dit te realiseren moeten een van de volgende specificaties worden geïmplementeerd:

- HTTP over TLS as documented in RFC 2818
- TLS Protocol Version 1.0 as specified in RFC 2246.
- SSL V3 as specified in The SSL Protocol Version 3.0
- WTLS as specified in WAP Forum Specifications June 2000 (WAP 1.2.1) conformance release - Wireless Transport Layer Security document WAP-199.
- WAP(TM) TLS Profile and Tunneling Specification as specified in WAP-219-TLS-20010411-a

Een specificatie kan gezien worden als een certificaat. Tijdens het tot stand brengen van de connectie wordt er gekeken of er een geldig certificaat aanwezig is. Wanneer dit niet het geval is wordt de verbinding verbroken.

3.5.1.3 Socketconnectie

Socketconnectie is een low-level protocol. HTTP is een extra laat die op een socketconnectie ligt. Hierdoor wordt er extra informatie meegestuurd. Een socketconnectie wordt vaak gebruikt bij het verzenden van binaire data.

3.5.1.4 Conclusie

Een vraag-antwoord protocol is het best geschikt voor dit project. Het gaat hier om verschillende soorten data en dan is het nodig om verschillende parameters mee te geven. De type dat zijn

- kaartdata;
- berichten;
    - sms;
    - locatie;
    - afstanden;
- opvragen van nieuwe locaties;

Een socketconnectie valt dus af. De keuze is dus tussen HTTPS en HTTP. Voor HTTPS moeten er een bepaalde specificatie worden geïmplementeerd. Dit was ten tijden van dit project niet beschikbaar. Bij aanbevelingen

3.5.2 Bericht formaat

Het netwerkprotocol voor het transport is nu gedefinieerd. De volgende stap is het vaststellen van de techniek waarmee informatie over het netwerk wordt verstuurd. De volgende zes technieken zijn onderzocht.

- Binair formaat
- Object serialization
- XML
- XML met binaire compressie
- XML- Remote Procedure Call [2]
- SOAP (Simple Object Access Protocol)

Deze technieken worden vooral bij client server toepassingen gebruikt tussen normale pc's. Met dit onderzoek wordt er ook naar de geschiktheid voor mobiele toepassing gekeken. De volgende criteria worden gehanteerd:

- Applicatie grootte
- Self descriptiveness
- Bandbreedte consumptie
- Processor belasting
- Ervaring binnen Geodan

3.5.2.1 Binair formaat

De informatie wordt over het netwerk vertuurd in een binaire vorm. Deze vorm van dataoverdracht

wordt standaard in J2ME ondersteund via sockets. Het voordeel van deze techniek is het efficiënte gebruik van het netwerkverkeer. Het nadeel is dat het geen transparante techniek is. Een ontwikkelaar heeft voorkennis van binaire data nodig om de applicatie te begrijpen. Het programmeer code zelf is moeilijk te interpreteren. Zowel aan de client als aan de server kant moet men precies dezelfde implementatie aanhouden. Het systeem wordt complexer naarmate het aantal berichten dat verzonden wordt toeneemt, want per bericht is er een aparte implementatie nodig. Bij meer dan vijf a tien berichten wordt de code groot en moeilijk onderhoudbaar.

3.5.2.2 Object serialization

Om de complexiteit bij binaire overdracht te verkleinen kan er voor object serialization gekozen worden. Bij object serialization worden er complete objecten over het netwerk gestuurd in plaats van binaire data. Ieder object komt maar een keer voor. Voor herkenning wordt er aan ieder object een ID meegegeven. Objecten worden in serie over het netwerk gestuurd. Een voorbeeld hiervan is het versturen van een lijst met gebruiker gegevens. Dan wordt er voor iedere gebruiker een apart object aangemaakt en verstuurd met een eigen ID. Aan de client kant worden deze objecten een voor een ingelezen.

Deze techniek maakt het mogelijk om objecten te ordenen en te versturen tussen client en server. Bij J2SE is er geen standaard ondersteuning voor object serialization. Door het ontbreken van deze standaard krijgt iedere object een eigen implementatie.

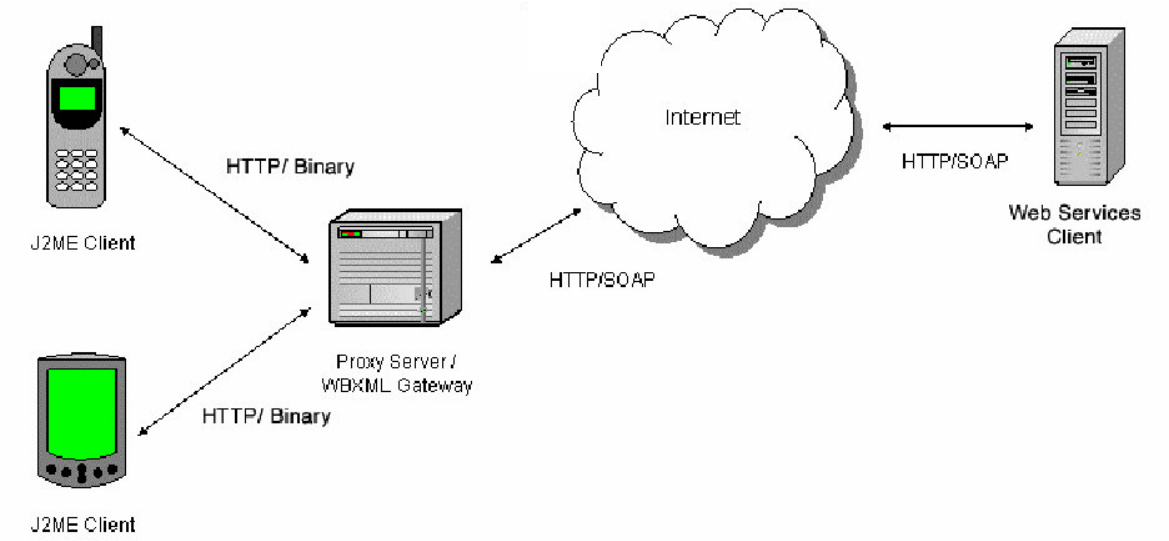
Zowel op de client als op de server moet dezelfde versie geïmplementeerd zijn. Wanneer een object door de server naar de client wordt verzonden moeten allebei de kanten de structuur van het object weten. In het geval van gebruikergegevens moeten client en server beide weten dat de volgorde van de data per object eerst het adres is daarna de naam en als laatst de plaats. Het updaten van de implementatie op het J2ME platform wordt bemoeilijkt door het preverify mechanisme [6,8,10] . De hele applicatie moet dan opnieuw gecompileerd worden.

3.5.2.3 Message exchange by XML

XML staat voor Extensible Markup Language [www.w3schools.org]. Het is ontwikkeld om de functionaliteit van het web te vergroten. Het wordt Extensible genoemd omdat het geen vast formaat is zoals HTML. Een bericht in XML formaat is zelfomschrijvend. De data uitwisseling tussen een client en server kan door middel van een bericht in XML formaat die over het netwerk wordt verstuurd. Dit komt omdat XML gestandaardiseerd is. Er is nog geen standaard XML ondersteuning binnen J2ME. Om XML-documenten in te lezen moet men gebruik maken van een extern pakket die de benodigde klassebestanden bevat die dit mogelijk maken. Doordat XML een populair medium is voor client-server communicatie is er een groot aanbod van externe pakketen die XML ondersteuning bieden voor J2ME.

3.5.2.4 Binary compression of XML

Het WBXML formaat reduceert de omvang van een XML bericht aanzienlijk. De tekst wordt omgezet naar een binair formaat [COMPXML]. Zowel de client als de server moeten het WBXML formaat ondersteunen. Indien de server dit formaat niet ondersteunt kan een proxy-server gebruikt worden met een WBXML gateway.



3.5.2.5 XML- Remote Procedure Call

XML-RPC is een berichtprotocol die het mogelijk maakt om remote procedures uit te voeren via het netwerk met behulp van het HTTP protocol. De client stuurt via HTTP post een bericht in XML formaat. De server parseert dit bericht en stuurt het resultaat in XML formaat terug. Er is nog geen standaard XML-RPC ondersteuning binnen J2ME. Om XML-procedure-calls in te lezen moet men gebruik maken van een extern pakket die de benodigde klassebestanden bevat die dit mogelijk maken

3.5.2.6 SOAP

SOAP is special ontwikkeld voor gedistribueerde applicaties. Verschillende partijen hebben meegedacht om zo met alle aspecten van client-server communicatie, XML-RPC en webservices rekening te houden. Het pakket heeft een hoge mate van complexiteit en is daardoor breed inzetbaar. Hierdoor heeft SOAP een aanzienlijk marktaandeel in de applicaties die gebruik maken van RPC in XML over HTTP. Tijdens de ontwikkeling is er weinig rekening gehouden met een mobiele implementatie. Vooral op het gebied van applicatiegrootte en geheugengebruik wordt veel overhead gegenereerd. Dit komt mede door extra mogelijkheden zoals namespace-awareness, data typing mechanism en de overdracht van aanvullende header informatie. Door deze extra mogelijkheden is SOAP een populair protocol geworden voor client-server applicaties. J2ME bied geen standaard onderseuning voor SOAP. Iedere applicatie die SOAP wil gebruiken moet dit bij de client installeren. Er is een oplossing beschikbaar via het Enhydra ME project. Deze hebben kSOAP ontwikkeld. Dit pakket vergt 41 KB aan geheugen. Een gemiddelde J2ME applicatie is rond de 50KB dus in verhouding tot een volledige J2ME applicatie is dit een substantieel deel. [36]

3.5.2.7 Conclusie

In de tabel 7.1 zijn alle criteria uitgewerkt. In overleg met Geodan zijn de voor en nadelen afgewogen. XML is het formaat geworden voor de opmaak van de berichten. Doorslaggevend waren de onderdelen:

- Package size
- self descriptiveness;
- ervaring binnen Geodan;
- onderhoudbaarheid van de code;

	Binair formaat	*Object serialization*	*XML*	*XML–RPC*	*Soap*
Applicatie grootte (klein + / groot -)	++	++	+	-	--
Self descriptiveness (groot + / klein -)	- -	-	++	++	++
Bandbreedte consumptie (hoog - / laag +)	++	+	+/-	-	---
Processor belasting (hoog - / laag +)	+	+	+	-	--
Ervaringen binnen geodan	+	- -	+++	+/-	++
Onderhoudbaarheid van de code	---	---	++	++	+
Standaard ondersteuning in J2ME	+	-	-	-	-

Tabel 7.1

3.5.3 XML-parser

Na het afwegen van bovenstaande feiten en overleg met Geodan is de keuze voor het bericht formaat XML geworden. Zoals gemeld is er bij J2ME geen standaard XML ondersteuning. Er zijn een aantal opensource XML-parsers op de markt met verschillende licenties. Om tot een goede keuze te komen worden de volgende onderzoekscriteria gebruikt

- Opensource licentie (in welke mate mag Geodan de software gebruiken / aanpassen)
- Model, manier waarop XML documenten worden ingelezen. Het model heeft invloed op
  - Performance
  - Geheugengebruik
  - Deployment code size
- Package size
- Complexiteit (Is het voor een programmeur makkelijk te begrijpen)

3.5.3.1 Licentie

De Opensource XML-parsers worden onder een aantal licenties op de markt gebracht.
In de onderstaande tabel staan de definities van deze licenties uitgewerkt.

Licentie	Omschrijving
Modified BSD	If you want a simple, permissive non-copyleft free software license, [14]
BSD	Moet ik nog opzoeken [15]
EPL	Enhydra PUBLIC LICENSE: [16]
Zlib/lippng	Op de software zit geen garantie dat het goed werkt. De auteur is niet aansprakelijk voor schade veroorzaakt door de software. Iedereen mag deze software gratis gebruiken, aanpassen en herdistribueren. [17]
GPL	General Public Licence: Iedereen mag deze software gratis gebruiken zolang er geen wijzigingen in de software worden gemaakt [18]

3.5.3.2 Methode

Het model geeft aan welke techniek wordt gebruikt om een XML-document in te lezen. Ieder type heeft zijn voor en nadelen op het gebied van geheugengebruik, snelheid en deploymend code size onderhoudbaarheid van de code. [35]

Type	Omschrijving
Model	Het hele document wordt ingelezen. De structuur wordt daarna in het geheugen opgeslagen. Een model parser gebruikt veel geheugen in verhouding tot de andere twee types.
Push	Het hele document wordt ingelezen. Ieder onderdeel wat de parser tegenkomt wordt naar een listener doorgestuurd.
Pull	Het document wordt in stukken ingelezen. De applicatie geeft het verzoek om een nieuw stuk in te lezen totdat het hele document is ingelezen. Deze methode heeft een laag geheugen gebruik.

3.5.3.3 Conclusie

Bij het onderzoeken naar de meest geschikte XML parser wordt er gekeken naar: Licentie; Grootte: De grootte van de klasse bestanden. Dit is een indicatie hoeveel ruimte het in het MIDLET project in beslag neemt. Een standaard MIDLET neemt ongeveer 9 KB in beslag.
MIDP: Het is wel J2ME compatible, maar draait het ook onder MIDP.

Naam	*Licentie*	*Grootte (kB)*	*MIDP*	*Methode*
ASXMLP 020308 [19]	Modified BSD	6	Nee	Push,model
kXML 2.0 alpha [20]	EPL	9	Ja	Pull
kXML 1.2 [21]	EPL	10	Ja	Pull
MinML [22]	BSD	14	Nee	Push
NanoXML 1.6.4 [23]	Zip/libpng	10	Patch	Model
TinyXML 0.7 [24]	GPL	12	Nee	Model
Xparse-J 1.1 [25]	GPL	6	Ja	Model
XMLtp 1.7 [26]	BSD	7	Nee	Push

Na zorgvuldige afweging is er voor kXML 1.2[8] gekozen. Er is gekozen voor een parser die de pull methode ondersteund. Een andere doorslaggevende factor was dat het aangeraden wordt door zowel sun als IBM. [35]

3.6 Ontwikkelomgeving

3.6.1 Clientomgeving

De volgende ontwikkelomgevingen zijn getest:
De onderzoekscriteria zijn:

- Is er een debugomgeving aanwezig
- Is er de mogelijkheid om externe emulatoren te integreren
- Is de ontwikkelomgeving goed getest en stabiel, of is het een bèta omgeving?
- Voor J2ME is de Sun Wireless Toolkit nodig. Hier worden een aantal standaard emulatoren meegeleverd.

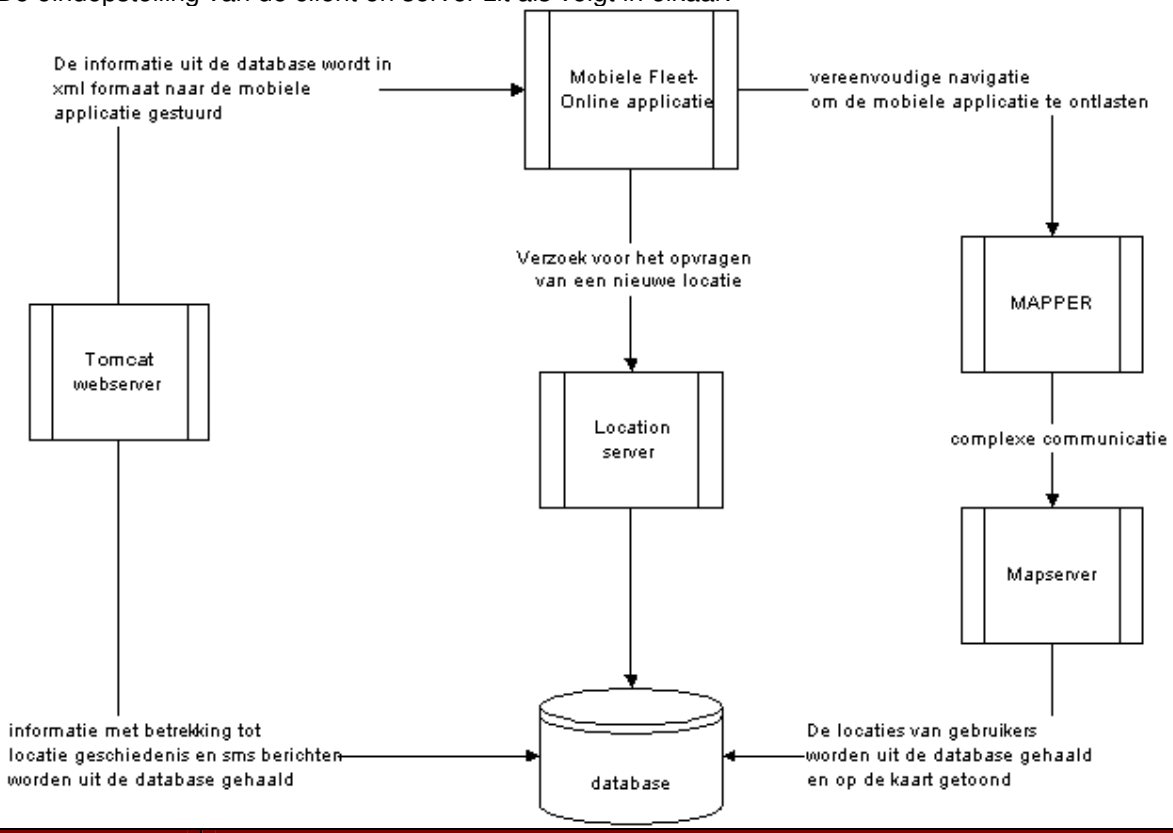
Ontwikkelomgeving	*Debugomgeving*	*Externe emulatoren*	*Bèta?*	*Sun Wireless toolkit integratie*
Sun Java Wireless toolkit	Nee	Ja	Nee	Ja
Siemens SDK	Nee	Nee	Nee	Nee
Motorola SDK	Nee	Nee	Ja	Nee
Nokia SDK	Nee	Nee	Nee	Nee
Eclipse met J2ME plugin	Ja	Nee	Ja	Ja
IBM Websphere device developer	Ja	Ja	Nee	Ja

3.6.2 Serveromgeving

De serveromgeving stond aan het begin van dit project al vast. De communicatie tussen de Client en Server gaat over HTTP en in XML formaat. Dit wordt gerealiseerd via Een Tomcat [4] webserver in combinatie met een Servlet. [11]

4 Resultaat

Het eindresultaat van dit project is een prototype met de benodigde documentatie.
De eindopstelling van de client en server zit als volgt in elkaar:



Component	*Omschrijving*
Mapper	De mapper component van Geodan die als wrapper dient voor de mapserver. Deze vereenvoudigt de navigatie en beperkt het rekenwerk voor de mobiele applicatie. Zo wordt de huidige positie en zoomfactor automatisch onthouden.
Mapserver	De mapserver is een geavanceerd geografisch product van Geodan. Er worden geografische bestanden ingelezen waar een afbeelding van wordt gemaakt. Deze afbeelding kan verschillende lagen bevatten. Bij dit project is de ondergrond het wegennet van Nederland met daar bovenop een laag die de positie van de gebruikers weergeeft.

Component	*Omschrijving*
Tomcat webserver	Hierop draaien de servlets die alle benodigde informatie uit de database haalt en in xml formaat naar de mobiele applicatie stuurt.
Locationserver	Hier komt Vodafone om de hoek kijken. Een nieuwe locatie wordt als volgt opgevraagd: 1 De mobiele applicatie doet een verzoek voor een nieuwe locatie aan de “Location server” 2 De “Location server” slaat de gegevens van de gevonden positie op. Deze locatie gegevens kunnen nu door de mapserver worden gebruikt om de nieuwe locatie op de kaart te tekenen. 3 De gebruiker krijgt ook via de Tomcat webserver de details van de nieuwe locatie
Database	In deze database zijn alle locatiedetails opgeslagen.

4.1 Configuratiebestanden

Een extra functionaliteit is de ondersteuning van configuratiebestanden en ondersteuning van meerdere talen. Aangezien J2ME geen bestandssysteem heeft zijn ook alle classes voor het inlezen van een configuratiebestand eruit gelaten. Om toch een configuratiebestand te krijgen moest er een alternatieve methode worden gezocht die de functie overneemt. Hiervoor wordt een hashtable gebruikt waar sleutels worden opgeslagen en opgezocht.

resources.put("Mapper", "http://fingers.geodan.nl:8008/mapper/map");
en daarna opzoeken met
String mapperAddress = ResourceBundle.getString("Properties_nl", "Mapper");

Dit is ook gebruikt voor de ondersteuning van meerdere talen.

resources.put("LoginFormTitle", "Movida-Online login");
daarna opzoeken met
String mapperAddress = ResourceBundle.getString("Language_nl", " LoginFormTitle ");
Of met
String mapperAddress = ResourceBundle.getString("Language_en", " LoginFormTitle ");

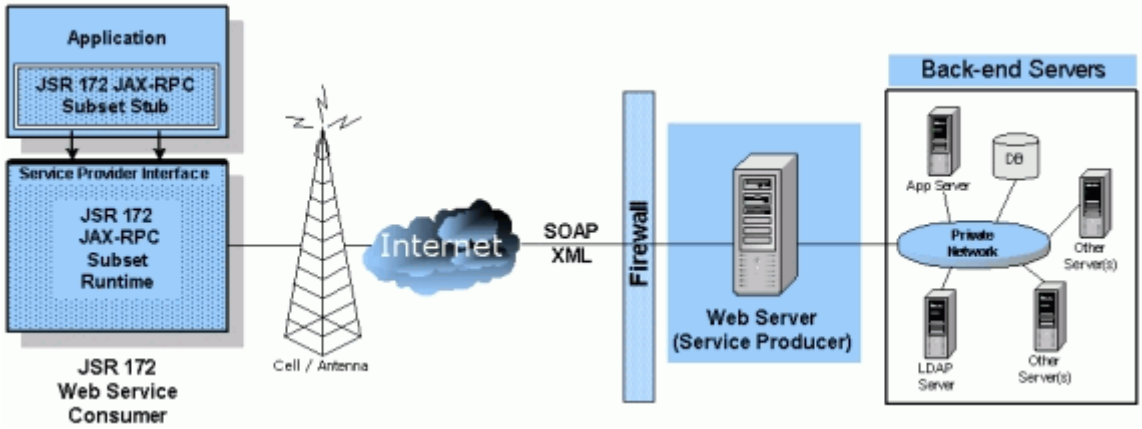
De eerste key geeft aan in welke class bestand er gezocht moet worden. De tweede key geeft aan welke waarde er geretourneerd moet worden.

5 Aanbevelingen

Hieronder volgen een aantal aanbevelingen die niet binnen deze stage pasten maar toch de vermelding waar zijn. Deze zijn niet in het onderzoek meegenomen, of waren nog niet beschikbaar gedurende dit onderzoek.

5.1 XML

Bij sun wordt er een aan standaard gewerkt waarmee J2ME applicaties gebruik kunnen maken van web services. De JSR 172 [1] is een specificatie request voor J2ME webservices. JSR 172 specificeert gestandaardiseerde client-side technologie die het mogelijk maakt om gebruik te maken van remote services.



5.2 Het gebruik van een “obfuscator”

Tijdens de implementatie op verschillende smartphones bleek dat er een limiet aan de applicatie grootte is. Dit is rond de 60Kb. De applicatie nu rond de 90k. Om dit probleem op te lossen kan een obfuscator [34] gebruikt worden. Deze biedt de volgende functionaliteit:

- alle onnodige classes worden verwijderd;
- alle onnodige methodes en variabelen worden verwijderd;
- de class bestanden en package namen worden hernoemd;
- het decompilen wordt tegen gegaan door corrupte data toe te voegen;

Met deze functionaliteit wordt er op twee fronten vooruitgang geboekt. De grootte van de MIDlet wordt geminimaliseerd en de applicatie wordt beveiligd tegen kopiëren. De optimalisatie heeft vooral nut wanneer er een third party pakket wordt gebruikt zoals een XML parser. Er wordt een klein deel van het pakket gebruikt, de rest kan er uit worden gehaald. Zo’n optimalisatie kan tot een halvering van de applicatie grootte lijden. Een aantal bekende obfuscators zijn JAX en Retroguard en Proguard. Na een korte test was de applicatiegrootte verkleint tot 45 Kb.

5.3 Binaire compressie van XML berichten

Tijdens dit project is er gekozen voor kXML 1.2. Dit pakket bied ook ondersteuning voor binaire compressie van XML berichten [33]. Om het prototype zo transparant mogelijk te houden is er gekozen voor een XML implementatie. Het ophalen van kaartdata is het grootste onderdeel van het dataverkeer die de applicatie gebruikt. Mocht er in de toekomst blijken dat de hoeveelheid data die via XML berichten wordt verstuurd te groot wordt kan men er voor kiezen om deze berichten te comprimeren. Aan de client kant hoeft slechts een klasse (XMLParsing.class) worden aangepast.

5.4 HTTPS

Voor deze project is er voort een HTTP connectie gekozen. Deze keuze is mede gebaseerd op het feit dat het binnen Geodan nog niet mogelijk is om een HTTPS connectie tot stand te brengen. Mocht het in de toekomst blijken dat er meer behoefte is naar veiligheid, kan men overstappen naar HTTPS.

6 Woordenlijst

Woord / afkorting	*Omschrijving*
JSR	Java Specification Request
CLDC	Connected Limited Device Configuration
CDC	Connected Device Configuration
KVM	Kilobyte Virtual Machine
CVM	Compact Virtual Machine
MIDP	Mobile Information Device Profile
PDAP	Personal Digital Assistent Profile
JCP	Java Community Process
JPDA	Java Platform Debugger Architecture
SOAP	Simple Object Access Protocol
XML	eXtensible Markup Language
JVM	Java Virtual Machine
preverify mechanisme	Bij een standaard JVM wordt er tijdens het uitvoeren van code nagegaan of het volgens bepaalde standaarden is geschreven. Deze standaarden hebben betrekking op beveiliging, geheugengebruik e.a. Dit vergt extra rekenwerk van de hardware waar de applicatie op draait. Bij J2ME wordt dit rekenwerk overgenomen door de compiler. Dus wanneer de software gecompileerd is kan er vanuit worden gegaan dat alles goed is. De applicatie is nu als een geheel gecompileerd. Er kunnen geen externe bestanden worden toegevoegd of huidige versies overschreven. Om veranderingen door te voeren moet de applicatie als geheel worden gecompileerd.
GPRS	GPRS staat voor General Packet Radio Service. GPRS is een techniek in het GSM-netwerk, die het mogelijk maakt om meer data te verzenden en te ontvangen dan tot nu toe mogelijk was met GSM-telefoons. Dit komt omdat bij GPRS de data niet in één keer over het netwerk wordt verstuurd, maar in stukjes. Op die manier wordt het netwerk efficiënter gebruikt en wordt het ook mogelijk om andere, grotere delen informatie te versturen. Met GPRS kan men met een speciale GPRS mobiele telefoon een permanente internetverbinding hebben. Hierbij betaalt de gebruiker niet - zoals normaal - per seconde dat er verbinding is, maar voor de overgedragen data. De datasnelheid over het huidige GSM-netwerk met GPRS is maximaal 115 Kb/s.

7 Literatuur

Websites

1. XMLCLDC [Parsing XML in J2ME] http://developers.sun.com/techtopics/mobility/midp/articles/parsingXML/
2. XML Remote Procedure Calls http://www.XML-rpc.org/
3. http://www-106.ibm.com/developerworks/library/wi-parsexml/
4. http://www.apache.org/
5. http://www.ez.nl/beleid/home_ond/dgtp/icn_telematica/docs/LBS2001.pdf

Boeken

6. John w Muchow - Core J2ME Technology & MIDP - Prentice Hall - December 21, 2001
7. James White, David Hemphill - Java In Small Things – Manning - 2002
8. Kim Toply - J2ME In A Nutshell – O'reilly – March 2002
9. Vartan Piroumian - Wireless J2ME™ Platform Programming – Prentice Hall – March 25, 2002
10. Dreamtech software team - Wireless Programming with J2ME : Cracking the Code – Hungry minds - 2002
11. Jason Hunter - Java Servlet Programming – O'reilly - 1998
12. Jos Warmer & Anneke Kleppe - Practisch UML- Pearson Education Uitgeverij, 2003
13. Christine Faulkner – The essence of Human Computer Interaction – Prentice Hall - 1998

Referenties Opensource licenties

14. http://www.gnu.org/philosophy/license-list.html
15. http://www.opensource.org/licenses/bsd-license.html
16. http://kXML.enhydra.org/software/license/
17. http://www.opensource.org/licenses/zlib-license.html
18. http://www.webreference.com/XML/tools/license.html

Referenties XML parsers

19. http://www.alsutton.com/software/XMLparser/
20. http://kXML.enhydra.org/
21. http://kXML.enhydra.org/
22. http://www.wilson.co.uk/XML/minml.htm
23. http://nanoXML.sourceforge.net/
24. http://www.gibaradunn.srac.org/tiny/
25. http://www.webreference.com/XML/tools/xparse-j.html
26. http://mitglied.lycos.de/XMLtp/

Overig

27. http://www.velcom.by/en/services/gprs/faq/
28. http://java-virtual-machine.net/other.html#embedded
29. http://www.comp.lancs.ac.uk/computing/users/fittond/ppcjava.html
30. http://kobjects.dyndns.org/kobjects/auto?self=%2381d91ea1000000f5d0738100
31. http://java.sun.com/products/midp4palm/
32. http://developers.sun.com/notfound.jsp?requrl=/techtopics/mobility/midp/ttips/
33. http://developers.sun.com/techtopics/mobility/midp/ttips/compressxml/
34. http://sourceforge.net/project/shownotes.php?release_id=175496
35. http://www-106.ibm.com/developerworks/library/wi-parsexml/
36. www.microsoft.com