Vodagone opleverdocument

Sjoerd Scheffer – 579392

Nijmegen, 6 april 2018

# Inhoudsopgave

Inhoudsopgave 2

1. Inleiding 3

2. Package diagram 4

2.1. Service 4

2.2. Data 4

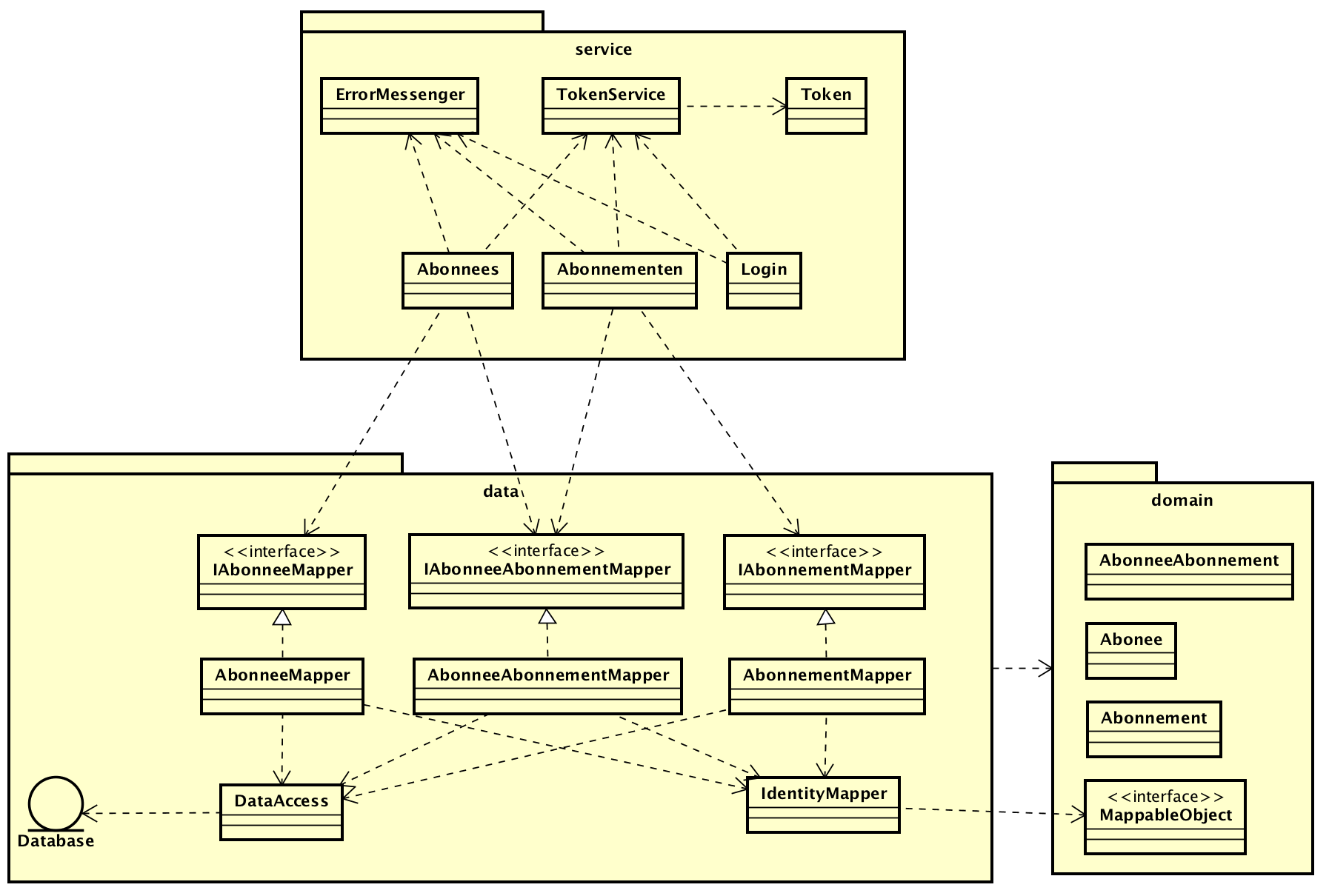
2.3. Domain 4

1. Deployment diagram 5

# Inleiding

Dit document bevat de vereiste diagrammen voor de Vodagone casus van OOSE-DEA. Elk diagram wordt kort toegelicht.

# Package diagram



## Service

De service package zorgt dat de opgevraagde informatie omgezet wordt naar JSON formaat, en zorgt ook voor authenticatie. **ErrorMessenger** is slechts een hulpmiddel om makkelijk foutcodes terug te sturen. **TokenService** zorgt voor authenticatie dmv tokens. **Abonnees, Abonnementen** en **Login** representeren de REST endpoints, en zetten de nodige informatie om van/naar JSON.

## Data

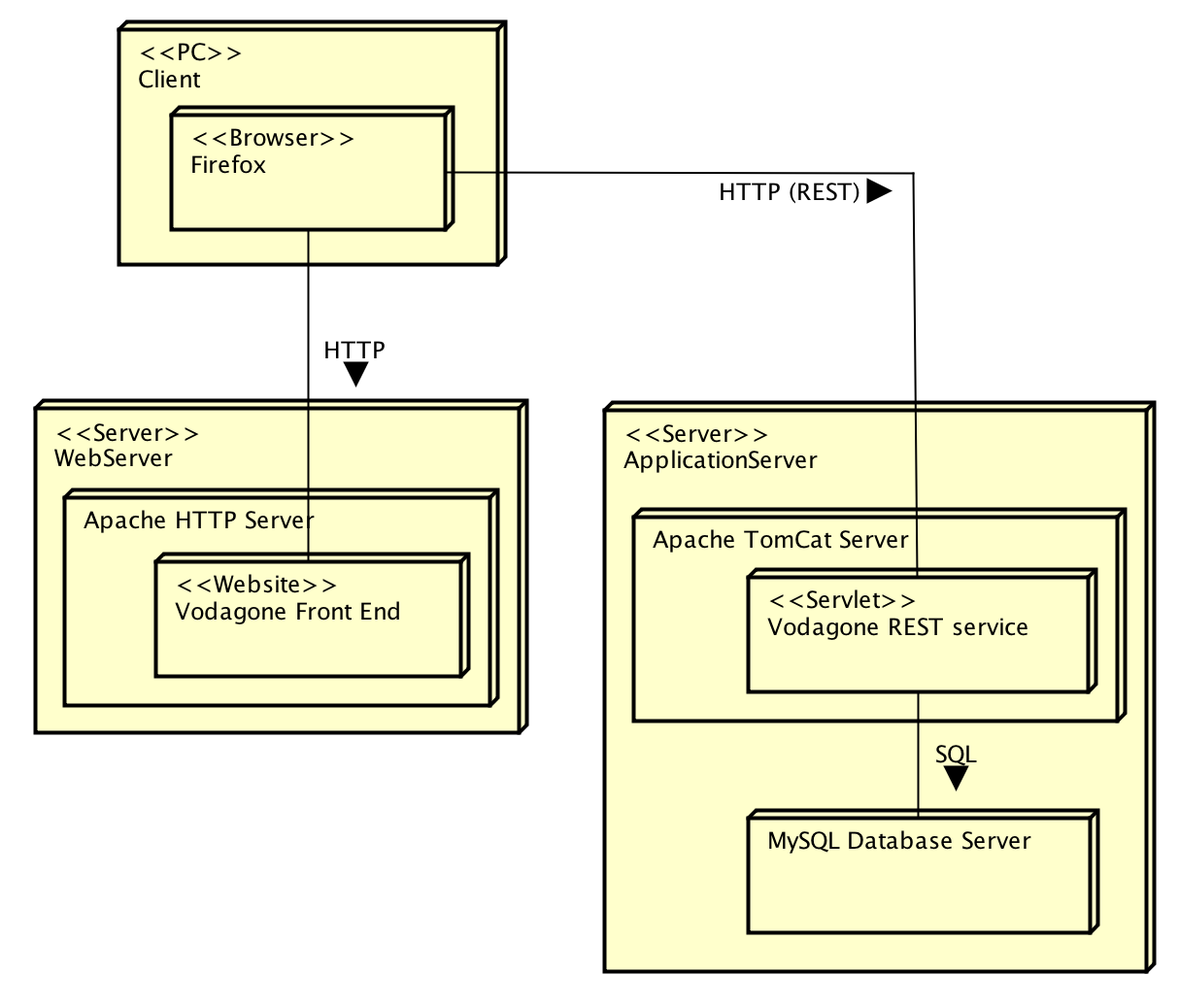
De data package zorgt ervoor dat de data uit de database omgezet wordt naar objecten in het domein. Deze zijn dan bruikbaar voor de service package. **AbonneeMapper, AbonnementMapper** en **AbonneeAbonnementMapper** (Data Mapper pattern) fungeren als een hulpmiddel om data te zoeken, wijzigen en te verwijderen. **IdentityMapper** wordt door de andere mappers gebruikt als een soort cache voor objecten die al opgehaald zijn.

## Domain

De domain package bevat de objectenstructuur van de data die uit de database gehaald wordt. Deze kunnen vervolgen gewijzigd worden en weer teruggeplaatst worden in de database. Ook zorgen deze objecten voor enige datavalidatie.

De huidige structuur zorgt ervoor dat de REST service niet hoeft te weten hoe de data opgehaald of verstuurt wordt naar de database. Wanneer er voor een andere database wordt gekozen hoeven alleen de DataMappers aangepast te worden.

# Deployment diagram



Op dit moment is dit de huidige infrastructuur. De client gaat met zijn browser naar de Vodafone front end. Deze draait op dit moment online op een andere server dan de rest. De front end vraagt (vanaf de browser) data op van de Vodagone REST service. Deze service maakt weer gebruik van een database, die in de huidige situatie op dezelfde server draait.

Om server load te verdelen zou het misschien beter kunnen zijn om de database op een aparte server te draaien. Ook is het mogelijk om de front end op dezelfde server te draaien als de TomCat server.