

Architecture of E-Health Flanders platform

March 21, 2010

Part I

Documentation Beyond
Views

0.1 Documentation roadmap

0.1.1 Description of the parts

0.1.2 How stakeholders might use the package

Platform tester

Patient

Government

0.2 View template

0.3 System overview

0.4 Mapping between views

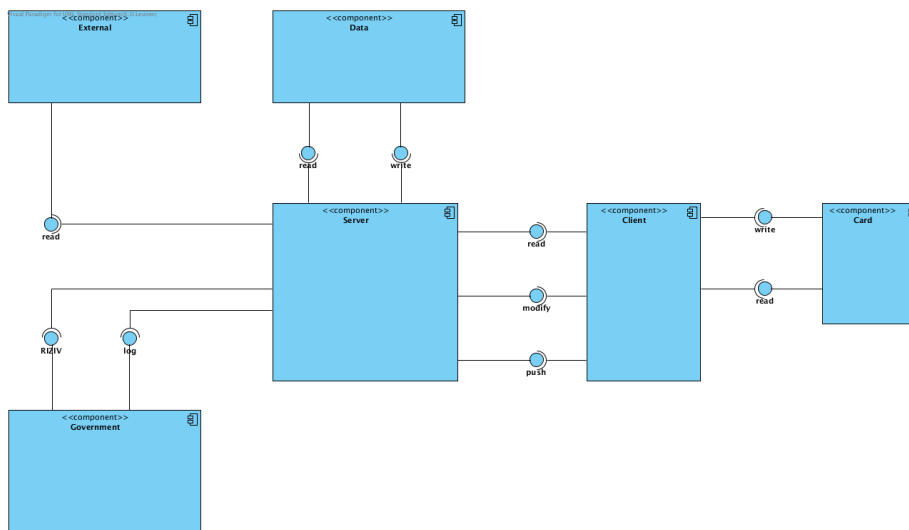
0.5 Directory

0.6 Glossary and acronym list

0.7 Background, design constraints, and rationale

Part II

Software Architecture
Views



0.8 Module Uses View

0.8.1 Primary presentation

0.8.2 Element catalog

Elements and their properties

Relations and their properties

Element interfaces

Element behavior

0.8.3 Context diagram

0.8.4 Variability guide

0.8.5 Architecture background

Rationale

Analysis results

Assumptions

0.8.6 Other information

0.8.7 Related view packets

0.9 C&C Client and Server View: Overview

0.9.1 Primary presentation

0.9.2 Element catalog

Server De server is de component waar clients verbinding mee maken. De interacties tussen clients en servers gebeuren op een veilige manier. De server is verbonden met clients, de overheid, eventuele externe componenten en de data server. Meer informatie kan gevonden worden in 0.10 Client and Server View: Server.

Client De client component is de component die gebruikers (dokters, patiënten of apothekers) gebruiken om op een veilige manier met de server te interageren. Meer informatie kan gevonden worden in 0.11 Client and Server View: Client.

Data De data component is een database waar alle informatie zoals patienten dossiers en dokter data op bewaard zijn. Hoe de data precies wordt opgeslaan is terug te vinden in het Deployment diagram.

Card De card component stelt de e-Card van de gebruiker voor. De e-Card bevat gebruikersinformatie alsook twee keys. Een key voor identificatie van de gebruiker en een key voor authenticatie. Naast de gebruikersinformatie en de keys heeft de e-Card ook ruimte voor een aantal voorschriften op te slaan. Meer informatie is terug te vinden in:

Government De government component bevat toegang tot de logging en RIZIV database.

External De external component bevat de validated data sources die gebruikers van de client kunnen opvragen via de server.

0.9.3 Context diagram

TODO

0.9.4 Variability guide

[None]

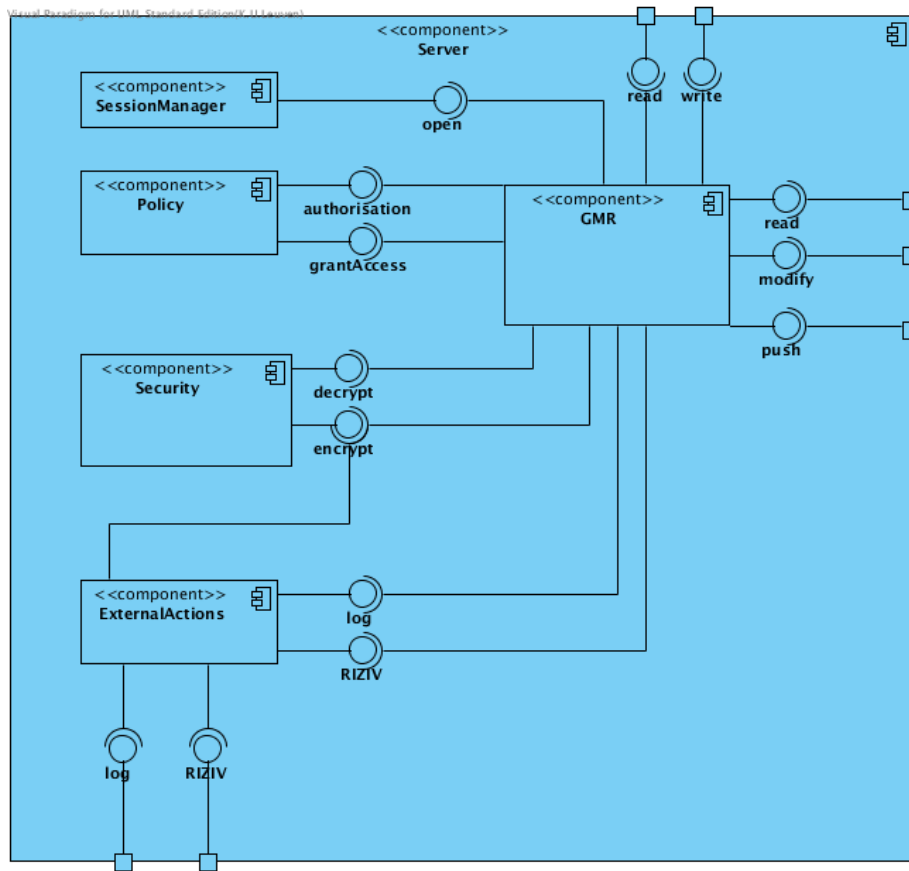
0.9.5 Architecture background

Rationale

Enkele design beslissingen die hier te zien zijn is het gebruik van de card reader voor het identificeren van de gebruikers op de centrale database, meer hierover is te vinden in het ...TODO.

0.9.6 Related view packets

De volgende client-server views bekijken dit overview diagram in meer detail. Ook in het deployment diagram is extra informatie te vinden vooral dan in verband met de opslag van de global medical records en dokter data.



0.10 C&C Client and Server View: Server

0.10.1 Primary presentation

0.10.2 Element catalog

GMR De GMR is de centrale component in de server component. De GMR delegeert alle inkomende *read*, *modify* of *push* acties naar de correcte andere componenten en doet dit uiteraard in de correcte volgorde.

Element interfaces

read TODO

modify TODO

push TODO

12

Element behavior

TODO

SessionManager

Element interfaces

open TODO

Element behavior

TODO

Policy

Element interfaces

authorisation TODO

grantAccess TODO

Element behavior

TODO

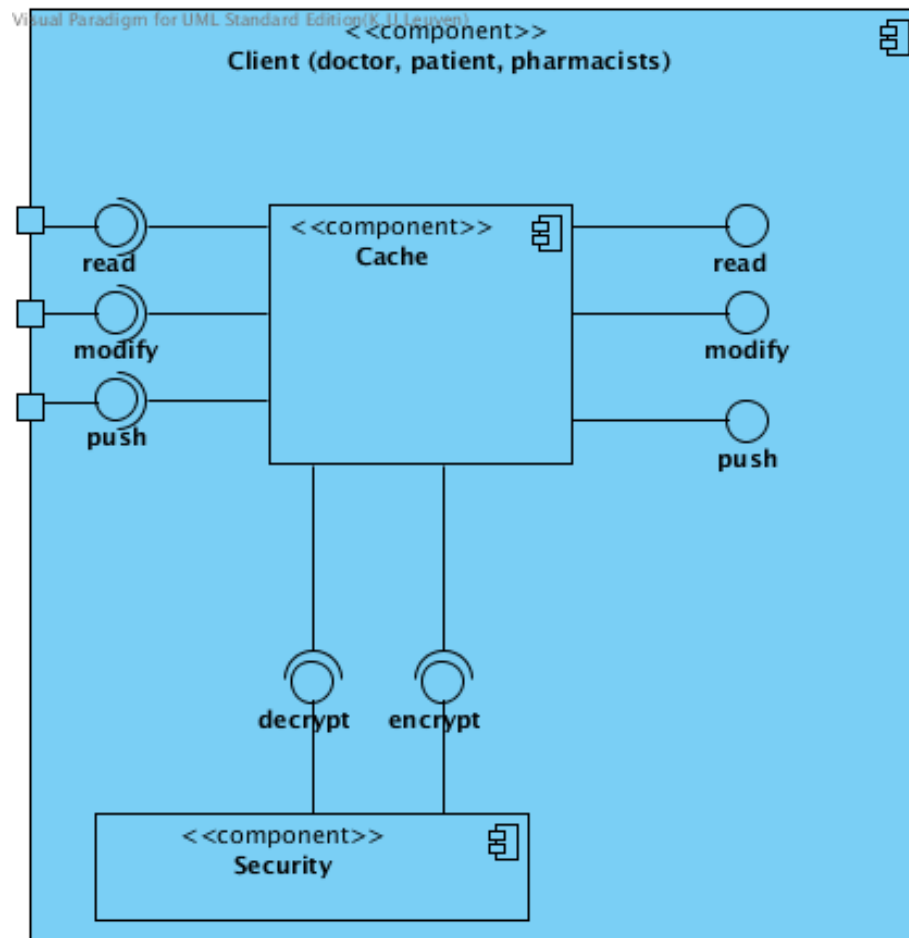
Security

ExternalActions

Element interfaces

log TODO

RIZIV TODO



0.10.3 Context diagram

0.10.4 Variability guide

0.10.5 Architecture background

Rationale

Analysis results

Assumptions

0.10.6 Other information

0.10.7 Related view packets

0.11 C&C Client and Server View: Client

0.11.1 Primary presentation

0.11.2 Element catalog

Cache De cache component is een lokale cache van bestanden die ook op de server aanwezig zijn.

Element interfaces

read TODO

modify TODO

push TODO

Element behavior

Wanneer een item uit de cache gelezen wordt (*read*), verstuurt de cache eerst de vraag naar de server als het gegeven item up-to-date is. Als dit het geval is dan wordt het bestand uit de cache aan de gebruiker weergegeven, is dit niet het geval dan stuurt de server het meest recente bestand terug. Dit wordt dan opgeslagen in de cache en dan teruggegeven aan de gebruiker.

Het schrijven van een item (*write*) gebeurt door het item eerst lokaal te schrijven, daarna verstuurt de cache een write operatie naar de server. Wanneer de cache een antwoord gekregen heeft van de server dat de schrijfoperatie correct verlopen is mag de cache het geschreven item uit de cache verwijderen.

Een *push* operatie is gelijkaardig aan de schrijf operatie van hierboven. Het enige verschil is dat het in het geval van een *push* gaat over een operatie waarbij de gebruiker geen toegang heeft tot het te schrijven document. Hij kan enkel een data-veld toevoegen, een veld wijzigen of overschrijven is niet mogelijk.

Security De security component is op zich gelijk aan deze in de server. De gedetailleerde uitwerking is te vinden in 0.12 Client and Server View: Security.

Elements and their properties

Relations and their properties

Element interfaces

Element behavior

0.11.3 Context diagram

0.11.4 Variability guide

Verschillende implementaties van de Client component zijn mogelijk. Wanneer de client lokaal op een pc wordt uitgewerkt voor een dokter zal de cache waarschijnlijk zo veel mogelijk data bijhouden.

Voor een client van de dokter die bijvoorbeeld op een smartphone draait is het onrealistisch om te verwachten dat deze alle data zal cachen. De client kan dan bijvoorbeeld zo uitgewerkt worden dat de cache van de smartphone iedere morgen voor de huisbezoeken gesynct wordt met de nodige patiënten van die dag.

Nog een ander geval is het geval van de client voor de apotheek of voor de patiënt. In het geval van een client voor de patiënt zal de cache enkel het dossier van de patiënt zelf bijhouden. Voor de apotheek hoeft de cache helemaal geen data te cachen.

Een laatste geval is wanneer de cache op een webclient draait. In dit geval kan de caching aangepast worden naar de mogelijkheden van de webserver. Hier moet dan wel rekening gehouden worden met de wetgeving die centrale opslag van patiënten data verbiedt. Het is mogelijk dat een webserver zoveel zou gaan cachen dat dit zou kunnen gezien worden als een centrale opslag. Omdat dit niet toegelaten is moet men er bij de implementatie van deze cache rekening mee houden. De cache zou zo kunnen aangepast worden om slechts een bepaald aantal documenten bij te houden of enkel de documenten die in een bepaalde tijdspanne opgevraagd zijn.

0.11.5 Architecture background

Rationale

Analysis results

Assumptions

0.11.6 Other information

0.11.7 Related view packets

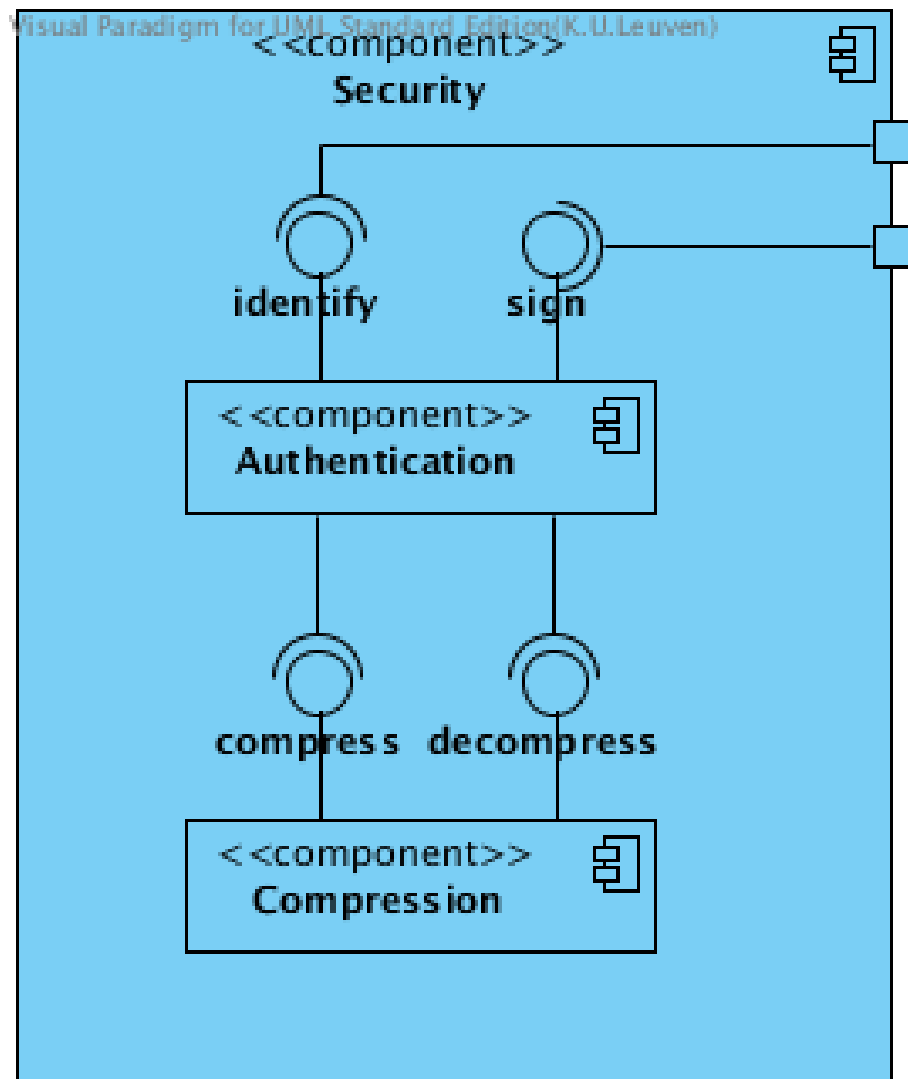
0.12 C&C Client and Server View: Security

0.12.1 Primary presentation

0.12.2 Element catalog

Authentication

Compression



Elements and their properties

Relations and their properties

Element interfaces

Element behavior

0.12.3 Context diagram

0.12.4 Variability guide

0.12.5 Architecture background

Rationale

Analysis results

Assumptions

0.12.6 Other information

0.12.7 Related view packets

0.13 Allocation Deployment View

0.13.1 Primary presentation

0.13.2 Element catalog

Elements and their properties

Relations and their properties

Element interfaces

Element behavior

0.13.3 Context diagram

0.13.4 Variability guide

0.13.5 Architecture background

Rationale

Analysis results

Assumptions

0.13.6 Other information

0.13.7 Related view packets

0.14 Interaction Diagrams

0.14.1 Primary presentation

0.14.2 Element catalog

Elements and their properties

Relations and their properties

Element interfaces

Element behavior

0.14.3 Context diagram

0.14.4 Variability guide