Mandatory Project: Software Architecture of the TM12 System

Mogens Habekost, Susie Agerholm Balle
Department of Computer Science, University of Aarhus
Aabogade 34, 8200 Aarhus N, Denmark
Gruppe: Spinebreaker
Student ID: 20070605, 19972253
Email: moh@solar.dk, susie.agerholm@gmail.com

H1. Architectural Description
18.09.2012

Abstract

The TM12 system implements an information system for supporting tele medicine, i.e. patients making measurements in their homes for review by general practitioners as well as hospital clinicians. This report gives a software architecture description of an architectural prototype of the TM12 system. The techniques used for architectural description are taken from [Christensen et al., 2012].

1. Introduction

Figure 1 shows a schematic overview of TM12:

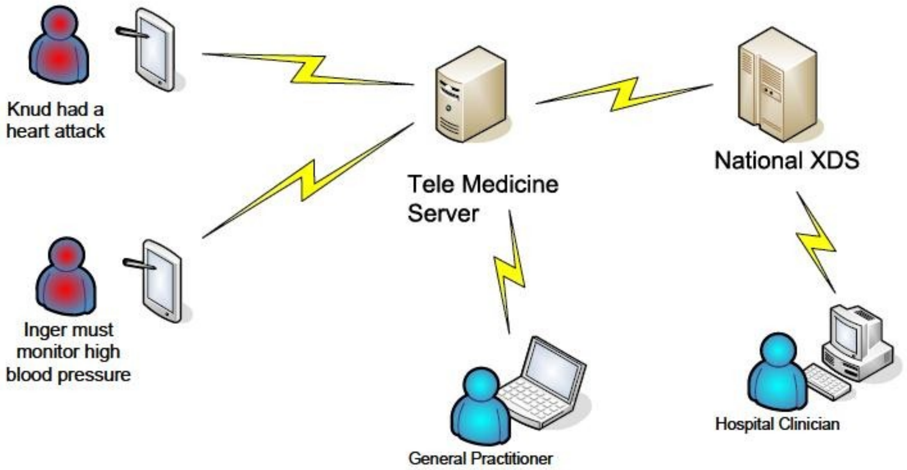


Figure 1: Rich picture of the TM12 architecture

TM12 is based on a device/tablet in the home that can make measurements of blood pressure and upload these to a TM12 web server. Such a measurement is denoted a tele observation or just observation. The software architect has made a decision to make the size of the message sent from home client to TM12 server small. The TM12 converts the observation into a standard format for clinical information, HL7 v.31[1], and stores it in an XDS2[2] storage system. A general practitioner can review observations for a patient using a webbrowser interface implemented on

[1] www.hl7.org

[2] Cross-Enterprise Document System, wiki.ihe.net/index.php?title=Cross-Enterprise Document Sharing

the TM12 server.

2. Architectural Requirements

For our purposes there is two main use case for the TM12 system:

- *Tele observation upload*: The patient makes a measurement in his/her home and uploads it to the TM12 server.
- *Tele observation review*: The general practitioner reviews all tele-observations for at given patient in a given time interval using a webbrowser.

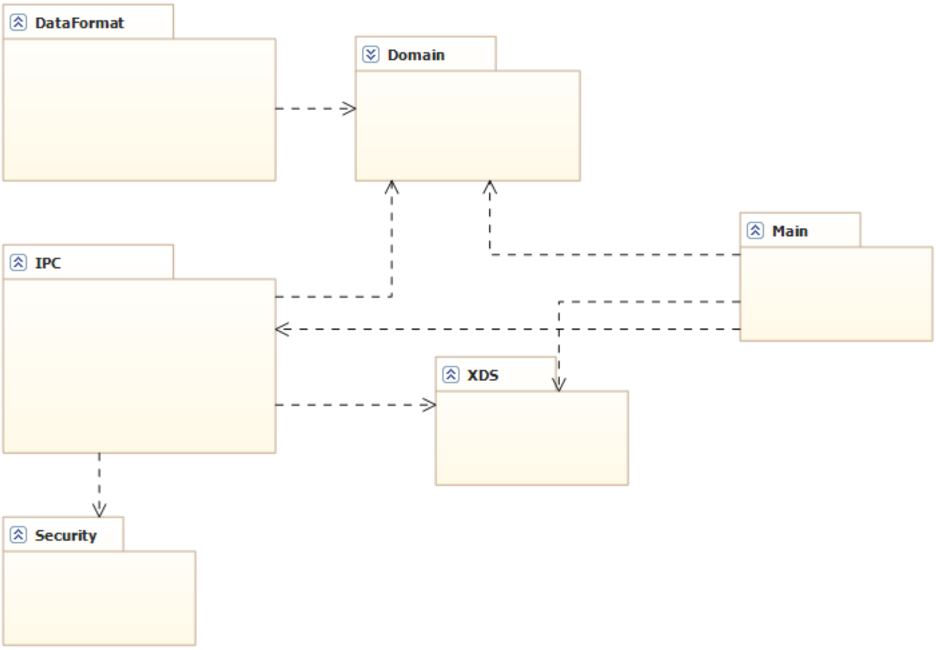
The major driving qualities attributes of the TM12 system are:

- *Performance.* TM12 should be performant so that a large number of patients may be part of the system.
- *Modifiability*. It must be possible to modify TM12 to include new types of clinical measurements or formats.
- *Security.* TM12 handles person-sensitive data and should only be accessed by authorized medical staff.
- *Availability.* System should have minimal downtime and not loose any measurements.

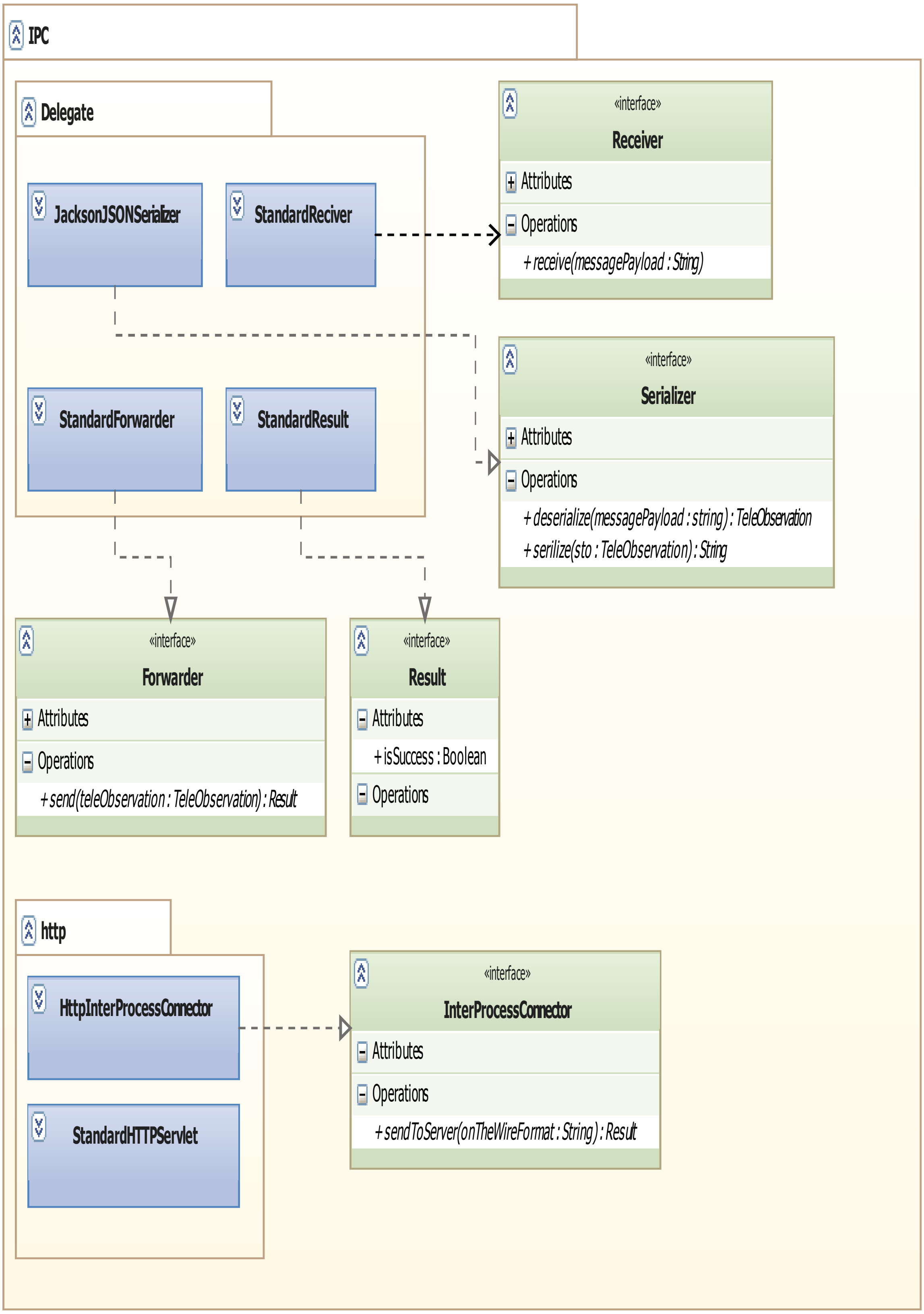
3. Architectural Description

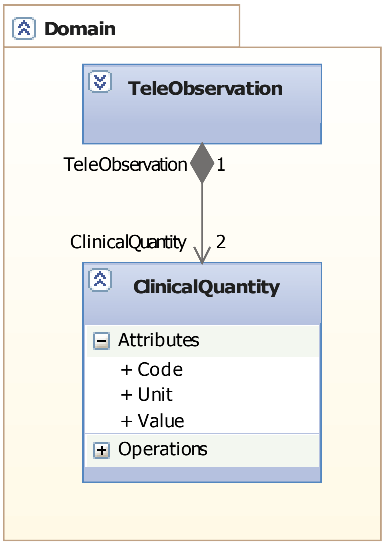
3.1 Module Viewpoint

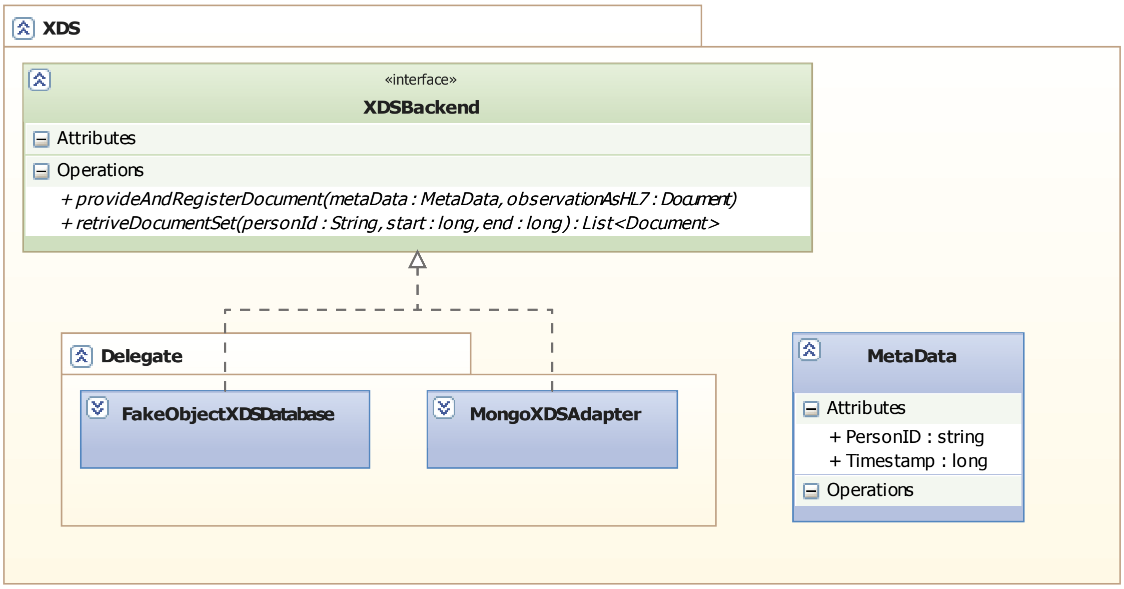
Overview – dependencies between packages:

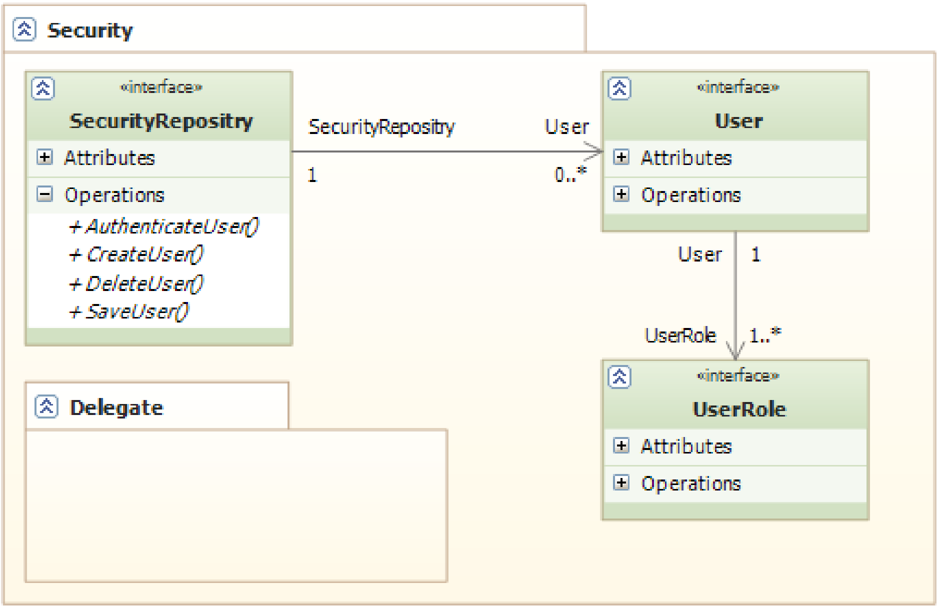


Packages:


IPC
Delegate
JacksonJSONSerializer
StandardReciver
StandardForwarder
StandardResult
«interface»
Receiver
Attributes
Operations
+receive(messagePayload : String)
«interface»
Serializer
Attributes
Operations
+deserialize(messagePayload : string) : TeleObservation
+serilize(sto : TeleObservation) : String
«interface»
Forwarder
Attributes
Operations
+send(teleObservation : TeleObservation) : Result
«interface»
Result
Attributes
+isSuccess : Boolean
Operations
http
HttpInterProcessConnector
StandardHTTPServlet
«interface»
InterProcessConnector
Attributes
Operations
+sendToServer(onTheWireFormat : String) : Result

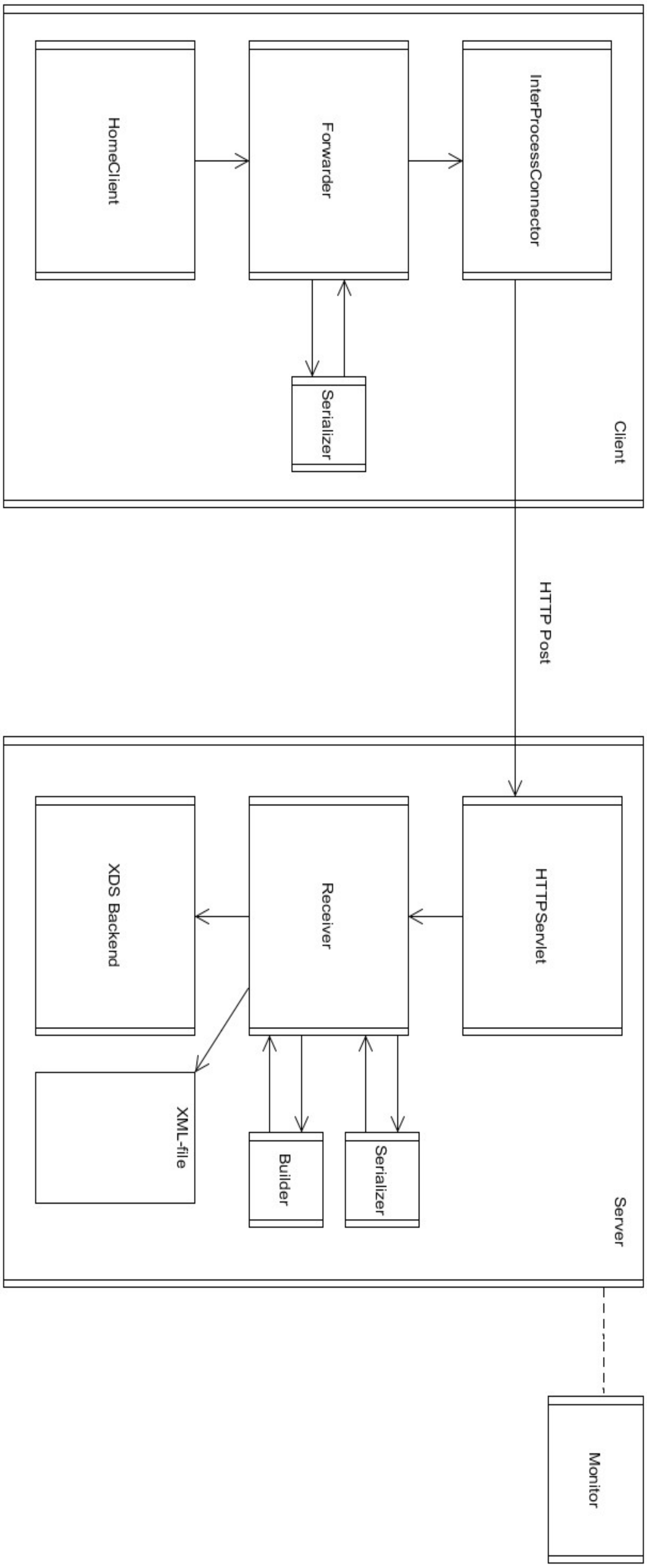

Domain
TeleObservation
TeleObservation
1
ClinicalQuantity
2
ClinicalQuantity
Attributes
+ Code
+ Unit
+ Value
Operations


XDS
«interface»
XDSBackend
Attributes
Operations
+ provideAndRegisterDocument(metaData : MetaData, observationAsHL7 : Document)
+ retriveDocumentSet(personId : String, start : long, end : long) : List<Document>
Delegate
FakeObjectXDSDatabase
MongoXDSAdapter
MetaData
Attributes
+ PersonID : string
+ Timestamp : long
Operations


Security
«interface»
SecurityRepository
Attributes
Operations
+ AuthenticateUser()
+ CreateUser()
+ DeleteUser()
+ SaveUser()
SecurityReposity
1
User
0..*
«interface»
User
Attributes
Operations
User
1
UserRole
1..*
«interface»
UserRole
Attributes
Operations
Delegate

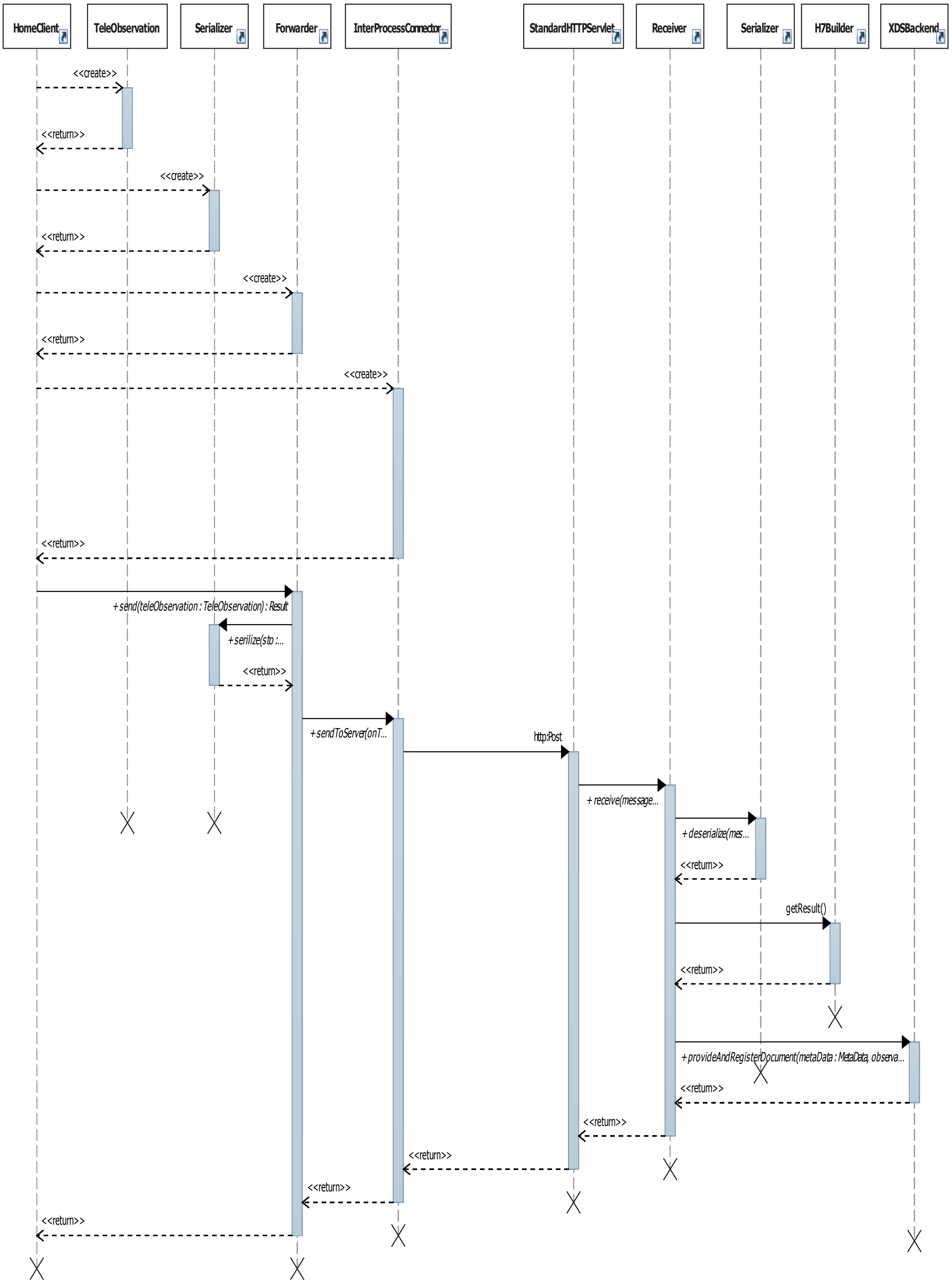
3.2 Component & Connector Viewpoint

Overview of active classes and responsibilities:

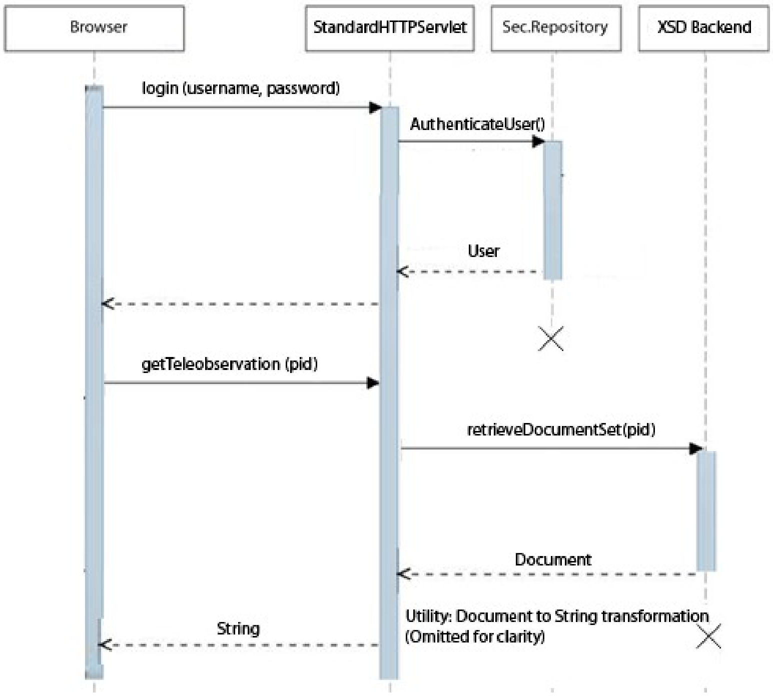


Sequence Diagram (selected classes):

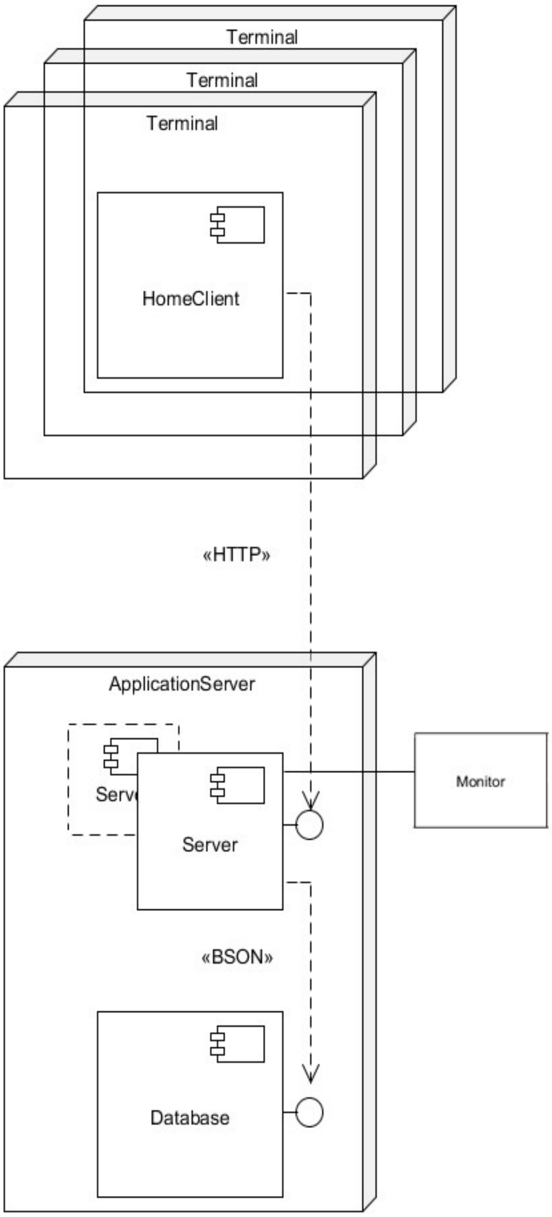
Posting data to TM Server:



Retrieving data from TM server:



3.3 Allocation Viewpoint



4. Discussion

Den abstrakte beskrivelse af begreberne 'elements, form og rationale' betyder, at der er et meget stort rum for fortolkning og dermed et stort potentiale for overlap mellem de tre kategorier. Alle elementer fra TM12 kan sagtens passes ind under mere end en kategori, alt efter hvordan man vælge at fortolke beskrivelsen - men her er vores bud :)

'Data elements' repræsenterer udvekslede data og udgøres groft sagt af klasserne fra domain package (Teleobservation, ClinicalQuantity)

'Processing elements' repræsenterer elementer, der 'transformerer data-elementerne' Det er klasser fra dataformat (builder, director)

'Connecting elements' er klasserne, der danner rammen om kommunikationen - IPC package, (serializer, forwarder, receiver) HomeClient, InterProcesConnector og HTTPServlet.

'Form properties' lægger en begrænsning på elementerne - vi går ud fra det skal være begrænsning i et arkitektonisk perspektiv. Vi har derfor valgt at koble den kvalitet til forskellen mellem interface og implementationsklasse, da det er denne kvalitet, der skiller de arkitektonisk betydende klasser (interfaces) fra klasser uden nogen arkitektonisk betydning (simple, implementerende klasser).
'Form relationships' lægger en begrænsning på elementernes placering i arkitekturen. Vi har derfor valgt at koble den kvalitet til evt. forhold omkring concurrency og de problemstillinger, der findes herunder.

Rationale: Beskrivelsen lægger op til en meget bred fortolkning af denne kategori - fra funktionelle aspekter til økonomiske. Vi har derfor valgt at sige at den kategori dækker over valget af infrastruktur bredt - en HTTP-baseret, client-server arkitektur som platform for dataudvekslingen.

5. Quality attributes – general scenarios

Scenario(s):	#1 Measurement posted by patient should not get lost even if communication channel to XSD database is down.
Relevant Quality Attributes:	Availability
Scenario Parts	**Source:** Internal to the system
	Stimulus: Crash
	Artifact: Communication channel to XSD storage
	Environment: Normal operations
	Response: Continue operation in degraded mode: Measurements are stored on TM server until communication channel to XDS Storage is restored.
	Response Measure: The TM server can operate in degraded mode for 100.000 measurements, after that the specific TM server will be unavailable.
Questions:	What effect will running in degraded mode have to the system? • The measurements will not be available for the Hospital Clinical personal. • The General Practitioner (GP) will not be able to review the measurements What happens when the server gets unavailable because more than 100.000 measurements have been received? • At that point the measurement equipment (TM client) will fail to connect to the TM server, and then try a backup TM server. The backup TM server is also needed for handling high loads scenario #11. Why store the data on the TM server, when a backup server is available? • Because it handles the case where it is the XDS servers that are unavailable, and all TM Servers has lost communications to the XDS server.
Issues:	• When storing the measurements on the TM server during downgraded mode, it is necessary that data is stored in such a way that they can be recreated in case of a crash of the TM server. • By storing data on TM server there is a higher security risc, when running in downgraded mode?
Tactics:	• Ping from TM Server to XSD Storage? • Passive redundancy (entails state resynchronization) • Journaling established within TM Server? XML file will be written if XSD connection down • Client side transaction: Client told to resend measurement if no acknowledgement received from server • State resynchronization for restart of failed server
Change to	• The receiver class will now have 2 options for storing an

original architecture:	measurement(Repository): • Option 1(Primary). Storing the measurement in XSD backend. • Option 2(Degraded mode). Storing the measurement in local file storage (Journaling Tactics). • The receiver class will need logic to decide which storage to use (Ping Tactics). • Logic to write the local stored measurements to XDS backend when it again is online(State resynchronization). • A fallback server will be added to handle cases where the primary server runs full (Passive redundancy).
Upsides and downsides:	Pros: • The system can handle loss of the XSD backend for short periods. • Safety-switch: If the primary webserver in downgraded mode has a full temporary storage, a second webserver takes over. Cons: • Added code complexity • Measurements will be temporarily unavailable to GP

Scenario(s):	#11 Measurements posted from 1000 simultaneous users should be reformated and persisted with a maximal latency of one minute.	
Relevant Quality Attributes:	Performance	
Scenario Parts		**Source:**
		Stimulus: 1000 simultaneous (simultaneous = are posting within one second of each other) clients/users uploading their measurement to TM server.
		Artifact: TM Server
		Environment: Normal operations, during runtime
		Response: If bottleneck occurs within processes executed on the webserver (i.e. data-serializing and reformating of data) a new webserver instance will start up. If a bottleneck occurs in relation to XSD Communication channel, files will temporarily be written to local storage (XML-file on webserver).
		Response Measure: Maximal latency of one second from a measurement arrives at webserver untill it is reformatet and stored in database or local storage (XML-file on webserver).
Questions:	How to pinpoint exactly where bottleneck occurs (webserver og communication to database) in order to choose suitable remedy?	
Issues:	• Webserver might get constipated. • XSD backend might get constipated.	
Tactics:	• Bound execution time • Increase available resources (shift to backup webserver) • Manage event rate (if backend is busy, events will be written to XML-file)	
Change to original architecture:	• An extra TM Server instance will start taking over processing requests if time to execute in HTTP Servlet or StandardReceiver class exceeds a certain timeframe. • TM server will write measurements to local file storage if backend response times exceeds a certain timeframe..	
Pros and cons:	Pros:	

	- We adress bottlenecks in system from a holistic perspective – no use optimizing storage if the servlet is slowing things down. - We use code from scenario #1 (availability) to achieve another goal i.e. performance optimization - The system can handle loss of the XSD backend for short periods. - If TM server crashes no data will be lost. Cons: - Added code and test complexity. - The cost of hardware will increase. - Performance tactics in relation to database bottleneck renders some of the measurements temporarily unavailable to GP (see scenario #1), so should only be short term solution to storage bottleneck.

Scenario(s):	#13 If a TM server crashes, system should be able to recreate data and clients should be still able to get their measurements processed.	
Relevant Quality Attributes:	Avaliablity	
Scenario Parts		**Source:** Internal event
		Stimulus: Server crash
		Artifact: TM Server
		Environment: Normal operation
		Response: Incoming requests should be directed to other secondary TM server (passive redundancy). Events already being processed on crashing server should be finalized after reboot.
		Response Measure: No requests should be lost due to server crash
Questions:		
Issues:	- Requests does not get processed - Server not accepting events	
Tactics:	- Process monitor - State resynchronization - Passive redundancy (new webserver ready to take over)	
Change to original architecure:	- Process monitor checks primary webserver process and initiates a restart if necessary and initiates backup server. - An extra webserver is added to the system and is ready to take over if the primary server is down for a restart (Passive redundancy)	
Pros and cons:	Pros: - If a server crashes a new one is available right away. Cons: - Double cost for hardware	

Scenario(s):	#15 Unauthorized users should not be able to access data on the server..	
Relevant Quality Attributes:	Security	
Scenario Parts		**Source:** Unknown user
		Stimulus: An unknown user is attempting to request data

	(measurements) from system**.**
	Artifact: TM 12 Server
	Environment: Normal operations
	Response: Attempt to access data without proper authorization should be blocked, detected and logged.
	Response Measure: All unauthorized access to the system should be denied.
Questions:	
Issues:	
Tactics:	• Authenticate user • Authorize user • Limit access (Webserver localized in DMZ while backend is hidden behind firewall)
Change to original architecture:	• A new user object will be added to the system. The user object must support roles. The following roles is needed: normal user, Medical user and admin user. • The admin user should have permissions to crud users. • The medical user has permission to review data and the normal user has permissions to add measurements. • The TM server needs to authenticate the user and enforce the roles of the user. The client needs to supply a valid login. • The TM server has to be placed in an DMZ zone with access to the XDS backend.
Pros and cons:	Pros: • The system will be able to distinguish users and deny unwanted users. • The DMZ will protect the XDS backend from the internet. Cons: • Added code and test complexity(Authenticate/Authorize the user).

Scenario(s):	#17 HL7 format is replaced/supplemented by a newer format HL8, which will be used from now on. However it should still be possible to read data in both formats.
Relevant Quality Attributes:	Modifiability
Scenario Parts	**Source:** System owner
	Stimulus: System owner wants to support new format with related new technologies – or system is exported to countries with other formats.
	Artifact: TM 12 System
	Environment: Design time
	Response: System is able to store new measurements in HL8 format. It is still able to display old measurements in legacy format to users.
	Response Measure: Version of system with new format should be developed, unit-and integration-tested within fourteen days.
Questions:	
Issues:	
Tactics:	• Use an intermediary (Builder pattern, Repository pattern) • Polymorphism

	• Hide information (Knowledge of how to construct the new HL8-format belongs exclusively to Director class – i.e. clear definition of responsibility)
Change to original architecture:	Add HL8 classes to the system.
Pros and cons:	Cons: • Need to support 2 formats. Extra code to support/debug/test. Pros: • Don’t have to make database conversion of existing documents. • Already stored data is still ‘valid’.

References

[Christensen et al., 2012] Christensen, H. B., Corry, A., and Hansen, K. M. (2012). An Approach to Software Architecture Description Using UML 2.3. Technical report, Computer Science Department, University of Aarhus.