An Integrated System to support Internet of Things (IoT) in a Smart Home

Software Architecture (Autumn 2015) – IT University of Copenhagen

Group Project Phase 2 Submission

Henrik Buch-Larsen

Michael Poulin Mortensen

Rasmus Løbner Christensen

1 TABLE OF CONTENT

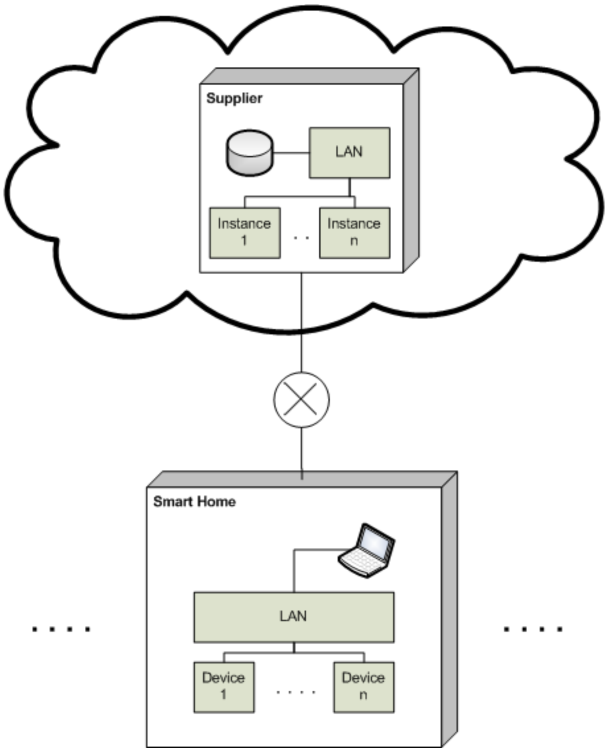
Change Traceability Table

	Enhancement description	**Feed back**	**Section**
1.	Personas added	Group and report feed back	3. User Stories and Personas

2 INTRODUCTION

This document contains a proposed architecture for an integrated system to support Internet of Things (IoT) in a Smart Home. The proposed solution takes base in the IoT reference architecture, handed out as part of the course learning material.

Since the term Internet of Things in its nature is about collecting and exchanging data across a network, we believe that a distributed cloud solution is the best way of designing such a system. As an introduction, we present the below picture which describes the proposed system in a high level way.



For each Smart Home connected to the distributed solution, the given supplier of such a system stores an instance. Through the cloud every Smart Home can interact with the given services and processes which the system provides. The laptop in the Smart Home depicts the fact that the given household users do not require any supplier made equipment in order to interact with the system, simply a laptop (in general a computer with internet connection), in order to configure the system the first time accessing the solution, and furthermore handling the ongoing configuration that such a system requires.

3 USER STORIES AND PERSONAS *

At the initial launch meeting, the project team developed the following *User Stories*.

For all *User Stories* the structure of: As a <Role> I want <Action> So that <Concern/Benefit>.

Not all requirements in the assignment are described in this section.

ID	Role	Action	Concern/Benefit
US0	Administrator	Restrict/Limit the use of Streaming/TV	So that children can watch and listen to music only at certain times.
US1	Administrator	Register a Device	It is useable.
US2	User	Register Consumable	It is consumable by other User's.
US3	Adminstrator	Register a Product	Orders can be made on specified stock thresholds.
US4	User	Notify Fire Department, in case of fire	The Fire Department is aware, and takes the needed action.
US5	User	Light up paths for an emergency exit, in case of fire	House residents know where to go in order to escape the house as fast as possible.
US6	User	Override system security, in case of fire	No security measures are hindering an easy escape for the house residents.

Additional brainstorm exercises were was done in this phase of the project, which can be seen under the ***notes*** section lastly in this document.

Quality Attributes	Description
Reliability Ease of use/intuitive interfaces Security Performance	John is a father, who sees himself as the head of his family. He works as a software engineer in a small software company, and in general, is very technology minded. Due to his fascination of technical gadgets, and what not, he's very interested in getting a Smart Home implemented for his family. User stories: 1. I need to ensure to my family, that a Smart Home solution is notifying the family members, in case of fire 2. I need to be able to see a log of who have entered/left the house over a given time period 3. I need to be able to set up rules for my children about the use of the households TV 4. I need to be able to monitor the use of electricity for my household, in a user-friendly fashion

Quality Attributes	Description
* Security * Ease of use/intuitive interfaces	Cindy is the husband of John. Cindy works as a sales officer for the local car insurance company. Due to her profession, she is concerned whether this proposed (by John) Smart Home solution is secure enough to guarantee the security of the household's automobiles. Furthermore she sees herself as the ones in charge of cooking as well as handling the purchase of groceries for her family. User stories: 1. I need to be able to easily set up thresholds for different items in the household's fridge 2. I need to be able to get statistics over the different used items in the household 3. I need to, at any time, monitor my garage

Quality Attributes	Description
Modifiability Scalability Multi-Tennancy Useability Performance	Paul is working as a Software Architect/Developer of the company that offers the Smart Home solution to customers. He has 2 years of experience as a software developer and 2 years as a solution architect in another company. He has taken the job to gain experince with cloud based solutions John is very motivated to build a high quality product. My user stories: 1. I need to be able to easily change and deploy the applications that are used to manage and monitor the Smart Home. 2. I need to be able to develop the same GUI for several type of devices, eg. smart phones and tablets. 3. I need the application to be responsive in a setup with many concurrent users 4. I need a modular and manageable architecture that separates the application into components that are easily replaceable 5. I need easy access to test environments that can be configured before testing and discarded after testing 6. I need the platform to provide natural support for the separation of model and view components.

Quality Attributes	Description
Useability Performance Security Availability	Giovanni is a 57-year-old solicitor in London. He is married to Elise and have no children. Together they own a modern apartment in London and a small cottage in southern France. They drive to France to stay in the cottage as often as possible, but they usually arrive late, so they have often had problems with buying food when they arrive. Giovanni's main interest is art, and he collects modern art witch he buys from galleries and at auctions. He collects mainly sculptures and abstract paintings. The value of his collection is now considerable, and he is worried of burglary, flooding, or fire. Giovanni wants a system that can monitor both the apartment and cottage. He also wants to use the system to check the stock level of food and other consumables so they can restock via the system or give him the possibility to buy I advance on the way to the cottage. It is therefore important that he can access the system on the internet. Furthermore, he wants the system to alert the fire brigade in case of a fire or the police in case of a burglary or a plummer in case of a flooding.

4 QUALITY ATTRIBUTE SCENARIOS

This section describes some of the relevant Quality Attribute Scenarios of the system.

The following tables describe the selected quality attributes and describe the motivation behind the choice.

Quality Attribute	Motivation
Availability	Availability covers the system's ability to survive situations like power or communications failure. The users of a Smart Home IoT system need to be guaranteed that the system acts correctly in the case of an emergency.
Scalability	A distributed cloud solution which should be able to manage and control potentially all the devices/sensors in all the households on the planet, face some pretty clear scalability challenges, which makes this quality attribute absolutely key.
Modifiability	A key function of the system is the ability to create and manage devices in the Smart Home. Furthermore the users should be able to manage e.g. food thresholds for the given household. Therefore we believe that modifiability is an important quality attribute of the proposed solution.
Performance	The performance of the system is also an important factor. The obvious example is in case of an emergency (e.g. fire), where the system *must* be able to take the correct action, as well with the desired response times.
Security	Security is the last quality attributed which we see necessary to include as relevant for the solution. The reason for choosing this is also quite obvious, since the system also acts as a security system for the given household as a whole.

4.1 AVAILABILITY

A device connected to the system must be able to communicate with the system when a power failure occurs.

Elements	Refined General Scenario
Source	Device
Stimulus	Service call to device or data logging from device
Environment	Power failure
Artifact	System
Response	A device must be able to route messages to the system when a power failure occurs.
Response Measure	System should be able to route service call within 1 seconds from sending system to device.

A device connected to the system must be able to communicate with the system when the internet connection fails.

Elements	**Refined General Scenario**
Source	Device
Stimulus	Service call to device or data logging from device
Environment	Internet failure
Artifact	System
Response	A device must be able to route messages to the system when a power failure occurs.
Response Measure	System should be able to route service call within 1 seconds from sending system to device.

4.2 Scalability

The system should be able to contain and manage at least 100 devices pr. customer instance.

Elements	**Refined General Scenario**
Source	Device
Stimulus	Service call to device or data logging from device
Environment	Normal condition
Artifact	System
Response	System should route messages to and from devices and application without delay
Response Measure	System should be able to route service call within 1 seconds from sending system to device.

The system should be able contain and manage concurrent devices pr. customer instance.

Elements	**Refined General Scenario**
Source	Device
Stimulus	Concurrent calls to device or data logging from device
Environment	Normal condition
Artifact	System
Response	System should route messages to and from multiple devices and application without delay
Response Measure	System should be able to route concurrent service calls within 1 seconds from sending system to devices

4.3 MODIFIABILITY

The system should enable dynamically to allocate and remove devices to and from the house management system.

Elements	Refined General Scenario
Source	User Interface
Stimulus	Register new device
Environment	Normal condition
Artifact	System
Response	System should be able to interact with new device without restart
Response Measure	New device should be immediately manageable

The system should enable dynamically to allocate and remove Consumable to and from the house management system.

Elements	Refined General Scenario
Source	User interface
Stimulus	Register new Consumable
Environment	Normal condition
Artifact	System
Response	System should be able to register new Consumables without restart
Response Measure	New Consumable should be immediately manageable

The system should enable dynamically to setup threshold values for Consumables to and from the house management system.

Elements	Refined General Scenario
Source	User interface
Stimulus	Setup threshold values for Consumable
Environment	Normal condition
Artifact	System
Response	System should be able to setup threshold values for Consumables without restart
Response Measure	Consumable threshold policy should be immediately in effect

4.4 Performance

The system must prioritize certain types of messages – e.g. fire alarms.

Elements	**Refined General Scenario**
Source	System
Stimulus	A fire alarm call from a device
Environment	System is busy
Artifact	System
Response	The system must ensure that certain kinds of messages must be prioritized, so fire alarms are not delayed in the system.
Response Measure	System should be able to route fire alarm calls to external service providers within 5 seconds.

4.5 Security

The system should identify the user based on the device logging on to the Wi-Fi network.

Elements	**Refined General Scenario**
Source	Security device
Stimulus	Device arrives at house Wi-Fi perimeter
Environment	Normal condition
Artifact	System
Response	User should be logged on the system
Response Measure	Request user pin code to verify identity

The system should enable setting up difference user groups for the system.

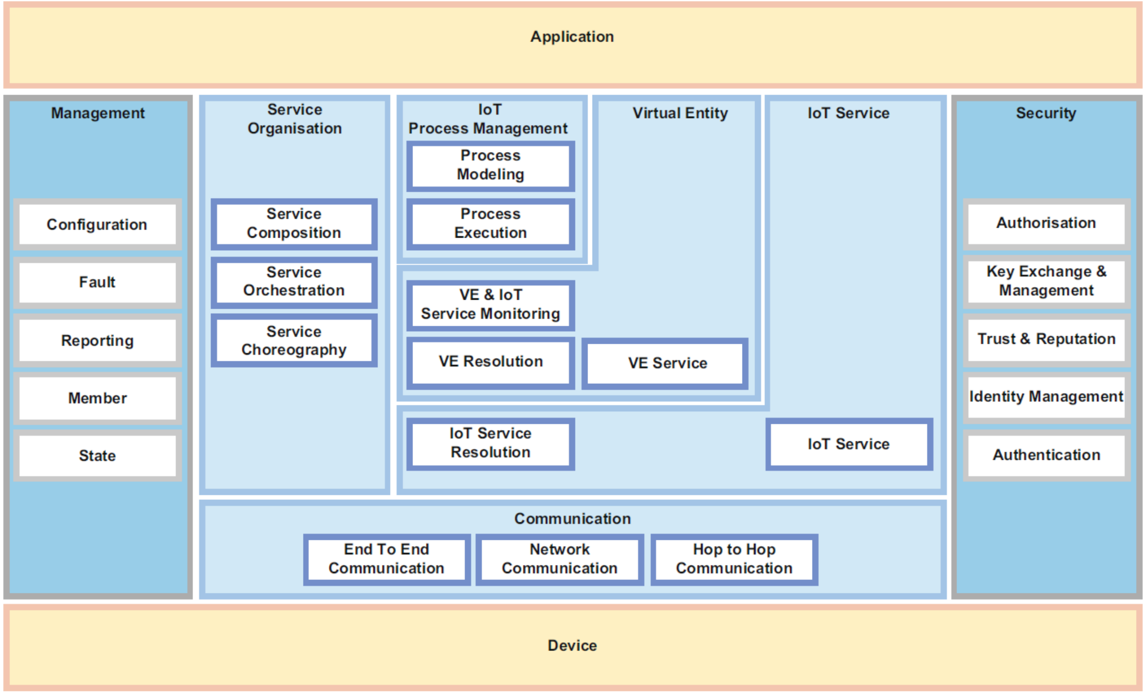
Elements	**Refined General Scenario**
Source	Administration
Stimulus	User wants to setup the different User groups to apply for the system
Environment	Normal condition
Artifact	System
Response	New User groups have been enabled without restart
Response Measure	New User groups security levels have been defined

5 IoT Architecture - IoT in a Smart Home

The section serves the purpose of explaining which key parts from the IoT reference architecture that is being used in the solution which we offer. Furthermore we argue why certain parts of the original IoT reference architecture is left out.

For each of the below Groups a description will be given, containing either an explanation of the purpose of the Group (*in our proposed architecture*), or reason for being omitted in the overall solution.

This is done in order to give the reader a high-level understanding of how we actually used the IoT reference architecture. In the next section we present the actual system architecture description.



5.1 Application

This is the top layer of the whole architecture - the application layer of the architecture.

5.2 Management

Configuration of the system is handled through the embedded components in this Group, but also the ability for the system to keep track of the different Devices current configuration. Furthermore this Group is responsible for error handling in the system. If a given component (in the Architecture) is triggering an error, all messages should go through this Group in order to identify the nature and severity of the problem. Error logging should also be done at this location.

5.3 Service Organization

This Group is not included in the proposed architecture due to the set of services being an already defined set. More precisely we see this Group as offering the possibility for the users to create and modify given services which the system offers, but in the proposed solution this is something that the users are not able to do. The reasoning behind this design choice is that we believe that handing out this functionality for the users, is something that would complicate the usage of the system to a level which is above the benefits which this may cause.

5.4 IoT Process Management

All the different processes exposed by the system are modelled in this Group. This is where the process management is handled, e.g. what happens when a Fire alarm is triggered. Execution of processes is triggered here, and the piping of information (from e.g. a device) to the correct receivers as well.

5.5 Virtual Entity

The Virtual Entity Group describes the actual entities in the system (e.g. devices). Furthermore entity specific services are also present in this Group. Registering and discovery of new devices, dispatching of relevant information between devices and getting the different device status is handled within this Group.

5.6 IoT Service

The concrete services within the system reside in this Group. We’ve chosen to split the exposed service functionality into six distinct groups, in order to visualize the services which the system offers in a more structured way. These are the following:

- Alert services
- Order services
- Climate control services
- Pet feeding services
- House Control Access services
- Device control services

Each of these contains the desired functionality, related to the different areas of functionality which the system offers.

5.7 Communication

This Group maintains communication between the different Groups in the IoT architecture. Furthermore this is where communication between the different devices is modelled. Later we present our Physical view of the communication layer, which goes more in-depth on how the actual communication is being handled. The devices are able to communication over Wi-Fi but also via 4G network, in case of the internet connection being unavailable.

5.8 Security

This Group of functionality handles User authentication, and should prevent any tempering of data and/or information. It should also enable the possibility of creating User Groups, so that the users of the system are able to differentiate on the level of administration rights within the system, e.g. changing preferences stores within the system.

5.9 DEVICE

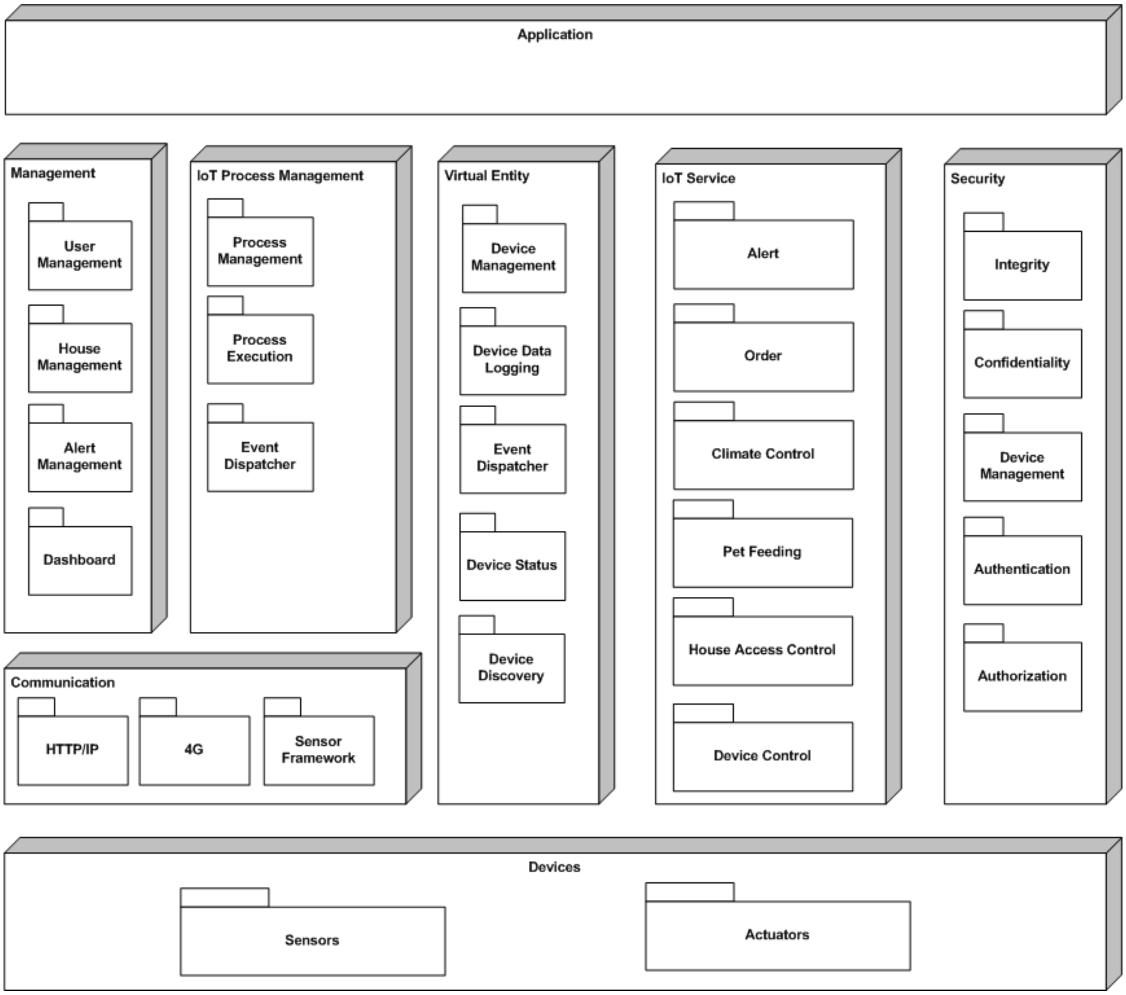
This Group is where the different devices are being handled. We've chosen to divide the actual physical devices into sub-groups, depending on role and behavior. In the following we present this division.

6 SYSTEM ARCHITECTURE DESCRIPTION

With the above in mind, we now present the system architecture description for the solution which we propose. For each of the above stated Groups, we present the related services which the architecture must enable. Responsibility and methods are also declared for each of the Groups.

First of all we present the context diagram (in relation to the above picture of the IoT reference architecture), which should give the reader an overview of our architecture.

6.1 CONTEXT DIAGRAM



6.2 MANAGEMENT SERVICES (AND MONITORING)

Service Name	User Management
Responsibility	Management of Users and Pets
Methods	- CreateUser - GetUser - UpdateUser - DeleteUser - CreatePet - GetPet - UpdatePet - DeletePet
Collaboration	House Management

Service Name	House Management
Responsibility	Management of House and Rooms
Methods	- CreateHouse - GetHouse - UpdateHouse - DeleteHouse - CreateRoom - UpdateRoom - DeleteRoom
Collaboration	User Management, Device Management

Service Name	Alert Management
Responsibility	Management of Alerts. Alerts are rules that describes who should be alerted in different situations. External and internal receivers are also managed.
Methods	- CreateAlert - GetAlert - UpdateAlert - DeleteAlert - CreateReceiver - AddReceiverToAlert - RemoveReceiverFromAlert - AddEventTypeToAlert - RemoveEventTypeFromAlert
Collaboration	User Management, Device Management

Service Name	Dashboard (not UI)
Responsibility	Listing relevant management and runtime information for use in applications like web applications or mobile apps
Methods	- GetAlerts - GetUserStatus - GetRessourceStatus - GetSensorStatus
Collaboration	User Management, Device Management ...

6.3 IOT SERVICES

Service Name	Alert
Responsibility	This service executes Alerts based on rules from alert configuration.
Methods	- Alert
Collaboration	Alert Management, Process Execution

Service Name	Order
Responsibility	This service executes an Order to a predefined Order service . It can order any type of products that are predefined in the product catalog.
Methods	- Order
Collaboration	Process Execution

Service Name	Climate Control
Responsibility	This service controls the devices that are used for climate regulation in the house. This includes aircondition, heat and light.
Methods	- Alert
Collaboration	Alert Management, Process Execution

Service Name	Pet Feeding
Responsibility	This service controls the pet food dispensers in the house.
Methods	- Feed(Pet)
Collaboration	Process Execution

Service Name	House Access Control
Responsibility	This service controls the devices that are used for accessing the house such as locks on doors and windows.
Methods	- Unlock(Door/Window) - Lock(Door/Window) - Status(Door/Window)
Collaboration	Dashboard service, Process Execution

Service Name	Device Control
Responsibility	This service controls the devices that are turned on or off such as coffee machines and tv's.
Methods	- On(Device) - Off(Device) - Status(Device)
Collaboration	Dashboard service, Process Execution

6.4 IOT PROCESS MANAGEMENT SERVICES

This package contains services for process management and configuration, process execution and the specific process services

Service Name	Process Management
Responsibility	This service manages and configures the predefined processes. The Fire Process is executed when a fire is detected The UnauthorizedAccess Process is executed when a UA is detected
Methods	- Enable(Process) - Disable(Process) - SetSchedule(Process) - FireProcessConfiguration(FireConfig) - UnauthorizedAccessProcessConfig(UAConfig)
Collaboration	Process Execution

Service Name	Process Execution
Responsibility	This service executes the specific activities in the processes
Methods	- Execute(Process)
Collaboration	

Service Name	Event Dispatcher
Responsibility	This service receives and parses events from the devices in the system and starts process execution based on the process configurations. If a fire event is received from a device in the system, the Fire Process

	is started
Methods	- EventHandler(Event)
Collaboration	

6.5 Virtual Entity Services

Service Name	Device Management
Responsibility	This service registers devices in the system
Methods	- CreateDevice - GetDevice - UpdateDevice - DeleteDevice
Collaboration	

Service Name	Device Data Logging
Responsibility	This service collects and stores data from devices that are logging data into the system
Methods	- Logevent
Collaboration	

Service Name	Event Dispatcher
Responsibility	This service dispatches events to relevant subscribers
Methods	
Collaboration	

Service Name	Device Discovery
Responsibility	This service is responsible for discovery of active devices in the network and make them available for management
Methods	
Collaboration	

Service Name	Device Status
Responsibility	This service is responsible for monitoring the status of devices in the system on a schedule. If a device is offline an alert should be generated.
Methods	

Collaboration	

6.6 DEVICES

The physical devices are divided into several groups depending on their role and behavior. All devices are constructed from two main types of components: Sensors and Actuators.

The following table shows the main device types that play a role in the smart house and the type of sensors and actuators needed to realize the functionality.

Device Type	Smart Home Functionality	Sensors	Actuator
TV	On/Off on schedule.	Power Status (On/Off)	On/Off Power Switch
Coffee machine	On/Off on schedule.	Power Status (On/Off)	On/Off Power Switch
Refrigerator	Status on inventory	Inventory type selector Inventory increase/decrease buttons	N/A
Pet Feeding Device	Feed Pet on schedule	Food in Dispenser Status. (Yes/No)	On/Off Switch to control motor on dispenser
Fire Alarm	Alert in case of fire	Smoke Sensor (On/Off)	N/A
Door/Window Lock	Control Door Locks	Door Status Open or Closed (On/Off)	Switch to control motor on/off and direction that locks or unlocks door
Card Reader	Provide User Identity	Smart Card Reader(Cert) Pin Code Pad (Pin)	N/A
Climate Control	Control aircondition and lights	Humidity and temperature sensors	Motor controlled termostats or aircondition devices Motor controlled light regulators
Garage Door	Open or close the garage door when car arrives or leaves the home	Photoelectric sensor	Garage door motor
Movement Sensor	Detecting movement in a room		
Rome Access Detection		Photoelectric sensor	

There are several standards emerging in the IoT domain that seeks to standardize the way these type of devices are connected to a larger system as the one being designed here.

As this assignment is not about these standards one that looks simple and appropriate for this case has been chosen as a base for further design activities.

6.7 Device discovery

Device discovery is needed to establish link devices to the Smart Home System. There are several standards for this process. As this is not a primary topic of the assignment this will not be described in detail, but the system needs to have this type of functionality.

The main technologies in this area are “Bluetooth beacons”, “Wifi Aware” and “Physical Web”.

Device registration and management is based upon a standard from Open Mobile Alliance called OMA Lightweight M2M Standard (LWM2M) wich can be used for both management and application data.

Device management includes:

- Bootstrapping
- Device Configuration
- Firmware Update
- Fault Management: Report errors from device, Query about Status

Application includes:

- Configuration and control: Configure settings, Send control commands
- Reporting: Notify changes in sensor values, Notify alarms and events

Objects/Resources are accessed with simple URIs: /{Object ID}/{Object Instance}/{Resource ID}

e.g. /3/0/1 (Device Object, Manufacturer Resource)

This type of Rest based interface makes it very easy to monitor or control a given device in the Home network.

On a practical level devices are registrered within the Smart Home application with a web application that is executed on a pc that is connected on the Home-network. Every device or device-hub in the network should send out a beacon to the PC wich makes it visible and managable. There are several standards for device discovery such as bluetooth beacons (e.g Google Eddystone, Apple iBeacon) or Wifi Aware from Wifi Alliance and Physical Web.

It is therefore assumed that devices relstively easy can be discovered registered and managed in a solution that is deployed either on premises (in House) or somewhere outside of the house.

http://openmobilealliance.org/

http://postscapes.com/iot-device-discovery

6.8 Security

We need the following security services in the system:

Integrity & Confidentiality:

The system must provide data integrity to ensure that altering of data is prevented. The system must ensure encryption between the Smart Home and the Cloud solution. Some of the information that are generated and processed by the system could be sensitive and therefore the system needs to provide data and message confidentiality. This should also be ensured by encryption.

Authentication:

The system should support secure authentication of the users. Both accesses to the system functionality and to the house itself should be based on proper authentication. In the proposed solution the user accesses a Smart Phone Application, on which he/she authenticate when arriving at the Smart Home. Furthermore this application requires a pincode in order to be accessed.

Authorization:

The system should provide possibility to setup user groups, so that certain users in the household can access functionality which other users cannot. E.g. changing food preferences for the household should only be done by certain users. The system must therefore enforce user roles, to handle this requirement.

7 Developing a High-Level Architecture

The overall system architecture is distributed by nature as it includes many different devices that need to communicate with the main system and the stakeholders of the system.

The candidates for the main architectural style for the solution where ***broker*** and ***layered*** architectures.

Broker:

This pattern is used where processes and services typically are distributed across multiple nodes. The pattern provides location transparency to the service consumers and a centralized repository for service resolution.

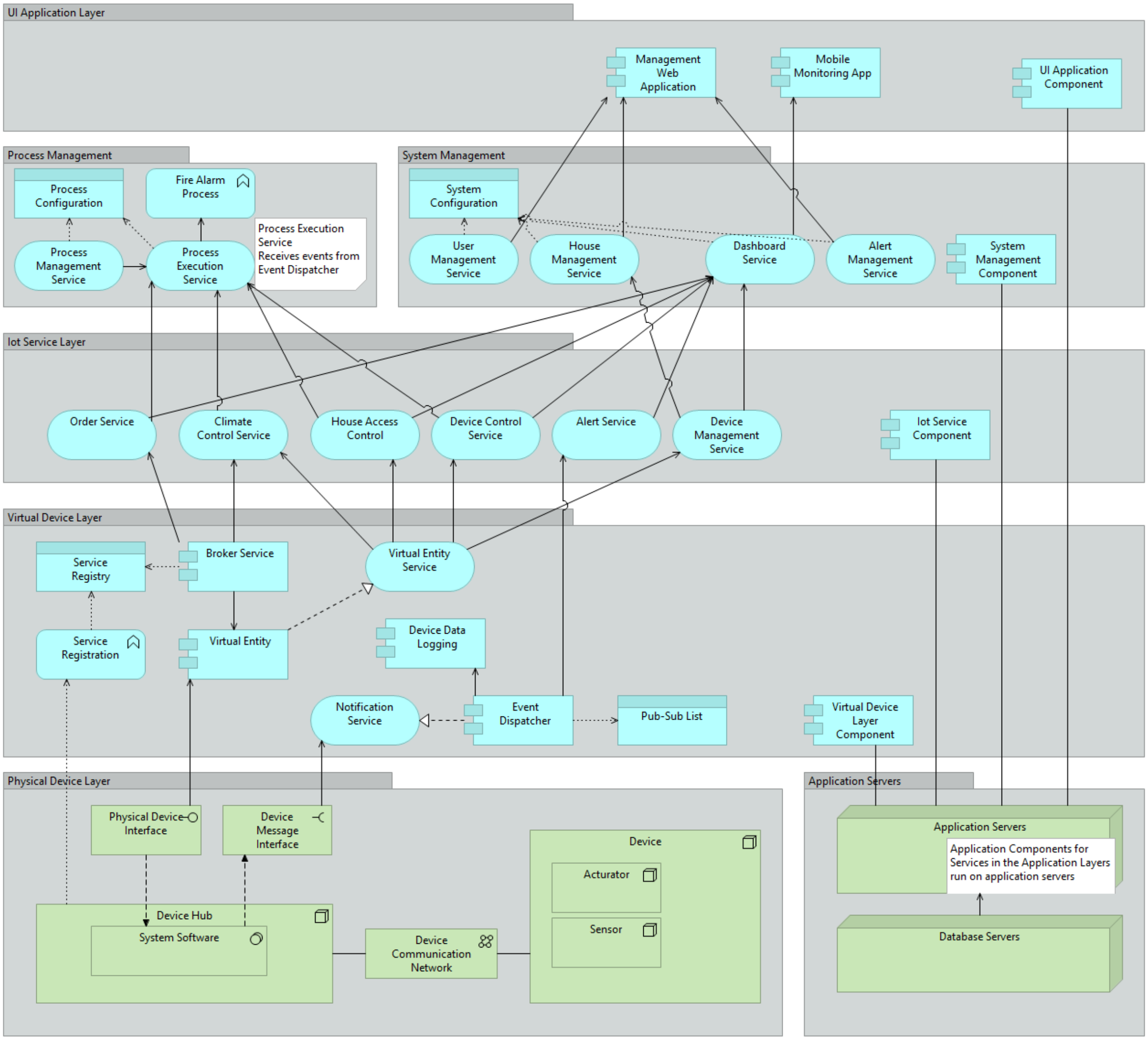
Layered Architecture:

The layered architecture is used to reduce complexity in a system. A layer typically includes related functionality such as resource access functionality, business logic or presentation.

The final design of the Smart Home System applies the two architectural styles together in the following way:

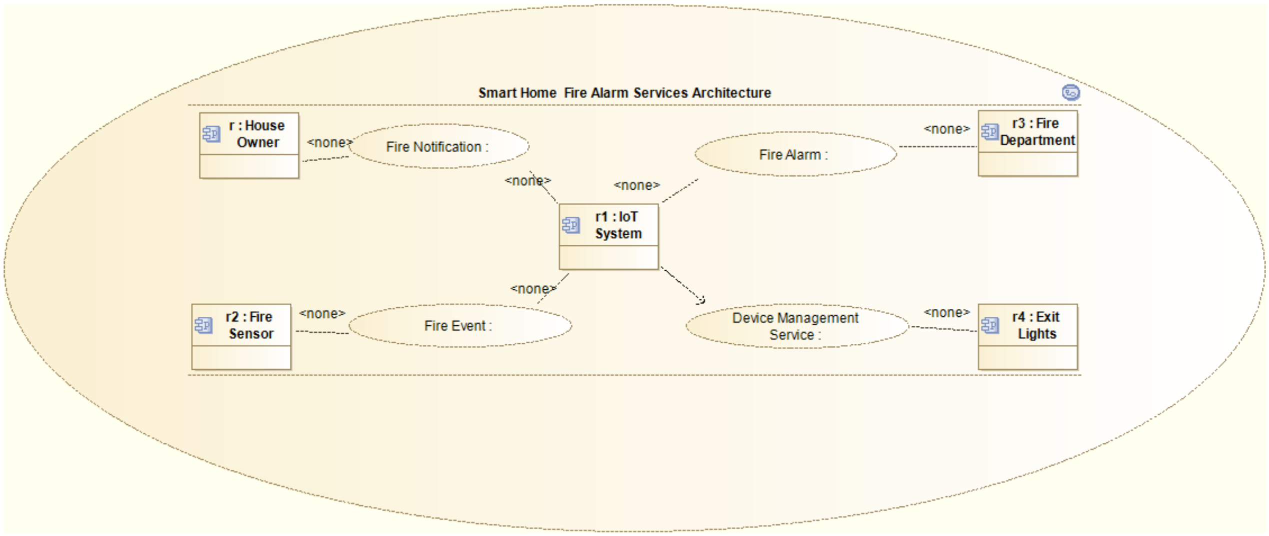
The discovery and registration of devices and services uses the broker architecture to provide easy and uniform access to these services for the other components in the system. As the reference architecture is divided into several areas that also represent different levels of granularity, the layered architecture is combined with the broker style to provide the final system architecture.

The overall system architecture is presented as a figure in the following page.

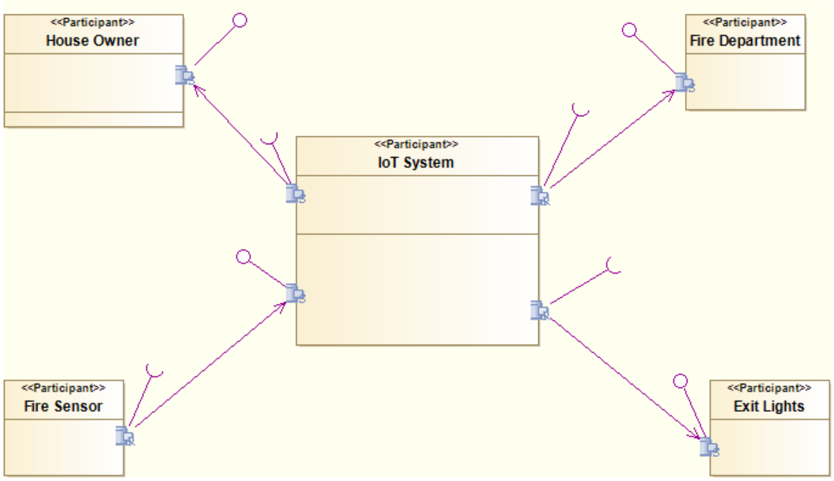

UI Application Layer
Management Web Application
Mobile Monitoring App
UI Application Component
Process Management
Process Configuration
Fire Alarm Process
Process Management Service
Process Execution Service
Process Execution Service Receives events from Event Dispatcher
System Management
System Configuration
User Management Service
House Management Service
Dashboard Service
Alert Management Service
System Management Component
Iot Service Layer
Order Service
Climate Control Service
House Access Control
Device Control Service
Alert Service
Device Management Service
Iot Service Component
Virtual Device Layer
Service Registry
Broker Service
Virtual Entity Service
Service Registration
Virtual Entity
Device Data Logging
Notification Service
Event Dispatcher
Pub-Sub List
Virtual Device Layer Component
Physical Device Layer
Physical Device Interface
Device Message Interface
Device
Acturator
Sensor
Device Hub
System Software
Device Communication Network
Application Servers
Application Servers
Application Components for Services in the Application Layers run on application servers
Database Servers

7.1 Service Architecture and patterns

This is an example of how the Fire Alarm Service Architecture looks like:



And the related participants diagrams:

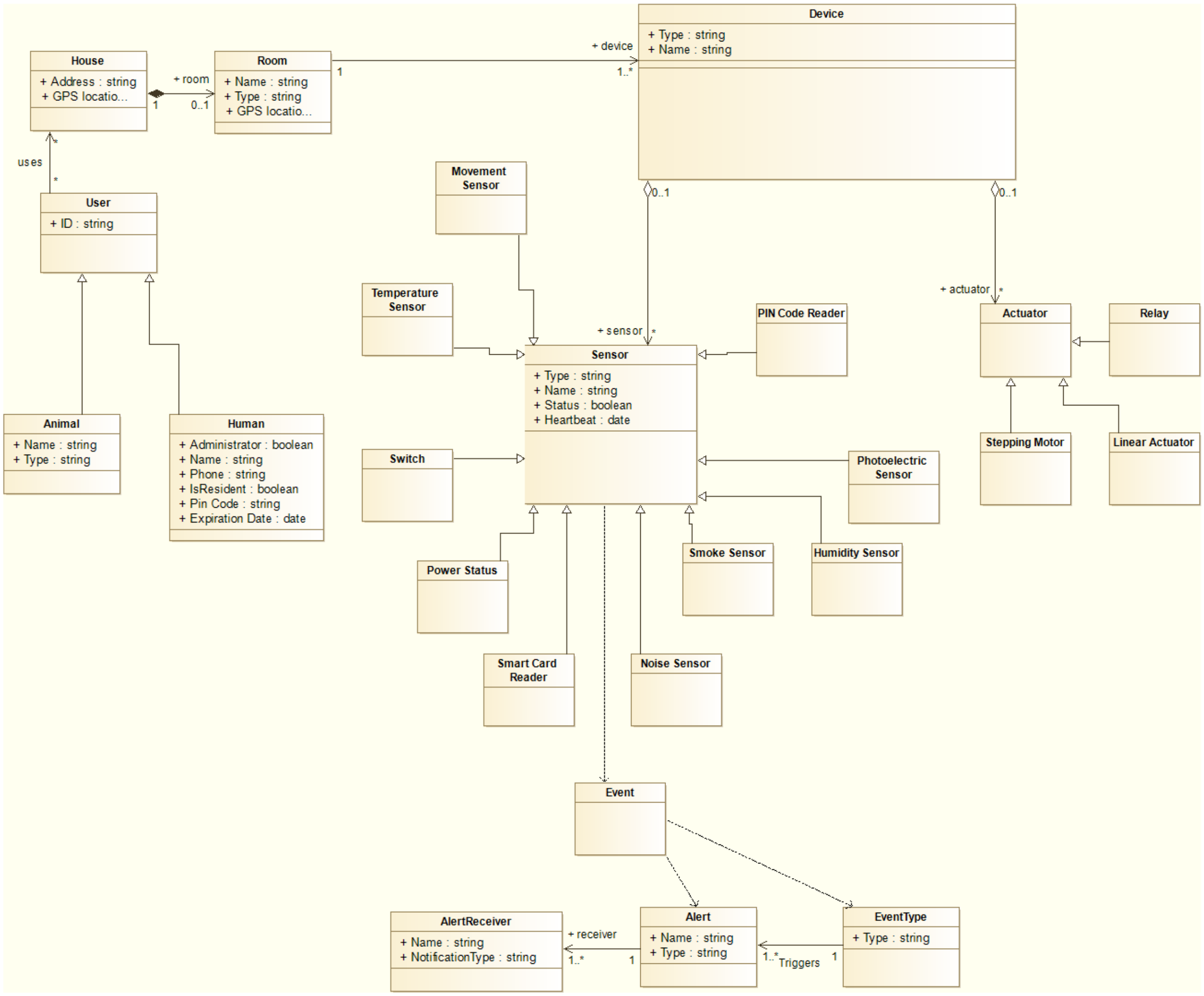


The following SOA patterns (soapatterns.org) have been applied to the solution:

- Service Layers: The services have been divided into layers as described in the overall system architecture.
- Entity Abstraction: This pattern has been used in the Virtual Entity Device Design
- Process Abstraction: Being used in the two business processes that interacts with external partners. These processes executes alone while utilising Task Services in the Iot Services Layer.
- Service Facade: The Device Control Service witch acts as a facade to the devices
- Event Driven Messaging: The notification service and the event dispatcher notifies subscribers of events coming from the devices
- Asyncronous Queing: The notification service and the event dispatcher should be connected with a message queue to ensure event persistens and possibly implement message priority.

The earlier depicted example on how to implement services with the SOA patterns. The accompanied SoaML diagrams are incomplete, but section 6 contains the different methods and services.

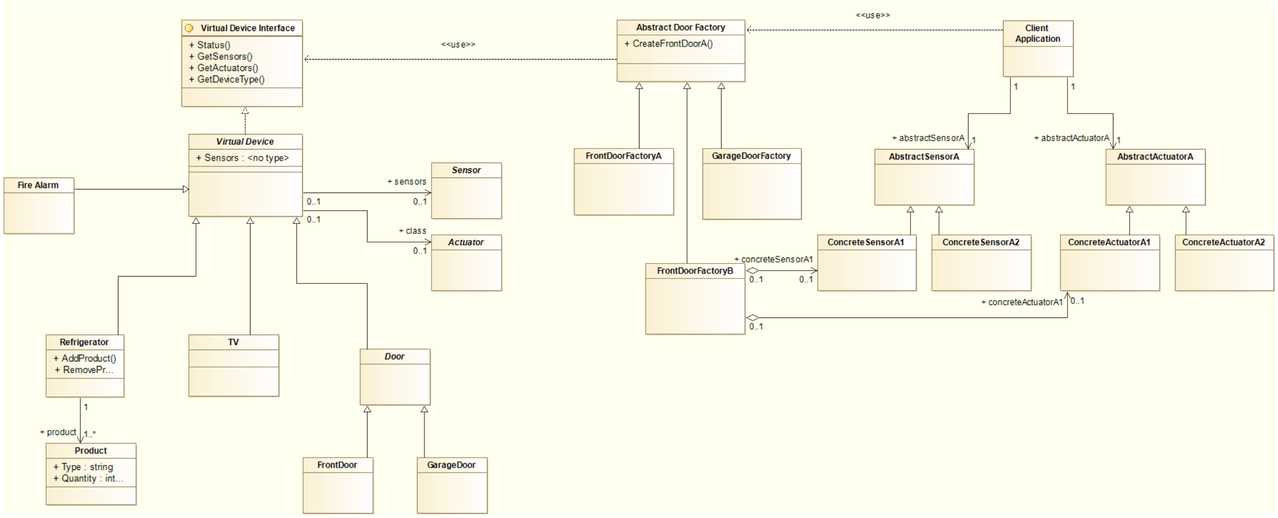
7.2 LOGICAL VIEW



The class diagram for the virtual devices is structured around a Virtual Device Interface that each of the devices implement. An abstract class called Virtual Device holds the list of sensors and actuators for each device.

The abstract factory pattern is deployed for creation of the virtual devices. This pattern supports the creation of many different types of virtual devices and device familes. The concrete factories holds the logic for the construction of the concrete devices and what sensors and actuators to install.

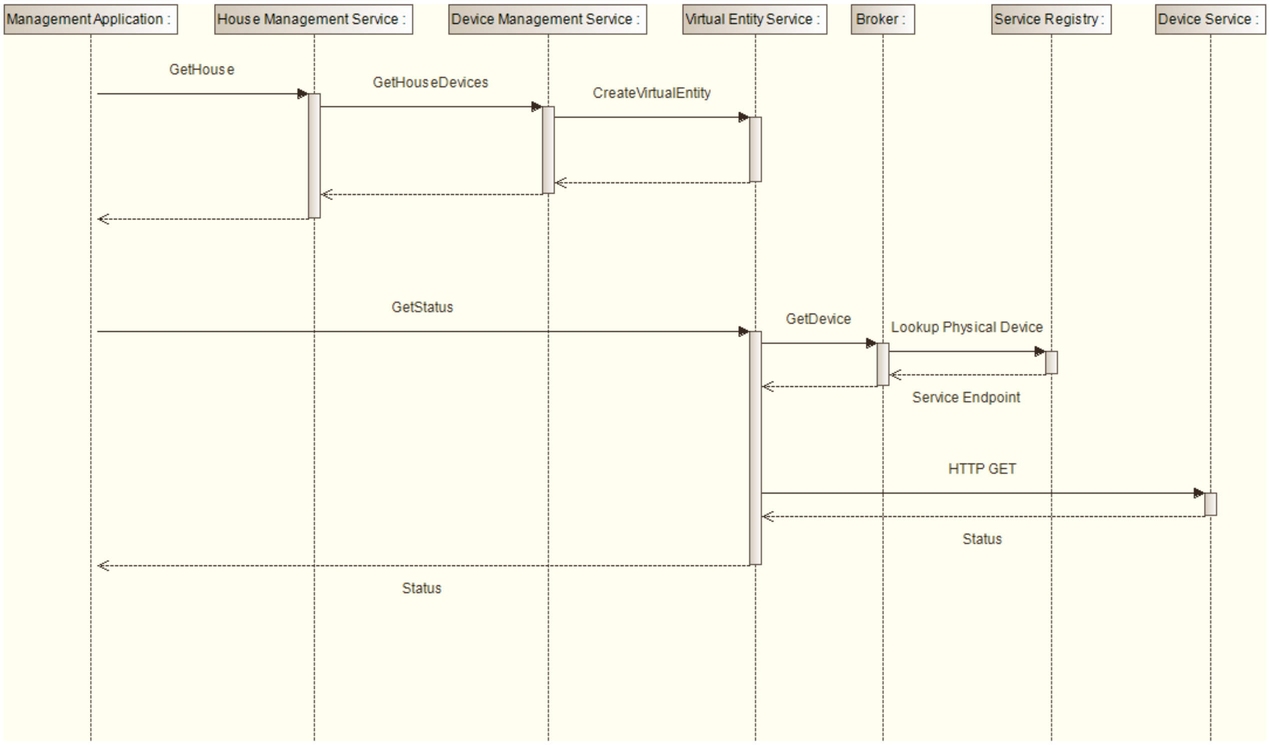
The figure is not complete as there would be a similar factory patter for each family of devices. Some of the simpler devices could be created with a traditional factory pattern.


Virtual Device Interface
+ Status()
+ GetSensors()
+ GetActuators()
+ GetDeviceType()
<<use>>
Abstract Door Factory
+ CreateFrontDoorA()
<<use>>
Client Application
1
1
Virtual Device
+ Sensors : <no type>
Fire Alarm
+ sensors
Sensor
0..1
0..1
0..1
+ class
Actuator
0..1
FrontDoorFactoryA
GarageDoorFactory
+ abstractSensorA
1
AbstractSensorA
+ abstractActuatorA
1
AbstractActuatorA
ConcreteSensorA1
ConcreteSensorA2
ConcreteActuatorA1
ConcreteActuatorA2
+ concreteSensorA1
FrontDoorFactoryB
0..1
0..1
+ concreteActuatorA1
0..1
0..1
Refrigerator
+ AddProduct()
+ RemovePr...
TV
Door
1
+ product
1..*
Product
+ Type : string
+ Quantity : int...
FrontDoor
GarageDoor

7.3 PROCESS VIEW

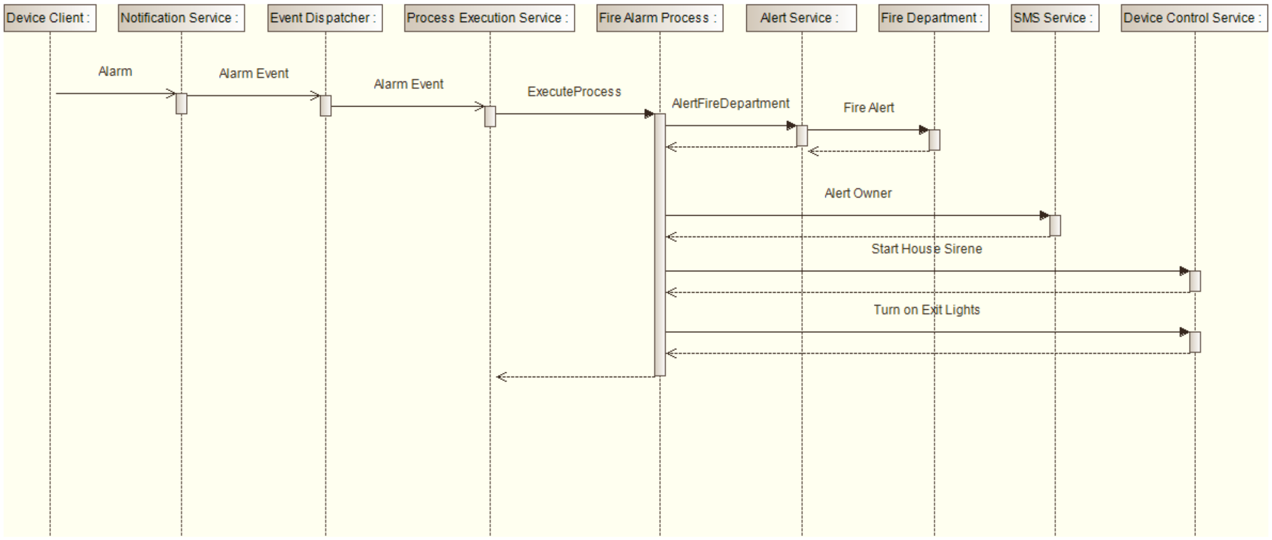
7.3.1 Virtual Device Sequence Diagrams

The following shows an example of how the management application retrieves status from a device in the house. The broker and the service registry decouples the client (Management application) from the server (Device)

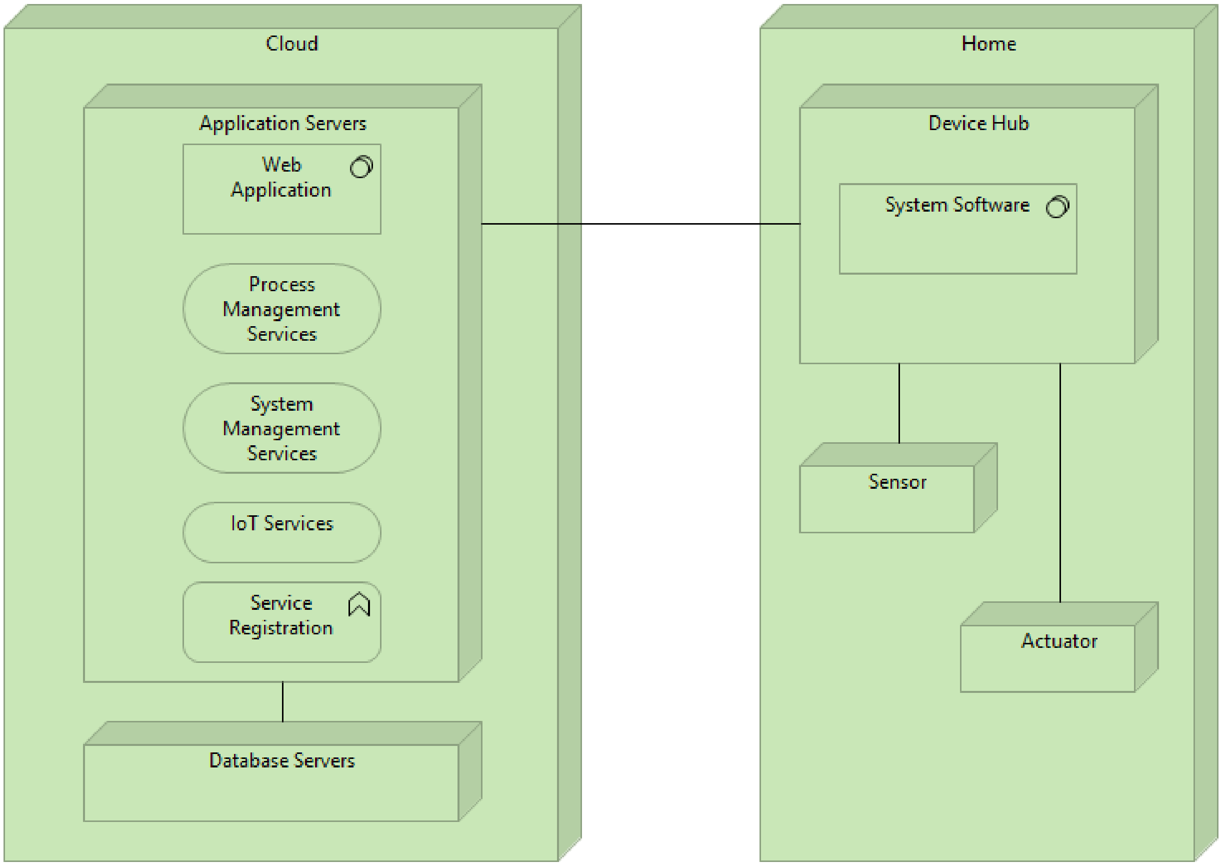


7.3.2 Alarm Sequence

The following shows an example event from a Fire Alarm, being recieved by the Event Handler, which notifies the process execution service. The process execution services starts the Fire Alarm process, which notifies the Fire Department and the owner and turns on the exit lights in the house.



7.4 Physical View



8 ARCHITECTURE DECISION LOG

This section contains descriptions of various design decisions we've made in order to identify the required services. The design decision descriptions will be divided into the five categories of quality attributes which we stated earlier in this report (Availability, Scalability, Modifiability, Performance & Security).

8.1 DESIGN DECISIONS

8.1.1 Availability

Concern (Identifier: Description)		T system must be able to communicate information about emergency alarms (e.g. fire alarm) even in the case of a power failure.
Ranking criteria (Identifier: Name)		1. Availability
Options	Identifier: Name	1. In-house emergency routine storing
	Description	A solution where the given Smart Home had some sort of device running (physical machinery, e.g. a dedicated computer), which could take over the handling of device information in case of power failure. This device should obviously be equipped with some sort of battery, in order to function without power. This way the different emergency processes would still be upheld by this device, even in case of power failure.
	Status	The option is declined.
	Relationship(s)	Alert handling
	Evaluation	The evaluation is that this solution would require additional support from the supplier. The supplier would have to ensure that this machinery would always be functioning, as well as guarantee that the hardware was not malfunctioned. Imagine having the entire worlds households connected to the solution, this would require some support.

	Rationale of decision	Since the system is thought out to be a complete cloud solution, which in theory should handle all the households in the world, this solution is simply not feasible.
	Identifier: Name	2. Cloud emergency routine storing
	Description	A solution where all emergency routines are stored in the Cloud, like the rest of the system. If a power failure occurs, the current devices would still be functioning, since they all have a backup battery. The given device would in case of an emergency realize that an internet connection was no available, and then use 4G (mobile communication) instead. This way the system would receive the emergency readings from the sensor, and proceed by activation the correct emergency process.
	Status	Option is decided.
	Relationship(s)	Alert handling
	Rationale of decision	This solution would require all emergency devices (the ones detecting e.g. a fire), to be equipped with a battery and a 4G SIM card. This way communication would always be ensured, even in the case of power failure.

8.1.2 Modifiability

Concern (Identifier: Description)		Should the users of the system be able to create their own processes? E.g. setting together some personal sequence of services which the system offers. For instance adding additional steps in one of the emergency routines.
Ranking criteria (Identifier: Name)		1. Modifiability
Options	Identifier: Name	Yes – the users should be able to create processes

	Description	A solution where each user has the ability to create and modify the different processes exposed by the system.
	Status	The option is declined.
	Relationship(s)	Process management
	Evaluation	*We believe that this would create a lot of uncertainty while using the processes. As a supplier you could no longer guarantee that your processes actually worked. Worst case scenario is imagining a fire emergency process which does not actually contact the correct Fire Department, due to the process being modified in an in-correct way.*
	Rationale of decision	*The reasoning behind this design choice is that we believe that handing out this functionality for the users, is something that would complicate the usage of the system to a level which is above the benefits which this may cause.*
	Identifier: Name	No – the system comes with predefined processes
	Description	A solution where all the given processes which the system exposes, is non-modifiable.
	Status	Option is decided.
	Relationship(s)	Process management
	Rationale of decision	Since we believe this would create a much more stable system, where users could actually trust the exposed processes, we have taken the decision of limiting modifiability in this regard.

8.1.3 Scalability

Concern (Identifier: Description)		The system must be able to handle a different numbers of devices from multiple users, and must be able to react on alarms without delays
Ranking criteria (Identifier: Name)		1. Scalability
Options	Identifier: Name	Own servers (Load balancing architecture)
	Description	It is an option to have a server solution to run the system. The System Owner will have a better control over the system and know where the data is stored. The system could for instance be based on a load balancing architecture where new servers can be added when new customers are added to the system.
	Status	This option is declined.
	Relationship(s)	
	Evaluation	If using a load balancing architecture new servers must be added when needed.
	Rationale of decision	This option is declined because it will require much work to keep the numbers of servers required if many there are many customers going to use the system. Furthermore, the number of devices for each customer can be as high as 100, so the impact of each new customer on the system can be hard to predict
	Identifier: Name	Cloud solution

	Description	It is an option to have the solution as a distributed cloud solution.
	Status	This option is decided.
	Relationship(s)	
	Rationale of decision	This option is decided because it will be easier to adjust the required capacity to the customer base. In the unlikely event of a reduced customer base, it is possible to scale down in the cloud.

8.1.4 Performance

Concern (Identifier: Description)		Ensuring that alarm messages (e.g. from a fire detector) is being handled within a limited timeframe
Ranking criteria (Identifier: Name)		1. Performance
Options	Identifier: Name	Prioritizing certain emergency messages
	Description	The Event Dispatcher shall be able to prioritize certain messages coming from the, potentially millions of, sensors. This way we can minimize the wait time for an emergency message.
	Status	This option is decided.
	Relationship(s)	Event Dispatcher

	Evaluation	Emergency processes must in nature be top priority in a system which handles the safety of people in a Smart Home. E.g. receiving messages about a low level of milk in the fridge, must never be evaluated before a message concerning e.g. a fire.
	Rationale of decision	Emergency messages are the single most important messages in the system; therefore these must be treated accordingly.
	Identifier: Name	No prioritizing needed
	Description	The event dispatcher simply computes messages as they come, and by guaranteeing fast computations (perhaps via a clever algorithm); we can ensure some maximum delay (worst case) off all message handling.
	Status	This option is declined.
	Relationship(s)	Event Dispatcher
	Rationale of decision	There simply is no argument for not handling emergency messages as top priority. The system, even using some "perfect" algorithm, should therefore always choose emergency messages over non-emergency ones.

8.1.5 Security

Concern (Identifier: Description)	The system shall be able to authenticate users in a secure way.
Ranking criteria (Identifier: Name)	1. Security

Options	Identifier: Name	The User uses a Smart Phone application, in order to request access to the Smart Home
	Description	When the given user arrives at his/her Smart Home, he/she is able to access a Smart Phone Application which can open up the House. This application is protected by a password, in order to be opened. This way the system knows when doors should be opened up, but also when a door is being opened without proper user authentication (hence starts a break-in process).
	Status	This option is decided.
	Relationship(s)	House Access Control
	Evaluation	The House Access Control is a key point in the security aspect of the solution. The system must ensure that no unauthorized people get access to the house. With that being said, different approaches might be considered, but since this one seems to be the one requiring the least support/interaction from the supplier, this is the best candidate.
	Rationale of decision	Other methods might also handle this requirement (see coming options), but this option separates itself from the rest, in a way that the supplier is not required to deliver additional hardware, in order to maintain a secure House Access Control. The assumption is, though, that users do have a Smart Phone, and in case it gets lost (the phone), that communication to other members of the house is possible, so that they can open up the house locks mechanisms.
	Identifier: Name	Card scanner
	Description	A solution where the users receive security cards (classic authentication security plastic cards) when buying the desired solution. Furthermore the given House gets a Card Reader installed, so that the users are required to swipe their card, as well as entering a e.g. 4-digit pin code.
	Status	This option is declined.
	Relationship(s)	House Access Control
	Rationale of decision	As a supplier this complicates things a lot. This creates the need for the supplier to manufacture these security cards, when new users get attached to the system, but also when existing cards gets lost. Furthermore the physical devices that are able to scan these cards must be installed at every Smart Home, which again, potentially could be the entire world population. The difference with a card reader device, compared to all other devices in a Smart Home, is that this would then be a minimum requirement for the system.

9 REFERENCES

Standards: OIC (Intel), Allseen (Qualcom)

http://www.datacenterknowledge.com/archives/2015/07/23/the-iot-standards-war/

Ziggbee – The wireless language

http://www.zigbee.org/

OMA (Open mobile Alliance)

http://openmobilealliance.hs-sites.com/lightweight-m2m-specification-from-oma

Device discovery Options

http://postscapes.com/iot-device-discovery

"DIAT: A Scalable Distributed Architecture for IoT" - by: Chayan Sarkar et. Al.

http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7000513

"IoT (Internet of Things) and DfPL (Device-free Passive Localisation) in a disaster management scenario" (Article in Simulation Modelling Practice & Theory, AUGUST 2012) – by: Gabriel Deak et. Al.

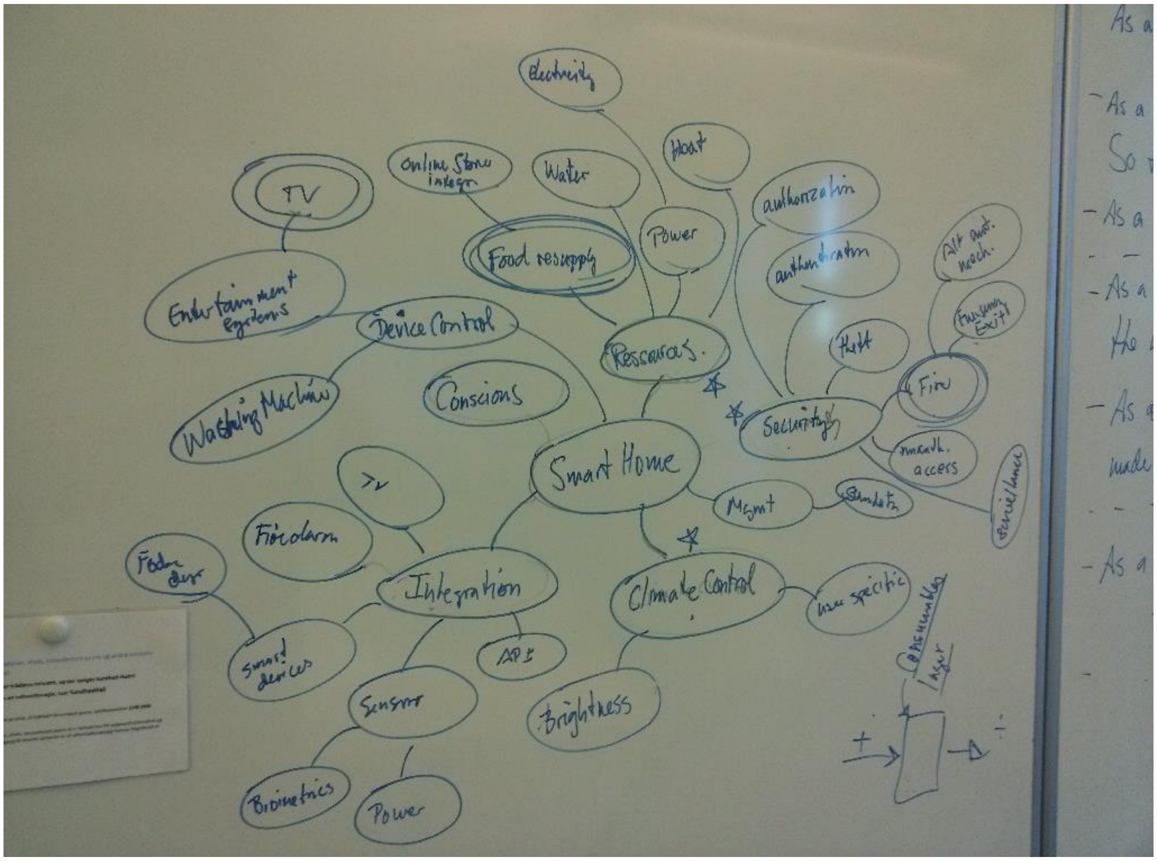
9.1 ADDITIONAL LINKS

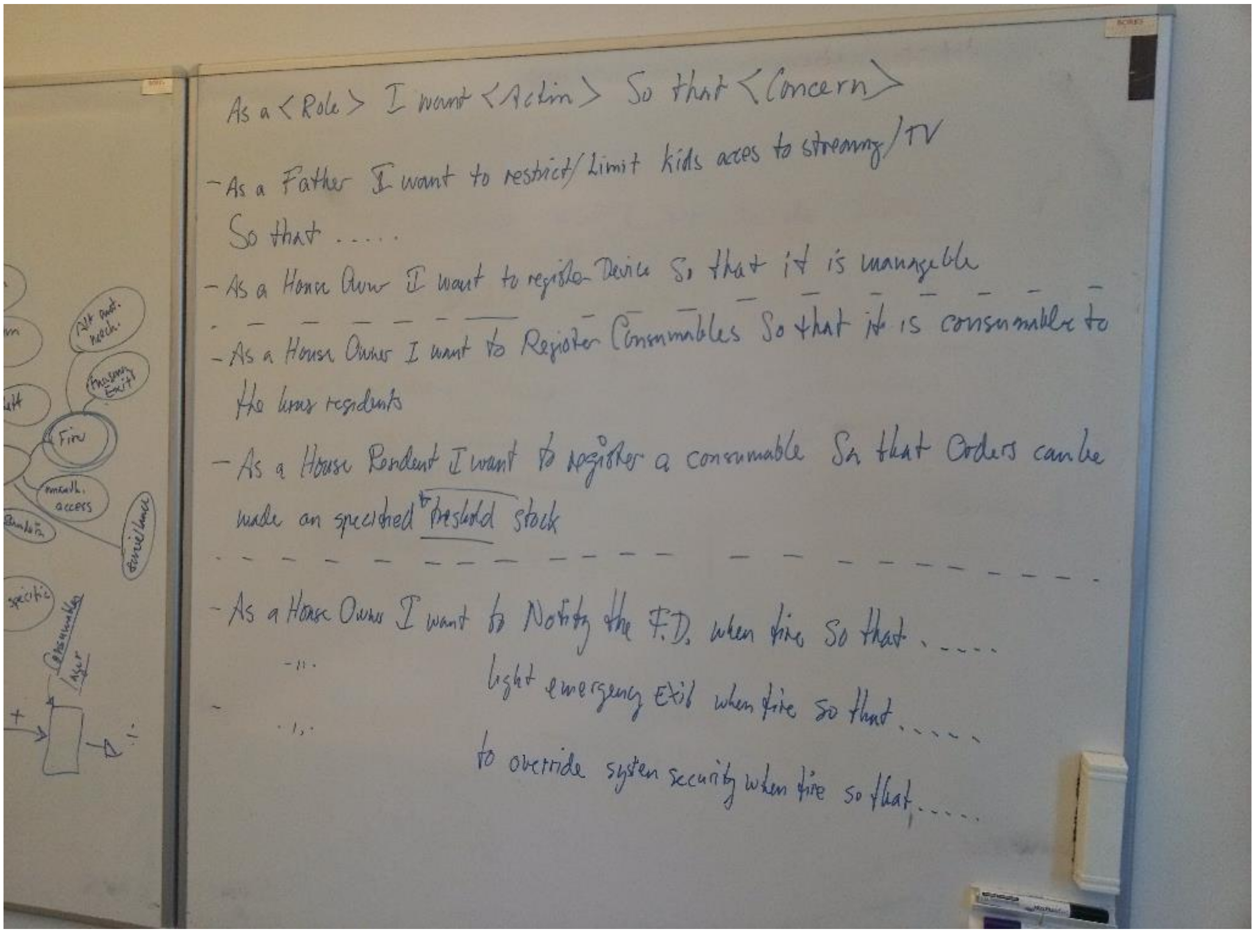
http://www.networkworld.com/article/2456421/internet-of-things/a-guide-to-the-confusing-internet-of-things-standards-world.html

http://electronicdesign.com/iot/open-standards-will-enable-iot-s-growth

10 NOTES

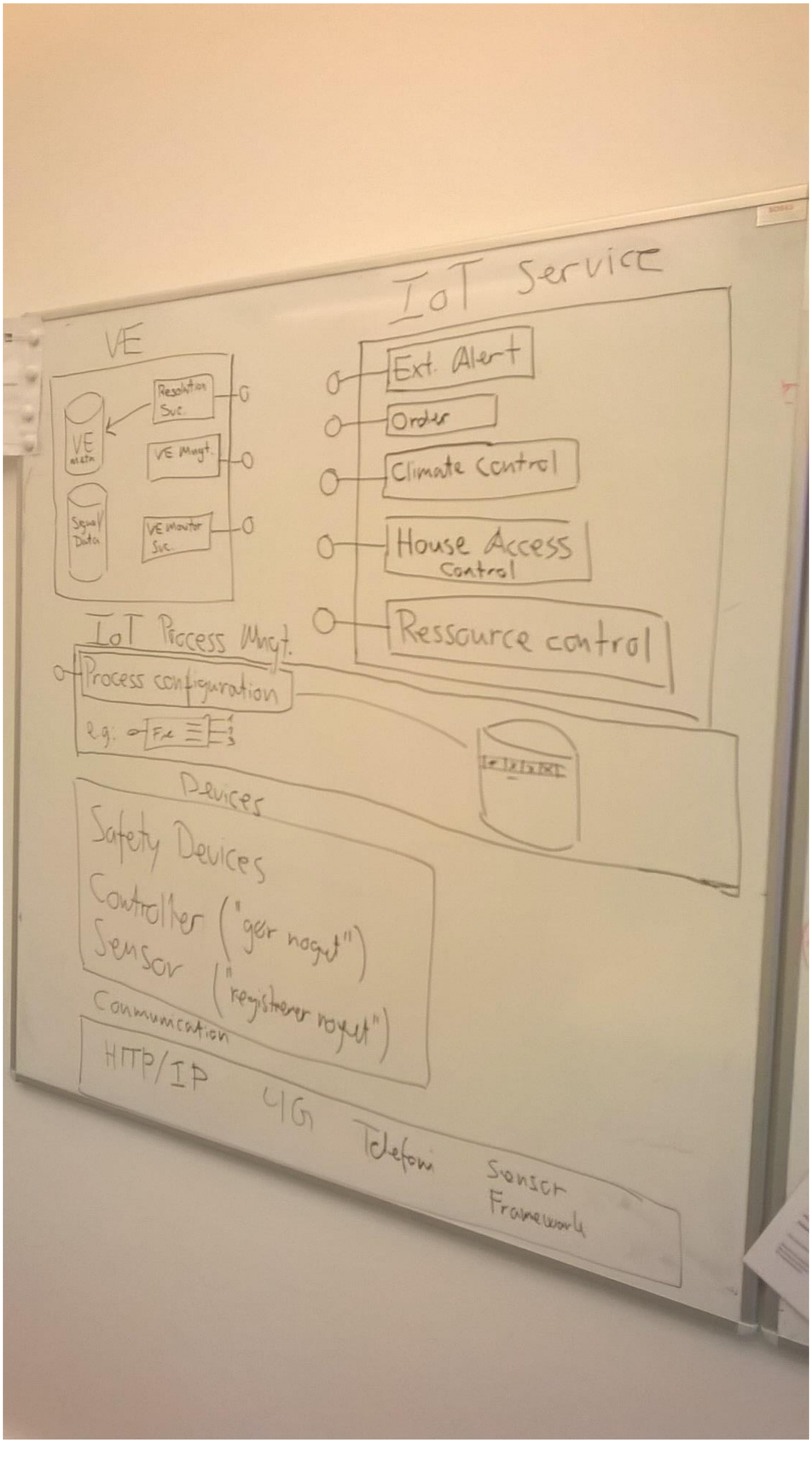
Pictures from meeting:





As a <Role> I want <Action> So that <Concern>

- As a Father I want to restrict/Limit kids aces to streaming/TV So that .....
- As a Home Own I want to register Device So that it is manageble
- As a House Owner I want to Register Consumables So that it is consumable to the home residents
- As a House Resident I want to register a consumable So that Orders can be made on specified threshold stock
- As a House Owner I want to Notify the F.D. when fire So that .....
- -"- light emergency Exit when fire so that.....
- -"- to override system security when fire so that.....


IoT Service
VE
Resolution Svc.
VE meta
VE Mngt.
Signal/Data
VE Monitor Svc.
Ext. Alert
Order
Climate control
House Access Control
Ressource control
IoT Process Mngt.
Process configuration
e.g.:
Fire
1
2
3
Devices
Safety Devices
Controller ("gør noget")
Sensor ("registrerer noget")
Communication
HTTP/IP
4G
Telefoni
Sensor Framework