Architecture Notebook v 1.4

Responsible team behind document: Architects

Important users of this document (Recipient): Developers

Short content summary: In this artifact, the architects maintain all of the important issues regarding the architecture of the application. The purpose is to document and explain architectural guidelines, decisions and constraints so that it is understandable for everyone involved in the project.

Version & Date: 1.4 2019-12-04

Inspected by Quality Coordinator: Emmelie Jeppsson
Date: 2019-12-09
Inspected by recipient: Anna Roth
Date: 2019-12-09

Version	Date	Authors	Reviewed by	Changes added
Pre study	2019-09	Philip Kantola, Anna Roth	-	First pre study version
1.2	2019-10	Philip Kantola, Anna Roth	Emmelie Jeppsson	New company doc. design. Added components and services modell.
1.3	2019-10	Philip Kantola, Anna Roth	Emmelie Jeppsson	Linked the requirements with their SRS id:s. Developed the comp. and ser. modell further.
1.4	2019-11	Philip Kantola, Anna Roth	Emmelie Jeppsson	Abstraction keys description. Developed the comp. and ser. modell further.

Author(s): (Development) Lead Architect Philip Kantola, Architect Anna Roth

MEDCOM

Table of contents

Author(s): (Development) Lead Architect Philip Kantola, Architect Anna Roth

MEDCOM

About the product

Medcom delivers a product tailored for Region Östergötland, the purpose of the product is to confirm if a burn injury is serious enough to send the patient to the national burn clinic located in Linköping. The product enables doctors from hospitals around Sweden to communicate pictures and information regarding specific cases of burn injuries with doctors located at the national burn clinic.

1. Architectural goals and philosophy

Architectural goals, often driven by software requirements, provide the motivation and rationale for decisions. Here it is defined how the system needs to respond to change over time.

- Aggregated pictures of burn injuries will be uploaded by the sending end and accessible for the receiving end in a cloud repository called FileCloud.
- The system will be secured, so that only authorized users can communicate with the backend. In order to access their APIs for storing and other things, the request has to be validated.
- The availability of the system is a natural key requirement since doctors would need to evaluate burn injuries with a quick response time at all hours.
- Application must be fully responsive to modern devices. The application shall work on standard sizes for phones, tablets, and computer screens, both in tilted and normal mode.
- The system will be implemented as a client-server system. The client portion resides on the user's device and the server portion must operate on a server belonging to Region Östergötland. Client-server communication will be done through APIs supplied by Region Östergötland.

2. Assumptions and dependencies

This section consists of a list of the assumptions and dependencies which drive architectural decisions.

- The technical scope of the project shall be adjusted to the skill level of the developers in the company. Most of the developers have not used angular before, and have to study this framework to gather new knowledge.

· The technologies used shall be discussed, reviewed and approved within the development team and with Region Östergötland.

· FileCloud shall be used as a cloud repository for storing all images and case data.

· Region Östergötland does not have a solution for authenticating the client. MedCom shall come up with a solution for authentication which shall be approved by Region Östergötland and follow Region Östergötlands rules and requirements.

· Region Östergötland will supply REST APIs for the client to connect with the backend, with functionality according to the needs of the team.

· The different components shall be loosely coupled for reusability.

· Reliability is a key factor and the application needs to be built to minimize downtime.

3. Architecturally significant requirements

Architecturally significant requirements are those requirements that play an important role in determining the architecture of the system. First parenthesis: Corresponding requirement in SRS. Second parenthesis: This requirement led to the following decisions in section 4.

· The application shall be supported in the browsers Google Chrome, Safari, Microsoft Edge and FireFox. (PR5, SRS version 1.3) (Led to decision 1)

· The application shall be single paged. (PR2, SRS version 1.3) (Led to decision 5)

· The front end of the application shall be implemented with Angular, also CSS, TypeScript and HTML5 will be used. (SIR1, SRS version 1.3) (Led to decision 2)

· The storing solution shall be FileCloud. (SIR4, SRS version 1.3) (Led to decision 6)

· **Input:** The sender presses the “Skicka ärende” (Send case) button in the preview mode.
Process: If at least one picture is attached to the case and a name and at least one contact information is added, the sender is redirected to the Oauth2 Authentication mode. In this mode, the sender chooses a way to authenticate himself/herself, either via personal number and code received over the phone from a doctor or via BankID.
Output: The system gets a verification that the sender is authenticated. (FR9, SRS version 1.3) (Led to decision 7)

4. Decisions, constraints, and justifications

1. The application will be a web application since Region Östergötland requested that the application without downloading an application.

2. Angular 2 or later should be used, on request of Region Östergötland. This makes it easier for them to keep maintaining our work in the future.

3. The client must be authenticated before getting access to the APIs of Region Östergötland, this is to make sure that the receiving end only receives relevant information.

4. We will not host our own backend, Region Östergötland wants us to develop only the client-side which communicates with their APIs to make it easier to change and replace.

5. The application will be single-paged. This is to save time reloading the page several times and to make for better user experience. The use of Angular makes the implementation of a single page strategy relatively simple.

6. Filecloud will be used as a storing solution since Region Östergötland already had implemented services that works with this solution.

7. The authentication will be done with either Bank-ID or code. These were the alternatives that suited Region Östergötland best.

8. All communication from the client with other parts will be done through services in Angular. This is to make the application modular in the sense that components can be replaced without interfering with server and API connections.

9. Data will be shared between components by using data service. This is a way to share data that is easy to implement and is convenient for our solution.

5. Architectural Mechanisms

5.1 Architectural analysis

Security

- Validation: There will be some sort of validation of the identity of the user. (FR9, SRS version 1.3)
    - Protect from malicious overloading: There is a need for a mechanism to prevent overloading to ensure system stability.

  - Prevent inappropriate content to be submitted: There is a need to prevent users from sending inappropriate content and use the application for purposes it is not meant for.

Availability

- Support different devices: The client should be able to operate on numerous devices (smartphone, tablet and PC). (PR3, SRS version 1.3)
- Support different operating systems: The client should be able to run on different systems.
- Support different browsers: The client should be able to run on numerous different web browsers (Google chrome, safari, Microsoft Edge and FireFox). (PR5, SRS version 1.3).

Communication

- Speed: Fast transfer of data between the sender and receiver. (SSAR1, SRS version 1.3)
- Ease of use: The system should be easy to use without prior experience. (A/AA-requirements, SRS version 1.3)
- Minimal setup: The users should be able to quickly gain access and start using the system. (A/AA-requirements, SRS version 1.3)

Error management

- Standardized testing: Using a structured and standardized approach during tests will be a suitable method to locate errors.

5.2 Architectural design and implementation mechanisms

Security

- IDP authentication system: The sender will have to authenticate oneself, using the IDP system provided by Region Östergötland.

Availability

- Angular web application: By developing a web application, using angular 8 combined with HTML5, the product will support the most common web browsers and be compatible with all the common operating systems.

Communication

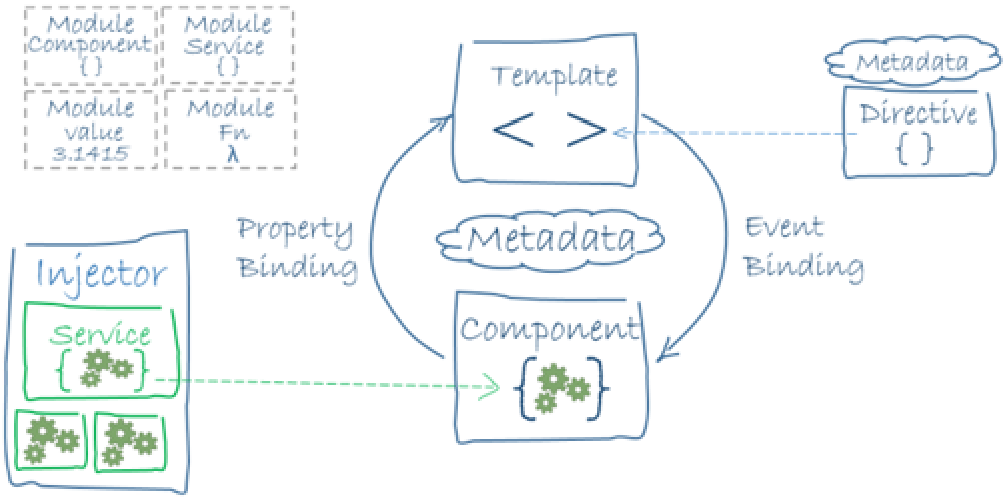
- Standards WCAG 2.1 (Web Content Accessibility Guideline): Developing a web application that is user-friendly for a broad target group (ease of use).

Error management

- Selenium: For automatic web application testing.
- Travis and “ng test” for testing: For automatically run unit tests when changes are being implemented.

6. Layers or architectural framework

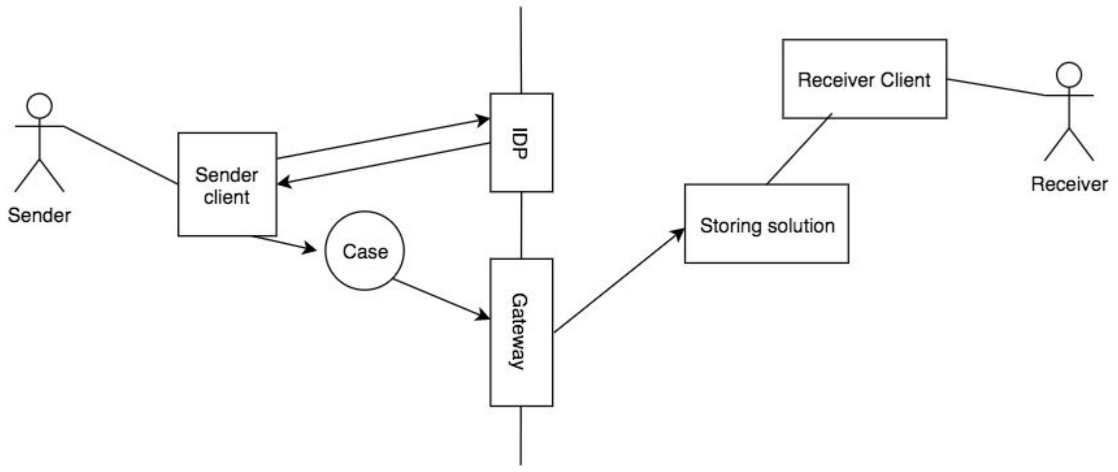
The front-end framework Angular will be used. More information about the architecture of Angular can be found [here](), at Angular architecture overview.



Source: https://angular.io/guide/architecture

7. Key abstractions

Key abstractions are the key concepts and abstractions that the system needs to handle. They are those things that, without which, you could not describe the system.



The figure above depicts the main abstractions. The direction of the arrows represents the high-level interaction between the components, it is not a detailed message diagram or a technical representation. The sender creates a case with the help of the sender client. The sender then receives an identity token which is sent with the case to the Gateway. The Gateway takes the identity token and lets the case pass to the storing solution. The receiver uses the receiver client to access the case which is stored in the storing solution.

Sender client

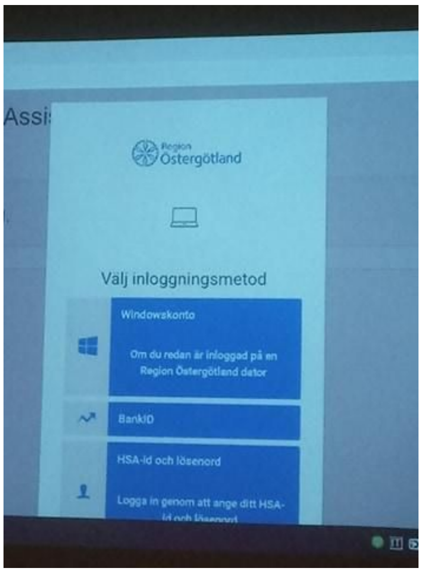
The sender client is a web application built in Angular which the sender uses to create a case, authenticate itself and send the case to the gateway. The sender client is described further in section 8, Architecture views.

Case

The case consists of all information about the current case which is relevant to send. Images, name of sender, email of sender, phone number of sender. Inside sender client, the case might be represented as a class but it is sent to the gateway in the form of a yaml file.

IDP

IDP stands for Identity provider and is an external service that offers users authentication as a service. Region Östergötland currently uses IDP to authenticate their users in their current solutions, which is shown as an example in the image below. The possible authentication alternatives will be BankID or a personal identification number with a 4-digit code.



Välj inloggningsmetod

Windowskonto

Om du redan är inloggad på en Region Östergötland dator

BankID

HSA-id och lösenord

Logga in genom att ange ditt HSA-id och lösenord

Gateway

Belongs to Region Östergötland and acts as a sort of security, which only lets validated users get through to access the storing solution. When the user authenticates oneself after entering credentials at the IDP a token is being generated (which is valid for five minutes) in return. This token is then being used to be able to access the APIs provided by the gateway. It is in the gateway the token is being validated.

Storing solution

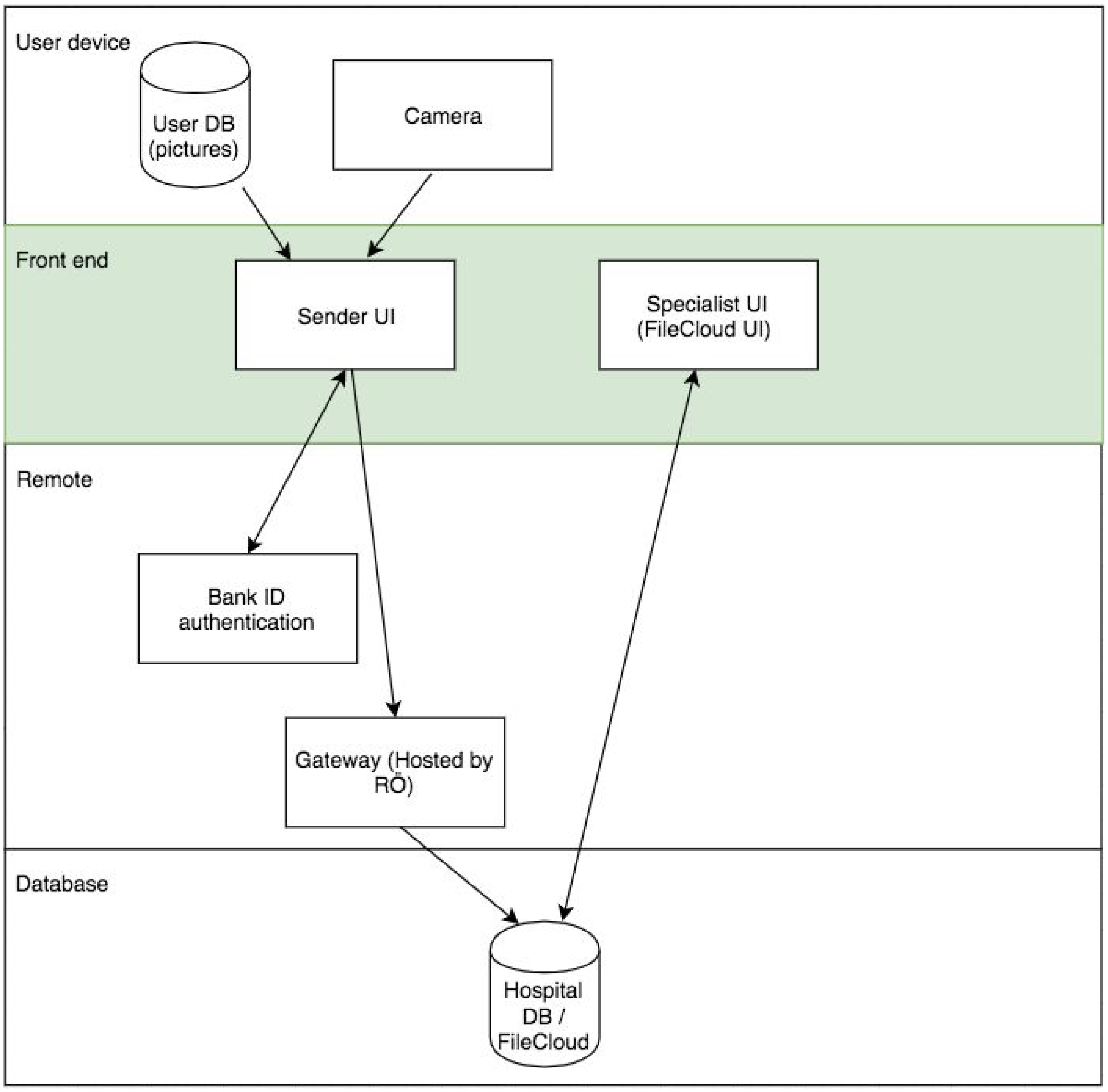
The storing solution is where the case will be stored permanently. The storing solution will at first be the cloud file server FileCloud.

Receiver client

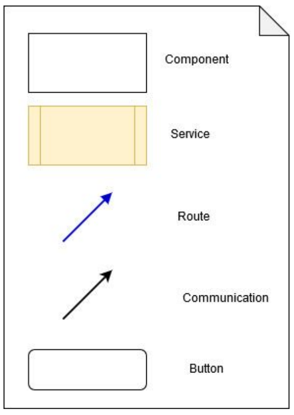
The receiver client will at first consist of FileCloud´s own user interface, where the receiver can interact with the stored case directly.

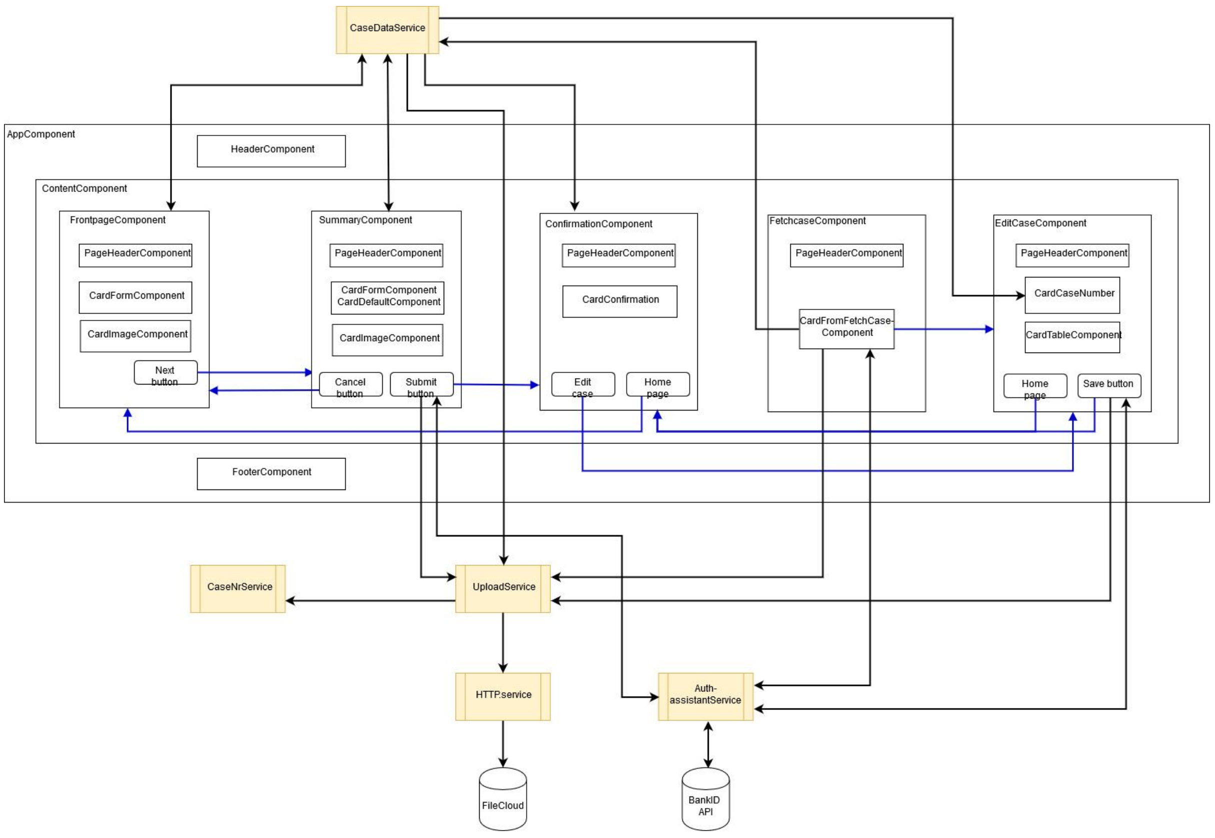
8. Architectural views

8.1 Logical View



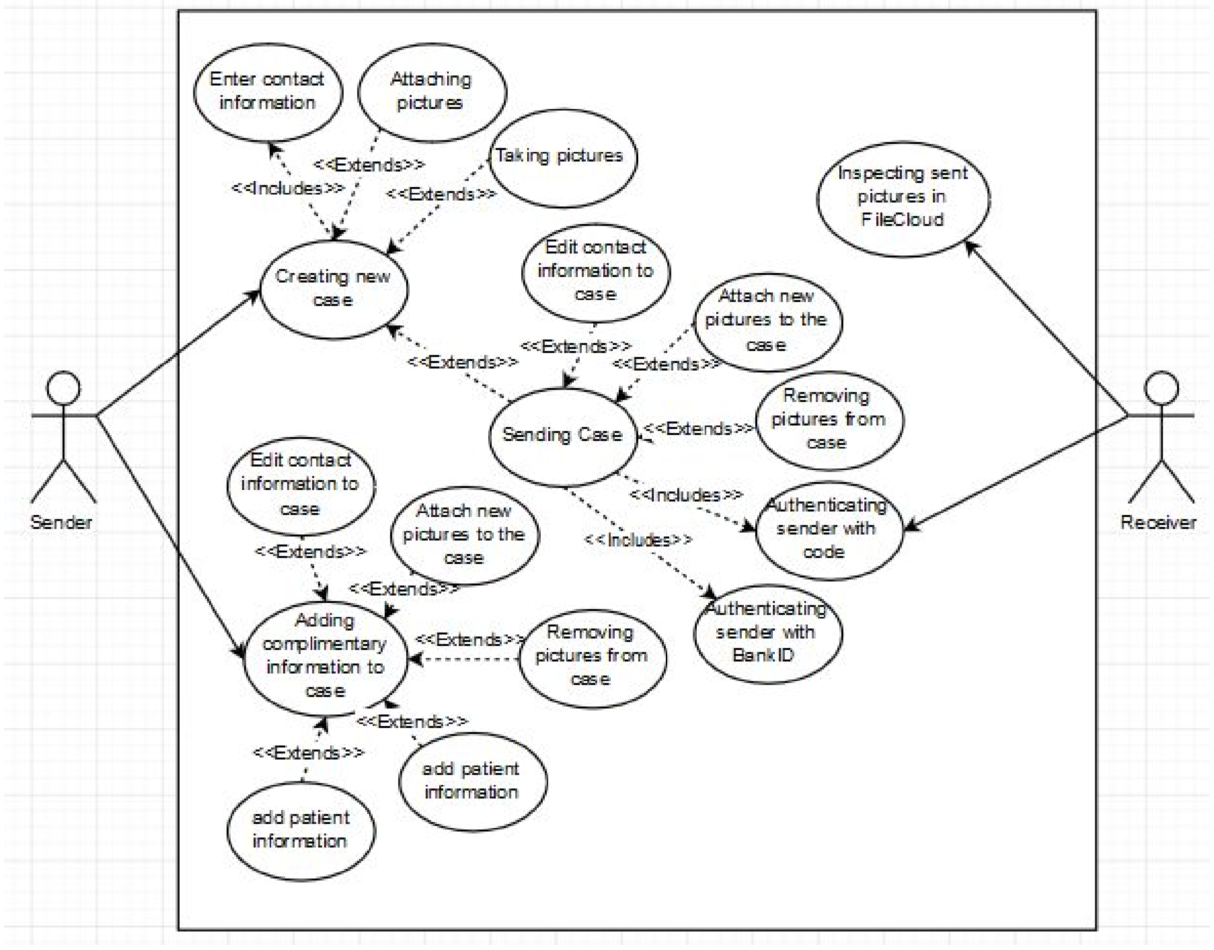
8.2 Sender UI (Components and services)





8.3 Use cases

Below is the use case diagram. The use cases are defined further in the SRS version 1.2.


Enter contact information
Attaching pictures
Taking pictures
<<Extends>>
<<Includes>>
<<Extends>>
Creating new case
Edit contact information to case
Attach new pictures to the case
<<Extends>>
<<Extends>>
<<Extends>>
Inspecting sent pictures in FileCloud
Sending Case
Removing pictures from case
<<Extends>>
<<Includes>>
<<Includes>>
Authenticating sender with code
Authenticating sender with BankID
Sender
Receiver
Edit contact information to case
Attach new pictures to the case
<<Extends>>
<<Extends>>
Adding complimentary information to case
<<Extends>>
Removing pictures from case
<<Extends>>
<<Extends>>
add patient information
add patient information

Source (The analyst & UX google drive folder, 2019-10-17)