D3.1.1

Trust Model for cloud applications and first Application Architecture

Project number:	257243
Project acronym:	TClouds
Project title:	Trustworthy Clouds - Privacy and Resilience for Internet-scale Critical Infrastructure
Start date of the project:	1st October, 2010
Duration:	36 months
Programme:	FP7 IP

Deliverable type:	Report
Deliverable reference number:	ICT-257243 / D3.1.1 / 1.0
Activity and Work package contributing to the deliverable:	Activity 3 / WP3.1
Due date:	September 2011 – M12
Actual submission date:	3rd October, 2011

Responsible organisation:	PHI
Editor:	Mina Deng
Dissemination level:	Public
Revision:	1.0

Abstract:	This deliverable describes the medical use case studied in the TClouds project. In particular, a home healthcare application in the cloud is analyzed. Technical requirements and legal issues are identified. The underlying legal implications are discussed. The deliverable also provides preliminary reference architecture and middleware architecture for enhancing security, privacy and resilience.
Keywords:	Home healthcare, medical, requirements, legal, architecture, security, privacy, middleware

Editor

Mina Deng (PHI)

Contributors

Mina Deng, Milan Petković (PHI)

Marco Nalin, Ilaria Baroni (HSR)

Eva Schlehahn (ULD)

Imad Abbadi (UOXF)

Disclaimer

This work was partially supported by the European Commission through the FP7-ICT program under project TClouds, number 257243.

Executive Summary

The TClouds healthcare use case focuses on developing a cloud-supported home healthcare application to provide collaborated services across different health care providers. In particular, this deliverable presents the home healthcare application that supports innovative services for the depressed patient's remote management, involving different actors and services.

Section 2 provides an overview of the home healthcare use case scenario, in which patient's monitoring information such as light, sleep, and daily activities is collected from patient's home by a mobile monitoring device provided by a Health and Wellness service provider, and shared with various healthcare service providers such as hospitals in the Cloud. Benefits of using cloud computing for the cloud subscribers are briefly listed, including but not limited to cost reduction, scalability, resilience, increased connectivity and pervasive availability.

Based on this scenario, Section 3 analyzes a list of security and privacy requirements derived from the proposed home healthcare application from a technical perspective. Security and privacy engineering methodologies are used to support requirement elicitation and build security and privacy in the system development lifecycle. In a nutshell, generic cloud characteristics that we have recognized are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. In addition, a semi-trusted security model is used assuming that cloud service providers are honest but curious. Specific use case oriented technical requirements are derived, including self-managed services, highly distributed data storage, data-centric protection, emergency access and availability, efficiency, data confidentiality, data integrity, accountability, patient-centric protection, data minimization, anonymization and data filtering.

The preliminary overview of legal issues in Section 4 aims at identifying the general legal issues of data protection in respect to cloud computing in the context of the TClouds medical use case. It takes into account the current European data protection legislation, introducing basic terminology and concepts. It then presents the general requirements. This part will focus on introducing the most important legal frameworks on European and national level, their basic principles, the requirements for technical conception and functionality of the cloud system as well as for its organisation, usage and configuration. The overview then highlights some still open questions that are needed to be discussed and worked on together with concerned project partners to present solutions for some of the legal issues. Finally, a conclusion and forecast to future in-depth legal analysis within the project work through reports and deliverables will be made.

Section 5 explains the architecture for the home healthcare application, including the usage scenario (issues, actors, benefits, etc.), but also all the services that will be developed, detailed by use cases. This last part describes first the generic architecture (for a possible final application), and then it specifies the features that will be implemented in the first year mock-up implementation.

The building block of middleware at application layer is discussed in Section 6. Specifically, the middleware functions in the Cloud in the context of home healthcare application are identified.
Section 7 discusses the issue of establishing trust in the Cloud. It requires building trustworthy middleware self-managed services and providing Cloud users with capabilities that enable users to assess the operation of the Cloud. It then identifies the security threats that can and cannot be covered by middleware. Additional mechanisms need to be provided in order to protect against some attacks, e.g. insider attacks, data input manipulation, denial of service attacks, and physical attacks. Middleware functions can help in addressing some but not all of these.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter Authors:

Mina Deng (PHI), Marco Nalin, Ilaria Baroni (HSR), Eva Schlehahn (ULD), Imad Abbadi (UOXF)

The main objective of the work package 3.1 “Cloud Applications Data Structures for Home Healthcare Benchmark Scenario” is to define the cloud architecture and specification from the application side of view, i.e., provide technical requirements for cloud computing in the healthcare sector, especially in the area of home healthcare services and leverage this application into an architecture, API, and protocols between application components and clients. The cloud supported home healthcare application architecture will be closely linked and integrated with WP2.1, WP2.2, WP2.3 and WP2.4.

1.1 Outline of the Work Done in Y1

This deliverable aims to address Task 3.1.1 “Technical requirements for privacy-aware and resilient home care” and partially Task 3.1.2 “Definition of application architecture: components, APIs and data structures”. Task 3.1.1 aims to develop a trust model for the targeted cloud application of home healthcare. First the TClouds home healthcare use case scenario is presented in Section 2. Based on the work from WP1 this task will identify the application specific trust issues, and derive the application level requirements related to security, privacy, and resilience. The results are presented in Section 3 and Section 4 on technical and legal requirements for the home healthcare case, and Section 7 on Trust model.

Task 3.1.2 aims to define the home healthcare application architecture and the border between the cloud and the application software, i.e., which trust related issues have to be performed on the application side and which on the cloud side. The results are presented in Section 5 on preliminary architecture for the home healthcare application and Section 6 on preliminary architecture of the application middleware.

1.2 Structure of this deliverable

The deliverable is organized as follows. Section 2 provides a description of the TClouds medical use case scenario. The application for the health use case will implement innovative services for the depressed patient’s remote home monitoring to support patient’s therapy. Monitoring information such as light, sleep, daily activities is collected from patient’s home by a mobile monitoring device provided by the Health and Wellness service provider, and shared with healthcare service providers in the Cloud.

Based on the aforementioned home healthcare scenario, Section 3 provides technical requirements derived from the home healthcare application with a focus on security and privacy. Security and privacy requirement engineering methodologies will be employed to support and requirement elicitation and fulfilment in the software development lifecycle. The

security and privacy requirements are analyzed both with the service-logic driven strategy and the architecture-driven strategy.

Section 4 provides a preliminary overview of the legal issues for the TClouds home healthcare use case. In this use case, an elementary aspect is the collection, processing and storing of depressed patient's personal data in a cloud computing environment. Due to the complexity of the use case and the difficulties to realise adequate protection of sensitive data in cross-border cloud systems, it is necessary to research the arising legal issues and look for possible solutions. This overview however is not intended as a complete analysis of the legal requirements concerning this scenario. Nevertheless, it already outlines roughly the arising problems for storing and processing medical data in a cloud computing environment. It also gives some first guidelines how the electronic patient file must be composed to comply with the general data protection framework on EU and national level.

Section 5 provides an overview of the reference architecture for the TClouds eHealth home monitoring scenario. First, some basic background information is introduced about the chosen target disease, i.e., depression. Next the innovative services provided by the home health use case to be delivered to depressed patients through the cloud based infrastructure are explained in detail. These services are further detailed through the definition of the use cases, illustrating also the use cases dependencies and involved actors. Then the reference architecture derived from the aforementioned use cases and scenario are described. A practical instantiation of the reference architecture that is implemented as the first year mock-up prototype will be illustrated as well.

Section 6 provides the preliminary architecture of the application middleware. In particular, it defines the generic application middleware and its services, it provides a generic definition of the functions which are needed to be implemented by the application middleware services, and identify the middleware services which are required by home health care application.

Section 7 provides a preliminary trust model which is required to establish trust in the Cloud for the eHealth home monitoring scenario. Specifically, it discusses the functions which are required to establish trustworthy middleware services and their role in establishing trust in the Cloud by continually enforcing security, privacy and resilience requirements of the reference architecture for the TClouds eHealth home monitoring scenario. Establishing trust in the Cloud includes having trustworthy middleware services and establishing supporting services to address other security, privacy and resilience requirements for the reference application.

Chapter 2

Scenario of Home Healthcare in the Cloud

Chapter Authors:

Mina Deng, Milan Petković (PHI), Marco Nalin, Ilaria Baroni (HSR), Imad Abbadi (UOXF)

2.1 Introduction on Depression

Depression is one of the most common Non-Communicable Diseases (NCDs) and it's affecting 121 million people around the world 1. Depression is nowadays the second leading cause of disability in the world for ages 15 to 44, and it is predicted to become the second leading cause, worldwide and for all ages, within 2020.

In spite of these facts, a vast gap exists between the patients' needs for treatments and the available services. Thereby, in developed countries between 44% and 70% of patients with mental health disorders do not receive treatment. Cloud based solutions could help to develop innovative services to fill this gap.

Drugs are not the only remedy for depressed patients; many other treatments emerged during the last decades, involving the correct synchronization between biological rhythms and the environment (chronobiology).

There is increasing evidence that 1) circadian disturbances are involved in common mental ailments such as bipolar disorder and depressive syndromes; that 2) keeping a correct synchronization between biological rhythms ("internal timing") is key to mental health; and that 3) in a therapeutic perspective, correcting abnormal circadian rhythms through exposure to light, melatonin pills, or sleep deprivation/sleep phase advance, can help to treat some of these disorders, as well as many other disorders, for example neurodegenerative illnesses such as Alzheimer's. Sleep manipulations such as sleep deprivation, light therapy, and phase advancement are non pharmacological chronobiology techniques able to ensure the clinical remission of the depressive syndrome (i.e. longitudinal studies revealed the presence of a link between the mood levels, the exposition time to sun light and the luminosity levels of the environment in two patients with a diagnosis of non seasonal affective disorder).

Alignment of internal rhythms is very important, and there are specific therapies that rely on that, like the Social Rhythm Therapy, an intervention that aims at keeping a daily log and regularize the time of five main daily events: (i) getting out of bed, (i) first contact with another person, (iii) start of work/school/volunteer/family care, (iv) dinner, (v) going to bed. Keeping regularity of these events demonstrated to be effective in controlling depressive episodes and reduce relapses.

As mentioned above, for depressed patients sleep habits and light exposure have proven to be critical factors impacting on rapid variations in the clinical figure. A proof of this is that nowadays Light Therapy is used as a common treatment, often to speed up normal drug

[1] World Health Organization. 2008-2013 Action Plan for the Global Strategy for the Prevention and Control of Noncommunicable Diseases, 2009.

treatment response, but in some case as a standalone treatment, without the use of drugs. Physical exercises is both a depression"sensor" (the more the patient is depressed, the less he moves), and an intervention (the more physical activity he does, the more endorphins are produced by his body). Finally a lot of eating problems are strongly correlated with Depression, having as a side effect that with aging the depressed patient develops also metabolic disorders or get cardiovascular diseases.

2.2 Scenario architecture

Empowering patients, allowing a continuous home-monitoring, and improving health professional-patients links will have a significant impact in patient management, including hospitalization and critical episodes prediction, and therefore in economical budget that depress meaning, without compromising the quality of care. In 2004 the cost of the depression has been estimated in 235 Euro for inhabitant (with a total cost of 118 billions of Euro in the EU 25 and EFTA countries). Moreover the direct costs on the Health Systems of the different EU countries are increasing, but about the 65% of costs concerns other sectors such as lost productivity due to absenteeism, disability pensions and early retirement. Mental ill health costs the EU an estimated 3%-4% of GDP. By the year 2020, depression is expected to be the highest ranking cause of disease in the developed world, and currently, in the EU, some 58,000 citizens die from suicide every year, more than the annual deaths from road traffic accidents, homicide, or HIV/AIDS.

The depressed patients need services able to early identify, counter fight and prevent potentially dangerous situations, and the current treatment model, consisting in monthly periodic visits, is not sufficient to cope with these needs. Furthermore there are several factors that can impact meaningfully on depression and that have been validated in medical literature, that still are not part (or are only partially part) of traditional treatment process, like behavioural parameters (e.g., physical activity, sleep), or environmental parameters (e.g., light). These additional factors can be monitored to provide better support to depressed patients.

Figure 1 describes the reference scenario for the home healthcare application that will be deployed in the TClouds infrastructure. The actors and their relationships will be deeper explained in Section 2.3, but for the moment the picture is useful to provide a general overview, as introduction to the following section describing the services designed to empower the patient over her treatment process. In particular the home healthcare application foresees services to support the patient in the management of:

- Drug therapies management, improving compliance with doctors' recommendations
- Sleep (and light) management
- Physical activity management

Furthermore, some proposed services will be dedicated also to Healthcare Professionals (HCPs), and to institutional organizations, like regional or national healthcare systems.

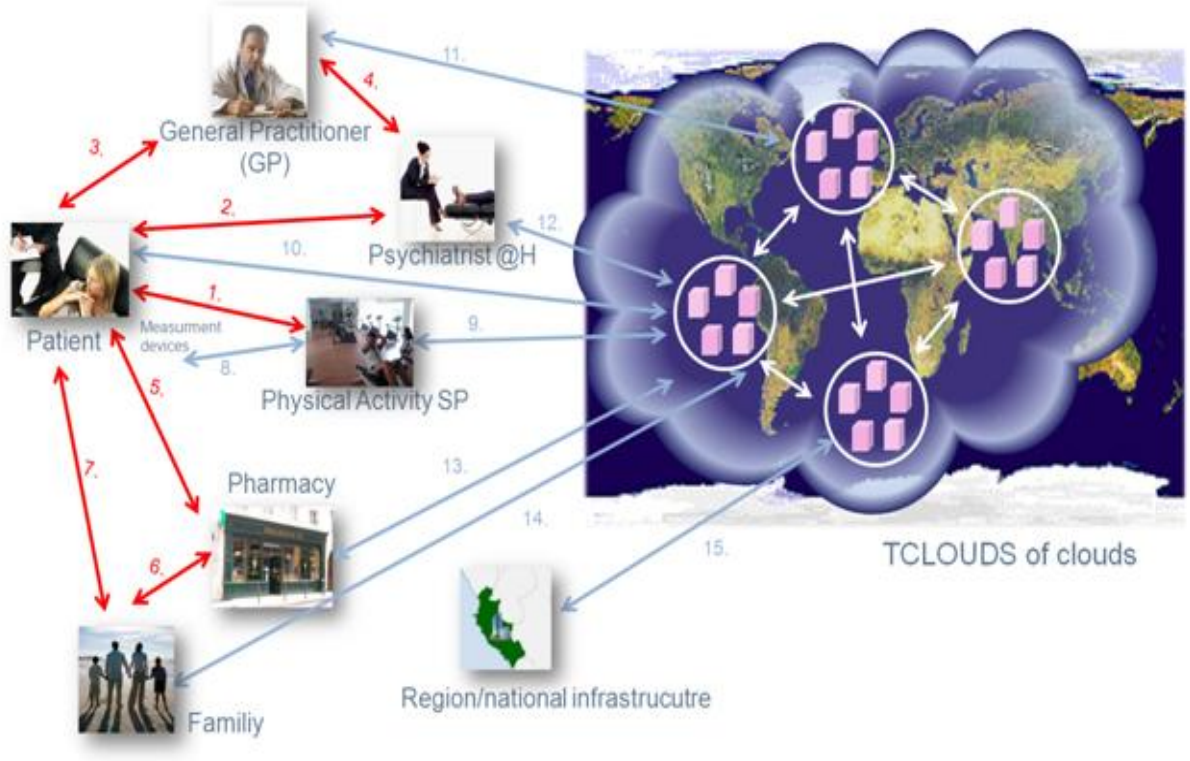

Figure 1 eHealth scenario architecture

2.3 Actors

The following actors are identified:

- General Practitioner
- Patient
- Medical professional (e.g. Psychiatrist at Hospital)
- Health and Wellness Service Provider (e.g. Activity monitoring)
- Pharmacy
- Family
- Region/national authorities and infrastructure (e.g. Department of public health)
- TCloud of clouds (EHR/PHR hosts): It hosts Personal Health Record (PHR) or Electronic Health Record (EHR) service(s) (e.g. Google Health, or EPIC EHR)

2.4 Links

The description of the links among different actors is given below. For the description of relationship between different actors, please refer to the use cases in Section 2.5.

1. Depressed Patient visits Health and Wellness Service Provider, registers with the home monitoring service and gets activity measurement devices (e.g. Pedometer).

2. Depressed Patient visits Psychiatrist at Hospital who prescribes medications and/or therapist.

3. Depressed Patient consults his GP regularly for treatment of depression. GP may prescribe medicines or refer Patient to a specialist (or Psychiatrist).

4. GP of the depressed patient may also contact his Psychiatrist at Hospital in an out-of-band fashion to discuss the case of the depressed patient.

5. Depressed Patient visits Pharmacy to get the medicines prescribed by healthcare professionals (e.g. GP, Psychiatrist).

6. Family members of the depressed Patient may also visit Pharmacy on the Patient's behalf to get the medicines prescribed by healthcare professionals (e.g. GP, Psychiatrist).

7. Often the depressed patient is living with his/her family members who are directly or indirectly affected by the patient's disease.

2.5 Description of services

- *Personal Diary*: the depressed patient can report and analyze his mood in an online diary, filling some values day by day (e.g., anxiety, stress, mood, depression, mania, etc.). This is helpful to elaborate the data with charts and graphs. The diary will also include daily timings useful for the Social Rhythm Therapy (clock time out of bed, first contact with another person, start of activities, dinner, and time to bed). An example of personal diary is indicated in Figure 2.
- *Self assessment questionnaires*: this service allows compiling some questionnaires specific for the depression that can be suggested to the patient in particular situations (e.g., in case of identification of suspect patterns in the Personal Diary). These questionnaires will be online implementation of existing self-assessment depression scales.

Figure 2 Example of a personal diary

- *Physical activity monitoring*: this service provides a monitoring system that collects and analyzes data related with physical activity, from wearable and non-wearable devices. Data can be inserted by the patients (or directly by patient's devices) or be provided by Physical Activity Service Providers. The patient will have the opportunity to correlate physical activity data with their personal diary.
- *Sleep management*: the patient can benefit from the use of sleep monitoring devices, both to monitor sleep therapy or simply sleep quality. Furthermore the system can rely on eventual actuators that the patients might have (e.g., wake up lamp), which values can be set automatically by the system (for example depending on the Light Therapy prescription, or on the sleep prescription, etc.).
- *Drug therapy management*: a virtual medicine cabinet service will be provided to the patient. Prescriptions prepared by healthcare professionals (e.g., psychiatrist, general practitioner, etc.) will be available to the patient, which will be able to receive these data and place a purchase order to the pharmacy. A prototype of integration with an automatic delivery service will be simulated, to demonstrate privacy protection and secure transactions. Furthermore the patient will have a reminder service for when it is time to take a drug, and compliance will be checked automatically and recorded in the system.
- *Epidemiological studies*: regional authorities can have access to anonymized data about patient's treatments, statistics about drug consumptions and correlations between drug (non-)compliance and disease development.
- *Patient portal*: patients will have a personal portal from where they can access all the services listed above. Furthermore patients can control the privacy settings, and establish who can access their data and what part of the data.
- *Healthcare professional portal*: Psychiatrist, General Practitioner and all the other professional actors involved in the patient's care, can access information about

patient's disease development, drug compliance, and physical activity (according to the permission level specified by the subscriber).

2.6 Final prototype description

The developed prototype will include:

- Drug management and related services;
- Sleep management functionality (in particular the ones related to smart wake up, and possibly dawn simulation), with the interaction between patient and psychiatrist implemented into the application;
- Examples of data collected from devices for physical activity monitoring, together with the correlation of physical activity results with the personal diary;
- The personal diary;
- Questionnaires for depressed patients and self-assessment depression scales;
- Privacy management functionalities, accessible through the patient's portal;
- Psychiatrist portal, with the patient's information about drug compliance, sleep monitoring, physical activity.

All the services will have interfaces for the described actors, in particular for the patient, the psychiatrist, the family, the pharmacy, region/national authority, and physical activity service provider.

2.7 Cloud computing perspective

IT systems have traditionally existed within the boundaries of the organization that owns the infrastructure. The costs of running an IT system is not limited to an initial capital investment, which is realized in the hardware, software and having appropriate environment. It extends to ongoing running costs realized in infrastructure updates due to adapting new technology and potential increase in business size. Additional running costs could also be due to (a) professional IT staff for infrastructure management, (b.) maintenance contracts with hardware and software suppliers, (c) system upgrades, and (d) in some cases annual payment for licensing costs. In a hospital setting, for example, an IT department would be responsible for managing the Infrastructure, which covers: (a) installing and configuring hardware and software packages to support the operations within the hospital, (b) capacity planning to ensure in advance preparation for potential increase in load/storage, (c) disaster recovery plans, and (d) security management. In addition, IT staff would be expected to keep abreast of latest developments such as security incidences that may affect their systems as well as perform necessary actions to counter these events, e.g. patch their systems. Supporting such systems is time consuming and requires substantial upfront as well as continuous investment. In addition, the IT staff that have full control over the organization assets can cause a potential impact on the system security, as in the case of leaking sensitive content to outsiders.

Outsourcing has been used as an economical approach for the above problems. Some organizations outsource the management part of their IT application infrastructure to third parties who are expert in the application domain. For example, most organizations outsource their hardware maintenance to the hardware suppliers or a third party. A hospital, for example, can have a contract with an external IT professional organization to manage and

maintain the whole of their IT system. The hospital would typically still own the infrastructure, but the contractor organization is in charge of carrying the IT management burden on behalf of the hospital. In return a hospital would pay a regular maintenance fee, which is much less than owning IT staff.

Although outsourcing IT services reduces the overall cost compared with internally managing them, outsourcing is still an expensive option for some organizations, as it does not react the real cost based on the actual usage of resources as used by the organization. This is because maintenance and support costs in an outsourcing model are based on subscription models where the customer agrees on periodic payments regardless of whether they receive any services or not and regardless of the utilization of the resources. For example, a support contract may require a hospital to pay a monthly fee of ' €X' per server for unlimited support. However, if in one month the hospital does not require any support, they would still pay ' €X' despite not receiving any support services. On the other hand, it might be advantageous to the hospital in a month where they receive a lot of support from the contractor or use more resources.

An even better approach is to enable outsourcing while paying only for the services received or resources used. Cloud computing is a complimenting approach to the above problem. It combines the outsourcing model with a pay-per-use model, enabling low entrance barriers and substantial cost reductions when no services are received or less resource are used. How much outsourcing service a cloud could provide would be based on the cloud type that an organization works on. Cloud computing supports three main types: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). IaaS provides the most flexible type for organizations who like to have the greatest control over their resources, while SaaS provides the most restrictive type for organizations where cloud provides have full control over the virtual resources. Cloud computing provides a full outsourcing support for the SaaS, a partial outsourcing support for PaaS (provides the virtual environment and software tools for the user to develop and deploy their applications), and a minimal outsourcing support for IaaS (managing physical resources and virtual resource monitoring).

In the home health care system rather than having dedicated IT infrastructure within the hospital, the infrastructure would be hosted by a cloud provider which then provides well defined (and sometimes restricted) interfaces into the infrastructure. The hospital can further define interfaces accessible to different entities within its operations. For example, they can host a drug inventory system and make it accessible to its pharmacy department or a patient registry system that is accessible to the registry staff.

2.8 Usage in the Home Health Care Setup

In this scenario, a hospital provides services, as described above, to clients for the purpose of treating patients for cases of depression. The services are accessible to clients through web portals that are provided through the hospital's website. We now consider how these services could be provided using a cloud infrastructure while making it transparent to the clients. Firstly, the hospital creates a software application either by using internal resources or by outsourcing the development to a third party. The developed application should provide the necessary functions to run all required service logics. Also, the outsourced application is better to be developed and customized to consider the available services that a cloud infrastructure can provide. With all the applications ready, the hospital has to deploy these applications to a cloud infrastructure. The cloud infrastructure provider (in the IaaS type) would allocate virtual resources and manage them to allow the applications to provide all the necessary functionality they are designed to do. For example a cloud provider should support high availability, load balancing, high performance, and disaster recovery. All these are examples of services that the cloud provider can provide and manage on behalf of the

hospital. The deployed application should interface with cloud provider supplied API's to take full advantage for such cloud provided services.

The hospital, on the other hand, need to have the necessary resources to deploy not only the application but also to install and manage the operating system and database management system. The hospital makes the applications accessible to clients by creating portals and including links in the portals. With the deployment complete, the services are available to clients through the web portals. Clients will not notice the application is hosted on the cloud, as for them they accessing the application by connecting to a URL. To make use of the hospital services, clients would have to register with the system. This can be done at the start of treatment, i.e. when client visits the psychiatrist or general practitioner, or the client can register online. The clients will then use the credentials provided by the hospital to login and access the services.

On the side of the hospital (e.g. general practitioners, pharmacists and psychiatrists) can access the services in a similar way to clients. This group of users should have access to part of or complete patients' records. They can view or update the records either in the presence of the client or without the client being around. All the changes are immediately committed to the system in the cloud and accessible to other parties that have access.

2.9 Benefits of Using Cloud Computing

Cloud computing provides economic incentives that are reacted in cost savings in terms of reduction in human resources required to run and maintain the infrastructure and costs of acquiring/upgrading a system. In addition, cloud computing is required to provide additional non-functional aspects, e.g. (reliability, availability, scalability, etc). In particular, the home care scenario which uses IaaS cloud type could benefit from cloud computing in the following ways.

1. Cost - reduces upfront investment for the following reasons: (a) the backend hardware infrastructure that runs the system is provided by the cloud provider in a pay-per-use model, (b) less dedicated IT staff would be needed to maintain the backend system and provide support, (c) in some licensing models in which payment is based on allocated resources, using on demand cloud reduces the overall licensing costs, and (d) reduces bills (e.g. electric costs, server room renting) as the consumer only uses the resources that are needed to run the business.
2. Scalability - with cloud computing an increase in demand for the services provided by the hospital can be met transparently to users by increasing the capacity of the resources provided (vertical scalability) or by adding additional resources (horizontal scalability). The real benefits come from the fact that a system running in the cloud can be setup to scale up and down automatically in response to demand. Research from a global survey (BridgeHead 2010) from hospitals and healthcare organizations worldwide revealed that medical images, scanned documents, email and advances towards the EHR are going to be the cause for a meaningful increase in healthcare data that is already challenging hospitals. Most of the participants in this survey (41%) claimed that they are expecting an increase in the data volume up to 25%, while approximately one fifth of them (18%) is expecting a growth from 25% to 50%. Besides traditional Healthcare Information Systems, there are other emerging fields of eHealth that could lead to exponential growth in the database size. For example remote monitoring, especially if the patient's monitoring is continuous (regardless of the activity to be monitored, e.g., ECG, physical activity, etc.), and with a lot of patients, we can expect rapid expansion of the data volume. Cloud Computing allows to easily scale storage capacity when needed.

3. Availability and Resilience - cloud computing would be based on huge underlying infrastructure, which is built to support and scale for large number of customers. Also, the infrastructure itself supports disaster recovery scenarios. The results from BridgeHead survey reported what are the top priorities in the next investments for IT budget in healthcare organizations. Disaster recovery, together with Data Backup and Business Continuity, was a priority for 44.3% of the respondents. Cloud Computing could offer backups and redundancy at lower costs with respect to legacy systems. Controlling such huge infrastructure and having well planned procedures in place enable cloud infrastructure to provide better availability and resilience in comparison with those provided by most small size to midrange organizations. Business continuity and availability are very important in most of the medical applications, especially those dealing with possible emergency situations detection (e.g., remote patients' monitoring) and management (e.g., availability of the EHR in a dangerous situation).
4. Increased Connectivity and Pervasive Availability - cloud computing is supported by high bandwidth connectivity to the Internet. It also benefit from having redundant Internet connections eliminating single point of failure. Most organizations do not have such connectivity in place due to excessive costs. An interesting IaaS feature that the cloud could offer is the creation of virtual networks to connect healthcare institutions (like in the case of Virtual Healthcare Professionals networks), or to connect patients and healthcare institutions (like in the case of remote monitoring, e.g., telemedicine, AAL, etc.).

2.10 Challenges

Cloud computing has the potential to lower costs. While assuring high performance/ availability; it is faced with a number of challenges. Some of the challenges are rooted from the nature of cloud computing, while others are a result of the nature of medical records and ethical issues associated with them. The main objection to the adoption of Cloud Computing (65%) in the BridgeHead survey was the hospitals' concerns about the security and availability of healthcare data given the great number of threats, including privacy breaches and identity theft. Other objections include cost (26.1%) and a lack of confidence that Cloud offers greater benefits with respect to local storage media (26.1%). Current cloud systems suffer from drawbacks and do not offer the expected cloud infrastructure characteristics. Research in this area focuses on individual capabilities rather than integrated systems and holistic middleware. The following are some of the challenges for using a cloud infrastructure for the homecare scenario.

1. Trust and Security - have been pointed out as the main barriers to cloud computing. A traditional IT system in a hospital would be strictly bound within the boundary of the hospital. This means that the hospital would be responsible for ensuring that appropriate access control policies are enforced. Using cloud computing means those different, possibly distrusting, organizations will share the same resources. This leads to possibilities of data leakage, policy breach or even unauthorized modification of data. Issues of trust come into play because one has to consider the mechanisms put in place by the provider to enforce the correct policy. In addition, one has to consider whether the service provider has the motivation to tamper with the data and the policy that the owner of the data wishes to enforce.
2. Data lock-in - APIs for Cloud Computing itself are still essentially proprietary, or at least have not been the subject of active standardization (Armburst et al, 2009). The fact that healthcare organization cannot easily migrate their data and software from one Cloud Provider to another is a major implicit risk in the adoption of a cloud infrastructure. For example, Hospitals are required by law to keep medical records for a long period of time, and the "survival" of the Cloud Provider is not guaranteed (as in any new IT market,

competitive pressure, inadequate business strategy, lack of financial support, etc, could lead some providers to go out of business or at least to force them to restructure their service portfolio offering).

3. Privacy - with the possibility of data leakage and policy breach, privacy issues become imminent. Medical records have a nature of being sensitive and therefore any possibility of unauthorized entities accessing the data brings up numerous privacy concerns.
4. Reliability - the promise of a near-to-unlimited storage implies that the hospital can store all the data in the cloud. Reliability issues come into play in such scenarios. For example the hospital would need to rely on the cloud to never lose the data and to make it available when needed.
5. Legal issues - cloud infrastructure, while appearing ubiquitous to users, has to reside in some physical location. The physical locations will be bound to some legislation. As a result questions of who has jurisdiction over cases of policy breach come into play. In addition, disclosure laws may imply that the cloud provider has obligations to disclose the data with or without permission from the hospital and/or patients. Furthermore auditability is another critical legal aspect: the possibility to ensure that the IT system is compliant with existing regulations is very important for eHealth applications, in particular for what concerns the management of patients' data in accordance with privacy protection directives. Cloud providers should ensure the auditability to attract Health Organization in investing in this kind of solutions. This is particularly critical for example in managing EHR or PHR applications, but also in case the Cloud will host and run Hospital Information Systems.
6. Protection of Personal Data and Reliability of Output of the Cloud - personal data are processed and stored in the cloud, which is controlled and managed by the cloud system administrators. A cloud system administrator could abuse his privileges by leaking personal data (confidentiality), removing data (availability) or altering data (Integrity). This means that a robust system should be in place to mitigate such attacks that could undermine the cloud advantages.
7. Multi-tenant Architecture - with cloud computing organizations are provided with virtual resources that share the hardware layer. An attacker could share the same physical resources as another competitor organization. This enables the attacker to learn sensitive information about other organizations (e.g. by exploiting covert channels).

Chapter 3

Security and privacy requirements from technical perspective

Chapter Authors:

Mina Deng, Milan Petković (PHI)

This section provides preliminary requirements derived from the e-Health application regarding the TClouds medical use case scenarios with a focus on security and privacy. This section is built on Section 2 and focuses on the security and privacy requirements from technical aspects.

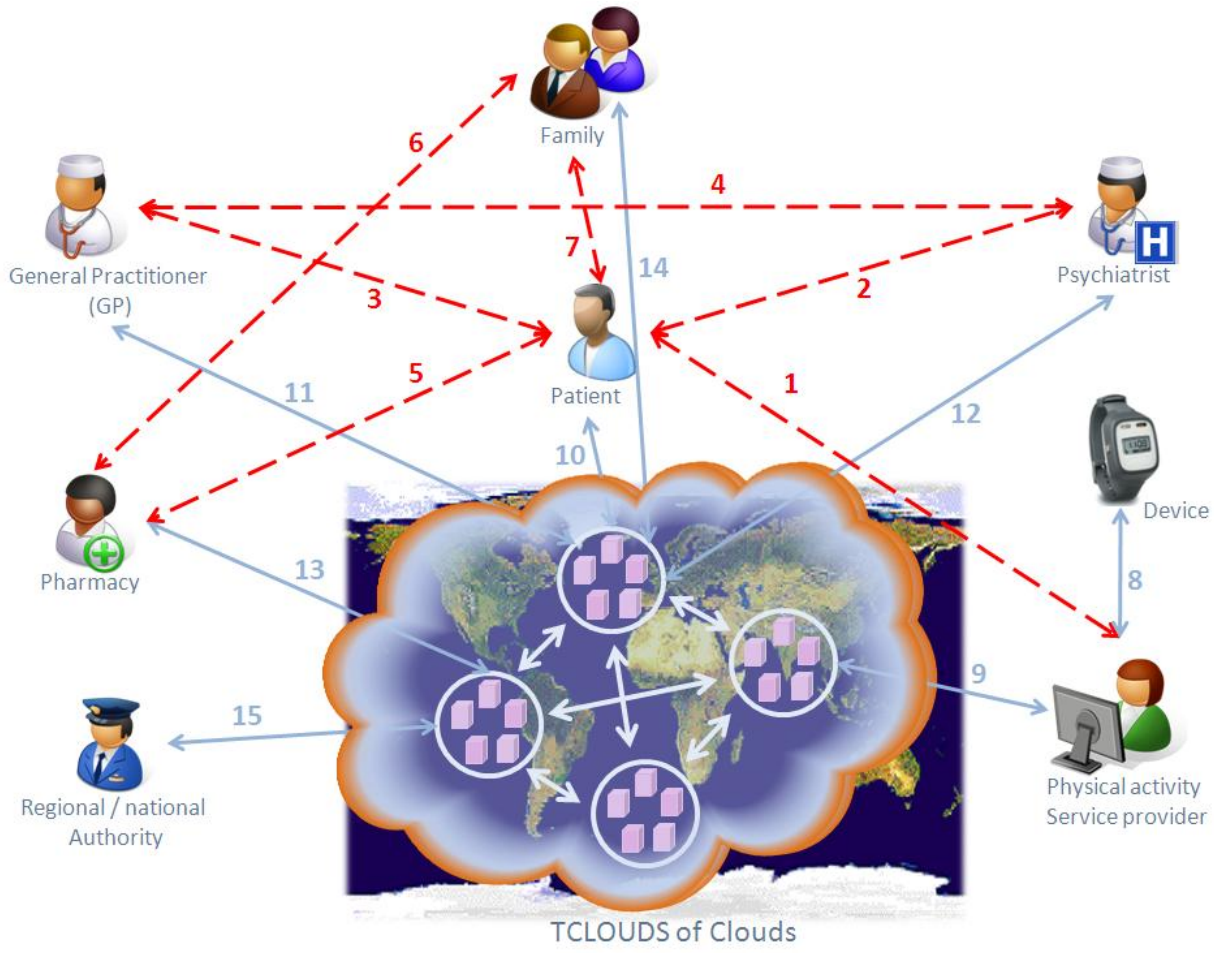

Figure 3 Medical use case architecture bird's eye view

Both the misuse cases and trust assumptions are based on the initial architecture of the TClouds medical use case scenarios as described in Section 2, in which the basic scenario

(in Figure 3) and use cases are described. A number of things need to be clarified for this report:

- The requirements presented in this report are preliminary and will be modified in line with the modifications of the healthcare applications as the project proceeds.
- The requirements are based on the healthcare application. However, no classifications are made to distinguish requirements that should be provided at the platform and infrastructure level (from Activity 2) and those provided at the application level (from WP3.1).

For eliciting the technical requirements for the TClouds healthcare system, two strategies are followed. We name the first one the service-logic driven strategy and the other one the architecture-driven strategy. In particular, the service-logic driven strategy is based on the system functionalities, i.e. analyzing the normal use of the system that are described as use cases and scenarios, and analyzing them for possible security and privacy threats. On the contrary, the architecture-driven strategy is based on system assets (i.e. the objects that need to be protected in the system) and then analyzes the potential security and privacy threats at these assets.

The requirement elicitation with the service-logic driven strategy is based on the system functionalities. The methodology for the architecture-driven strategy uses Data Flow Diagram (DFD) to identify system assets, and a number of security and privacy threat modelling analysis methods to identify the requirements (see Section 3.2 for details).

3.1 Service logic-driven technical requirements

According to the definition from NIST (NIST, 2011) cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The recognized characteristics of the Cloud, as suggested by NIST (NIST, 2011), include on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service.

1. On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed, automatically without requiring human interaction with each service's provider.
2. Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.
3. Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in the cloud, and the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, computation, memory, network bandwidth, and virtual machines.
4. Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, quickly scaled out, and rapidly released or quickly scaled in. Depending on consumer's demands, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

5. Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, and bandwidth). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Cloud computing is expected to offer a number of benefits, including multi-tenancy, scalability, resilience, availability, flexibility, and cost reduction. In this section, we point out a number of specific requirements for healthcare systems in the cloud.

Table 1 Service-logic driven requirements of the healthcare use case

Generic cloud-specific requirements	
Self-managed services	Cloud computing should facilitate automated self-managed services to support clouds' virtual resources availability, reliability, resilience, scalability, security and privacy, and adaptability.
Highly distributed data storage	Data are not stored at local data stores, but data stores are highly distributed in the cloud.
Requirements for healthcare in the cloud	
Semi-trusted (or honest but curious, passive) model	Semi-honest model is assumed that the cloud providers (including cloud employees and system administrators) are semi-trustworthy (or honest but curious).
Data-centric protection	• Electronic health record (EHR) data have to be protected in a highly distributed way by different systems with complex and maybe legacy architectures, even if some of which may not have a trustworthy data management system. • The center of the protection is at data stores/centers.
Emergency access and availability	It is important to guarantee the timely availability of medical data, especially under emergency cases. This in term requires the availability of the decryption key if data are encrypted at data stores.
Efficiency	Access control mechanism must be sufficiently efficient to be leveraged in the processes of medical care. Given the short time doctors currently have to spend with patients, it is unacceptable if the system performance is too slow to satisfy business needs.
Data confidentiality	• Fine-grained access control is required to provide confidentiality of data. • Unlike multimedia or entertainment data, even partial leakage of patients' medical data is undesirable. • The access control policy should not only be role-based, but highly context-based (or rule-based). For instance, patients may have a trust relationship with their current medics, while disregard the relationship with their former medics. • The access control and key management mechanism should be secure and efficient. Private / secret keys should be securely stored and protected. • Data can be potentially accessed by a variable set of parties from different domains with different rights. There is a large uncertainty in who will eventually need to access a data object. It is thus

	implausible to implement central management. • Potential side channel leakage of medical data should be prevented. (For example, the fact that someone takes an HIV test demonstrates that he/she is considered at risk.) It is a desirable to define rules that protect side channel information without disrupting normal healthcare.
Data integrity	• The integrity of medical data should be guaranteed to facilitate the correct medical care for patients. • The integrity of logging / auditing data should be guaranteed to ensure system accountability / auditability.
Accountability	Data access and usage or certain operations in the system have to be logged. In many cases, the context allowing data access cannot be determined automatically, but only verified by a human after the incident. In this regard, auditing is desired with some automated verification procedures.
Patient-centric protection	• Access control: Patients should be able to specify/delegate the access control rights / policies of their medical data. • Usage control: Patient should be able to control how their data is used and to which party it is distributed. • Patients should be aware of their privacy rights (i.e. refer to legal requirements in Section **Fehler! Verweisquelle konnte nicht gefunden werden.**).
Data minimization & anonymization / filtering	• According to the European Data Protection Directive 95/46/EC (EU, 1996), the principle of data minimization means that "a data controller should limit the collection of personal information to what is directly relevant and necessary to accomplish a specified purpose. They should also retain the data only for as long as is necessary to fulfill that purpose. In other words, data controllers should collect only the personal data they really need, and should keep it only for as long as they need it". • Data needs to be anonymized or filtered (i.e. to remove personal identifying information) under certain scenarios, e.g. for clinical research or studies that require data secondary use. Or it is according to patient's privacy preferences, e.g. when PHR is shared with healthcare institutions, it may be necessary to remove part of the data before sharing with healthcare institutions.

3.2 Architecture-driven technical requirements

3.2.1 Security and privacy threat and requirement analysis methodology

We apply a systematic approach for integrating security and privacy threat and requirement engineering process into the system development lifecycle. In Figure 4, the requirement engineering methodology applies STRIDE analysis approach (Lipner, 2006) for security and the LIDDUN analysis approach (Deng, 2010) for privacy threat modelling and requirement elicitation. More detailed description of the security and privacy requirements elicitation methodology can be found in (Lipner, 2006) and (Deng, 2010).

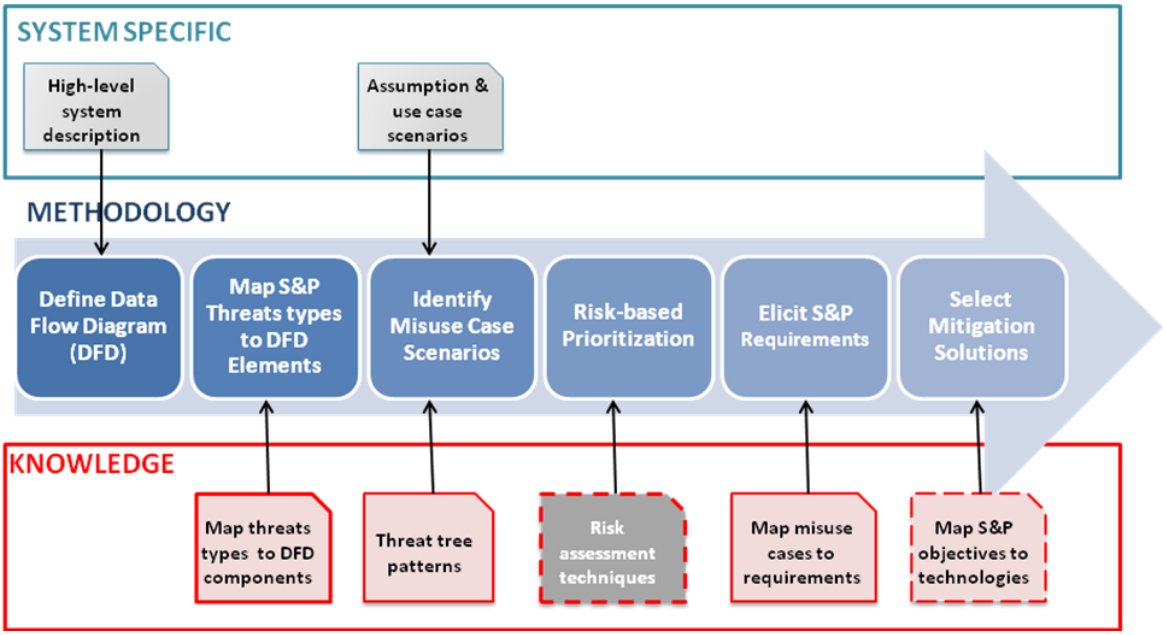

Figure 4 Security and privacy engineering methodology: integrating the security and privacy threat and requirement analysis into the system development lifecycle

First, a data flow diagram is created based on the system architecture. The data flow diagram of the TClouds healthcare system is presented in Figure 5.

Then security and privacy threats are mapped to the DFD elements using Table 2 and Table 3 as a guide to determine the corresponding security and privacy threats. In particular, a number of tree patterns, providing an overview of the most common preconditions of each threat, are applied to detail the security and privacy threat instances in the TClouds healthcare system. The exhaustive list of security and privacy threat tree patterns will not be elaborated in this report. Please refer to (Lipner, 2006) and (Deng, 2010) for detailed explanation.

Next, the identified security and privacy threat types that are relevant to the designated system are documented as misuse cases (cf. Section 3.2.4). A misuse case presents a collection of threat scenarios in the system. The identified privacy threats need to be evaluated and prioritized via risk assessment. Indeed, due to both time and budget constraints, not all threats are worthy further treatment. Note that details on the risk-analysis process will not be discussed in this deliverable. Finally, the security and privacy technical requirements are elicited from the misuse cases.

Table 2 Mapping security threats to DFD element types.

Security threat	Entity	Data flow	Data store	Process
Spoofing	x			x
Tampering		x	x	x
Repudiation	x			x
Information disclosure		x	x	x
Denial of Service		x	x	x
Elevation of privilege				x

Table 3 Mapping privacy threats to DFD element types.

Privacy threat	Entity	Data flow	Data store	Process
Linkability	x	x	x	x
Identifiability	x	x	x	x
Non-repudiation		x	x	x
Detectability		x	x	x
Information disclosure		x	x	x
Content unawareness	x			
Consent / policy noncompliance		x	x	x

3.2.2 TClouds healthcare system functional architecture

The TClouds healthcare use case scenario, as depicted in Figure 1, can be graphically represented using data flow diagrams (DFD), using following elements: data flows (data communication), data stores (logical data or concrete databases, files, etc.), processes (units of functionality or programs) and external entities (end-points of the system like users, external services, etc.). For threat modelling trust boundaries are introduced which represent the border between trustworthy and untrustworthy elements.

A data flow diagram (DFD) is created based on the specified use case scenario. The bird's eye view of the TClouds healthcare system DFD is shown in Figure 5.

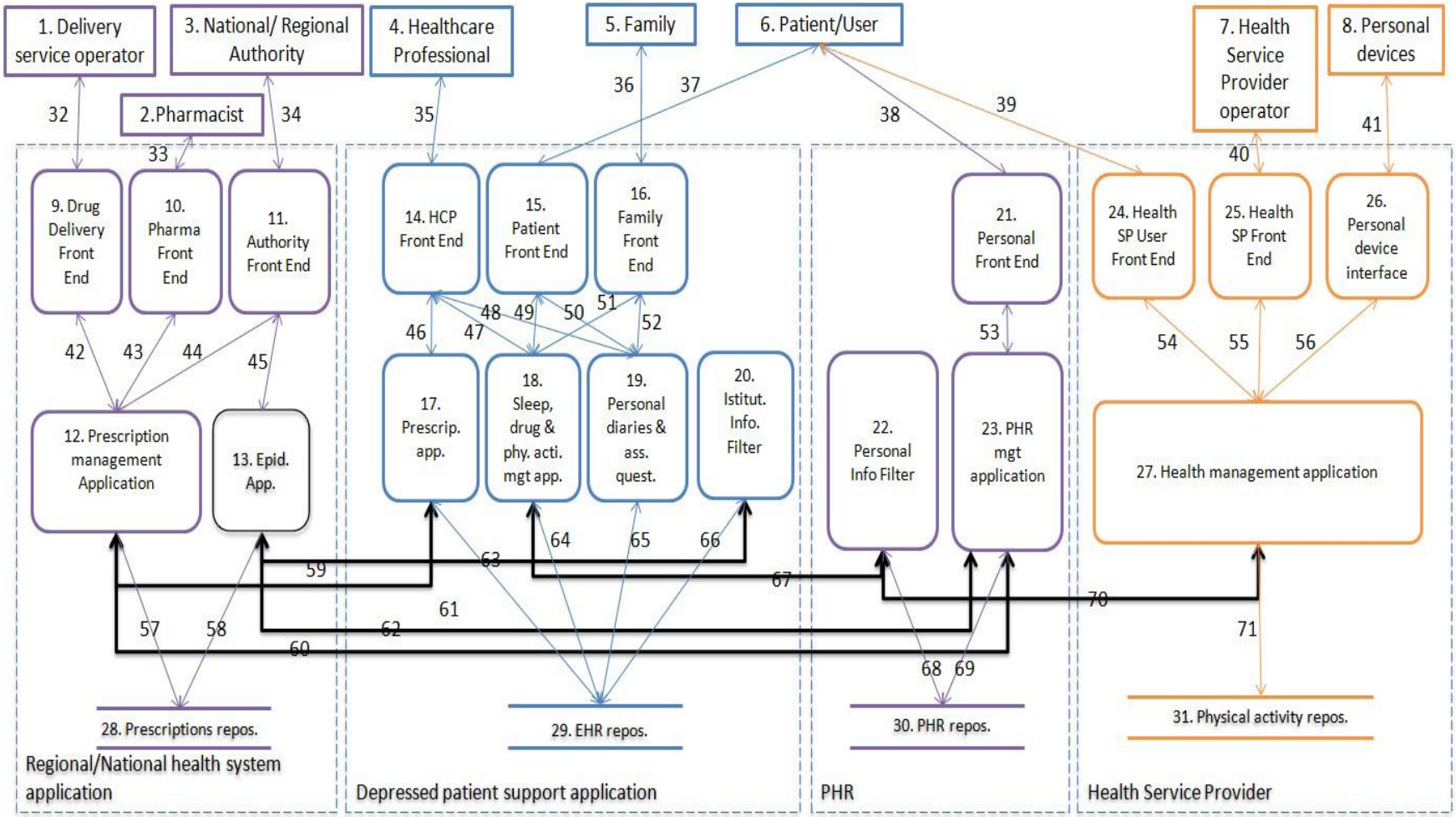

Figure 5 TClouds healthcare system data flow diagram bird's eye view

3.2.2.1 System stakeholders

1. Patient
2. Family
3. Healthcare professionals (including GP and medical professionals such as psychiatrist)
4. Health and wellness service providers operator
5. Pharmacists
6. Regional and/or national authority
7. Delivery service operator

3.2.2.2 Data flow diagram elements

Once stakeholders are specified and the system DFD is defined, all the DFD elements need to be listed in order to protect these elements from security and privacy threats. DFD elements include external entity (e.g. the system stakeholders), data flow (e.g. data communication), data store (e.g. database), and process (e.g. program).

The frames marked with dashed-lines indicate different application domains, including health and wellness service provider, traditional healthcare service provider, PHR service provider, and regional/national institutional service providers. Table 4 lists all the elements in the preceding DFD diagrams.

Table 4 Specification of DFD elements in the TClouds healthcare system data flow diagram

	DFD element (assets)	Description
External Entity	1. Delivery service operator	Providers for delivery services, e.g. to deliver drugs to patients, such as DHL.
	2. Pharmacist	Pharmacist at a pharmacy
	3. National / regional authority	Users (employees or system administrators) of the regional / national authority and infrastructure
	4. Healthcare professional	General practitioner, medical professional, e.g. psychiatrist at hospital, all other healthcare professionals involved in patient's care
	5. Family	Family members of patient
	6. Patient / user	Patient that receives the healthcare services
	7. Health service provider operator	Operators (employee or system administrators) of the health and wellness service provider (e.g. that provides patient's physical activity monitoring service)
	8. Personal devices	Personal monitoring devices, such as monitoring device that collects patient's activity information, such as sleep monitoring information
Process	9. Drug delivery front end	Front end that connects the drug delivery service operator to the cloud
	10. Pharmacy front end	Interface that connects the pharmacist

	DFD element (assets)	Description
	11. Authority front end	Front end that connects the national / regional authority
	12. Prescription management application	Application executed by national/regional authority to manage prescriptions within a particular nation / region
	13. Epidemiology application	Application to perform epidemiology studies
	14. Healthcare provider front end	Front end that connects a healthcare professional
	15. Patient front end	Front end that connects a patient
	16. Family front end	Front end that connects a patient's family member
	17. Prescription application	Application executed by healthcare professionals to issue/manage prescriptions for their patients.
	18. Sleep, drug, and physical activities management application	Application executed by healthcare professionals to provide drug management, sleep management, and physical activities management for their patients.
	19. Personal diaries and assessment questionnaires management application	Application executed by healthcare professionals to manage patient's personal diaries and self-assessment questionnaires.
	20. Institutional information filter	Information filter for data anonymization used by healthcare institutions
	21. Personal front end	Front end that connects a patient
	22. Personal filter	Information filter executed by a patient
	23. PHR management application	Application executed by a person/user to manage the user's personal healthcare record
	24. Health service provider user front end	Front end that connects a user of a health service provider
	25. Health service provider front end	Front end that connects a health service provider operator
	26. Personal device interface	Interface that connects a personal monitoring device
	27. Health management application	Application executed by health service provider to provide health management services to its users
	28. Backend management for Prescriptions repository	Backend process for Prescriptions repository
	29. Backend management for EHR repository	Backend process for EHR repository
	30. Backend management process for PHR repository	Backend process for PHR repository
	31. Backend management process for Activity repository	Backend process for Activity repository
Data	32. Prescription repository	Data store that contains patients' data within a

	DFD element (assets)	Description
Store		nation/region. Data types include: • patients therapy / drug prescriptions (from health professionals) • patients medication purchase history information (from pharmacies)
	33. EHR repository	Data store that contains electronic health records of a healthcare institution's patients (i.e. only used for clinical purposes). Data types include: • patients medical and medication information • patients therapy / drug prescriptions • personal online diary, self-assessment questionnaires (from patients) • organizational policy • patient consent (ref. Section **Fehler! Verweisquelle konnte nicht gefunden werden.** definition)
	34. PHR repository	Data store that contains personal health records of users (note that a user might not necessarily be a patient). Data types include: • drug compliance, patient privacy policy • patients therapy / drug prescriptions (from healthcare professionals) • physical activities (from patients) • personal medical information & medication information • personal information • personal privacy policy
	35. Physical activity repository	Data store that contains data of health service providers. Data types include: • users monitoring data such as sleep monitoring information and physical activities • personal health and wellness advice for users
Data Flow	36. Drug delivery front end data stream	Data stream between Drug Delivery Front End and Drug Delivery Service Operator
	37. Pharmacy front end data stream	Data stream between pharmacy front end and pharmacists
	38. Authority data stream	Data stream between authority front end and national/regional authority
	39. HCP data stream	Data stream between healthcare professional front end and healthcare professionals
	40. Family data stream	Data stream between family front end and family members
	41. Patient data stream	Data stream between patient front end and patients
	42. Personal data stream	Data stream between personal front end and users
	43. Health SP user data stream	Data stream between health service provider user front end and users

	DFD element (assets)	Description
	44. Health SP data stream	Data stream between health SP front end and health SP operators
	45. Personal device data stream	Data stream between personal device interface and personal device
	46. Prescription – delivery data stream	Data stream between prescription management application and delivery service front end
	47. Prescription – pharmacy data stream	Data stream between prescription management application and pharmacy front end
	48. Prescription – authority data stream	Data stream between prescription management application and national/regional authority front end
	49. Study – authority data stream	Data stream between epidemiology study application and national/regional authority front end
	50. Prescription – HCP data stream	Data stream between prescription application and healthcare professionals front end
	51. Management – HCP data stream	Data stream between Sleep, drug, and physical activities management application and healthcare professionals front end
	52. Diary – HCP data stream	Data stream between personal diaries and assessment questionnaires management application and healthcare professionals front end
	53. Monitor – patient data stream	Data stream between Sleep, drug, and physical activities management application and patients front end
	54. Diary – patient data stream	Data stream between personal diaries and assessment questionnaires management application and patients front end
	55. Management – family data stream	Data stream between Sleep, drug, and physical activities management application and family members front end
	56. Diary – family data stream	Data stream between personal diaries and assessment questionnaires management application and family members front end
	57. PHR management – person data stream	Data stream between PHR management application and personal front end
	58. Health management – user data stream	Data stream between health management application and health service provider users front end
	59. Health management – SP data stream	Data stream between health management application and health service provider operator front end
	60. Health management – device data stream	Data stream between health management application and personal device interface
	61. Management – prescription backend data	Data stream between prescription management application and the backend management for

	DFD element (assets)	Description
	stream	prescription repository
	62. Study – prescription backend data stream	Data stream between epidemiology study application and the backend management for prescription repository
	63. Management – prescription data stream	Data stream between prescription management application prescription and application
	64. Prescription management – PHR data stream	Data stream between prescription management application and PHR management application
	65. Study – filter data stream	Data stream between epidemiology study application and institutional information filter
	66. Study – PHR data stream	Data stream between epidemiology study application and PHR management application
	67. Prescription – HER backend	Data stream between prescription application and the backend management for EHR repository
	68. Monitor – EHR backend	Data stream between Sleep, drug, and physical activities management application and EHR repository backend management
	69. Diary – EHR backend	Data stream between personal diaries and assessment questionnaires management application and EHR repository backend management
	70. Filter – EHR backend	Data stream between institutional information filter and EHR repository backend management
	71. Diary – filter	Data stream between personal diaries and assessment questionnaires management application and institutional information filter
	72. Filter – PHR backend	Data stream between personal information filter and PHR repository backend management
	73. Management – PHR backend	Data stream between PHR management application and PHR repository backend management
	74. Health management – Filter	Data stream between personal information filter and health management application
	75. Health management – physical activity backend	Data stream between health management application and physical activity repository backend management
	76. Prescription data stream	Data stream between Prescription backend management and Prescription repository
	77. EHR data stream	Data stream between EHR backend management and EHR repository
	78. PHR data stream	Data stream between PHR backend management and PHR repository
	79. Activity data stream	Data stream between Activity backend management and Activity repository

3.2.3 List of the potential adversaries

This section describes the types of potential attackers the misuse cases (Section 3.2.4). An attacker is someone who intentionally or unintentionally initiates the misuse case. We categorize the attackers in Table 5.

Table 5 Potential attackers

Attacker type	Description
Outsider	A person that is outside of the system, usually with little technical skills (e.g. probably using simple downloaded tools or following a hacking tutorial), that tries to attack the system.
Skilled Outsider	A person that is outside of the system, with advanced technical skills and broad knowledge of security who performs complex attacks.
Clumsy User	A user of the system that performs some actions which unintentionally lead to system failures or security breaches and so on.
Insider	A malicious person within the organization (e.g., a malicious system administrator or employee, a general practitioner, member of the medical staff, etc.), usually with little technical skills (e.g. probably using simple downloaded tools or following a hacking tutorial), that tries to perform attacks to the system.
Skilled Insider	A person that is inside of the system with advanced technical skills and broad knowledge of security who performs complex attacks.

3.2.4 Architecture-driven security threat types and misuse cases

Security threats are identified using the STRIDE analysis (Lipner, 2006) as a part of the Threat Modelling Process. STRIDE is an acronym for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service and **E**levation of privilege.

To identify the security threats based on the system architecture, following Table 2, we map security threat types to the DFD elements that are listed in Table 4.

The starting point for the threat analysis is the architecture of the TClouds healthcare system. The security threat analysis is performed by applying the STRIDE Threat Modelling Process. The resulting threats types at each DFD element is listed in Table 6 and the threat instantiations are documented by misuse cases.

Table 6 Mapping security threat to system assets (S – Spoofing, T – Tampering, R – Repudiation, I – Information disclosure, D – Denial of service, E – Elevation of privilege).

DFD element (assets)		Security threats					
		S	T	R	I	D	E
External Entity	1. Delivery service operator	x		x			
	2. Pharmacist	x		x			
	3. National / regional authority	x		x			

	DFD element (assets)	Security threats					
		S	T	R	I	D	E
Process	4. Healthcare professional	x		x			
	5. Family	x		x			
	6. Patient / user	x		x			
	7. Health service provider operator	x		x			
	8. Personal devices	x	x	x	x	x	x
	9. Drug delivery front end	x	x	x	x	x	x
	10. Pharmacy front end	x	x	x	x	x	x
	11. Authority front end	x	x	x	x	x	x
	12. Prescription management application	x	x	x	x	x	x
	13. Epidemiology application	x	x	x	x	x	x
	14. Healthcare provider front end	x	x	x	x	x	x
	15. Patient front end	x	x	x	x	x	x
	16. Family front end	x	x	x	x	x	x
	17. Prescription application	x	x	x	x	x	x
	18. Sleep, drug, and physical activities management application	x	x	x	x	x	x
	19. Personal diaries and assessment questionnaires management application	x	x	x	x	x	x
	20. Institutional information filter	x	x	x	x	x	x
	21. Personal front end		x		x	x	
	22. Personal filter		x		x	x	
	23. PHR management application		x		x	x	
	24. Health service provider user front end		x		x	x	
	25. Health service provider front end		x		x	x	
	26. Personal device interface		x		x	x	
	27. Health management application		x		x	x	
	28. Backend management for Prescriptions repository		x		x	x	
	29. Backend management for EHR repository		x		x	x	
	30. Backend management process for PHR repository		x		x	x	
	31. Backend management process for Activity repository		x		x	x	
Data Store	32. Prescription repository		x		x	x	
	33. EHR repository		x		x	x	

	DFD element (assets)	Security threats					
		S	T	R	I	D	E
Data Flow	34. PHR repository		x		x	x	
	35. Physical activates repository		x		x	x	
	36. Drug delivery front end data stream		x		x	x	
	37. Pharmacy front end data stream		x		x	x	
	38. Authority data stream		x		x	x	
	39. HCP data stream		x		x	x	
	40. Family data stream		x		x	x	
	41. Patient data stream		x		x	x	
	42. Personal data stream		x		x	x	
	43. Health SP user data stream		x		x	x	
	44. Health SP data stream		x		x	x	
	45. Personal device data stream		x		x	x	
	46. Prescription – delivery data stream		x		x	x	
	47. Prescription – pharmacy data stream		x		x	x	
	48. Prescription – authority data stream		x		x	x	
	49. Study – authority data stream		x		x	x	
	50. Prescription – HCP data stream		x		x	x	
	51. Management – HCP data stream		x		x	x	
	52. Diary – HCP data stream		x		x	x	
	53. Monitor – patient data stream		x		x	x	
	54. Diary – patient data stream		x		x	x	
	55. Management – family data stream		x		x	x	
	56. Diary – family data stream		x		x	x	
	57. PHR management – person data stream		x		x	x	
	58. Health management – user data stream		x		x	x	
	59. Health management – SP data stream		x		x	x	
	60. Health management – device data stream		x		x	x	
	61. Management – prescription repository data stream		x		x	x	
	62. Study – prescription repository data stream		x		x	x	
	63. Management – prescription data stream		x		x	x	
	64. Prescription management – PHR		x		x	x	

DFD element (assets)	Security threats					
	S	T	R	I	D	E
data stream						
65. Study – filter data stream		x		x	x	
66. Study – PHR data stream		x		x	x	
67. Prescription – EHR		x		x	x	
68. Monitor -- EHR		x		x	x	
69. Diary – EHR		x		x	x	
70. Filter – EHR		x		x	x	
71. Diary – filter		x		x	x	
72. Filter – PHR		x		x	x	
73. Management – PHR		x		x	x	
74. Health management – Filter		x		x	x	
75. Health management – physical activity repository		x		x	x	
76. Prescription data stream		x		x	x	
77. EHR data stream		x		x	x	
78. PHR data stream		x		x	x	
79. Activity data stream		x		x	x	

Misuse cases (or “abuse” cases) illustrate attacking scenarios on a targeted system. A misuse case is the inverse of a use case, i.e. a misuse case can be considered as a use case from a point of view of an attacker hostile to the system. These misuse cases are used as input to elicit the architecture-driven technical requirements. Misuse cases indicate what countermeasures are needed in the system. Defining misuse cases is an iterative process: when the system changes and new components are added, new threats can emerge, and therefore, misuse cases and requirements need to be updated accordingly. To describe misuse cases, we use the template, proposed by G. Sindre and A. L. Opdahl (Opdahl, 2001), as described in Table 7.

Table 7 Misuse cases template

Summary	**Short descriptions of the performed attack interactions.**
Assets, stakeholders and threats	This field merges the worst case threats, stakeholders and risks. This field describes what the threats are when the misuse case succeeds. The threats occur at some assets that the stakeholders want to protect.
Primary attacker	This field describes the type of attacker performing the misuse case. For instance, some misuse cases require insiders or people with a certain technical skill, other misuse cases only require a person in general. Some misuse cases could accidentally occur whereas others are most likely performed intentionally. Different types of attackers used in the misuse case description are listed in Table 5.
Basic flow	This field describes the typical flow of the attack interactions, resulting in a

Summary	**Short descriptions of the performed attack interactions.**
	success for the attacker.
Alternative flows	This field describes alternative ways for the misuse case to occur.
Triggers	This field describes how and when the misuse case is initiated. This field is important when something else than the primary attacker (e.g., viruses with time trigger) initiate the misuse case.
Preconditions	This field describes conditions that can be guaranteed by the system, i.e. states of the system that ensure the misuse case to be possible.
Assumptions	This field describes conditions that cannot be guaranteed by the system, i.e. states of the system's environment that ensure the misuse case to be possible.
Capture points and guarantees	The capture points specify the ways in which the threat can be avoided. Capture points could be categorized in detection and prevention. Prevention would simply make the misuse (nearly) impossible, while detection requires actions (or reactions mitigating the threat) to be taken. This field, besides the specification of capture points, also specifies the detection and prevention guarantees. These guarantees specify the guaranteed outcome independent on which prevention or detection path is followed.

The methodology for security threat modelling and requirement analysis can be summarized by the following steps:

1. The TClouds healthcare system is modeled with the Data Flow Diagram (DFD), and the resulting diagram is presented in Figure 5.
2. The assets corresponding to the DFD elements are identified and listed in Table 4.
3. Security threats are determined and each security threat type is mapped to system assets. An overview of all assets is provided in Table 6.
4. The generic security threats from the previous step are instantiated into specific contexts using the threat tree patterns. An exhaustive list of security threat patterns can be found in (Lipner, 2006). These threat instantiations are documented in the form of misuse cases (Section 3.2.4), which are sorted by the assets that certain threat applies to.

3.2.4.1 Misuse cases: external entities

The misuse cases described in this section provide one example to analyze the threats at the patients / users. Similar approaches can be used to analyze misuse cases for the rest of entities (1—8).

3.2.4.1.1 Spoofing the users of the system

Summary	**The attacker gains access to the system by pretending to be an authentic user of the system. The attacker prepares fake credentials or uses credentials of an existing user. To gain access to the credentials of an existing user, the attacker steals the credentials (e.g. by breaking into the computer of a user), guesses the password, etc.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient.
	  • Information disclosure.
	Asset: internal information, electronic health records (EHR), medication history, patient activity data
	• The health care institution, pharmacy, regional infrastructure, health service provider, family, TClouds administration
	  • Information disclosure.
	Asset: Credentials
	• The user of the system.
	  • Credentials disclosure.

Primary attacker	outsider/skilled outsider
Basic flow	1) The attacker prepares fake credentials. 2) The attacker logs to the system using the fake credentials. 3) The attacker gains access to services with the fake identity.

Alternative flows	1) access to authentic credentials
	  a) The attacker gains access to a user's authentic credential (e.g. by guessing or stealing the password, etc.).
	  b) The attacker uses the authentic credentials to log into the system.
	  c) The attacker gains access to the system with spoofed identity.
	2) exploiting a bug
	  a) The attacker circumvents verification of the credentials by exploiting a bug in the authentication system (e.g. a bug in the authentication protocol).
	  b) The attacker gains access to the system.

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	1. Credentials are weak (can be falsified) or are not sufficiently protected (being compromised). 2. There is no authentication system or the authentication system is too weak (e.g. bug in the authentication protocol allows the attacker to omit verification of the credentials).
Assumptions	1. Entities do not sufficiently protect their credentials (e.g. by storing their password in plaintext only without encryption). 2. Entities chose low entropy passwords.
Capture points and guarantees	• **Prevention capture points:** A strong authentication system is present (such as with biometrics, smart-cards, one-time-passwords, using verified protocols, etc.). • **Prevention guarantee:** Only authentic entities gain access to the non-public domain of the system.

3.2.4.1.2 User's Repudiations

Summary	**The attacker performed certain actions (e.g., modified data in data store, or modified cloud subscriber's program), but denies the fact that he/she has ever performed the action.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient.
	  • Information disclosure
	  • Information tampering
	Asset: internal information, electronic health records (EHR), medication history, patient activity data
	• The health care institution, pharmacy, regional infrastructure, health service provider, family, TClouds administration
	  • Information disclosure
	  • Information tampering
	Asset: programs
	• Information flows and executions of the system
	  • Information disclosure
	  • Information tampering

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker logs in to the e-health system. 2) The attacker performed some action (such as accesses the data store, and modifies data in the store). 3) The system has no mechanism or fails to log the attacker's action. 4) The attacker is able to deny that he/she has ever performed the action (such as accessed the data store or modified the data entries).
Alternative flows	1) The attacker logs in to the e-health system with someone else's authentic credentials. 2) The attacker performed some action (such as modifies cloud subscriber's VM). 3) The system has no mechanism or fails link the attacker's action with the right credential (such as the log information is tampered). 4) The attacker is able to deny that he/she has ever performed the action.
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	1. Credentials are weak (can be falsified) or are not sufficiently protected (being compromised). 2. There is no logging system or the logging system is unsecure (or the logging information can be falsified).
Assumptions	1. Entities do not sufficiently protect their credentials (e.g. by storing their password in plaintext only without encryption). 2. The logging information is not well protected.
Capture points and guarantees	• **Prevention capture points:** 1) Each data upload / download / modification / deletion / access (disclosure) / search should be registered / logged with sufficient details (e.g. actor ID, time of action, purpose, action specification, etc.)

	2) Integrity of the logging and registered data need to be guaranteed. Logged data is trustworthy and reliable or is verified (e.g. the user is authenticated) before writing the log, the logging file can serve as a proof. • **Prevention guarantee:** A strong logging mechanism is used to generate logging proofs. All actions performed in the system programs or on data stores are logged. Integrity of logging information is assured.

3.2.4.2 Misuse cases: data flows

The misuse cases described in this section provide an example to analyze the threats at the PHR management – person data stream. Similar approach can be performed to analyze misuse cases for the other data flows (36 – 79).

3.2.4.2.1 Tampering with the PHR management – person data stream

Summary	**The attacker gains access to the communication channel where data is uploaded / downloaded from the personal front end to the PHR management application. The attacker alters the transmitted data, e.g. by a man-in-the-middle attack. The altered data get stored in the patient data repository.**
Assets, stakeholders and threats	**Asset: patient health records (PHR)** • The patient • Information disclosure.
Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker gains access to the communication channel used for transmitting data between the PHR management application and the personal front end and sets up a proxy. 2) Patient starts uploading / downloading his / her PHR data to /from the repository. 3) The attacker captures the message using the proxy. 4) The attacker manipulates the data contents. 5) The attacker forwards the altered message to the repository (man-in-the-middle attack). 6) The manipulated data is stored in the repository.
Alternative flows	1) The attacker directly access the communication channel a) The attacker access to the communication channel between the PHR management application and the personal front end. b) The attacker alters the message by collisions, data gets corrupted.
Triggers	• Initiated by the attacker, i.e. this can happen whenever the data stream is transmitted.
Preconditions	• Communication takes place in a public network or the attacker can gain access to wire.
Assumptions	• Patient does not verify the data he/she has uploaded / downloaded.
Capture points and guarantees	• **Prevention capture points**: a) Accessing the communication channel is protected by encryption or access control or by using private networks (e.g. VPN). (Skilled

	Outsider would need physical access to the wire). b) The message integrity is protected at the application level, e.g. by computing hash or MAC value of the document. c) The channel integrity is protected at the middleware / OS level by using security functionality provided by these layers. • **Prevention guarantee**: Only authentic entities gain access to the non-public domain of the system. • **Detection guarantee**: By checking the hash/MAC value of the message, the corrupted data can be detected and then the message transmission can be retransmitted.

3.2.4.2.2 Information disclosure of the PHR management – person data stream

Summary	**The attacker gains access to the communication channel where data is uploaded / downloaded from the personal front end to the PHR management application. The attacker sets up a sniffer and monitors the traffic on the channel. When the data is transmitted, the attacker saves the content that passes by the sniffer. The attacker keeps a copy of the transmitted data.**
Assets, stakeholders and threats	**Asset: patient health records (PHR)** • The patient • Information disclosure.
Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker gains access to the communication channel used for transmitting data between the PHR management application and the personal front end. 2) The attacker sets up a sniffer on the channel, and monitors the traffic on the channel. 3) When data is transmitted between the personal front end and the PHR management application, the attacker saves the data content that passes by the sniffer. 4) The attacker keeps the copy of the uploaded data.
Alternative flows	1) Man-in-the-middle attack a) The attacker performs a man-in-the-middle attack b) The attacker forwards unaltered data and keeps a copy. c) The attacker breaks the encryption cipher, e.g. by brute force attacks or cryptanalysis, to obtain the plaintext of the data stream. 2) Attacker gains side channel information of the transmitted data stream.
Triggers	• Initiated by the attacker, i.e. this can happen whenever the data stream is transmitted.
Preconditions	1. Communication takes place in the public network or the attacker can gain access to the wire. 2. No or weak encryption / access control of the data stream
Assumptions	There are no side channels that can be used by the attacker.
Capture points and guarantees	• **Prevention capture points:** a) The confidentiality of the data stream is provided at the application level by access control / encryption of the transmitted data, taking patient's privacy policy and consent into consideration.

	b) The confidentiality of the channel is protected by using the security functionality provided by OS/middleware layer. • **Prevention guarantee:** The data on the communicated channel is protected by access control and encrypted using secure access control and encryption mechanisms.

3.2.4.2.3 DoS (denial of service) of the PHR management – person data stream

Summary	**The attacker gains access to the communication channel where data is uploaded / downloaded from the personal front end to the PHR management application. The attacker floods the channel with messages. Once the actual load on the channel exceeds its capacity, the communication channel is rendered useless and the patient data repository becomes unreachable.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Unreachable information.
	Asset: internal network
	• The TCloud of clouds healthcare system
	  • The communication channel between the patient repository and the patient becomes unavailable.

Primary attacker	insider/skilled outsider/clumsy user
Basic flow	1) The attacker gains access to the communication channel where data is uploaded / downloaded from the personal front end to the PHR management application. 2) The attacker floods the channel with messages, e.g. by broadcasting a large amount of Address Resolution Protocol (ARP) requests. 3) The traffic load on the communication channel exceeds its capacity. 4) The communication channel becomes unavailable and the patient repository becomes unreachable.
Alternative flows	1) Scripting a) To automate the uploading /downloading process from the PHR management application, the patient may use a script. b) The script has a bug and sends the documents in an infinite loop. c) The amount of traffic on the channel exceeds its capacity. d) The communication channel becomes unavailable and the patient repository becomes unreachable.
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	1. Communication takes place at the public network or the attacker can gain access to the wire. 2. Scripting is allowed.
Capture points and guarantees	• **Prevention capture points**: a) Custom scripting (e.g., performed by patient) is impossible or prohibited. b) Only authenticated entities can use the communication channel. c) Only private networks / secure channel are used for communication between the patient and the patient repository. • **Detection capture points**: Access to the repository is securely logged.

	• **Prevention guarantee**: Outsiders/skilled outsiders cannot harm the network. Authenticated users cannot accidently flood the network. • **Detection guarantee**: Users flooding the channel can be identified.

3.2.4.3 Misuse cases: data stores

The misuse cases described in this section provide an example to analyze the threats at the PHR repository. Similar approach can be performed to analyze misuse cases for the other data stores (32 – 35).

3.2.4.3.1 Tampering with personal health record (PHR) data

Summary	**The attacker gains access to the distributed patient repository. The attacker puts falsified data directly into the database, deletes data from the database or floods the repository with irrelevant data, resulting in the discarding or overwriting of the stored data.**
Assets, stakeholders and threats	**Asset: patient health records (PHR)** • The patient • Loss, corruption/falsification of the personal health record data
Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker gains access to the patient repository (e.g., by bypassing the PHR management application process). 2) The attacker injects falsified data directly in the PHR repository.
Alternative flows	1) Deleting data a) The attacker removes data directly from the PHR repository.
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	The patient repository data store is accessible from the outside of the PHR management application (i.e., direct access to the data store is possible). Access to the PHR repository is not monitored (e.g. by a firewall) or it can be bypassed. The PHR repository is insufficiently protected by internal security policies.

Capture points and guarantees	• **Prevention capture points**:
	  a) Data modification and deletion is securely logged.
	  b) The patient health data store is protected by internal security policies (e.g. only administrators can perform maintenance tasks, only the PHR management application can write/read data from the data store).
	  c) Extra-monitor access is impossible (e.g. the data store is protected by a reliable firewall).
	  d) Overcapacity failures are handled properly (e.g. system administrators got informed when a certain amount of data is stored).
	  e) Only private networks / secure channel are used for communication between the patient repository and the PHR management application.

	• **Detection capture points**: there is a system monitoring unusual behavior of the users (e.g., sudden uploading a large amount of data), and the system administrator is informed in time. • **Prevention guarantee**: Direct access to the patient health data store is prohibited. The chance of overcapacity failure is significantly lowered. • **Detection guarantee**: The overcapacity failure attack can be detected and stopped in time.

3.2.4.3.2 Information disclosure of PHR repository

Summary	**The attacker gains access to the patient data repository. The attacker searches for hidden data (i.e. the data that is not erased and stays on the storage medium), or accesses the patient health data directly (e.g. by bypassing the portal).**
Assets, stakeholders and threats	**Asset: patient health records (PHR)** • The patient • Information disclosure
Primary attacker	skilled insider/skilled outsider
Basic flow	• Attacker as a skilled outsider: 1) The attacker gains access to the TClouds healthcare system (e.g. by spoofing an authorized user, see “Spoofing the users of the system”) 2) The attacker gains direct access to the PHR repository by bypassing the PHR management application (e.g. the attacker communicates directly with the machine where the data store is located). 3) The attacker reads sensitive data (e.g. patient health data of a VIP) directly from the PHR repository (e.g. the attacker executes SQL queries with a remote access). • Attacker as a skilled insider: 1) Some operations on the data store leave data on the storage medium as hidden data, e.g. when the data store is recovered from a failure, the data is rewritten but it uses different sectors on the hard drive. The data is left on the hard drive, however invisible in the system. 2) The attacker gains access to the hardware where the data store is located (e.g. by stealing the administrator credentials). 3) The attacker searches for hidden data on the data store, e.g. reads empty space of the hard drive sector by sector. 4) The attacker obtains confidential data while leaving no trace of the access logging files (i.e., no files of the data store were accessed, but only the hidden data).

Alternative flows	• Attacker as a skilled insider:
	1) Scanning an old data storage medium.
	  a) When a backup storage medium is replaced, the old medium is erased. However, it still contains hidden data.
	  b) The attacker gains access to the old data storage medium.
	  c) The attacker searches for hidden data on the old data storage medium, e.g. reads empty space of the hard drive sector by sector.
	  d) The attacker obtains confidential data.
	2) Stealing the data storage medium.

	a) The attacker gains physical access to the hardware where the data store is placed. b) The attacker steals the hardware where the patient health data is stored. c) The attacker accesses the data files directly. d) The attacker obtains confidential data.
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• Attacker as a skilled outsider: 1. The PHR repository is accessible from outside the portal (i.e., direct access is possible, PHR management application can be bypassed). 2. Access to the PHR repository is not monitored or it can be bypassed. 3. The clinical data store is insufficiently protected with internal security policies (e.g. policies allowing only the PHR management application to access the PHR repository). • Attacker as a skilled insider: 1. Data store management leaves hidden data on the storage medium.
Assumptions	• There are no side channels that can be used by the attacker. • The PHR repository is physically accessible.
Capture points and guarantees	• **Prevention capture points**: a) Data reading is securely logged. b) The patient health data store is protected by internal security policies (e.g. only administrators can perform maintenance tasks, only the PHR management application can write/read data from the data store). c) Fine-grained control mechanisms are required to access the data storage medium, e.g. considering personal privacy policy. The access control policy should be context-aware. d) For data with a medium or high confidential level, data should be encrypted with secure encryption algorithms, and secure key management schemes should be applied to manage private / secret keys. e) Extra-monitor access is impossible (e.g. the data store is protected by a reliable firewall). f) Only private networks / secure channel are used for communication between the patient repository and the PHR management application. g) During the data backup or deletion, hidden data (i.e. the data that is not erased and stays on the storage medium) should not be produced. Old data storage medium on which the data was stored should be destroyed, and cannot be used for data recovery. • **Prevention guarantee:** a) Bypassing the access control mechanism to access the PHR repository is prohibited (i.e., bypassing the PHR management application to access patient health data is prohibited). b) There is no hidden data on the system. c) The storage medium that stores patient data is inaccessible by unauthorized entities. d) Old data storage medium on which patient data was stored are destroyed and cannot be used for data recovery.

3.2.4.3.3 DoS (denial of service) against PHR repository

Summary	**The attacker causes the patient repository (data store) to be unavailable by sending invalid input (e.g. queries) to the data store, resulting in a crashed or damaged repository, modifying the PHR management application (e.g. changing internal security policies), or sending a large amount of messages directly to the data store.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Unavailability of PHR
	Asset: Patient repository (data store)
	• The TClouds healthcare providers.
	  • Damage of the PHR repository

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker gains access to the TClouds healthcare system (e.g. by spoofing a authorized user, see “Spoofing the users of the system”) 2) The attacker gains direct access to the PHR repository by bypassing the PHR management application (e.g. the attacker communicates directly with the machine where the data store is located). 3) The attacker crashes/damages the data store by sending invalid input directly to the data store (e.g. SQL queries) or performs administrative tasks to lock the data store (e.g. changes internal security policies, locks data files, etc.).

Alternative flows	1) SQL injection attack
	  a) The attacker puts a SQL injection (or any other type of injection) in the message to the repository.
	  b) The injection reaches the data store and crashes/damages it.
	2) Flooding the PHR repository
	  a) The attacker floods the data store by sending a large amount of queries.
	  b) The data store cannot process regular queries (e.g. coming from the repository) and becomes unavailable.

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	1. The PHR repository is accessible from outside the portal (i.e., direct access is possible, PHR management application can be bypassed). 2. Access to the PHR repository is not monitored or it can be bypassed. 3. The clinical data store is insufficiently protected with internal security policies (e.g. policies allowing only the PHR management application to access the PHR repository). 4. Invalid input / queries are able to reach the PHR repository (e.g. SQL injections are possible, and there is no input validation at the PHR management application).

Capture points and guarantees	• **Prevention capture points**:
	  a) Invalid input / queries are filtered out, e.g. by using input validation at the PHR management application.
	  b) The patient health data store is protected by internal security policies (e.g. only administrators can perform maintenance tasks, only the PHR management application can write/read data from the data store).
	  c) Extra-monitor access is impossible (e.g. the data store is protected

	by a reliable firewall). d) Only private networks / secure channel are used for communication between the patient repository and the PHR management application. • **Prevention guarantee:** Bypassing the access control mechanism to access the PHR repository is prohibited (i.e., bypassing the PHR management application to access patient health data is prohibited). The data store is highly available.

3.2.4.4 Misuse cases: processes

The misuse cases described in this section provide an example to analyze the threats at the PHR management application. Similar approach can be performed to analyze misuse cases for the other processes (9—31).

3.2.4.4.1 Spoofing the PHR management application

Summary	**The attacker runs a process (on a machine outside or within the TCloud of clouds healthcare network) that is recognized by the TCloud of clouds healthcare system entities as a legitimate PHR management application.** **Using this process the attacker can** • **communicate with the patient and family front ends,** • **perform passive attacks: eavesdrop / intercept data that passes through the PHR management application, etc.** • **perform active attacks: manipulate / falsify data communicated through the PHR management application, etc.** • **interact with the health virtual machine and the health service provider portal,** • **access to the patient repository (data store), access / falsify / delete the data stored at the patient repository** • **read / set / modify patient's privacy policy settings and establish who can access the patient's data and which part of patient's data.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Falsifying information.
	Asset: patient's privacy policy
	• Patient & the TClouds healthcare providers.
	  • Falsifying information.

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker prepares the process that can spoof a PHR management application. 2) The attacker deploys the process on a TClouds server (performed by a Skilled Insider) 3) The attacker registers the process as an authentic PHR management application (e.g. the process pretends to be a legitimate PHR management application on the system by identity theft). Therefore,

	the process is viewed by the patient and family front ends, the health virtual machine, and the health service provider portal as an authentic source. 4) The attacker runs the process and is able to: a) communicate with the patient and family front ends, b) passive attacks: eavesdrop / intercept data that passes through the PHR management application, etc. c) active attacks: manipulate / falsify data communicated through the PHR management application, etc. d) interact with the health virtual machine and the health service provider portal, e) access the patient repository (data store), access / falsify / delete the data stored at the patient repository f) read / modify patient's privacy policy settings and establish who can access the patient's data and which part of patient's data.
Alternative flows	1) attacks performed by a skilled outsider: a) in Step 2 of the basic flow, the process is deployed on an external server by a skilled outsider
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• Credentials needed for running/registering code are weak (can be falsified) or are not sufficiently protected (can be stolen). • Leverage insufficient or no entity authentication
Capture points and guarantees	• **Prevention capture points**: a) Strong entity authentication is required at the PHR management application, such that only administrators of the TCloud of clouds healthcare system can run / execute / register code on the TClouds servers. b) Credentials needed for running/registering code are strong and well protected c) Only the code that is running on trusted locations can be allowed and executed as a part of the system. d) Messages that are transferred out of the PHR management application are signed. • **Prevention guarantee**: the PHR management application cannot be spoofed. • **Detection guarantee:** Fake PHR management application can be detected in the system (and removed).

3.2.4.4.2 Tampering with the PHR management application

Summary	**The attacker tampers with the PHR management application to modify its functionality. For example, the attacker at the compromised PHR management application could falsify the patient's privacy policy (e.g. to share the patient's PHR data with unauthorized parties) or access and falsify the PHR data stored in the patient repository (e.g. change the suggested medication in the PHR files to an advertised one). Corrupted PHR management application could provide false PHR data (containing irrelevant or fake PHR documents) or perform requests from the patient and family front ends, the health service provider front end, and the health virtual machine.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Information disclosure, unavailability, or falsifying information.
	Asset: patient's privacy policy
	• Patient & the TClouds healthcare providers.
	  • Information disclosure, falsifying information.
	Asset: internal information
	• TClouds healthcare providers.
	  • Information disclosure.
	Asset: availability of the PHR management application and patient repositories
	• TClouds healthcare providers.
	  • Falsifying information (e.g. all returned search results are the same or corrupted), unavailability (e.g. DoS attacks).

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker sends invalid input to the PHR management application process. 2) The message corrupts the state of the PHR management application. 3) While the PHR management application is in the corrupted state, the attacker controls the behavior of that process. 4) The attacker reads the information kept on the PHR management application or modifies the information that is provided by the PHR management application to other parties.
Alternative flows	1) Spoofing of TCloud of clouds healthcare system administrator a) The attacker presents false credentials (i.e., spoofs an administrator) and modifies the functionalities of the PHR management application.
Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• Credentials needed for accessing the PHR management application are weak (can be falsified) or are not sufficiently protected (can be stolen). • Invalid input reaches the PHR management application process. • The administrative part of the PHR management application can be accessible from the outside, regular terminals or clients.

Capture points and guarantees	• **Prevention capture points**:
	  a) Strong entity authentication is required at the PHR management application, such that only administrators of the TCloud of clouds healthcare system can modify the processes at the PHR management application. (It is impossible to spoof the administrators.)
	  b) Credentials needed for modifying the processes at the PHR management application are strong and well protected.
	  c) All inputs are validated.
	  d) Access control mechanism is used such that administrative part of the PHR management application cannot be accessed by unauthorized parties.
	• **Prevention guarantee**: The state of the PHR management application cannot be corrupted by tampering.
	• **Detection guarantee:** The corrupted state of the PHR management application can be detected and the correct state can be restored.

3.2.4.4.3 Repudiate the actions at the PHR management application process

Summary	**The attacker denies having ever made the PHR data available to unauthorized parties (i.e. those who are not supposed to view the data). In the alternative scenario, the attacker gets access to data using the emergency procedures (without any real emergency) or abuse his/hers privileges to access the PHR data. After accessing the data, the attacker denies the act.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Information disclosure
	Asset: internal information
	• TCloud of clouds health network of institutions
	  • Information disclosure

Primary attacker	Insider/Clumsy user
Basic flow	1) The attacker uploads confidential data to the PHR repository. 2) Once on the PHR repository, the PHR is accessed by several unauthorized users. 3) The attacker denies having uploaded the data.

Alternative flows	1) Repudiation of accessing PHR data
	  a) The attacker gets access to data using the emergency procedures (without any real emergency) or abuse his/hers privileges to access the PHR data.
	  b) The attacker denies ever accessing the data.

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• The attacker has a user account. • There is no logging mechanism, or there is a logging system that generates logs that cannot be used for audit.
Assumptions	There are no side channels that can disclose data.

Capture points and guarantees	• **Prevention capture points**:
	  a) The data transmitted by the process should be verified / signed using secure / strong signature mechanism that generates signatures as proof.
	  b) Strong and secure logging mechanism is used that generates logging file as proof.
	  c) Each action or request for data upload / download / modification / deletion / access (disclosure) / search should be registered / logged with sufficient detail (e.g. actor ID, time of action, purpose, action specification, etc.)
	  d) Integrity of the logging / registered data. Logged data is trustworthy and reliable or is verified (e.g. the user is authenticated) before writing the log, the log file can serve as proof.
	  e) Strong authentication scheme is present (such as with biometrics, smart-cards, one-time-passwords, using verified protocols, etc.), such that authentic entities gain access to the non-public domain of the system.
	• **Prevention guarantee:** Uploading and downloading data through the PHR management application cannot be denied.

3.2.4.4.4 Information disclosure of the PHR management application process

Summary	**The attacker gains access to the process data and the state of the PHR management application processes. Once that the attacker has access to the process, the attacker can access the documents or monitor who is accessing what document, etc.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Information disclosure
	Asset: internal information
	• TCloud of clouds health network of institutions
	  • Information disclosure
	Asset: privacy of users
	• Users of the system
	  • Breaching the privacy

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker corrupts the PHR management application process (e.g. the process responsible for retrieving the documents) by e.g. sending invalid input or accessing process memory. 2) The corrupted process allows the attacker to view the state and/or actions at the PHR management application. 3) The attacker discovers confidential information by analyzing the information gained from (and about the states of) the process.

Alternative flows	1) Spoofing an administrator
	  a) The attacker spoofs an administrator or tampers with the persistent storage of the PHR management application (see spoofing the users (MC 3.2.4.1) and tampering with the PHR management application (MC 3.2.4.4.2))
	  b) The attacker modifies the processes of the PHR management application.
	  c) The attacker gains access to the PHR data storage from the PHR management application through the modified processes.

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• It is possible to remotely access the administrative interfaces of the PHR management application processes and/or the memory of the processes. • It is possible to physically access the machine on which the processes.
Assumptions	There are no side channels that can disclose data.

Capture points and guarantees	• **Prevention capture points**:
	  a) All inputs are validated.
	  b) Enforce process confidentiality by means of strong / secure access control. The state, the memory and the actions of the processes at the PHR management application is only accessible by those who have explicit permission (e.g. system administrators).
	• **Detection guarantee:** Access to the state, memory or actions of the processes running at the PHR management application can be detected and the responsible attacker can be identified.

3.2.4.4.5 DoS (Denial of Service) against the PHR management application

Summary	**The attacker performs attacks that crash the PHR management application processes or overload its processing capacity. As a result, the PHR management application becomes unavailable, the requests from the patient and family front ends, the health service provider, the health professional, and the health virtual machine can no longer be processed, and the PHR data stored at the patient repository cannot be reached by the authorized parties i.e. patient repository becomes unreachable.**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Information unavailability (i.e., Data needed for the treatment of the patient cannot be queried or accessed by authorized parties.)
	Asset: the PHR management application
	• TCloud of clouds health network
	  • Damage of the PHR management application

Primary attacker	skilled insider/skilled outsider
Basic flow	1) The attacker sends an invalid input to the PHR management application. 2) The PHR management application process (e.g. the process responsible for managing the patient's privacy policy) crashes. 3) The PHR management application is rendered unavailable.

Alternative flows	2) Tampering with the PHR management application
	  a) The attacker tampers with the PHR management application process.
	  b) The PHR management application crashes
	3) Consuming resources
	  a) The attacker consumes fundamental or application specific resources (e.g. sends more requests than the PHR management application can handle).
	  b) The PHR management application is rendered unavailable
	4) Overloading the PHR management application
	  a) Unusually a large amount of non-malicious calls causes overload on the capacity of the PHR management application
	  b) The PHR management application is rendered unavailable.

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• Anonymous / unauthorized parties can consume resources (e.g. flood the internal network, send requests to the PHR management application).

Capture points and guarantees	• **Prevention capture points**:
	  a) All inputs are validated.
	  b) The PHR management application is load-balanced.
	  c) Only authenticated parties can consume resources from the PHR management application.
	• **Detection capture points**: Access to the PHR management application is securely logged.
	• **Prevention guarantee**: The PHR management application is highly-available.

	• **Detection guarantee:** Attacker flooding network and consuming all of the repository resources can be detected and identified.

3.2.4.4.6 Elevation of privilege at the PHR management application

Summary	**The attacker (as a user of the system) performs actions to provide him/her with privileges for the patient that should not be assigned to him/her. For example the attacker presents false credentials (e.g. a stranger that pretends to be a patient's family member or a nurse pretending to be a doctor), spoofs a user with more privileges (e.g. a system administrator) or tampers with the PHR management application to falsify its security / privacy policies. The attacker then abuses the gained privileges by performing actions that he/she are unauthorized for (e.g. accessing sensitive information, performing administrative tasks on the PHR management application).**

Assets, stakeholders and threats	**Asset: patient health records (PHR)**
	• The patient
	  • Information disclosure
	Asset: the PHR management application
	• TCloud of clouds health network
	  • Damage of the PHR management application

Primary attacker	skilled insider
Basic flow	1) The attacker logs in. 2) The attacker presents false credentials (e.g. a stranger that pretends to be a patient's family member or a nurse pretending to be a doctor). 3) The attacker gains more privileges to the repository (e.g. .a stranger that gains the privileges of a patient's family member, or a nurse that gains the privileges of a doctor). 4) The attacker abuses gained privileges (e.g. by searching / accessing patient's health data files that otherwise would be inaccessible by the attacker).

Alternative flows	1. Tampering with the PHR management application
	  a) The attacker tampers with the PHR management application to gain more privileges.
	  b) The attacker abuses gained privileges (e.g. by accessing medical files that otherwise would be inaccessible by the attacker).
	2. Spoofing an authorized party
	  a) The attacker spoofs a user with more privileges (e.g. an administrator).
	  b) The attacker abuses gained privileges (e.g. by changing the security / privacy policies at the PHR management application, for instance, to access the patient's sensitive health data).

Triggers	• Initiated by the attacker, i.e. this can happen at any time.
Preconditions	• The attacker has an account.

Capture points and guarantees	• **Prevention capture points**:
	  a) There is an entity authentication mechanism that verifies user's credentials.
	  b) Sufficient authorization is leveraged.

	c) All inputs are validated. d) Spoofing the users and tampering with the PHR management application (see misuse cases “spoofing the entity” and “tampering with the PHR management application”) are impossible. • **Detection capture points**: Access to the PHR management application is securely logged. • **Prevention guarantee**: Users can only have the privileges as specified in the policy. • **Detection guarantee:** The attacker elevating privilege at the PHR management application can be detected (e.g. by comparing the logs and the policy) and identified.

3.2.5 Architecture-driven security requirements

Table 8 Mapping of security threat analysis to security objectives

Security threat types (STRIDE)	Elementary security objectives
Spoofing an external entity or process	Authentication
Tampering with data store	Integrity of stored data
Tampering with data flow	Integrity of transmitted data
Tampering with a process	Integrity of application
Repudiation by entities	Non-repudiation
Repudiate a process	Auditability
Information disclosure of data store	Confidentiality of stored data
Information disclosure of data flow	Confidentiality of transmitted data
Information disclosure of a process	Confidentiality of application
DoS against data store	Availability of stored data
DoS against data flow	Availability of transmitted data
DoS against a process	Availability of application
Elevation of Privileges for processes	Authorization

Security requirements are elicited from capture points and guarantees of the security misuse cases and are summarized in Table 9.

Table 9 Architecture-driven security requirements for the healthcare use case in the cloud

Security threats	Security requirements
Spoofing an entity (1 — 8)	Authentication of the entities, including patient, family, health professional, pharmacist, regional infrastructure user, delivery service operator, and health service provider operator. a) Strong authentication scheme is present (such as with biometrics, smart-cards, one-time-passwords, using verified protocols, etc.), such that authentic entities gain access to the non-public domain of the system.

Security threats	Security requirements
Entity repudiations (1 — 8)	Non-repudiation services of the entities (1 — 8) (i.e. accountability and integrity of logging data): • A strong logging mechanism is used to generate logging proofs. • Each data upload / download / modification / deletion / access (disclosure) / search should be registered / logged with sufficient details (e.g. actor ID, time of action, purpose, action specification, etc.) • Integrity of the logging and registered data need to be guaranteed. Logged data is trustworthy and reliable or is verified (e.g. the user is authenticated) before writing the log, the logging file can serve as a proof. • Strong authentication scheme is present (such as with biometrics, smart-cards, one-time-passwords, using verified protocols, etc.), such that authentic entities gain access to the non-public domain of the system. (i.e., to prevent spoofing the entity)
Spoofing the process (9 – 31)	Entity authentication at processes (9 –31): a) Strong entity authentication is required at processes (9 –31). b) The monitoring device needs to be authenticated before transmitting data to / from the personal device interface. c) Only administrators (or authorized parties) can run / execute / register code on the TClouds servers. d) Credentials needed for running/registering code both at the client side and in the TCloud of clouds health system are strong and well protected. e) Only the code that is running on trusted locations can be permitted and executed as a part of the system. f) Messages that are transferred out of the process are signed.
Tampering with the process (9 – 31)	Integrity of processes (9 – 31): a) Strong entity authentication is required at processes (9 – 31). b) Communicating devices (e.g. the personal monitoring device) must be able to assess the trustworthiness of the process / platform they are communicating with. c) Credentials needed for modifying the processes are strong and well protected. d) All inputs are validated. e) Access control mechanism is used, such that administrative part of the processes is inaccessible by any unauthorized parties.
Repudiation against the process (9 – 31)	Non-repudiation services (i.e., accountability & integrity of logging data) at processes (9 – 31): a) The data transmitted by the process should be verified / signed using secure / strong signature mechanism that generates signatures as proof. b) Strong logging mechanism is used that generates logging file as proof. c) Each action or requests for data upload / download / modification /

Security threats	Security requirements
	deletion / access (disclosure) / search should be registered / logged with sufficient detail (e.g. actor ID, time of action, purpose, action specification, etc.) d) Integrity of the logging / registered data. Logged data is trustworthy and reliable or is verified (e.g. the user is authenticated) before writing the log, the log file can serve as proof. e) Strong authentication scheme is present (such as with biometrics, smart-cards, one-time-passwords, using verified protocols, etc.), such that authentic entities gain access to the non-public domain of the system. (i.e., to prevent spoofing the entity
Information disclosure of the process (9 – 31)	Confidentiality of processes (9 – 31): a) Enforce process state information confidentiality by means of strong / secure access control. Only those who have explicit permission (e.g. system administrators) can access the state, the memory and administrative interfaces of the aforementioned process (or the process is inaccessible). b) All inputs are validated.
DoS against process (9 – 31)	Availability of processes (9 – 31): a) All inputs are validated. b) The patient portal is load-balanced. c) Only authenticated parties can access / consume resources of the process. d) Access to the process is securely logged.
Elevation of Privilege for the process (9 – 31)	Authorization at processes (9 – 31): a) Entity authentication mechanism is available to verify user's credentials. b) Sufficient authorization is leveraged at the process. c) All inputs are validated. d) Spoofing the users and tampering with the process (see misuse cases 3.2.4.1.1 and 3.2.4.4.2) are impossible. e) Access to the process is securely logged.
Tampering with data store (32 – 35)	Integrity & Confidentiality of stored data in (32 – 35): a) Data modification and deletion is securely logged. b) The data store is protected by internal security policies (e.g. only administrators can perform maintenance tasks). c) Extra-monitor access is impossible (e.g. the data store is protected by a reliable firewall). d) Overcapacity failures are handled properly (e.g. system administrators got informed when a certain amount of data is stored). e) Only private networks / secure channel are used for communication between data stores (32 – 35) and middle-tier applications (12, 13, 17—20, 22, 23, 27)
Information disclosure of data store (32 – 35)	Confidentiality of stored data in (32 – 35): a) Data stores are protected by internal security policies (e.g. only administrators can perform maintenance tasks). b) Fine-grained control mechanisms are required to access the data storage medium, e.g. considering personal privacy policy. The access control policy should be context-aware. c) For data with a medium or high confidential level, data should be encrypted with secure encryption algorithms, and secure key

Security threats	Security requirements
	management schemes should be applied to manage private / secret keys. d) Data access / modification / deletion actions are securely logged. e) Extra-monitor access is prohibited (e.g. the data store is protected by a reliable firewall). f) Only private networks / secure channel are used for communication g) During data backup or deletion, hidden data (i.e. the data that is not erased and stays on the storage medium) should not be produced. Old data storage medium on which the data was stored should be destroyed.
DoS (Denial of service) against data store (32 – 35)	Availability of the stored data in (32 – 35): a) Invalid input / queries are filtered out, e.g. by using input validation at the patient portal. b) Data stores are protected by internal security policies (e.g. only administrators can perform maintenance tasks, only the patient portal can write/read data from the data store). c) Extra-monitor access is prohibited (e.g. the data store is protected by a reliable firewall). d) Only private networks / secure channel are used for communication accessing the data store.
Tampering with the data stream (36 – 79)	Integrity of the transmitted data in (36 – 79): a) Accessing the communication channel is protected by encryption or access control or by using private networks (e.g. VPN). (Skilled Outsider would need physical access to the wire). b) The message integrity is protected at the application level, e.g. by computing hash or MAC value of messages. c) The channel integrity is protected at the middleware / OS level by using security functionality provided by these
Information disclosure of the data stream (36 – 79)	Confidentiality of the transmitted data in (36 – 79): a) The confidentiality of the data stream is provided at the application level by encryption of the transmitted data. b) The confidentiality of the communication channel is protected by using the security functionality provided by OS/middleware layer.
Dos of the data stream (36 – 79)	Availability of the transmitted data in (36 – 79): a) Custom scripting (e.g., performed by patient) is prohibited. b) Only authenticated entities can access the communication channel. c) Only private networks / secure channel are used for communication.

3.2.6 Architecture-driven privacy requirements

For the healthcare system in the cloud, we apply the methodology for privacy threats modelling and privacy requirements elicitation proposed in (Deng, 2010). Classic privacy threat types include linkability, Identifiability, non-repudiation, detectability, information disclosure, content unawareness, and consent / policy noncompliance. The corresponding privacy objectives are presented in Table 10.

Each privacy threat can be mapped to the system data flow diagram component following the relation presented in Table 3. Certain privacy-enhancing properties, namely repudiation and undetectability, are not desired in this healthcare system. On the contrary, non-repudiation is required to guarantee accountability (audibility).

Table 10 Privacy threats and objectives

Privacy threat types	Privacy objectives
Linkability	Unlinkability
Identifiability	Anonymity / pseudonymity
Non-repudiation	Plausible deniability
Detectability	Undetectability
Information disclosure	Confidentiality
Content unawareness	Content awareness
Consent / policy noncompliance	Consent / policy compliance

Information disclosure threat is a threat for both security and privacy, and the related misuse case is already discussed in the section of security requirements.

In the following, we will consider linkability, Identifiability, content unawareness and consent / policy noncompliance as privacy threats. Following mapping between the privacy threat types and each element in the system model, privacy misuse cases (i.e. threat scenarios) can be identified. Note that this report will not discuss privacy misuse cases in detail. Instead, we will provide an overview of the identified privacy requirements based on the aforementioned privacy objectives. Privacy requirements are summarized in Table 11.

Table 11 Architecture-driven privacy requirements for the healthcare use case in the cloud

Privacy threats	Privacy requirements
Consent / policy noncompliance	Consent / policy compliance: a) The stakeholders of the TClouds medical system should process patient's personal data in compliance with patient's consent, e.g., should not disclose the EHR database to third parties for secondary use. b) Design system in compliance with legal guidelines for privacy and data protection and keep internal policies consistent with policies communicated to the stakeholders c) Legal enforcement: user can sue the responsible party (accountability) whenever his or her personal data is processed without consent / against the user's privacy policy. d) Employee contracts: employees / entities that share information with unauthorized third parties will be penalized. e) Note: Legal enforcement will lower the threat of an insider leaking information but it will still be possible to breach user's privacy.
User's content awareness	User's content awareness: a) Patient-centric protection: patient should be able to specify his/her privacy policy, who access which part of his/her data, e.g. Psychiatrist, GP, and pharmacy, and all other professional actors involved in the patient's care. These parties should be able to access information about patient's (e.g. diseases development, drug compliance, physical activity) according to the permission

Privacy threats	Privacy requirements
	level specified by the patient/user. b) right to request restrictions to access PHR c) right to receive an accounting of disclosures of PHR d) right to amend, inspect and copy PHR e) right to receive confidential communications regarding PHR (e.g. specify location, means) a) Patient (or healthcare professional) ensures his/her data is updated when it is expired, to prevent from wrong decisions made on incorrect data, e.g. using system reminder. b) Patient (or healthcare professional) is aware that providing too much personal data brings risk to his/her privacy. User provides only minimal set of required information (i.e. to follow the data minimization principle).
Linkability of an entity	Unlinkability of pseudonyms (user IDs) of TClouds healthcare system privacy-sensitive users (they are scenario specific). a) **Conditional**: only for privacy concerned (sensitive) users, such as patient or health professionals, depends on trust model b) Pseudonymize users IDs. Ensure user's pseudonyms change over time. c) User privacy awareness: inform users that using real ID has a risk for privacy violation. d) Message and channel confidentiality should be provided.
Identifiability of an entity	Conditional (revocable) anonymity of TClouds healthcare system privacy-sensitive users (they are scenario specific), such as patient or health professionals. a) **Conditional**: only for privacy concerned (sensitive) users, such as patient or health professionals, depends on trust model b) Anonymity can be revoked if necessary in order to ensure system accountability (non-repudiation). c) Under emergency cases, patient never stays anonymous.
Linkability of data flow (36—79)	Unlinkability of the transmitted data (36—79) (e.g. anonymous delivery of drug / therapy prescriptions): a) Channel confidentiality should be ensured by deploying secure communication channel, such as mutual certificates with secure encryption mechanism. b) (Assume there is no need to deploy anonymous communication channels.)
Identifiability of data flow (36—79)	Anonymization (or pseudonymization) of the transmitted data (36—79): a) Channel confidentiality should be ensured by deploying secure communication channel, such as mutual certificates with secure encryption mechanism. b) (Assume there is no need to deploy anonymous communication channels.)
Linkability of documents stored in the data store (32—35)	Unlinkability of data entries / documents in the data stores (32—35) against unauthorized parties or for secondary use: a) **Conditional**: depends on the trust model, unlinkability of documents, i.e. against unauthorized parties or for the purpose of secondary use. b) Use data anonymization techniques to anonymize the documents

Privacy threats	Privacy requirements
	stored in the data store. c) Enforce data protection by means of access control, while taking patient's privacy policy and consent into consideration.
Identifiability of data store (32—35)	Anonymization (or pseudonymization) of the data entries / documents in the data stores (32—35) against unauthorized parties or for secondary use: a) **Conditional**: depends on the trust model b) Use data anonymization techniques to anonymize the documents stored in the data store. c) Enforce data protection by means of access control, while taking patient's privacy policy and consent into consideration.
Linkability of the process (9—31)	Unlinkability and confidentiality of the process (9—31) (aligned with the corresponding security requirement): a) **Conditional**: Different actions / accesses to the process cannot be linked to the same actor except for parties who have explicit permission (e.g. system administrators). b) Enforce process confidentiality by means of strong / secure access control. Only those who have explicit permission (e.g. system administrators) can access the state, the memory and administrative interfaces of the process (or it is inaccessible). c) All inputs are validated.
Identifiability of the process (9—31)	**Conditional** (Revocable) anonymity of privacy sensitive users (patients or healthcare professional) such that the entity will not be identified from the application process memories, by unauthorized parties (those without explicit permission to access the process memory states, etc.)

3.3 Discussions

1. The requirements for the TClouds medical use case that have been discussed in this report are to summarize the service-logic driven technical requirements, architecture-driven technical requirements, as well as the requirements identified from the legal perspective.
2. We analyzed both privacy and security requirements together as part of the Security Development Lifecycle (Lipner, 2006). Security is a necessary means to achieve privacy. In addition, in spite of the coexistence of security and privacy properties in one system, some security objectives might conflict with some privacy objectives. Therefore, it is important to consider the service-logic driven requirements to identify the desirable objectives of the system (e.g. repudiation and plausible deniability as privacy properties are not desired in the TClouds healthcare system). It is thus useful to consider requirements both for security and privacy together.
3. There are some tradeoff between security and privacy with system performance, e.g. in terms of cost and efficiency (Deng, 2010). To facilitate this, it is important to find a proper balance between requirements and system performance, while taking the system's practical constraints into consideration. One tradeoff is between privacy and efficiency; the other is between privacy and cost. These two tradeoffs are interactive. Generally speaking, building security privacy in is usually at the price of increasing the implementation budget or lowering the performance efficiency of the system.

Therefore, it is important to identify what are the relevant requirements and to apply risk assessment (ENISA, 2009 November) to prioritize these requirements.

3.4 Conclusion

This section provides preliminary requirements derived from the e-Health application with respect to the TClouds medical use case scenarios with a focus on security, privacy and legal issues.

The requirements presented in this report are preliminary and will be modified in line with the modifications of the healthcare applications when necessary.

The technical requirements are based on the healthcare application. Further investigation is necessary to distinguish requirements that should be provided at the platform and infrastructure level (from A2) and those provided at the application level (from WP3.1). Moreover, the identified technical requirements still need to be prioritized and balanced with the complexity of the middleware and system performance. These aspects will be discussed in Section 7.

Chapter 4

Preliminary view of legal issues of the medical use case

Chapter Authors:

Eva Schlehahn (ULD), Mina Deng (PHI)

4.1 Introduction and scope of the legal issues overview

This overview is intended as preliminary input for WP 3.1 regarding the legal issues for the TClouds medical use case. In this use case, an elementary aspect is the collection, processing and storing of personal data of depressed patients in a cloud computing environment. Due to the complexity of the use case and the difficulties to realise adequate protection of sensitive data in cross-border cloud systems, it is necessary to research the arising legal issues and search for possible solutions. Still, this overview is by no means a complete analysis of the legal requirements concerning this scenario. Nevertheless, it already outlines some arising problems for storing and processing medical data remotely in a cloud computing environment. It also gives some first guidelines how the electronic patient file must be composed to comply with the general data protection framework on EU and national level.

4.2 Basic terminology and concepts

In this part we define some basic terminology and concepts. Some of these terms were already defined in the TClouds deliverable D1.2.2 [Marnau, Schlehahn, *Cloud Computing: Legal Analysis*, see there under Annex A - Exemplary role model and Annex B - Basic terminology and concepts]. Nevertheless, there are some additional ones which are specifically relevant in regard to the medical use case of the project. This list is not complete yet and will be reworked for the report R1.2.2.2 [*Legal analysis and requirements "Patient monitoring"*]. So far, it focuses only on the main ambiguities that may exist regarding the general terminology.

4.2.1 Electronic patient file

The electronic patient file is the sum of all administrative and medical information in regard of a certain patient in the cloud environment.

4.2.2 Anonymisation

"Rendering anonymous" shall mean the alteration of personal data so the comprised information cannot be referenced to an identified or identifiable natural person or that such

reference would require an exorbitant amount of time, expense and effort (see also definition of pseudonymisation).

4.2.3 Pseudonymisation

"Pseudonymisation" or "aliasing" means the replacement of the data subject's name and other identity-related features with a dissimilar identifier to preclude or hinder the identification of the data subject. In contrast to anonymisation, the data is still related to a specific identifier. So there is still the danger of linkability and traceability back to the real identity of the patient. Thus, whenever possible, the method of anonymisation should be preferred to protect the patient's privacy.

4.2.4 Data minimisation

"Data minimisation" means that the patient is not forced to disclose more personal data that is absolutely necessary for the medical treatment. For example, this also means that from the medical professional's or physical activity service provider's point of view, they only obtain the data that is absolutely necessary to perform their specific duties. As a consequence, the main objective shall be the need-to-know or need-to-retain principle in an adequate and non-excessive manner. For example, a general practitioner shall not learn the content of a psychiatrist's notes on counselling. Pseudonymisation as well as anonymisation is a measure that supports data minimisation.

4.2.5 Deletion of data

"Deletion" means the irreversible removal or obliteration of data, so the access to this data is by no means possible anymore. Tagging data as "deleted" and only blocking the access to is not considered as deletion.

4.2.6 Blocking of data

"Blocking of data" is the labelling of data to limit the further processing. The access to blocked data may only be possible under narrow preconditions. An example may be the limitation of access to the concerned operation department of the hospital after the completion of the medical treatment. In this case, the access of other departments or even externals is blocked.

4.2.7 Blanking of data

"Blanking of data" means the possibility to exclude certain items of information to they will not be displayed at all to someone without authorisation. This could be useful for use cases where the patient does not want to disclose information of individual clinical or other events to one certain doctor. For example, the information that the patient receives mental health treatment by a psychologist could be "blanked" (hidden) towards a general practitioner. This possibility is proposed the Italian DPA [Il Garante per la protezione dei dati personali, "Linee guida in tema di Fasciolo sanitario electtronico (Fse) e di dossier sanitario", 2009].

4.2.8 Duplication of data

The duplication of data means the copying of data or parts of it into another database or category. This procedure may be useful in cases where different entities need access to the same information while having access rights only to separate parts of the electronic health file. Nevertheless, the duplication of data is counterproductive in respect of data minimisation. Instead, the primary goal must be to establish purpose-bound access authorisations in regard to the concerned data parts. This view is strongly supported by the European data protection authorities [exemplary: Il Garante per la protezione dei dati personali, “Linee guida in tema di Fasciolo sanitario electtronico (Fse) e di dossier sanitario”, 2009].

4.2.9 Data portability

Data portability means that in cases where the patient decides to get the medical treatment by another medical professional or hospital, it must be possible to migrate the content of the electronic patient file out of the cloud environment into another system.

4.2.10 Separation of data parts

The separation of data parts means the organisation of the data set as a functional separation for further processing. The separation may be enforced by physical separation (e.g. storing in different data bases or on different servers), but also by separation within a complete set by logical differentiation via tagging of data. However, the separation must always be purpose-bound in regard to the collection, processing and storing of the data and must be enforced by corresponding access controls. So only the professional duty of each entity determines the nature and amount of its access rights.

4.2.11 Multi-tenancy

"Multi-tenancy" means the sharing of resources across a large pool of customers and/or users by measures that enable each user to only access and process his own data without interfering with other users. The separation of data parts is an effective measure to enforce the multi-tenancy.

4.2.12 Processing context

The processing context is closely related to the purpose-binding of the processing of personal data and refines it. Examples for processing contexts are patient admissions, medical treatment, hospital and home care, quality management, etc. Correlating to these processing contexts, refined access control functionalities are mandatory.

4.2.13 Sticky policies

Sticky policies are a way to cryptographically associate policies to encrypted (personal) data. These policies function as a gate keeper to the data. The data is only assessable when the stated policy is honoured [Leenes, Schallaböck, Hansen, *PRIME* - White Paper v3, 7].

4.2.14 Break Glass procedure

A “Break glass procedure” is a way to temporarily overrun an access restriction in regard to a patient’s data against the internal access control policy and without the patient’s specific consent. This procedure is primarily meant for emergency cases, where the acting party, e.g. a licensed practitioner, has no emergency access privileges but must be able to access the electronic patient file as a whole or parts of it for a potentially life-preserving treatment. For such emergency circumstances, the electronic patient file must have a well-defined procedure to allow access via alternate and/or manual methods, if a medical professional has no regular access authorisation. Other possible use cases might be account problems (forgotten username/password) or system failures. In all cases of this “break glass” procedure, the access-demanding person must get a warning that no regular authorisation is possible and with further proceeding, unauthorised access will be obtained. A precise declaration about, who desires access and why an emergency case is assumed, must be given. Furthermore, the access must be temporary and the event will be logged as well as reviewed, ideally be the local DPA authority [cf. Yale University, Introduction to HIPAA]. In some European countries, the log review may be performed by entity-internal supervisors. However, in some EU member states, the local data protection authorities are obliged and authorized to scrutinize the process of emergency accesses.

4.3 General requirements

This section addresses a first outline of precise requirements that the electronic health system in the cloud must provide. The Section is sub-sectioned in four main parts: Legal groundwork, basic principles and technical as well as organisational requirements and measures. These parts complement and are partially built upon each other.

4.3.1 International and national law as groundwork

The legal issues addressed in this paper are primarily focusing on the requirements as stated in the EU directives and the OECD guidelines listed below:

- EU Data Protection Directive 95/46/EC
- EU E-Privacy Directive 2002/58/EC and its amending Directive 2009/136/EC
- OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, adopted on 23 September 1980, in its respective latest version

Of these three frameworks, the European Data Protection Directive is the main element to elicit the fundamental requirements of data protection and privacy in a cloud computing related scenario. However, the articles of this directive are not coercively binding laws that do apply directly in all EU member states. Rather, the legal nature of the directive is in such a way that it gives a rough groundwork and guidelines for the realisation of its objectives in the national data protection laws of the EU member states. As the member states adopt the objectives of the EU directive, 95/46, they have the liberty to make individual regulations in certain areas, such as in the field of health data. So, in respect to article 8 paragraph 1 EU Data Protection Directive, member states are authorised to even apply narrower regulation on the collection, storage and processing of health data than the directive does. As a consequence, since the TClouds project medical use case involves the Italian hospital San Raffaele, the national data protection law of Italy must be considered. Therefore, the laws of the Codice in material di protezione dei dati (personali Decreto legislativo 30 giugno 2003, n.

196), must be included into this first overview of legal requirements as well to work out the country-specific data protection requirements, which go beyond the EU framework.

4.3.2 Basic principles

According to Section 75 et Section 76 DP Code, the processing of personal data in the health field always must be compliant with the self-determination principle. This means that every data subject - in the medical use case the patient - must have the possibility to decide freely if his or her data can be collected, stored and processed in the electronic patient file. Denied or withdrawn consent shall have no negative effects for the patient in any way, e.g. via withheld benefits concerning the assumption of costs or the quality of the medical treatment.

To ensure protection of an individual's personal data, the legal framework EU level as well as in national law laid down some basic principles. These principles are:

- Purpose
- Transparency and information
- Informed consent
- Control functionalities

These principles apply generally in all cases of personal data processing. However, they become increasingly important in cloud computing contexts due to the outsourced factual control over the data. The following principles form the basis of security and privacy for personal data processing, regardless of whether the processing takes place in an internal hospital information system or within a cloud computing environment.

Purpose

The EU directive 95/46/EC states in Article 28, that the collection and processing of data must be predetermined for a specific purpose. Generally, a purpose is the main goal or motivation of an activity or behaviour. In this context, this means that the purpose of the data processing must coincide with the purpose for which the data was originally collected. This predetermination shall last for the complete life-cycle of the data processed in the cloud. Such a purpose-boundary applies especially for particular sensitive data, such as health-related data. Nevertheless, there are some exceptions and also restrictions to this general regulation. So for example, under certain circumstances some latitude is given for a further data processing for historical, statistical or scientific purposes as far as it is not evidently incompatible with the original purpose of the data collection. Furthermore, under certain circumstances related to the European Convention's most fundamental guarantees of Human Rights and Freedoms, purposes of journalistic, literary or artistic expressions also qualify for some exclusion of the predetermination. Also, legal obligations of professional secrecy facilitate derogation from the predetermination [EU directive 95 46 EC recitals (28), (29), (33), (37)].

Transparency and information

A pre-condition to exercise any rights related to the data subject's self-determination is the transparency and information of the concerned individual. The transmission and remote processing of data in the cloud computing environment comes with a significant loss of control over the factual execution of the data processing. In this context, transparency shall be understood as the possibility for the data subject to learn which of his personal data will

be stored in the cloud, which processing occurs and who has access to this data. This information also includes the applicable law that concerns the individual medical case. Information means that the medical professional or any other entity, who is data controller, actively informs the data subject about events in relation to his personal data [so-called “Breach Notification", see Sections 13, 78, 79 + 80 DP Code]. This especially concerns a notice to the data subject in case of a data breach incident. Ideally, the means of transparency and information is supported by correlating functionalities in the cloud computing environment, in which the data will be stored and processed.

Informed consent

The consent of the individual as data subject is always mandatory [Section 76 paragraph 1, subsection a) DP Code]. Before the giving of this consent, the concerned individual must be informed about the purpose, the means and the extent of the data collection, processing and storing before giving consent. In the cloud computing context, this implies higher demands to patient information before a valid consent can be given. Also, it is necessary that the individual learns who will have access to this data. For the TClouds medical use case, it is also important to keep in mind that if consent is given, it must not only include the current health data but also explicitly past clinical events, if such a history shall be set up in the electronic patient file. The consent can be given verbally, but in this case, it must be explicitly documented by the medical professional [Section 81 paragraph 1 DP Code].

Exceptions can only be made in case of:

- **Emergency** [Section 82 paragraph 3 DP Code]: If the medical care or the health of the individual may be negatively affected, information and consent requirements may be complied without delay after the service has been delivered
- **Impossibility** [Section 82 paragraph 2, subsection a) DP Code]: If the data subject is physically impaired, legally incapable, unable to distinguish right and wrong and it is impossible to obtain the consent from a person or entity representing the patient.
- **Danger of an immediate serious and irreparable damage** [Section 82 paragraph 2, subsection b) DP Code]: This also concerns the physical health and integrity of a person

In all these cases, the consent of the patient must be obtained without delay after the service. If this is not possible, the consent can be given by the responsible data protection authority [Section 76 paragraph 1, subsection b + paragraph 3 DP Code] if

- the protection of a third person is necessary or
- the protection of the general public is necessary

Regarding minors, it must be considered that once majority sets in, the consent must be given by the individual anew.

Control functionalities

To enforce the protection of the data subject’s rights, corresponding and effective control functionalities should be implemented into the cloud system. Examples for such control functionalities are data upload, modification, blanking and deletion. Furthermore, access authorisation and withdrawal, the right to object the data processing in general or related to parts of it, and the possibility to display an exhaustive log and list of data uploads, accesses, alterations and entities that have access should be implemented. Some exemplary and more detailed explanation of these control functionalities will be addressed in the following text

under Section 3.4 (Organisation, usage and configuration). However, all of these control functionalities should be implemented as an integrated, all-embracing solution and in an understandable and in an easy to handle manner.

4.3.3 Technical conception and functionalities of the cloud system

This part addresses the general conception and mandatory functionalities of the cloud system. These are mostly technical implementations to support the patient's rights and the basic principles of data protection framework on EU and national level. As far as precise requirements in regard to the conception and the functionalities were made, so are these exemplary implementations.

Data separation model

The databases of cloud servers consist of data objects. Each of these objects can be assigned to a certain data subject, which is the patient, and to his or her medical case. The collection of all data, which are assigned to a medical case, constitute the electronic patient file. The single attributes of the patient data have some semantics, which can be divided into fine-grained, different data parts. Such data parts are:

- Demographics (e.g. gender, age, disabilities etc.)
- Contact information
- Insurance data and other administrative data
- Medical data
- Hospital and home care data

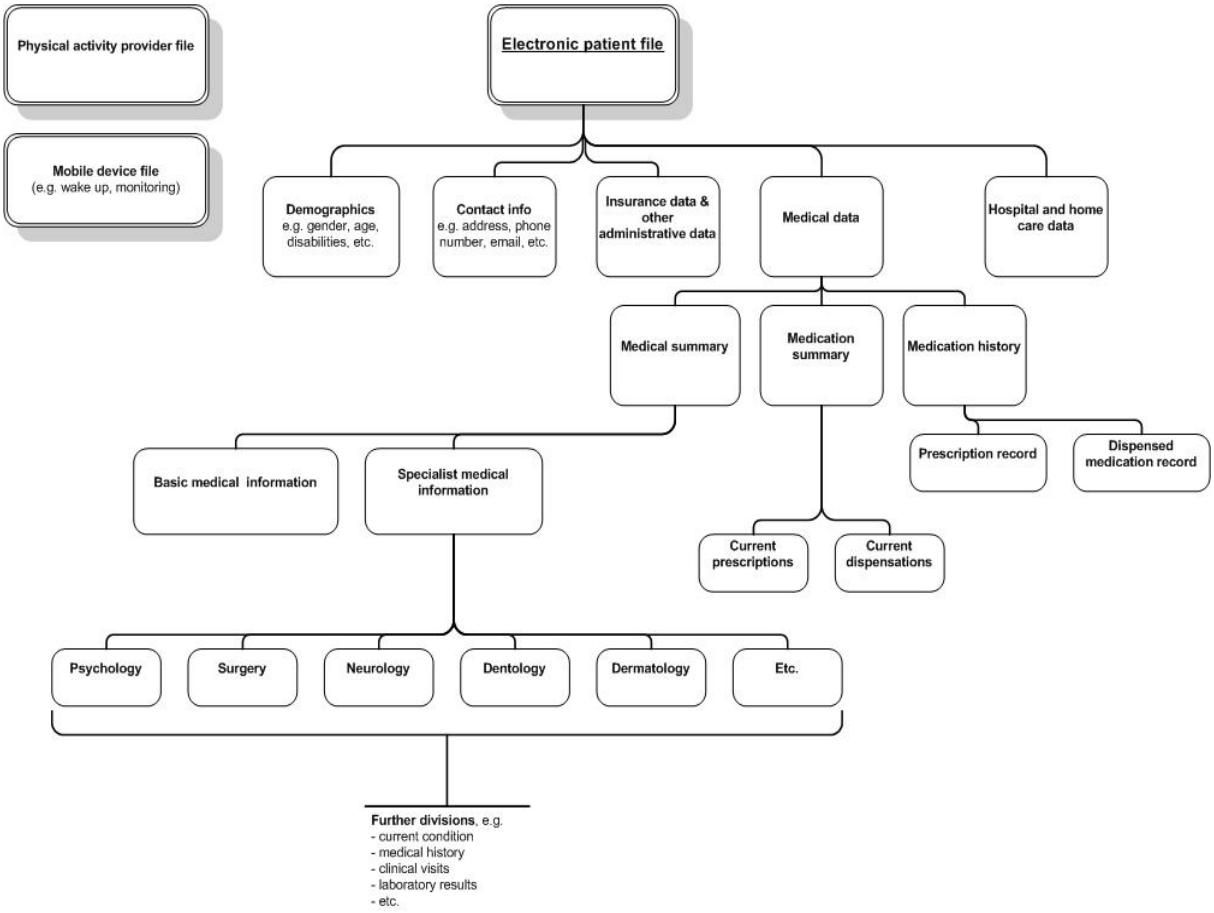

Figure 6: Exemplary data separation model

These superordinate divisions have several sub-divisions in the “Medical data” category. Such sub-divisions are necessary, since the TClouds scenario refers to situations where several medical professionals and pharmacies might be involved.

So, the exemplary sub-divisions would be:

- **Medical summary**
  - Basic medical information (typically, general practitioner information)
  - Specialist medical information (divided into discipline categories)

  Both parts above contain further sub-divisions, such as:
    - Current patient condition (allergies, current illness or disease, etc.)
    - Medical history
    - Clinical visits
    - Laboratory results
    - Etc.

- **Medication summary**
  - Current prescriptions
  - Current dispensations

- **Medication history**
  - Prescription record
  - Dispensation record

Additional data parts could provide extra files for externals, which are not bound by medical confidentiality or are not pharmacies which execute the dispensation of medicines based on prescriptions. Also, the data transmitted by mobile devices could be separated from the electronic patient file. Thus, examples for such additional data files would be:

- **Physical activity service provider file**
- **Mobile device file** (e.g. wake up device, monitoring devices)

Beyond these diversions, the data objects may contain meta data, which can reveal the status of the medical case, the responsible entity or medical professional, the creator of a data object or link to another data objects. Every data object must provide attached meta data which contains information about who created the data object and if it was verified by a medical professional. The data model as such must enable the creation of a clinical basic data set. Which data will be included into this basic data set will be the decision and the responsibility of the subscriber, e.g. the hospital. It must be discernable for any data object if the medical treatment is current or has ended. Information, if the accounting process in regard to the medical service is finished or if the case is closed, must be provided. Also it must be discernable if a data object is assigned to a blocked or archived medical case. Such tagging also requires corresponding access authorisation regulations.

If medical patient data will be exported into another part for purposes of statistical analytics and scientific research, it must be anonymised. In relation of the processing context, it must be possible to display data of the patient file with or without the identity data. A possibility to temporarily or permanently pseudonymise the patient data must be given. This especially applies for cases, in which medical data will be used for training and statistical, scientific or quality management purposes.

Authorisation concepts

The authorisation concept and access controls must support this modular approach of data parts. For example, the physical activity service provider shall not access any medical data. The general practitioner shall not access psychiatrist's files and the other way around. In general, no information should be exchanged without the patient's consent or legal obligation. Any collection, processing and storing of the data must be bound by the original purpose. In the cloud computing context, this also means that access to medical data by actors that are not bound by professional secrecy, e.g. administrators and other cloud and application provider staff, should be prevented. Furthermore, the access authorisation of entities i.e. persons shall only be possible as long as it is absolutely indispensable for the duration of the medical treatment.

Data parts must either be physically or logically separated. Since several parties will be using data from the electronic patient file, also separate user accounts must be set up. It must be possible to define processing function, access authorisation and configuration setting for each user account. Such a definition may be enabled by sticky policies.

Furthermore, it must be possible to set up some labels for parts of the electronic patient file or the whole file. Such labels are supplemental to the meta data and could include several status signals. Exemplary for such status signals could be the following information:

- If the patient has objected against the inclusion of pre-treatment data into the file
- If an information block regarding this data exists

- If the data or the whole patients file requires a higher protection level. This for example concerns data of patients who are employees of the hospital in which they are treated medically. It must be possible to connect such labelling to specific authorisation regulations.

All of these labels must be changeable or deletable at any time. Furthermore it should not be possible to use this labelling to actively and systematically search for specific patient. If such a function will be necessary though, it should be connected to a corresponding authorisation regulation.

In general, the access authorisation in respect to the parts of the electronic patient file must be determined only individually and not generally by categorisation. This means that a patient and his/her data must be assigned to a specific and functions-focused organisation entity. A distinction between treatment and co-treatment is necessary. Flexible multiple allocations of patients to specific doctors or nursing and medical organisation entities must be possible. All-embracing access authorisation should be avoided. Instead, authorisation should always be associated to the specific treatment. The allocations should always be oriented towards the specific function, not a person. Helpful can be the classification and division of involved entities and persons into specific user categories and subsidiary roles. Such categories could be (exemplary):

- Medical professionals
- Nursing staff
- Administration staff
- Training staff
- External staff
- Technical administration staff

Some examples for subsidiary roles are:

- Administrative hospital admission
- Medical hospital admission
- Secretary staff
- Nursing staff (leading nurse)
- Stand-by emergency staff
- Attending physician
- Treating doctor
- Research staff
- QS management
- Controlling
- Internal data protection officer
- Application administration
- Authorisation administration

These lists are by no means conclusive. Instead, it must be possible to add other categories and roles to depict the case-specific authorisations of each involved entity. It also must be possible to define organisation entities in a flexible and overlapping manner if necessary. The implementation of such an authorisation concept must be documented, so the necessity of each the access authorisation and its extent are comprehensible. The administration of

this access authorisation must enable the terminated or permanent de-activation withdrawal of access rights.

Beyond this documentation and authorisation concept, a functionality to display the assigned authorisation and access rights of each user must be implemented.

Generally, it shall not be possible that a user can claim unauthorised access rights beyond the rights already to her/him. Nevertheless, there may be individual situations, where it may be necessary to neutralise access restrictions or extend access authorisation. This especially concerns emergency accesses. In these cases, such overrun of an access restriction must follow a well defined and documented procedure. An example for such a procedure would be the “Break Glass” procedure [Yale University Introduction to HIPAA]. One may think other procedures as long as they follow standardised incident management processes, like the ITIL incident management standards. However, mandatory are several steps to process such an emergency access:

- The application must show a warning that the person claiming access does not have regular authorisation and is about to overrun an access restriction
- A warning must be shown that this procedure will be logged and reviewed
- A written explanation for the reason and necessity of this access claim must be given
- The person claiming the access must be identifiable by username
- If possible, a second person should affirm the access (four eyes principle)

Only then, if all of the above mentioned steps have been processed, the access shall be granted. The procedure must be well documented by emergency event logs, so the event can be reviewed. The mandatory content of such logs is described below under pt. 3.3.4 (Logging functionalities). The application interface on user side should provide optical differences of standard processing contexts in emergency cases.

System functionalities

This Section describes some technical functionality, which could be implemented to support the protection of patient’s personal data. These functionalities are not conclusive and can be complemented by other organisational and security features. Some of them are already standards for security in IT systems, but could also apply for cloud computing cases.

Data stored and processed in the cloud should be encrypted as far as possible to prevent the disclosure of personal data to cloud service providers, data centre admins, etc. Also this is a tool to protect data in cases of unlawful data breaches from outsider attackers. In an electronic patient file, the duplication of data should be avoided. If it is necessary that different entities must get access to specific data, but only have access authorisation to different data parts, the concerned data objects should be tagged with effective access labels. If a redundant storing of data is unavoidable to warrant the system functionality, any blocking, blanking or deletion must be equally considered for each data part.

A single-sign-on-service should be integrateable for the patient file. The necessary access credentials for sign-on must provide an adequate security level. A migration of data, for example in case of changing medical professionals taking care of the treatment, must be possible. If data is transmitted between parties, the transmission must be encrypted. The decryption keys must be solely under control of the responsible medical professional or hospital. Storage media, on which the data will be included, must be encrypted. This applies especially for mobile storage devices. The decryption keys shall not be accessible for externals.

The electronic patient file should enable an overview of all data stored in respect to a specific patient. It must be possible to block or archive time- or event-controlled files or parts of it to withdraw them from operational access. In cases of archiving closed cases, a restoring into the active status is not necessary since a read-only access should usually suffice. For any exceptions, the emergency access procedure as described above under pt. 3.3.2 could be used to achieve further access rights.

Closed medical cases should be deleted after a determined archiving time period. It must be possible to verify the effective deletion of the data. One such verification tool can be logs provided by the cloud service vendors to prove to the healthcare provider that the data really has been deleted. The duration time of data archiving must be conform to the specific legal requirements on EU as well as on national level. Blocking and deletion functionalities shall only be performed by specifically authorised entities.

If the replication and transmission of databases into a test system is necessary to search for performance and system failures, the databases should be pseudonymised. The electronic patient file should provide a pseudonymisation service, which is able to generate purpose-related temporary or permanent pseudonyms. Also, the provision of an anonymisation function should be possible [See Chase, Lauter, *An Anonymous Health Care System*, for an exemplary approach to an anonymisation credential procedure]. For the evaluation of system and functionalities performance, effective common criteria, such as the *Common criteria for Information Technology Security Evaluation*, should be determined beforehand.

Logging functionalities

The system must provide some logging, which gives information about who (i.e. which organisational entity) at which time has accessed the personal data and how it was processed. Besides the collection and initial storing of the data, modification, blocking, blanking, deletion and read-only-access of data must be documented.

Generally, the type and extent of the logging must be tailored to the specific nature of the processing and the protection level of the data. In this context, the logging should also be aimed at data minimisation. A logging concept should be provided to determine the nature and scope of the logging, the means and duration for storing and evaluation and the protection mechanisms in respect to the concerned persons. The evaluation of the logs should be possible any time and related to a certain incident as well as on a random basis. The determination of logging concept and evaluation should be done with consultation of the internal data protection officer of the hospital.

The logging should take place on the level of the application functions to enable a traceability and comprehensibility related to the professional and operational functions of the involved parties. A logging on the level of the database or a solely technical logging would not be as effective.

The logging should at least contain the following information:

- Time of a data access and end of transaction (login/logout)
- Account name of the accessing user
- Number or other identifier of the used working station
- Accessed transaction (display/inquiry function, screen display mask, report on data)
- Which medical case and patient is concerned

Emergency event logs shall also contain additional information about:

- Explanation for reason of access and the assumption of an emergency case

Also, this information must be logged even if the access procedure was cancelled.

In cases of search function usage, the log shall contain the following information instead of the above mentioned general information in standard access situations:

- Used search or inquiry criteria (e.g. patient number, case number, name, birth date, address, diagnosis etc.)
- Result of the search/inquiry (e.g. number of results, case numbers, etc.)
- Eventual following actions (e.g. choice of specific data from the results list, screen display mask, print, data export)

It must be differentiated between the professional use of the procedure by medical or administrative staff. Logs shall not contain any medical data. The access to logging data shall be restricted. Cryptographic methods to protect the logs should be implemented. Persons which actions are documented by the logs should not have access to these. Access to logging data shall only be granted to persons for whom the evaluation of those is their explicit duty.

Regarding the deletion of data, the log shall only contain for single data objects the time of deletion and the ordering user, for data sets additionally the case number or similar identification characteristics. For the evaluation of logging data, the system must provide the possibility to review the information in regard to:

- Processing context
- Eventually explanation obligation for a transaction (esp. emergency cases)
- User account name
- Working station
- Functions/transactions
- Search criteria
- Patient number/case number
- Time frame

An evaluation about which entity had which specific user rights must be possible. Sufficient evaluation functionalities must be provided to enable a reviewing of the logs in regard to any oddities, e.g. related to frequency of access, unusual search criteria etc. The structure and format of the logging data form must also enable a flexible evaluation if necessary. Therefore, the logging data should be in a standardised form for analytics tools or database functions (e.g. CSV-format).

4.3.4 Organisation, usage and configuration

The user of a cloud environment carries out transactions to enter data into the cloud, modify, present, transmit, import, export and delete it. Each transaction must be assigned to a specific processing context. In this context, each processing function must be aligned to the access rights of the particular person.

It must be possible to encrypt single data parts in some cases, e.g. diagnosis, laboratory results, hospital and home care data. The decryption keys shall only be held by the entity which conducted the encryption in the first place.

The application to access the electronic patient file should support a fast change of users, i.e. user accounts. The single-sign-on service should be based on a two-factor authorisation. The implementation of an automated work station block would be useful. Inside the hospital, the application should support the procedure of saving and resumption of the work at another work station. Transactions for access rights delegation or documentation of medical instructions by the doctor should be possible.

The application interface on the screen should look clearly arranged. This especially applies to the display of administration interface for rights and user roles. The consequences of rights allocations should be clearly viewable. An easy process of backup and restoring access rights configuration as a whole or parts of it should be possible. Any changes should come into effect directly without delay.

The design of the application should be compliant with the requirements in regard to ergonomics standards like ISO 9241 (Ergonomic requirements for office work with visual display terminals) and ISO 14915 (Software ergonomics for multimedia user interfaces). For the data protection relevant functions, help and explanation functions should be implemented.

4.4 Open questions

This Section addresses some open questions that need to be answered to provide more precise input regarding the TClouds medical use cases. These questions concern legal issues as well as detail aspects in regard to this specific medical scenario.

4.4.1 Completeness of the medical information

A problem is that a medical professional has no legally pre-assigned right regarding the completeness of the patient's data. This is relevant because in cases of patient monitoring with mobile devices or grids, a constantly 24/7 monitoring is a massive intrusion in respect to the patient's privacy. Though the patient will have to express explicit consent to this monitoring process, there may be situations where the constant monitoring via the device is not wished for. So, to enable the patient's self-determination in regard to his/her personal data, it must be possible to temporarily switch off or divest the monitoring device. An exemplary situation would be that a patient possibly does not want his body functions to be monitored during sexual activities. As a consequence, the possibility to intervene must exist. In this context, medical professionals fear the burden of legal accountability in case of a false or not precise enough diagnosis on the basis of not complete or false data. Nevertheless, the accountability of the medical professional will be limited, preconditioned he asked the patient for completeness and elucidated the consequences of a diagnosis based only partial information. Thus, the possibility to enter incomplete data into the cloud via the mobile device must be given. However, in which ways this can be realised, must be discussed further during the project.

4.4.2 Information duties concerning medication interdependencies

The open question is how the medication system works in Italy. Does the medical professional instruct the pharmacies to give a specific medicine via the prescription - or does he only issue the necessary active pharmaceutical ingredients, so the pharmacy will mix together the medicine based on this information? This is relevant for the specific information duties regarding the interdependencies of several medicines.

4.4.3 Documentation duties of medical professionals

Another open question concerns the documentation duties of medical professionals in Italy. The electronic health file must provide a possibility for the medical professional to ensure the necessary documentation to comply with the applicable law. Eventually existing differences in health care law as well as e.g. tax law in regard to the legally required duration of documentation storage in comparison to other EU countries, e.g. Germany must be researched (in Germany, tax law requires a storage over ten years).

4.4.4 Accounting for health care insurances

The further research work in respect of the legal requirements for the TClouds medical use case must also cover the question how the health care accounting system of the insurances in Italy exactly works. This encompasses clarifying which information and to which extent the health insurances need for accounting medical services. In the light of the data minimisation objective, an accounting of the single health care activities outside the cloud computing environment would be preferable.

4.4.5 Encryption of certain data types

Data stored and processed in a cloud computing environment should be encrypted. However, especially the processing of certain data types, such as x-ray pictures and the automated evaluation of medical examinations, can prove difficult. In the current state-of-the-art, fully homomorphic encryption techniques still need improvement in execution and performance. Therefore, it would be desirable to support further research of problematic issues about fully homomorphic encryption in regard to medical data to find suitable solutions.

4.4.6 ePrescription in Italy

So far, the data protection law in Italy requires a specific procedure for prescriptions. Section 87 of the DP Code regulates the drugs paid for by the National Health Service. There it states that a certain paper form must be used. An open question is if there are exceptions provided in some special law sections and which is the procedure for drug prescriptions, which do not fall under the regulation of Section 87 DP Code.

4.4.7 Continuity management

In situations where a medical professional does not continue his/her practical exercise, the issue of missing authorisation of another doctor taking over the office and practice must be solved. The problem is that the patient's consent to collect, store and process the medical data is not automatically transferred to the new doctor. Therefore, measures to prevent unauthorised access of newly involved parties must be developed. In Germany, a so-called "Two-cabinet-solution" was developed, where printed/written patient files are stored separated in different cabinets. Once a patient visits the practice, the new doctor can then ask consent of the patient to access his/her file from the cabinet of the former doctor to continue the medical treatment. In a cloud computing system, one may think to an automated notification of all patients in case of an abandoned practical exercise. However, a procedure for this automated notification must be developed. This applies especially for cases when the former doctor cannot set up this notification himself anymore, e.g. due to death or severe

injury through an accident. A possible solution would be to involve another person or entity with notification authorisation in such cases.

4.4.8 Hospital organisation in Italy

During the further progress of the legal analysis for this medical use case, it is necessary to explore the specifics of internal organisation in Italian hospitals. This is especially important in respect to the further refinement of user categorisations and subsidiary roles. So, all partners involved in WP 3.1 should discuss the optimal realisation of the user categories and roles in regard to this aspect.

4.4.9 Legal access of data by unconcerned third parties

Aside from the concerned parties with direct relation to the cloud infrastructure, in which the data is stored and processed, some external parties may desire access to this data. Depending on the nature of the external request, it is possible that some legally stipulated access rights may interfere with the basic principles of data protection and privacy of the patient. The national legislation of the EU member countries provides different regulations of preconditions and extent in regard to such external accesses. Some exemplary third parties that may desire insight to the data inside the cloud are as follows [Marnau, Schlehahn, D1.2.2, Annex A]:

1. Supervisory authorities;
2. Investigation authorities;
3. Policy makers;
4. Auditors;
5. Certification bodies;
6. Licensure authorities;
7. Other official and business entities;
8. Attackers,

Subsequently, the consequences of such legal accesses with influence on the data protection issues need to be discussed. Solutions, how such an access must be established and with procedures must be followed during the process, must be found.

4.5 Conclusion

To refine this preliminary overview regarding possible issues and solutions for the TClouds medical use case, further research work must be done. The legal requirements and exemplary solutions presented in this preliminary overview will be discussed with the other partners involved in WP 3.1 to work out problem fields, explore open questions and present tangible results. Parallel to this overview, these results will be presented in the context of the in-depth analysis of the general legal data protection requirements on European level within the deliverable D1.2.2 [*Cloud Computing: Legal Analysis*] and report R1.2.2.2 [*Legal analysis and requirements "Patient monitoring"*]. These documents will go more into detail regarding the general and use-case specific legal requirements to enable a data protection compliant realization of the specific cloud-related scenarios. They also will take into account the current and ongoing developments in the context of the revision of the European data protection framework.

Chapter 5

Preliminary Architecture of the home healthcare application

Chapter Authors:

Mina Deng (PHI), Marco Nalin, Ilaria Baroni (HSR)

5.1 Introduction

The goal of this Section is to provide to the other work-packages, especially those of A1 and A2, an overview of the reference architecture for the eHealth home monitoring scenario of the TClouds project.

In the Section 5.2 will describe detail the scenario (described in the Section 2) and services through the definition of the use cases, illustrating also the use cases dependencies and involved actors. Section 5.3 is the heart of this section, and it describes the reference architecture derived from the above mentioned use cases and scenario. A practical instantiation of the reference architecture, which most likely will be implemented in the first year prototype, will be illustrated too.

5.2 Use Case Specification

5.2.1 Use Case Model

This section will explain in detail the scenario and the services described in the previous Sections. The Use Cases methodology will be described to highlight the actors involved in the system and their relationships and experience in using the final system, and they will be used for the definition of the reference architecture for the home healthcare scenario.

5.2.2 Actors

In this section we identify different actors important for the use case model. Actors are parties outside the system that interact with the system; an actor can be a class of users, a role users can play, or another system. Note that, depending on the use case, some parties or actors may not be involved.

- Patient
- GP: The patient's general practitioner.
- Hospital: the hospital is considered as an actor for legal issues, e.g., when auditing.
- Hospital psychiatrist: this is the psychiatrist following the treatment of the patient, employed by the hospital.

- HWSP: Health and Wellness Service Provider. This could be the one who provides the devices (e.g. like Philips) or the service (e.g. the hospital when considering the sleep management, or the gym when monitoring physical activity, etc.).
- Pharmacy: Any pharmacy providing the drugs needed by the patient for following correctly her treatment
- Family: patient's family members or close friends which are also users of the system, and which are explicitly authorized to see the personal data by the patient.
- Public authority: This is a general category to include public authorities that may need to access the data in specific conditions, like Regional or National Healthcare Systems for reimbursement, or law court for auditing during possible legal actions, etc.
- Cloud SP: Cloud Service Provider. It is implicitly involved in all the use cases, as it is providing the platform on which the applications are running, however it will be mentioned only when the Provider itself is protagonist (or co-protagonist) of the use case action (e.g., in case of auditing requests from its users, etc.)
- Carrier: It is the one bringing the drug from the pharmacy to the patient's house. It may be the patient itself, a patient's relative/friend, or a Drug Delivery Service Provider.
- Bank: A payer for the drug bought by the patient. It may be the patient's bank, her insurance company bank (in case the drug is to be reimbursed by the insurance company), regional/national healthcare systems accounts, etc.

5.2.3 Use Case Overview

In the following, the identified use cases are discussed grouped according to the following functional packages. Each package represents a service.

- Patient management portal
- Personal diary
- Self assessment questionnaire
- Activity monitoring (physical activity, sleep)
- Drug therapy management (prescription and anonymous deliver)
- Epidemiological studies
- Auditability

5.2.4 Patient management portal

This section includes all the use cases related to the use of the online portal that the project will realize for the management of the patient. Some services offered by this portal will be available only to the Hospital psychiatrist, other only to the patient (or her family/GP), other to both.

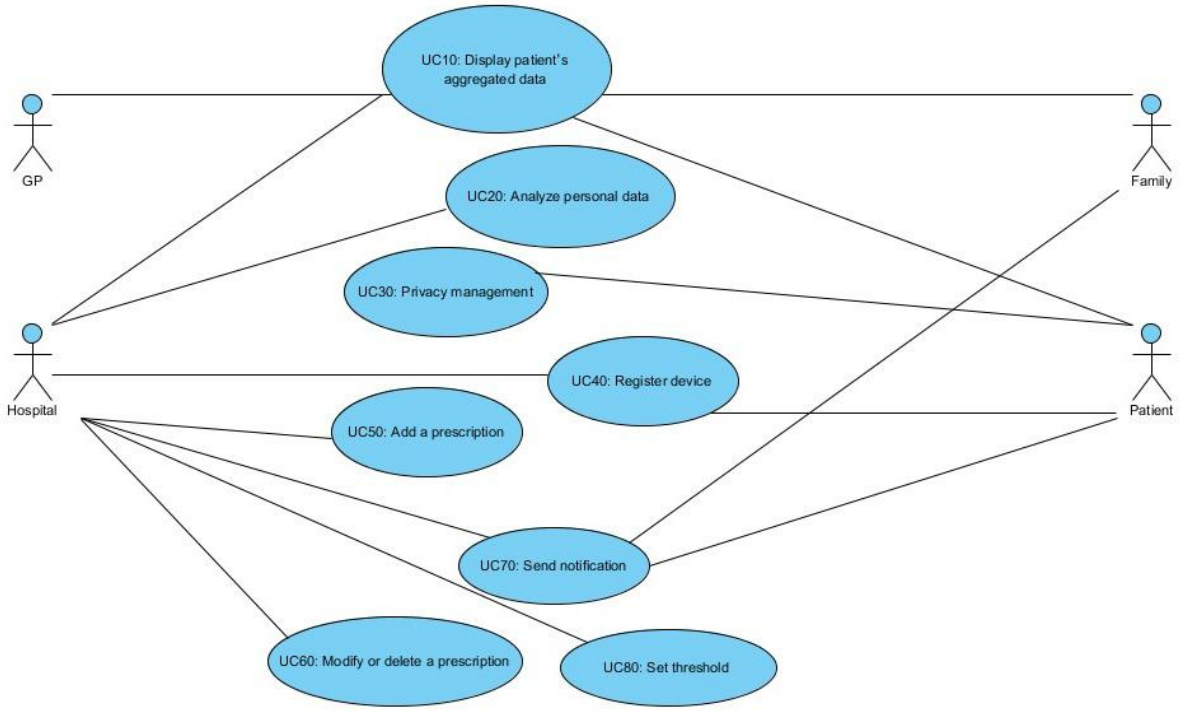

Figure 7 Patient management portal actors

USE CASE UNIQUE ID	/UC 10/ (Display patient's aggregated data)
DESCRIPTION	The patient, while accessing the personal portal, is able to visualize in a graphical format correlations between her personal parameters, both monitored through devices and collected with online questionnaires, diaries, etc. The family and the healthcare professionals that are authorize will be able to do the same
ACTORS	• Patient • Hospital psychiatrist • Family • GP
PRECONDITIONS	Some data was collected from devices or online portal
POSTCONDITIONS	The patient (or a user authorized by the patient) visualized the data
NORMAL FLOW	1. The patient access the page on the portal where she can visualize personal historical data 2. The patient should specify a time frame be analyzed (by default the last month) 3. All the personal data collected in the specified time frame will be displayed in a graphical form 4. The patient is able to filter the data displayed, selecting only specific parameters (e.g., depression scale values and sleep values, etc.)
ALTERNATIVE FLOW (DATA ACCESSED BY A THIRD USER)	1. Same as before but with another user accessing patients' data (e.g., Hospital psychiatrist, family, or GP) 2. In this case the precondition is that the patient authorized this user in advance

USE CASE UNIQUE ID	/UC 20/ (Analyze personal data)
DESCRIPTION	The application will be able to correlate one or more indicators (e.g., physical activity, sleep quality, depression development, etc.). This correlation can be automatically performed or triggered by a user. Data are checked against the thresholds set by the psychiatrist
ACTORS	Hospital psychiatrist
PRECONDITIONS	• Some data was collected from devices or online portal. • UC 80
POSTCONDITIONS	If a bad situation is identified, the proper countermeasure will be activated. In particular: • If the patient forgot to take a drug -UC 210 • If the patient forgot to do physical activity -UC 210 • If a dangerous situation is identified -UC 70 • If a situation that requires additional assessment is identified -UC 140
NORMAL FLOW	1. New data is collected from patient's devices or inserted by the patient herself through the personal portal 2. The application analyzes the newly received data and correlates it with data already present in the database 3. The application checks the possible problems, as specified by the thresholds set by the psychiatrist
ALTERNATIVE FLOW (MANUAL CORRELATION)	1. The psychiatrist accesses the patient's personal historical data 2. The psychiatrist selects a time frame to be analyzed (by default the last month) 3. The correlation of the selected data is displayed in a graphical form 4. The psychiatrist is presented with a set of options that he can activate as reaction of the data visualized (see post-conditions)

USE CASE UNIQUE ID	/UC 30/ (Privacy management)
DESCRIPTION	The patient will be able to configure her privacy settings for deciding who can access her data
ACTORS	Patient
PRECONDITIONS	The patient decides to specify or change her privacy settings
POSTCONDITIONS	A user is added/removed to the authorized list
NORMAL FLOW	1. The patient accesses her privacy settings page on the portal 2. The patient is able to create lists of users with the

USE CASE UNIQUE ID	/UC 40/ (Register device)
DESCRIPTION	The actor will be able to add devices on her profile. Once registered, the devices will be authorized to upload data to the patient's profile (but not to download any data).
ACTORS	• Patient • Hospital Psychiatrist
PRECONDITIONS	The patient has a new devices to add
POSTCONDITIONS	• A new device is registered to the patient's profile • Newly registered device is able to send data directly to the patient's profile
NORMAL FLOW	1. The patient accesses her personal devices page on the patient portal 2. The patient registers the device, by inserting the device serial number and type in the web page
ALTERNATIVE FLOW (REGISTRATION DONE BY CONNECTING THE DEVICE)	1. The patient connects her new device to the PC or Smart Phone (e.g., through USB connection, or Bluetooth connection, etc.) 2. The client on the PC or Smart Phone asks for credential to upload data to patient's portal 3. The patient inserts the credentials
ALTERNATIVE FLOW (REGISTRATION DONE BY THE HOSPITAL PSYCHIATRIST)	1. The hospital psychiatrist accesses the patient's personal devices page 2. The hospital psychiatrist registers the device, by inserting the device serial number and type in the web page

USE CASE UNIQUE ID	/UC 50/ (Add a prescription)
DESCRIPTION	The hospital psychiatrist will be able to register a new prescription on the patient's profile
ACTORS	Hospital psychiatrist
PRECONDITIONS	The psychiatrist visits the patient or monitors patient's results and data
POSTCONDITIONS	• A prescription is added to the patient's profile • The patient is notified that a new prescription was added UC 70
NORMAL FLOW	1. The hospital psychiatrist accesses the patient's management page 2. The hospital psychiatrist compiles the prescription form. Prescription may be relative to: • Drug intake; • Physical activity recommendations; • Sleep habits; 3. The hospital psychiatrist saves the prescription in the patient's profile

USE CASE UNIQUE ID	/UC 60/ (Modify or delete a prescription)
DESCRIPTION	The hospital psychiatrist will be able to modify or delete any prescription
ACTORS	Hospital psychiatrist
PRECONDITIONS	A prescription must already exist -UC 50
POSTCONDITIONS	• The prescription is modified or deleted • The patient is notified that a prescription was modified or deleted -UC 70
NORMAL FLOW	1. The hospital psychiatrist accesses the patient's management page 2. The hospital psychiatrist selects the prescription that should be modified or deleted. Prescription may be relative to: • Drug intake; • Physical activity recommendations; • Sleep habits; 3. The hospital psychiatrist modifies (or deletes) the prescription 4. The hospital psychiatrist saves the prescription in the patient's profile

USE CASE UNIQUE ID	/UC 70/ (Send notification)
DESCRIPTION	An actor receives a notification from the system
ACTORS	• Hospital psychiatrist • Patient • Family
PRECONDITIONS	• An event that should be notified happened • Some thresholds for the patient were specified -UC 80
POSTCONDITIONS	The actor (patient, psychiatrist, family member...) received the notification
NORMAL FLOW	1. An actor (e.g., psychiatrist, patient) execute an action that must be notified to other actors (e.g., prescribing a new therapy, filling a questionnaire, etc.) 2. An instant message is sent to the recipient actors
ALTERNATIVE FLOW (THE NOTIFICATION IS SENT AUTOMATICALLY BY THE SYSTEM)	1. The system, after processing stored data, identifies that thresholds specified by the psychiatrist are reached 2. An instant message is sent to the recipient actors specified by the psychiatrist

USE CASE UNIQUE ID	/UC 80/ (Set threshold)
DESCRIPTION	The hospital psychiatrist fixes some thresholds to monitor the patient's data
ACTORS	Hospital psychiatrist
PRECONDITIONS	The psychiatrist wants to configure the automatic monitoring of patient's data
POSTCONDITIONS	Some thresholds for patient's data monitoring are set
NORMAL FLOW	1. The hospital psychiatrist accesses the patient's management page 2. The psychiatrist selects data that should be monitored 3. The psychiatrist set the threshold values 4. The psychiatrist decides the automatic action to be performed when the thresholds are reached. An action can be for example sending a notification to the patient, or asking the patient to fill a self-assessment questionnaire, etc. -UC 70, UC 140

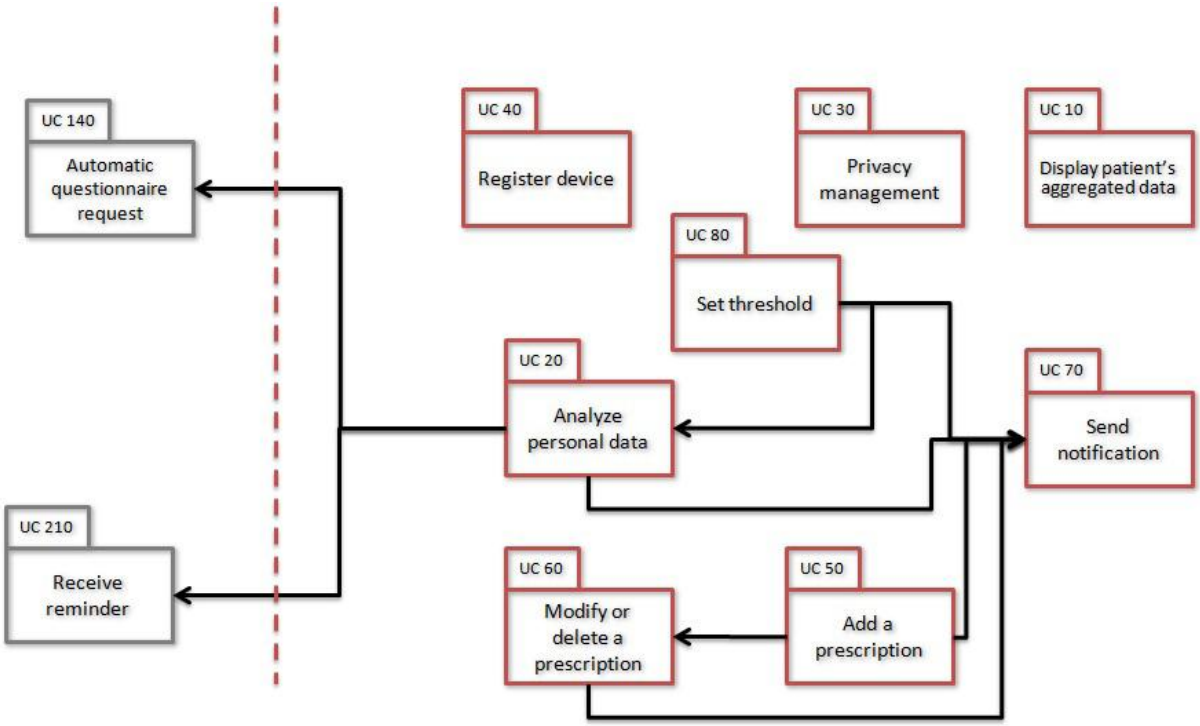

Figure 8 Patient management portal dependencies

5.2.5 Personal diary

This section includes the use cases related to the patient's personal diary. Here the patient can keep track of her daily mood and main events, especially those important for the Social Rhythm Therapy.

USE CASE UNIQUE ID	/UC 90/ (Insert diary record)
DESCRIPTION	The patient inserts data, compiling the mood diary or adding other information
ACTORS	Patient
PRECONDITIONS	The patient wants to add a new entry in her diary
POSTCONDITIONS	A new diary record is added
NORMAL FLOW	1. The patient accesses her personal diary on application web site 2. The patient chooses what kind of data should be added (daily events, daily mood, daily feelings, etc...) 3. The patient enters data and adds some comments if needed

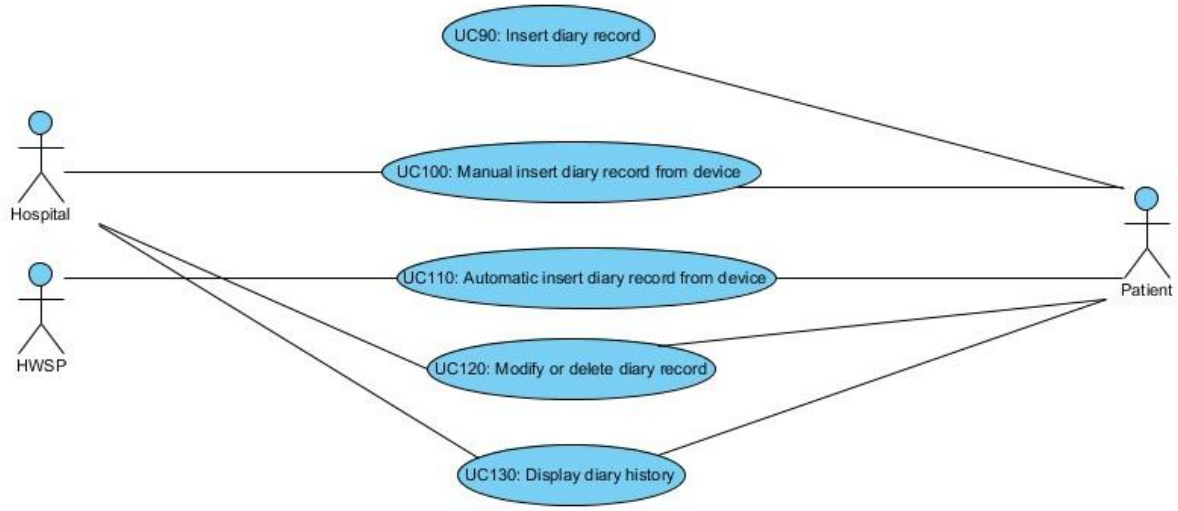

Figure 9 Personal diary actors

USE CASE UNIQUE ID	/UC 100/ (Manual insert diary record from device)
DESCRIPTION	The patient wears a monitoring device, and periodically she sends the information collected to her diary
ACTORS	• Patient • Hospital psychiatrist
PRECONDITIONS	The device is registered to the patient's profile -UC 40
POSTCONDITIONS	A new diary record is added
NORMAL FLOW	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily or weekly), the patient connects her device to the PC or Mobile Phone and uploads the data to her personal portal
ALTERNATIVE FLOW (DATA ENTRY FROM PSYCHIATRIST)	1. The patient goes to her periodic visit at the Hospital, carrying her personal device 2. The psychiatrist connects the patient's device to his PC and uploads the data to the patient's personal portal

USE CASE UNIQUE ID	/UC 110/ (Automatic insert diary record from device)
DESCRIPTION	The patient wears a monitoring device that periodically sends information to the patient diary
ACTORS	• Patient • HWSP
PRECONDITIONS	The device is registered to the patient's profile -UC 40
POSTCONDITIONS	A new diary record is added
NORMAL FLOW	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily), the device sends the measured data to the patient's personal portal
ALTERNATIVE FLOW (DATA PASSES THROUGH HWSP)	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily) the device sends the data to the HWSP 3. The HWSP sends the data to the patient's personal diary

USE CASE UNIQUE ID	/UC 120/ (Modify or delete diary record)
DESCRIPTION	The patient (or her doctor) accesses her diary and changes (or delete) previously inserted data
ACTORS	• Patient • Hospital psychiatrist
PRECONDITIONS	Data was previously inserted somehow in the patient's diary -UC 90, UC 100, UC 110
POSTCONDITIONS	A diary record is modified or deleted
NORMAL FLOW	1. The patient accesses her personal diary page on the application website 2. The patient selects the data that she wants to modify (or delete) 3. The patient inserts the new value for the selected data (or confirms the delete)
ALTERNATIVE FLOW (DATA IS MODIFIED OR DELETED BY THE PSYCHIATRIST)	1. The patient goes to the Hospital for her periodic visit with the psychiatrist 2. During the visit, the psychiatrist accesses the patient's personal diary page on the application website 3. The psychiatrist selects the data that should be modified (or deleted) 4. The psychiatrist inserts the new value for the selected data (or confirms the delete)

USE CASE UNIQUE ID	/UC 130/ (Display diary history)
DESCRIPTION	The patient (or the psychiatrist) accesses her personal historical data on her personal diary website
ACTORS	• Patient • Hospital psychiatrist
PRECONDITIONS	Data was previously inserted somehow in the patient's diary -UC 90, UC 100, UC 110
POSTCONDITIONS	Historical data are visualized
NORMAL FLOW	1. The patient accesses her personal diary page on the application website 2. The patient selects the type of data to be shown 3. The patient selects the time frame to be displayed 4. The system displays the requested data
ALTERNATIVE FLOW (THE PSYCHIATRIST ACCESSES THE PATIENT'S DATA)	1. The psychiatrist accesses the patient's personal diary page on the application website 2. The psychiatrist selects the type of data to be shown 3. The psychiatrist selects the time frame to be displayed 4. The system displays the requested data

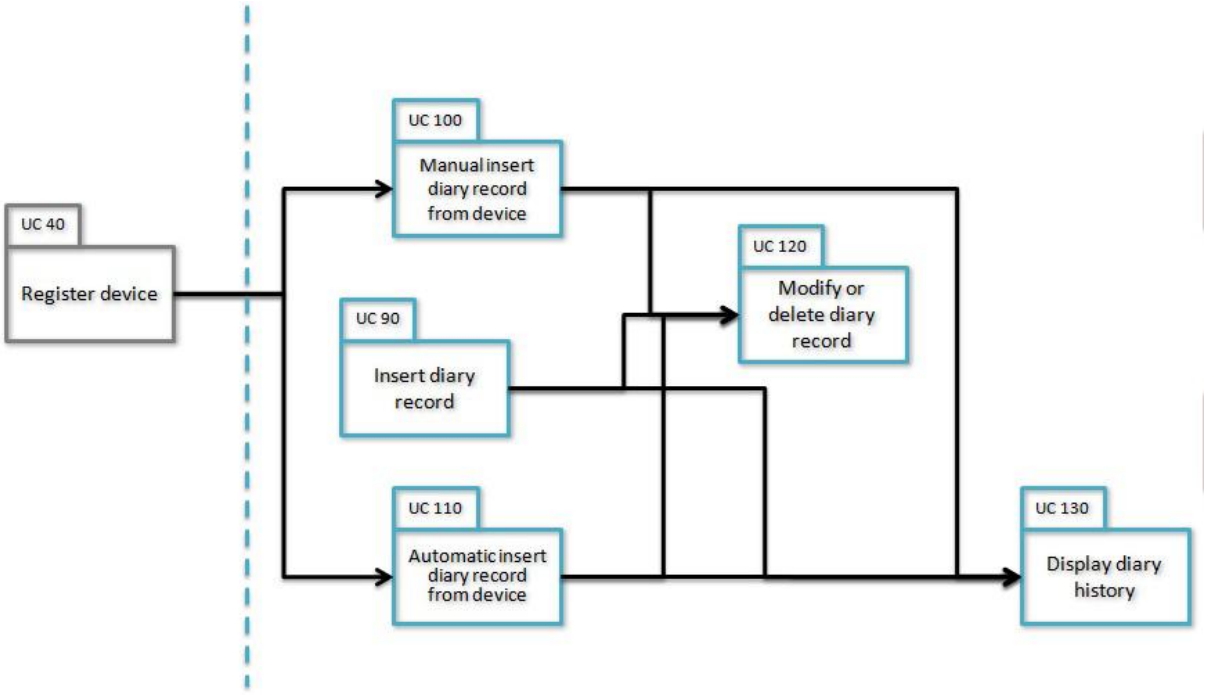

Figure 10 Personal diary dependencies

5.2.6 Self assessment questionnaire

This section includes the use cases related to the self assessment questionnaires and depression evaluation scales available to the patient. Some of these questionnaires can be requested by the psychiatrist or by the system.

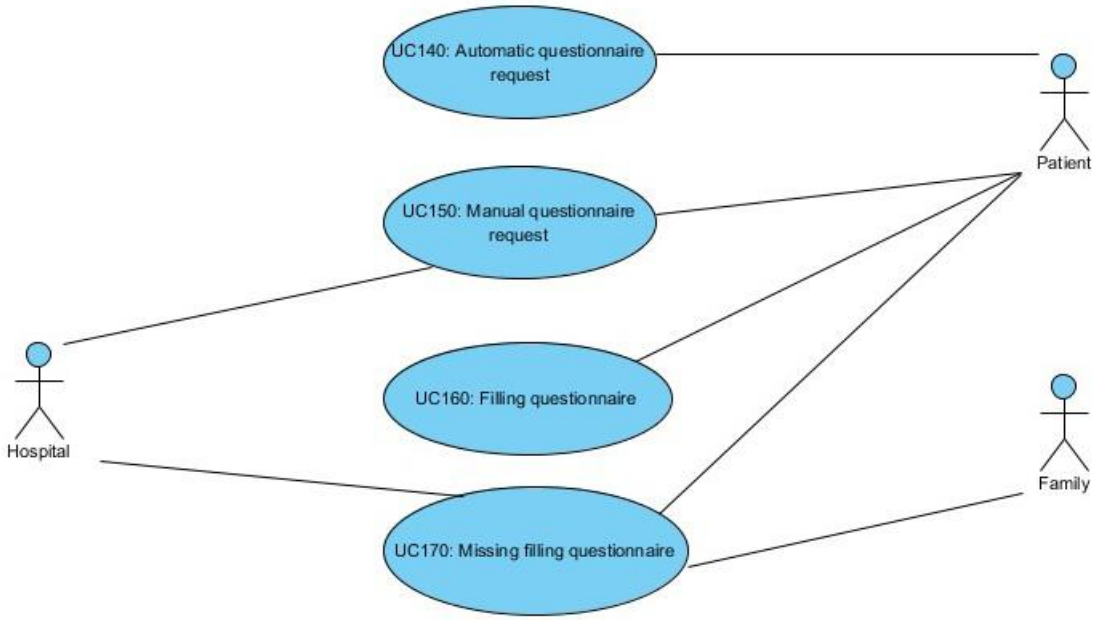

Figure 11 Self assessment questionnaire actors

USE CASE UNIQUE ID	/UC 140/ (Automatic questionnaire request)
DESCRIPTION	Based on some newly inserted data (or analyzed trends), the system requires the patient to fill some self-assessment questionnaires
ACTORS	Patient
PRECONDITIONS	• The psychiatrist specified some thresholds for some values that, if reached, should generate the request for the patient to fill the questionnaire -UC 80 • Some values inserted in the patient's diary reached the specified thresholds -UC 90, UC 100, UC 110
POSTCONDITIONS	The patient receives a questionnaire request -UC 70
NORMAL FLOW	1. Some new data are received (or analyzed) in the patient's diary 2. The system identifies a threshold overcoming, that requires as a counteraction that the patient fills a questionnaire 3. The system sends a request to the patient to fill the questionnaire

USE CASE UNIQUE ID	/UC 150/ (Manual questionnaire request)
DESCRIPTION	The psychiatrist requires the patient to fill some self-assessment questionnaires
ACTORS	• Hospital psychiatrist • Patient
PRECONDITIONS	Some values were previously inserted in the patient's diary -UC 90, UC 100, UC 110
POSTCONDITIONS	The patient receives a questionnaire request notification -UC 70
NORMAL FLOW	1. The psychiatrist accesses the patient's diary historical data -UC 130 2. Based on the displayed data, the psychiatrist decides to send a questionnaire request to the patient 3. The psychiatrist selects the type of questionnaire to be filled by the patient 4. The psychiatrist selects the questionnaire administration options (e.g., periodicity, deadline, etc.) 5. The psychiatrist confirms the previous choices and sends the request -UC 70

USE CASE UNIQUE ID	/UC 160/ (Filling questionnaire)
DESCRIPTION	The patient fill a self-assessment questionnaire
ACTORS	Patient
PRECONDITIONS	The patient wants or was requested by the system to fill a self-assessment questionnaire -UC 150, UC 140
POSTCONDITIONS	A new questionnaire is filled
NORMAL FLOW	1. The patient decides to fill a self-assessment questionnaire 2. The patient accesses her personal portal page dedicated to self-assessment questionnaires 3. The patient chooses what questionnaire she wants to fill 4. The patient fills the questionnaire 5. (If needed) The psychiatrist is notified that the patient filled the questionnaire -UC 70
ALTERNATIVE FLOW (QUESTIONNAIRE FILLED ON REQUEST)	1. The patient receives a request to fill a self-assessment questionnaire -UC 70 2. The patient accesses her personal portal page dedicated to self-assessment questionnaires 3. The patient fills the requested questionnaire 4. (If needed) The psychiatrist is notified that the patient filled the questionnaire -UC 70

USE CASE UNIQUE ID	/UC 170/ (Missing filling questionnaire)
DESCRIPTION	If the patient didn't fill a questionnaire that she was requested to, she is reminded by the system to fill it. If she doesn't fill it, the psychiatrist is notified
ACTORS	• Hospital psychiatrist • Patient
PRECONDITIONS	The patient was requested to fill a self-assessment questionnaire -UC 150, UC 140
POSTCONDITIONS	• The patient is reminded to fill a questionnaire • The psychiatrist is notified that the patient didn't fill her questionnaire
NORMAL FLOW	1. The patient has not filled the requested questionnaire within the 5p.m. of the due day 2. The system reminds her to fill her questionnaire -UC 70 3. If the day after she still hasn't filled it, the system can notify the psychiatrist (UC 70), depending on the psychiatrist specified action (UC 80), and on the patient's privacy settings (UC 30)

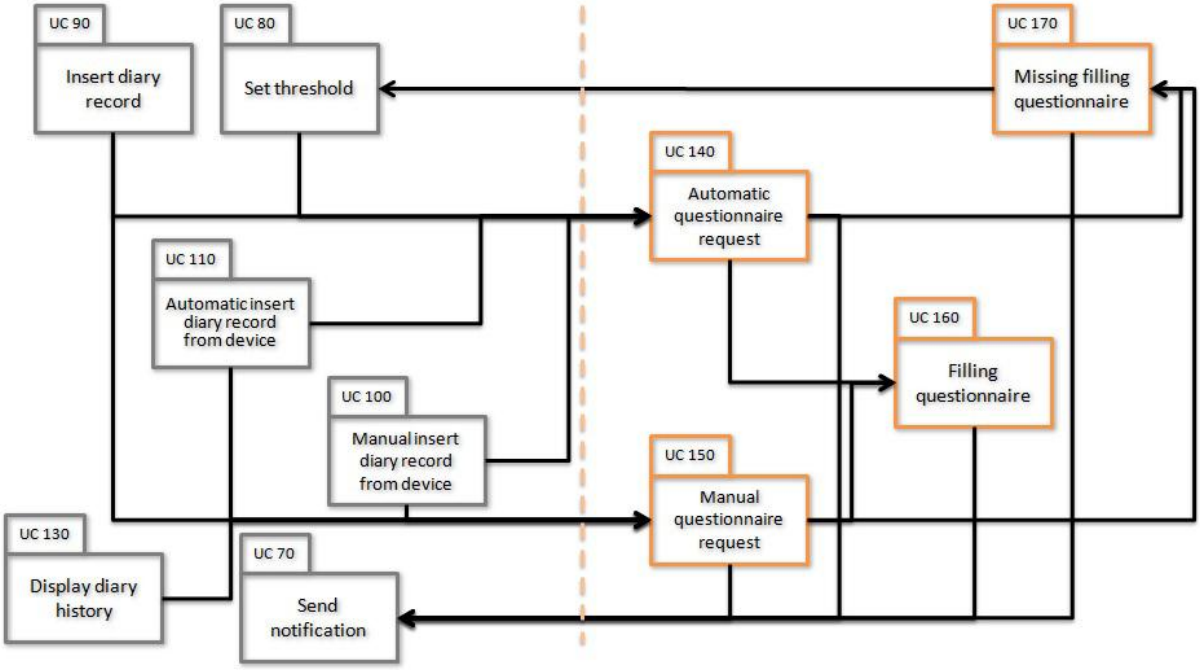

Figure 12 Self assessment questionnaire dependencies

5.2.7 Activity monitoring

This set of use cases encompasses all the services related to the management of physical activity, sleep activity and drug intake activity. These three will be referred generically as ”activity” in the use cases.

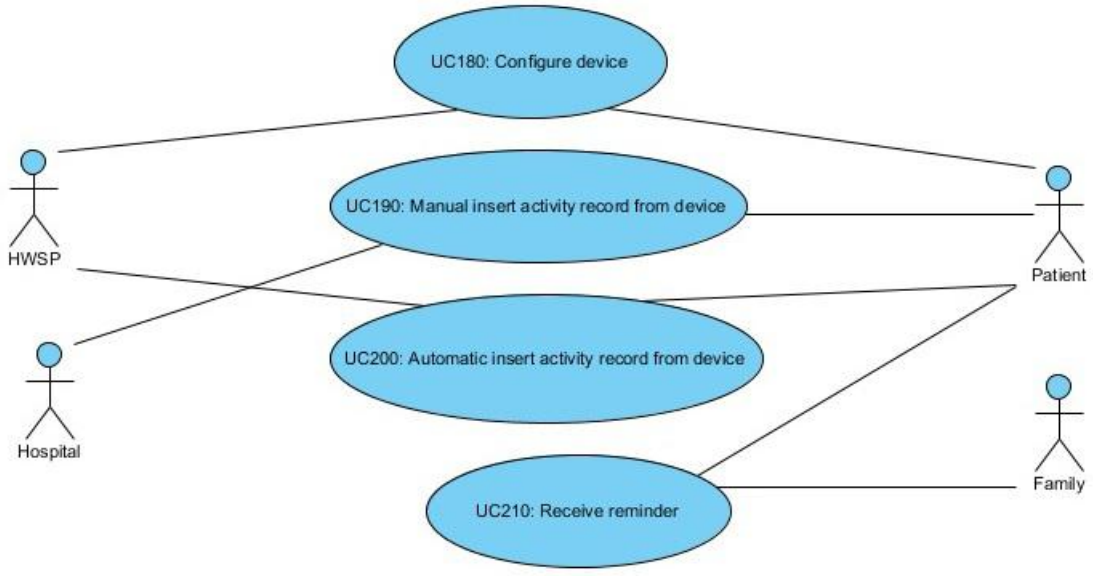

Figure 13 Activity monitoring actors

USE CASE UNIQUE ID	/UC 180/ (Configure device)
DESCRIPTION	A device for enforcing a prescription of an activity (sleep, physical activity, drug intake) is configured
ACTORS	• Patient • HWSP
PRECONDITIONS	A prescription is present in the system -UC 50
POSTCONDITIONS	A device gets configured
NORMAL FLOW	1. The HWSP monitors the patient's prescription of physical activity, sleep, and drugs 2. If there are changes in the prescription, the HWSP creates a new configuration for the patient's devices 3. Once the patient connects her devices to internet, the configuration is downloaded 4. If the patient doesn't connect her devices for three days (from when the prescription was issued), she is notified that she should update her devices configuration - UC 70

USE CASE UNIQUE ID	/UC 190/ (Manual insert activity record from device)
DESCRIPTION	The patient uses a monitoring device, and periodically she sends manually the information collected to the portal
ACTORS	• Patient • Hospital psychiatrist
PRECONDITIONS	• The device is registered to the patient's profile -UC 40 • The device is configured properly -UC 180
POSTCONDITIONS	A new activity record is added
NORMAL FLOW	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily or weekly), the patient connects her device to the PC or Mobile Phone and uploads the data to her personal portal
ALTERNATIVE FLOW (DATA ENTRY FROM PSYCHIATRIST)	1. The patient goes to her periodic visit at the Hospital, carrying her personal device 2. The psychiatrist connects the patient's device to his PC and uploads the data to the patient's personal portal

USE CASE UNIQUE ID	/UC 200/ (Automatic insert activity record from device)
DESCRIPTION	The patient wears a monitoring device that periodically sends information to the portal
ACTORS	• Patient • HWSP
PRECONDITIONS	• The device is registered to the patient's profile -UC 40 • The device is configured properly -UC 180
POSTCONDITIONS	A new activity record is added
NORMAL FLOW	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily), the device sends the measured data to the patient's personal portal
ALTERNATIVE FLOW (DATA PASSES THROUGH HWSP)	1. The patient wears or uses her monitoring device 2. Periodically (e.g., daily) the device sends the data to the HWSP 3. The HWSP sends the data to the patient's personal portal

USE CASE UNIQUE ID	/UC 210/ (Receive reminder)
DESCRIPTION	If the patient didn't comply with the prescribed activity, she is reminded by the system to do it properly. If she doesn't do the activity at all, the day after she is requested to specify a reason why she didn't
ACTORS	• Patient • Family
PRECONDITIONS	The doctor prescribed some activity to the patient -UC 50
POSTCONDITIONS	• The patient is reminded to do her activity • The patient is requested to add a justification
NORMAL FLOW	1. The patient has not done the prescribed activity within the time of the day specified in the prescription 2. The system reminds her to do her activity properly (e.g., at 5p.m. it reminds her to do 10000 steps, or at 11p.m. it reminds her to go to sleep, or it reminds her to take the medicine at 10a.m.) -UC 70 3. If the she doesn't comply at all with her prescription in the due day and time, the day after the system requires her to specify in the portal a justification of the reason why she didn't performed properly (e.g., "yesterday it was rainy and I couldn't go out to run", or "I was out and I forgot my medicine at home", etc.)

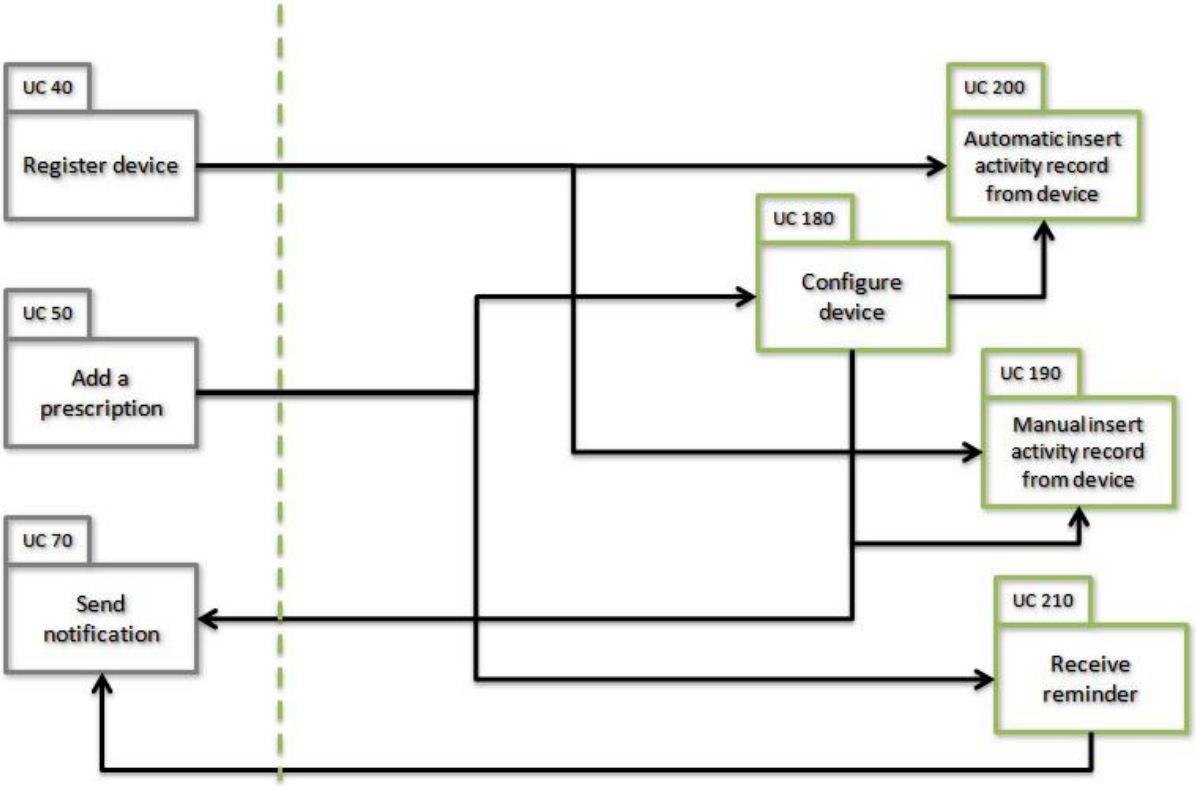

Figure 14 Activity monitoring dependencies

5.2.8 Drug therapy management

In addition to the previous section of use cases, this section details a set of additional use cases specific for the drug management, which includes additional services with respect to physical activity and sleep prescriptions which are mostly done via online purchasing.

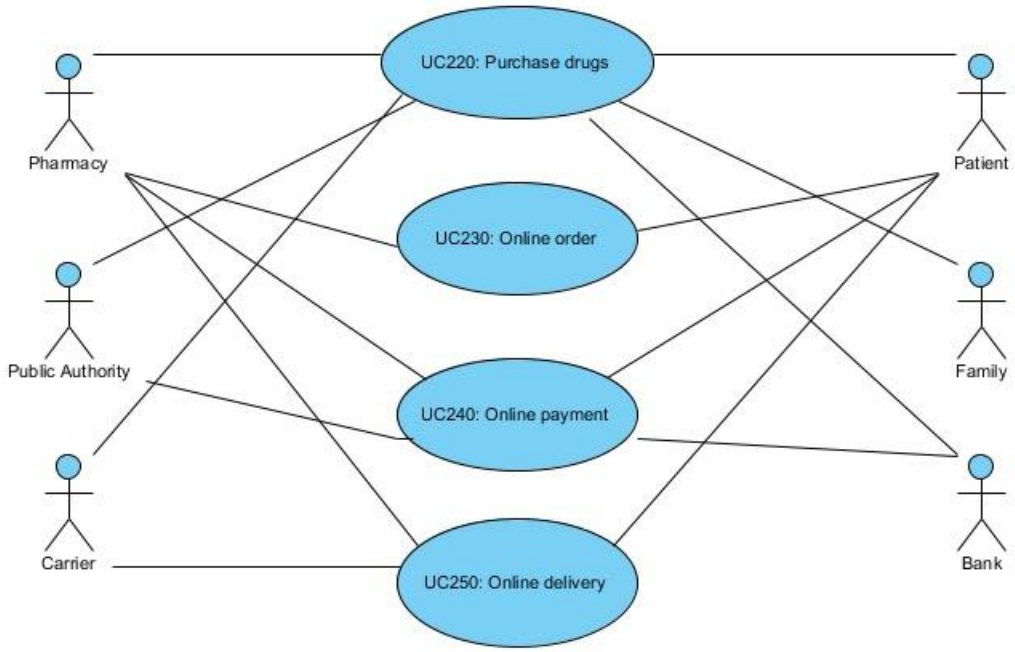

Figure 15 Drug therapy management actors

USE CASE UNIQUE ID	/UC 220/ (Purchase drugs)
DESCRIPTION	The patient can purchase the drug, if she has a valid prescription in the system. As alternative, all the steps in the purchase (ordering, payment and delivery) can be anonymous
ACTORS	• Patient • Pharmacy • Family • Bank (optional) • Public authority (optional) • Carrier (optional)
PRECONDITIONS	A drug prescription exists -UC 50
POSTCONDITIONS	The patient receives her drugs
NORMAL FLOW	1. The patient (or a patient's family member) goes to the pharmacy and she asks for her drugs 2. The pharmacist checks in the system whether the patient has a valid prescription or not 3. Once the prescription is verified, if the drug isn't already paid (e.g., by the Insurance Company's bank, by the patient's bank, by a Public authority, etc.), the pharmacist asks the customer to pay 4. The patient pays and takes her drugs home 5. Once the payment is done, the system automatically updates the validity of the prescription
ALTERNATIVE FLOW (ONLINE PURCHASE)	1. The patient can issue an online order through her portal -UC 230 2. The payment can be managed online and anonymously by the system -UC 240 3. The patient can ask for the delivery of the package, done by a carrier -UC 250

USE CASE UNIQUE ID	/UC 230/ (Online order)
DESCRIPTION	The patient carries out the order, with the possibility to select an anonymous procedure
ACTORS	• Patient • Pharmacy
PRECONDITIONS	A drug prescription exists -UC 50
POSTCONDITIONS	An order is placed
NORMAL FLOW	1. The patient accesses her personal portal page, dedicated to drugs management 2. The patient checks the possible pharmacies that can satisfy the drug request and she chooses one of them (if she doesn't choose, one is selected randomly) 3. The patient opens the request form, indicating the privacy order settings. In this section she can decide that her personal data have to be anonymous for the pharmacy 4. The patient forwards the order

USE CASE UNIQUE ID	/UC 240/ (Online payment)
DESCRIPTION	The patient forward the payment, with the possibility to select an anonymous procedure (this use case is just for completing the scenario, pseudonymous/anonymous online payment use case is out of the interest of the eHealth scenario, and it should need further investigation)
ACTORS	• Patient • Bank • Public authority • Pharmacy
PRECONDITIONS	The order is placed -UC 230
POSTCONDITIONS	The order is confirmed and paid
NORMAL FLOW	1. The patient accesses her personal portal page, dedicated to drugs management 2. The patient selects the drugs order that she wants to pay 3. The patient opens the request form, indicating the payment preference (it may be that the patient's insurance company is paying, or that a public authority is paying, like regional healthcare system, or that the patient is paying by herself, etc.). In this section she can decide that every actor involved in the payment has to know only the necessary personal data to execute the procedure correctly (e.g., the bank knows the patient's account ID, the pharmacy details and the amount, but not the kind of drugs; the pharmacist receives the payment but he can't knows the patient's personal data, etc.) 4. The patient confirms the payment

USE CASE UNIQUE ID	/UC 250/ (Online delivery)
DESCRIPTION	The patient selects the carrier for the drug, with the possibility to decide for an anonymous procedure
ACTORS	• Pharmacy • Carrier • Patient
PRECONDITIONS	The order is placed (UC 230) and payed (UC 240)
POSTCONDITIONS	The patient receives her drugs
NORMAL FLOW	1. The patient accesses her personal portal page, dedicated to drugs management 2. The patient selects the drugs order that she wants to receive at home 3. The system presents a list of possible carriers and the patient selects one of them (if she doesn't choose, one is selected randomly) 4. The patient confirms the options and forward the complete order 5. The selected carrier receives a notification that a new (anonymous) package is to be delivered. It just know the sender, the receiver and the package identifier. 6. The carrier performs the delivery

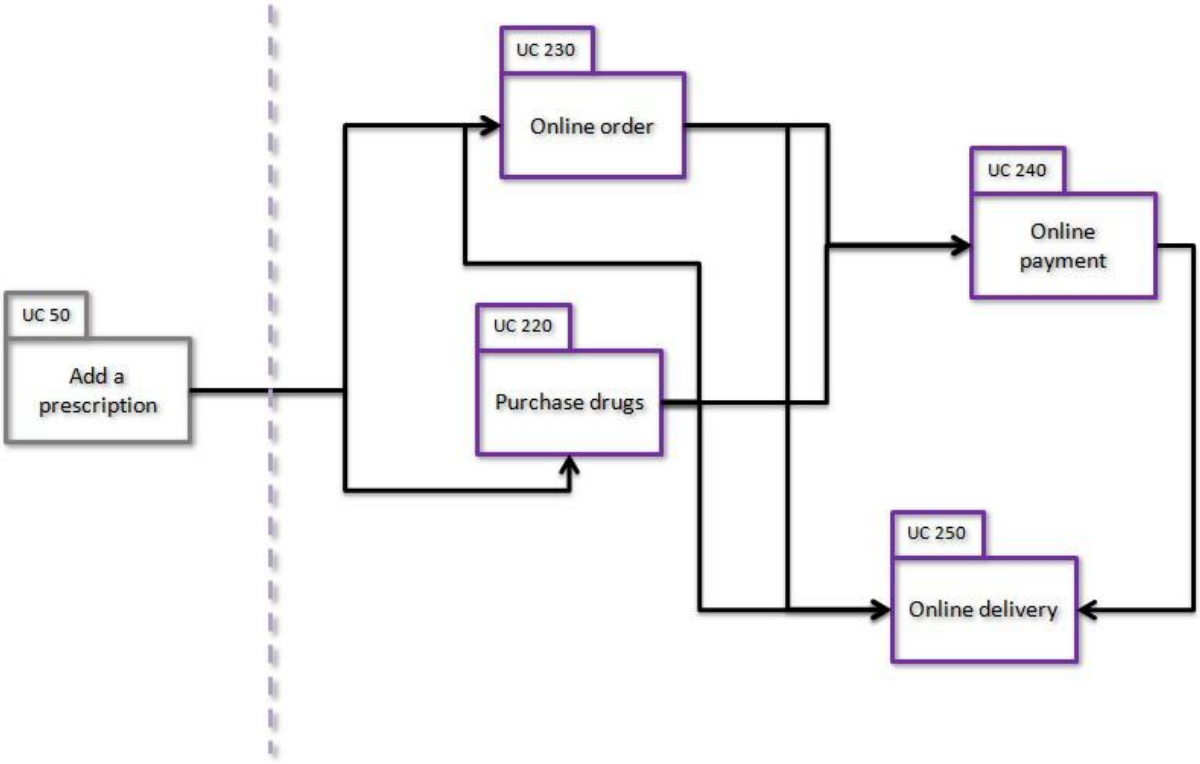

Figure 16 Drug therapy management dependencies

5.2.9 Epidemiological studies

Data present in the Cloud can be shared by different hospitals with public bodies (like regional or national healthcare systems) for epidemiological studies, for example on compliance with treatments, or on effectiveness of drug treatments, on diseases development, etc. The next set of use cases are related to this scenario, with the addition of the reporting of the adverse drug reactions (ADR).

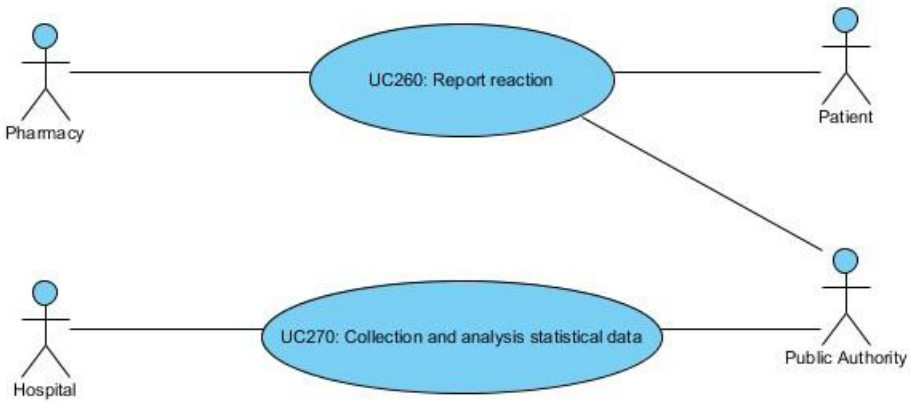

Figure 17 Epidemiological studies actors

USE CASE UNIQUE ID	/UC 260/ (Report reaction)
DESCRIPTION	An adverse drug reaction experienced by the patient is reported to the National Healthcare System
ACTORS	• Patient • Pharmacy • Public authority
PRECONDITIONS	Patient experienced a reaction to a drug
POSTCONDITIONS	The reaction is reported to the National Healthcare System
NORMAL FLOW	1. The patient goes to the pharmacy where she bought the drug and communicate the adverse reaction to the pharmacy (e.g., red dots appeared on the skin, etc.) 2. The pharmacist compile the proper form, indicating only the drug taken by the patient and the adverse reaction. Only patient's name initials, sex, birthday and ethnicity of the patient are reported 3. The system complete the rest of the information, including the other prescriptions and drugs that the patient was using, without showing these information to the pharmacist 4. The full report is sent to the public authority (e.g., national or regional healthcare system)

USE CASE UNIQUE ID	/UC 270/ (Collection and analysis statistical data)
DESCRIPTION	Different hospitals can send their data to public authority that collects and analyzes them
ACTORS	• Hospital • Public authority
PRECONDITIONS	• Different hospitals subscribed some kind of agreement with the public authority (e.g., national or regional health-care system) • The public authority created a protected repository in the Cloud, and specified an interface through which the Hospitals can send data to it (but they cannot access the repository directly)
POSTCONDITIONS	The public authority receives the hospital data
NORMAL FLOW	1. Periodically, the Hospital Information System (HIS) of each Hospital participating to the network sends information to the public authority repository 2. The public authority decides to start an investigation (e.g., drug compliance after three months from the prescription) 3. The public authority runs statistical analysis over anonymized data received from the different entities

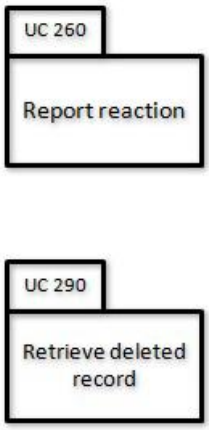

Figure 18 Epidemiological studies dependencies

5.2.10 Auditability

This section includes use cases related to system auditability.

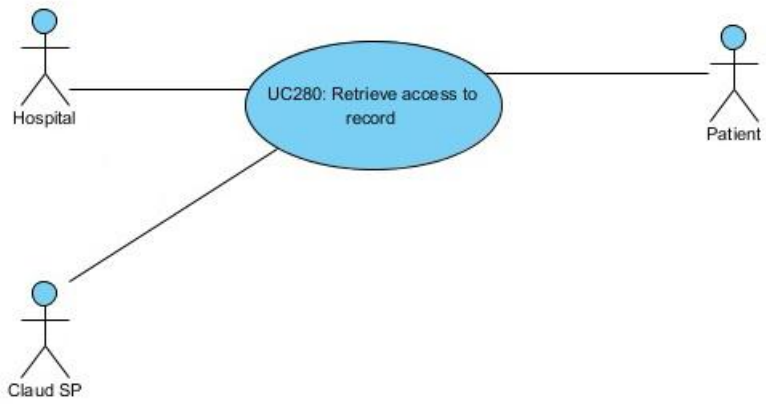

Figure 19 Auditability actors

USE CASE UNIQUE ID	/UC 280/ (Retrieve access to record)
DESCRIPTION	Patient retrieves a list of all entities that accessed a record of her data
ACTORS	• Patient • Hospital • Cloud SP
PRECONDITIONS	The patient wants to verify who accessed her data
POSTCONDITIONS	The patient receives the log of the accesses to her data
NORMAL FLOW	1. The patient accesses her privacy settings page on the portal 2. Close to the different type of data, where the patient can specify who can access them, she can find also a button to verify the access log for the specific data. The patient presses this button 3. The system displays all the accesses to the specific data, with the identifier of the subject that accessed it and date and time, and which action was performed (e.g., modification, deletion, etc.)

5.3 Home healthcare application architecture

5.3.1 Introduction

Preamble on eHealth services

Although San Raffaele Hospital (HSR) is very much interested in offering support and motivational services to its depressed patients, home healthcare services are not yet part of HSR portfolio. The scenario described in Section 2 is a very complex one and requires several services, which may be offered by a single entity (like HSR), but also as a joint offer between different service providers. While the practical implementation, of which a draft description will be illustrated in Section 5.4.2, could be on a single application or (virtual) machine, it make probably sense to differentiate between the different roles in the application management, as well as in the data ownership, control and liability. Section 5.3.3 will describe this allocation in detail, providing a possible rationale for each allocation.

Testbed approach

In the eHealth scenario, being a step into the future with respect to traditional healthcare service models, several services have to be created from scratch. In the TClouds project it

was decided to implement also other functionalities already available in the HSR legacy system, for different reasons:

- Easiness of integration: a newly developed EHR system, only with the functionalities needed for this specific scenario, will be much easier to be integrated with the newly developed services, with respect to the legacy system, which has also the limitation that, being a delivery system in use, it could be unavailable in the project and bring additional delays.
- Legal constrains of the database: while integrating with the actual EHR, one of the possible problems is related to the access to real patients' data. This problem could also lead to reduced (or delayed) availability of the system during the integration or validation activities. On the other hand, the realization of an ad-hoc testbed application, with realistic data (in place of real data) will provide a smother development and integration process.
- Availability to the partners: building an application (or a set of application) that will implement the scenario in a testbed environment will have the advantage that it can be deployed and instantiated also by other partners, that can test their own prototypes and solutions during the project lifetime.

Organization of this Section

The main content of this Section is expressed in Section 5.3.3, where the conceptual architecture for the eHealth scenario is described. Beside the conceptual model of the reference architecture, in the first year TClouds (and WP3.1 in particular) will produce a first prototype of the scenario applications. This prototype will very likely implement a minimum subset of the described functionalities and information flows, thus we added Section 5.4.2 to describe how the first instantiation of the reference architecture will (probably) look like.

Finally Section 5.3.4 will provide a graphical representation of the data flows derived from the reference architecture.

5.3.2 Notation

In the architecture picture (Figure 21) the following notation has been used:

- **Characters:** the different characters are the actors external to the system, as described in Section 2.3
- **Green boxes:** these are all the front end of the different applications with which the end user will interact with the system, including also personal devices interfaces.
- **Blue boxes:** the blue boxes represent all the applications and functions implemented by each Service Provider. In general, this is the service logic framework.
- **Orange boxes:** the orange components are the technical interfaces of the applications (blue boxes) they belong to, exposed toward other applications.
- **Purple cylinders:** they have been used to describe databases and repositories. Of course each system will have its own database to make the application run (e.g., log), but the ones shown in the picture are important to be explicitly mentioned.
- **Black arrows:** connections of the different applications. The arrow identifies a first tentative direction for the data flow. Indeed not all the data flows are bidirectional.

5.3.3 Application architecture

Figure 20 shows an overview of the eHealth application architecture. The functionalities have been divided in four different applications depending on who owns the data, who should protect or disclose it, and who could reasonably offer specific services. The fact that the reference architecture for the eHealth scenario includes different service providers doesn't necessarily imply that a single service provider (like HSR, but also Philips, etc.) could decide to implement one or more (or even all) the proposed services.

As described in the introduction of this Section, the actual implementation of the scenario is a futuristic view of the evolution of traditional healthcare service models, not constrained by pre-existing solutions or services to be integrated. This leaves open the possibility to speculate on the best allocation for each functionality or service, making arbitrary decisions. This Section will further describe this allocation and the rationale behind it.

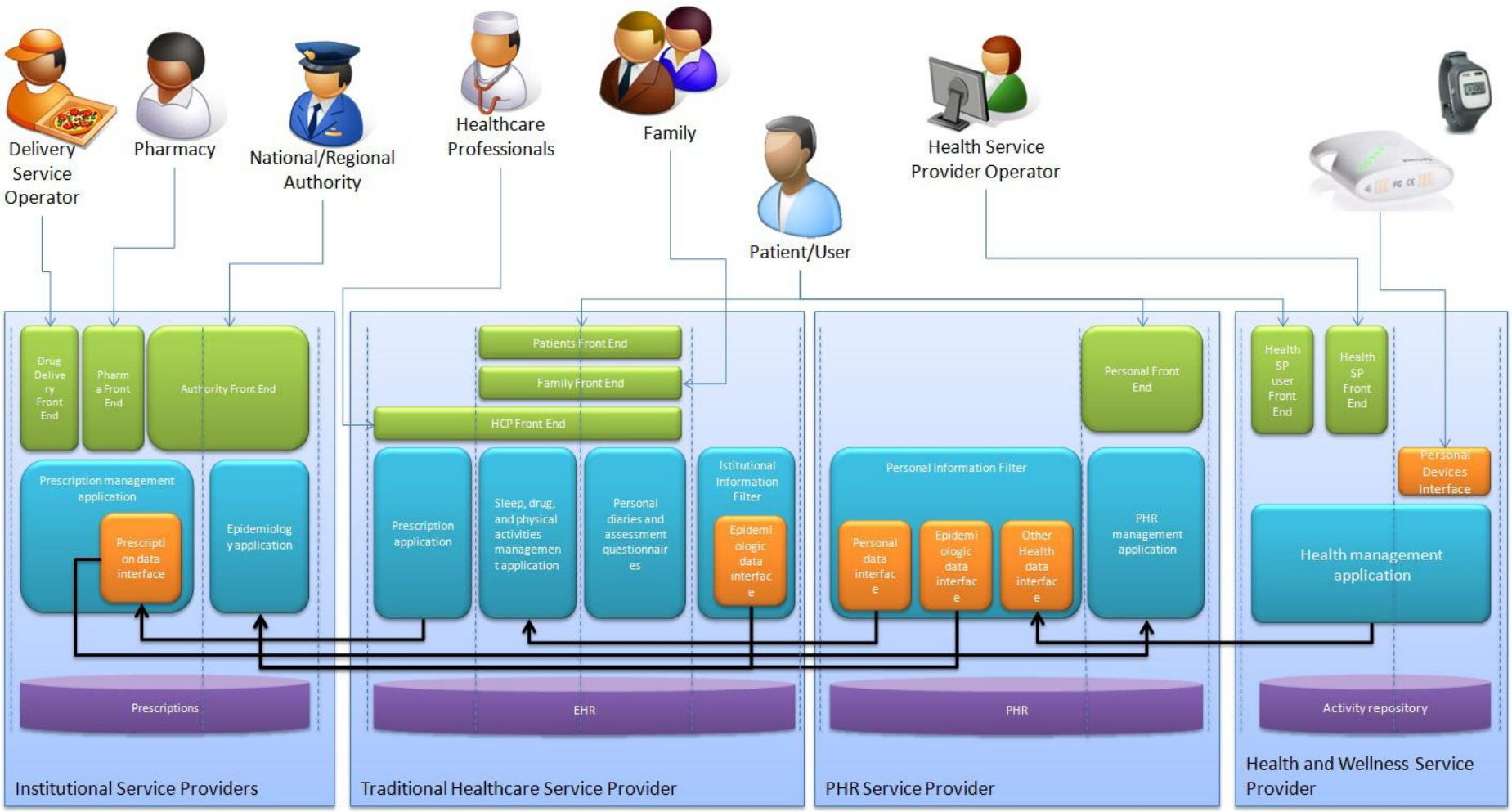

Figure 20 TClouds eHealth application architecture

Traditional Healthcare Service Provider (e.g., San Raffaele Hospital)

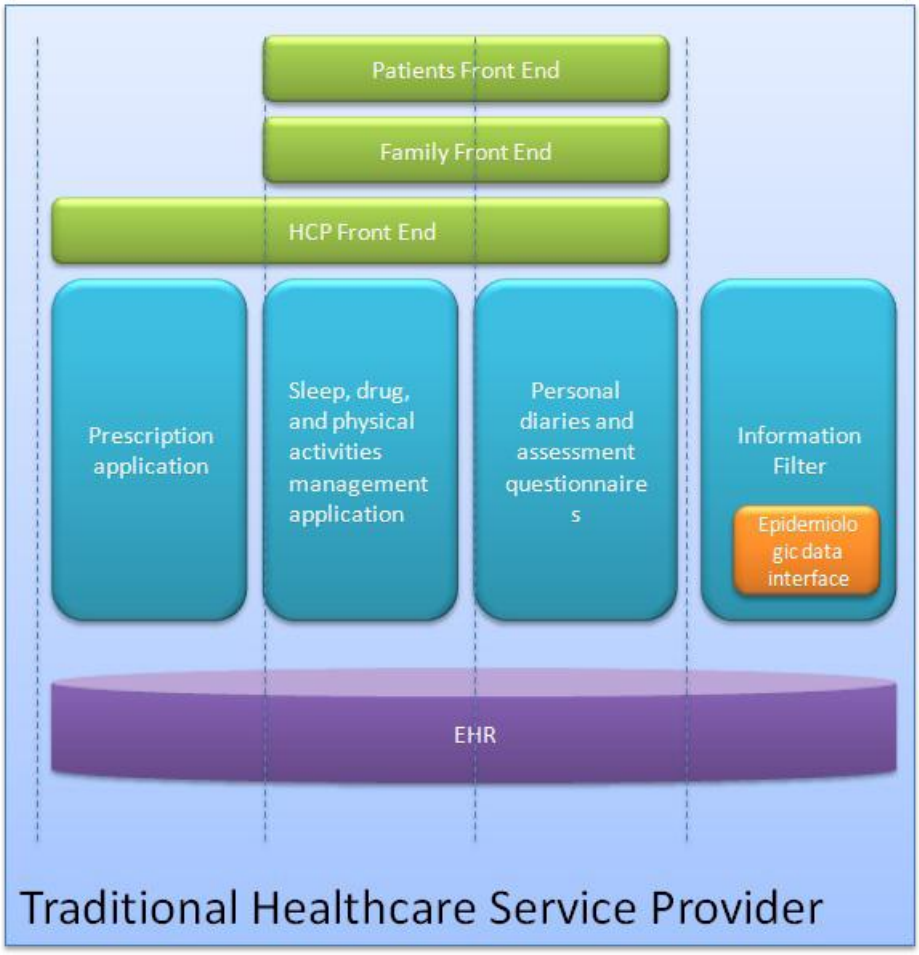

Figure 21 Application architecture for Traditional Healthcare Service Providers (e.g., Hospital)

The first group of functionalities is represented by a traditional healthcare Service Provider (e.g., San Raffaele Hospital), that would like to offer added value services to its depressed patients. This service provider is in general responsible for having an EHR repository, and maintaining and protecting it as requested by national regulations.

Beside traditional services (visit booking, exams results storage, etc.), the scenario foresees patient's remote support in different aspects of daily life, in particular those related to drug intake, sleep management and physical activity, which are known from medical literature to have a meaningful impact on depression development and treatment. This service provider could offer these services for the patient, together with additional diagnostic services like mood questionnaires and depression self-assessment questionnaires.

Furthermore the scenario foresees additional support services for ePrescription (i.e., electronic prescription), including drug ordering, purchasing and delivery to the patient's house. The Healthcare Service Provider could offer part of these services, in particular those dedicated to Healthcare Professionals (HCP), like the hospital psychiatrist, to create and modify prescriptions.

The main components for this Service Provider are:

- **Patients Front End:** This is a postal that allows the patient to manage her activities and personal data, connecting her to the services offered.
- **Family Front End:** The family can access to the patient's data through this front end, that permits to interact only with the activity monitoring application and personal diaries and assessment questionnaires application. The system checks the privacy settings for every access.
- **HCP Front End:** This component allows the HCP to use the system, permitting him the access to the prescription application, activity monitoring application, and personal diaries and assessment questionnaires application.
- **Prescription application:** When a doctor prescribes drugs, exercises, or something else to the patient, the information are managed by the prescription application, that will also take care of the communication of the data to the Institutional Service Provider.
- **Sleep, drug, and physical activities management application**: The Healthcare Service Provider offers a personal management and monitoring service for the activities that the depressed patient can execute during her therapy. This application will collect and organize personal data, to be displayed by the Front End. The application can also collect data from the PHR Service Provider (in according with the privacy settings selected by the patient).
- **Personal diaries and assessment questionnaires:** This application will manage the questionnaires and the personal diaries functionalities, used by patient through her Front End. This application will save the questionnaire data in the EHR.
- **Information Filter:** This application will manage services of information exchange with public institutions, through proper data filtering, depending on privacy regulations, especially for epidemiological studies. This part will be detailed in sections below, while presenting the Institutional Service Providers.

PHR Service Provider

A second possible service provider is the PHR (Personal Health Record) Service Provider. PHR can be considered as an extension of the EHR concepts, including also several other personal data directly related to health, like physical activity, sleep habits, etc. There are many different players in this field, but two main examples are Microsoft HealthVault and Google Health.

Obviously, many data present in the PHR can be duplicated in the EHR. Furthermore in the prototypes implemented in TClouds, probably these two applications will be offered by a unique Service Provider (i.e., San Raffaele), creating practically only one complete repository (and corresponding set of services) with patients' personal data.

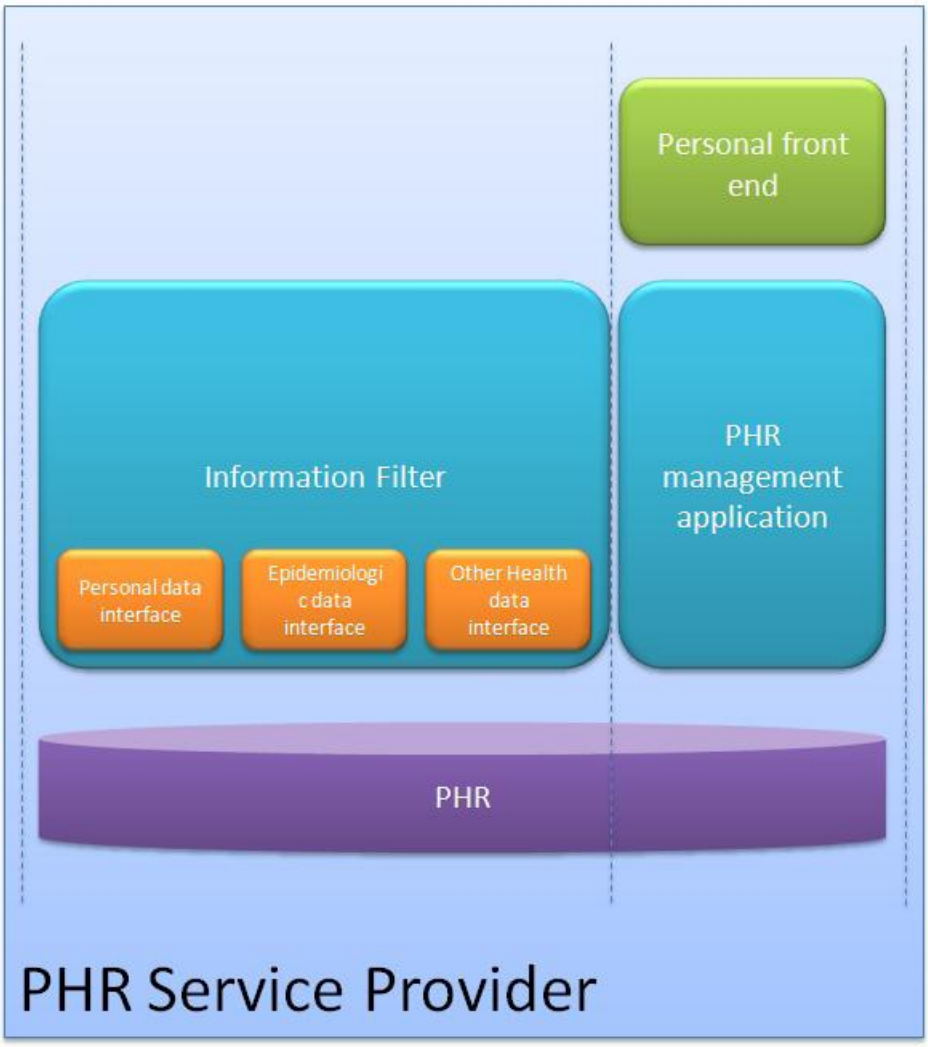

Figure 22 Application architecture for PHR Service Provider (e.g., Microsoft HealthVault)

However it was considered more useful to distinguish these two types of providers mainly for three reasons: 1) it is important to distinguish between data used in clinical practice (EHR) which have particular legal constrains, not only in terms of protection, but also in terms of integrity, preservation, etc. (while this doesn't apply to personal data, like physical activity); 2) it is interesting to distinguish between data that are responsibility of the Hospital, like the EHR (on which the patient has of course many rights, but their management has also other implications and obligations for the Hospital, for example for accountability, reimbursement, etc.), and data whose management may be responsibility of the patient, like PHR; and 3) it may be that for the first prototypes and mock-ups off-the-shelf PHR services will be used (e.g., Microsoft HealthVault). In the end the 3rd point didn't materialize, but at the time of the choice this wasn't sure.

Here a brief description of the boxes in the PHR Service Provider architecture:

- **Personal Front End:** Through this component, the patient can access her personal data in PHR Service Provider, where all data are owned by her. For this reason, the Personal Front End needs to interact with the PHR management application.
- **Information Filter:** This is the application to manage information exchange with other Providers, through proper filtering, depending on privacy regulations. In particular the

data can be shared with Institutional Service Provider, for example for epidemiological studies; or with Traditional Healthcare Service Providers, for example for sending personal data (e.g., physical activity performances, sleep data, etc.), or to receive issued prescriptions, etc.; or with Health and Wellness Service Providers, for example to receive personal data (e.g., physical activity performances, etc.).

- **PHR management application:** With this application, the users can manage the data and services in the PHR. This application will be used to specify the access rights to the PHR repository, and thus to configure the Information Filter.

Health and Wellness Service Provider

The third possible service provider is the Health and Wellness Service Provider, sometimes referred only as Health Service Provider, that shouldn't be confused with the Traditional Healthcare Service Provider, described above. The Health and Wellness SP is a service provider that provide services which are not part of traditional healthcare system, but are anyway related with the persons' health. In general most of these services are related to physical activity and nutrition. An example of this kind of service providers can be for example Philips, and in particular the DirectLife service for physical activity monitoring, helping the person in tracking how much she moves every day, setting personal goals and track progresses. Furthermore the DirectLife service support the person to increase the activity levels, providing a personal coach who can help to stay motivated. Another example can be a normal gym, with a training program and advanced equipments (treadmills, etc.), able to collect the person's training data. Of course also a Hospital, if it owns an infrastructure that allows remote patient's physical activity monitoring, can act as Health and Wellness Service Provider.

The Health and Wellness SP usually will have its own interface (toward end users, its own operators or its devices), as well as its own application and data repository. However it is possible that such Service Providers decide, for business opportunity reasons, to share data with other services and entities (e.g., traditional healthcare service providers). In this case they become interesting for our scenario.

Here the information related the boxes in the Figure 23:

- **Health SP user Front End:** The users can access to the services offered by the portal trough this front end.
- **Health SP Front End:** This component allows the Health Service Provider Operator to use the system, managing the services through the Health management application.
- **Health management application:** With this application, the actors can manage the data and services in the activity repository. It also allows sending the data downloaded from devices to others service providers.
- **Personal Devices interface:** The devices need to interact with the system to upload patient's data. Personal Devices Interface allows this communication.

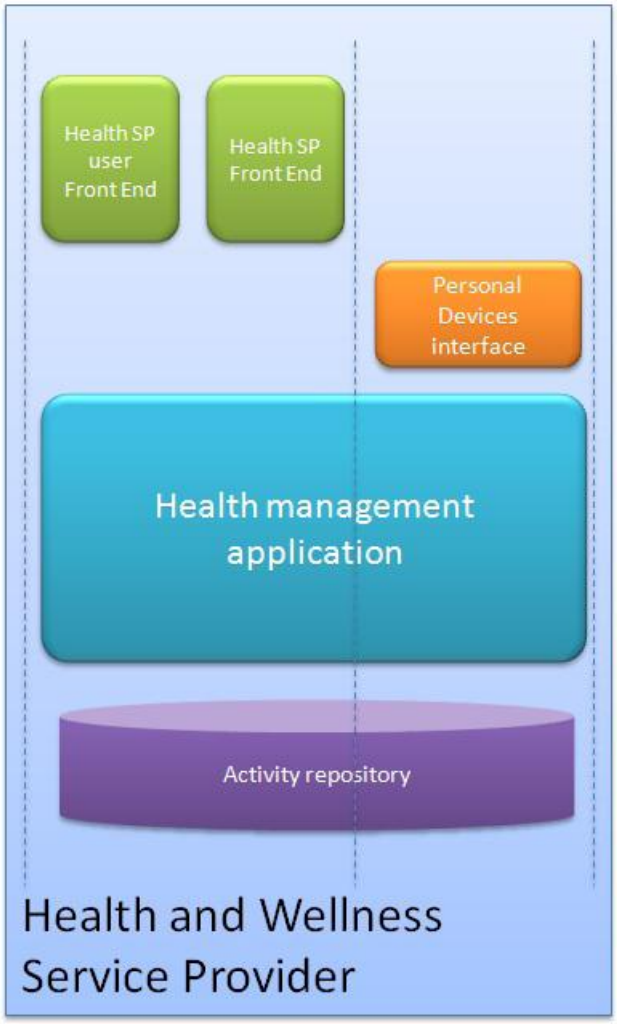

Figure 23 Application architecture for Health and Wellness Service Provider (e.g., Philips)

Institutional Service Providers (e.g., National or regional healthcare system)

The last type of service providers involved in this scenario is the Institutional Service Providers, for example, regional and national healthcare systems. From now on they will be referred to (also) as authority.

Figure 24 Application architecture for Institutional Service Providers (e.g., National or regional healthcare system)

In general, regional and national authorities are slower in the process of adoption of new technologies, as well as in the computerization of their processes. However for the specific scenario, it makes sense to hypothesize an independent and trusted third party for specific services. In case this kind of independent entity is not available or too slow in the adoption of the proposed solutions, of course the Traditional Healthcare Service Provider can include and implement these services, asking the patients to rely on its reputation only. However, from a conceptual model, it is reasonable to assign some functions of the scenario to an application provided by Institutional Service Providers. In this case two main roles were assigned to this SP. The first is to be an independent entity that can manage a regional and national database of ePrescriptions. Prescriptions will be issued by entities known and authorized by the regional/national authority, like general practitioners (GPs), hospital doctors, etc. Moreover it makes sense to imagine that this prescription service is inserted into a regional/national care plan, which is already present in many regions of Italy, and probably also in Europe. This kind of plans sometimes includes also meal delivery at home for elderly people, or similar services. It is reasonable to assume that also drugs can be delivered at home through the same care plan, and that (trusted) pharmacies and deliverers can participate (through appropriate authentication and interfaces) to this regional/national

infrastructure. This role was assigned to a regional/national healthcare system also because in Italy (and often in other countries), some drugs are reimbursed by these entities.

The second role is to be active in epidemiology and population wide studies. The choice of assigning this role to the Institutional SP is arbitrary, but it was justified by the previous role assignment. Indeed, an authority that owns data on how many drugs were prescribed and purchased in a region or country, can decide to start studies on compliance with drug treatments, regional/national burden of a specific disease, etc. Furthermore this application could benefit from the interaction with other Service Providers, like hospitals, correlating for example the impact for noncompliance with therapies and re-hospitalizations and derived costs, etc. Moreover it can probably interact also with PHR Service Providers that, if the users agrees and provide an explicit consent, could provide additional data related to habits, like physical activity, sleep, etc.

This second application in particular seems to be very suitable to be hosted on a Cloud infrastructure, as it has very elastic needs in terms of computation, data transfer load, etc. Indeed this activity is not expected to be continuous over the normal authority activities, but to be done seldom, for example when strategic choices (e.g., budget allocations) or corrective actions must be undertaken.

In the Institutional Service Providers the boxes are:

- **Authority Front End:** The employees of the public Institution offering the services here described will have its own dedicated Front-End to manage the services. In particular a specific user interface can be designed to monitor epidemiological studies.
- **Pharmacy Front End:** The pharmacist can access the data related to the prescription using this front end, thanks to the services included in the Prescription management application.
- **Drug delivery Front End:** The deliverer, responsible to bring the drugs to the patient, takes the shipment details from this front end.
- **Prescription management application:** Through this application, the actors can manage the data and services in the Prescription repository. This application is in charge of the management of the regional/national database of the prescriptions. It will also check the external Service Providers identity and authorization to the access of the prescription data.
- **Epidemiology Application:** There are some service providers that send material for epidemiological studies to the Institutional Authority. This application elaborates this information to produce statistical data.

5.3.4 Data Flows

Figure 5 shows a Data Flow Diagram (DFD) derived from the above mentioned architecture. A DFD is a graphical representation of the flow of data through the information system. The following notation has been used:

- **Squares:** Entities, agents or persons external to the system
- **Rounded corner squares:** System functions and application (service logic)
- **Double lines:** Repositories and databases
- **Arrows:** Possible information flows

Furthermore the colours have been used to distinguish between the different Service Providers, in particular:

- **Light Blue:** Traditional Healthcare Service Provider
- **Green:** Health and Wellness Service Provider
- **Orange:** PHR Service Provider
- **Purple:** Institutional Service Provider

Finally we used tick black lines to highlight information flows which are potentially happening between different service providers.

5.4 Architecture instantiation for the first year mock-up

The first year prototype is just a mock-up implementing just part of the functionalities depicted in the architecture shown in Figure 5. The scope of this Section is to provide an insight of a realistic view of what the first year prototype looks like, as well as to describe the technologies that are used.

5.4.1 First year mock-up scenario

The first year mock-up application will be kept as simple as possible, in order to provide a solid starting point for the integration and realization of the proposed services. Only two Service Providers will be involved:

1. Hospital Service Provider: this provider is a combination of the Traditional Healthcare Provider and PHR Service Provider. From now on it will be referred to also as ”TH+PHR Service Provider”.
2. Health and Wellness Service Provider: as described in the previous Sections.

Each of the two Service Providers will run its own independent application, on separate Virtual Machines. Two additional actors will be involved: the patient and the hospital psychiatrist.

Overall, the scenario based on the expected mock-up will look as follows:

- The patient will be provided with a smart bracelet (the ActiWatch) that will measure physical activity, sleep, and environmental light data continuously. The patient will also be provided with a docking station that can be connected to her PC and that will download the data from the device and upload the data to the Health and Wellness Service Provider (HWSP). The patient’s PC will have a client that will be developed and that will take care of a secure connection with the HWSP.
- A minimal web interface for the database investigation will be realized on the HWSP side.
- The patient will also subscribe to the TH+PHR SP portal. There she will find a page where she can report her mood conditions on mood diaries (structured data, where she will be able to evaluate her levels of depression, discomfort, anxiety, etc.). Furthermore on the same portal she will also find validated medical scales for measuring depression levels.
- The patient can decide to provide to TH+PHR SP the credentials to download data from HWSP. In that case an unidirectional connection between the two Service

Providers will be created and the data will be sent from HWSP to TH+PHR SP whenever new data is generated.

- The patient will be able to check her personal data collected from the HWSP correlated with her medical data, through the TH+PHR SP portal.
- The patient will also be able to customize her privacy settings on her web page.

As far as the psychiatrist is concerned:

- The doctor will be able to see all the TH+PHR SP data, as it will be an employee of this provider with proper authorizations.
- If the patient provided the proper authorization through her privacy settings, the doctor will be able also to see patient's physical activity, sleep and environmental light data.
- The doctor will also be able for each patient to specify some thresholds that should be monitored (e.g., sleeping at least 7hours, doing at least 20 min of physical activity every day, etc.), as well as some countermeasures that should be activated in case the thresholds are (or are not) met. Some examples are sending an SMS or sending an email, to the doctor or the patient, etc.

5.4.2 Draft of the architecture instantiation for the first mock-up

Figure 25 shows the subset of Figure 20 that will be part of the 1st year prototype. Figure 26 shows a cleaner version of this implementation, without the old components (that won't be developed in the first year) overlapped.

As shown in the picture, for the moment we envisage to merge PHR functionalities with the traditional health services. The option to use an existing PHR service provider (in particular Microsoft Health Vault) was dismissed and we implemented our own PHR.

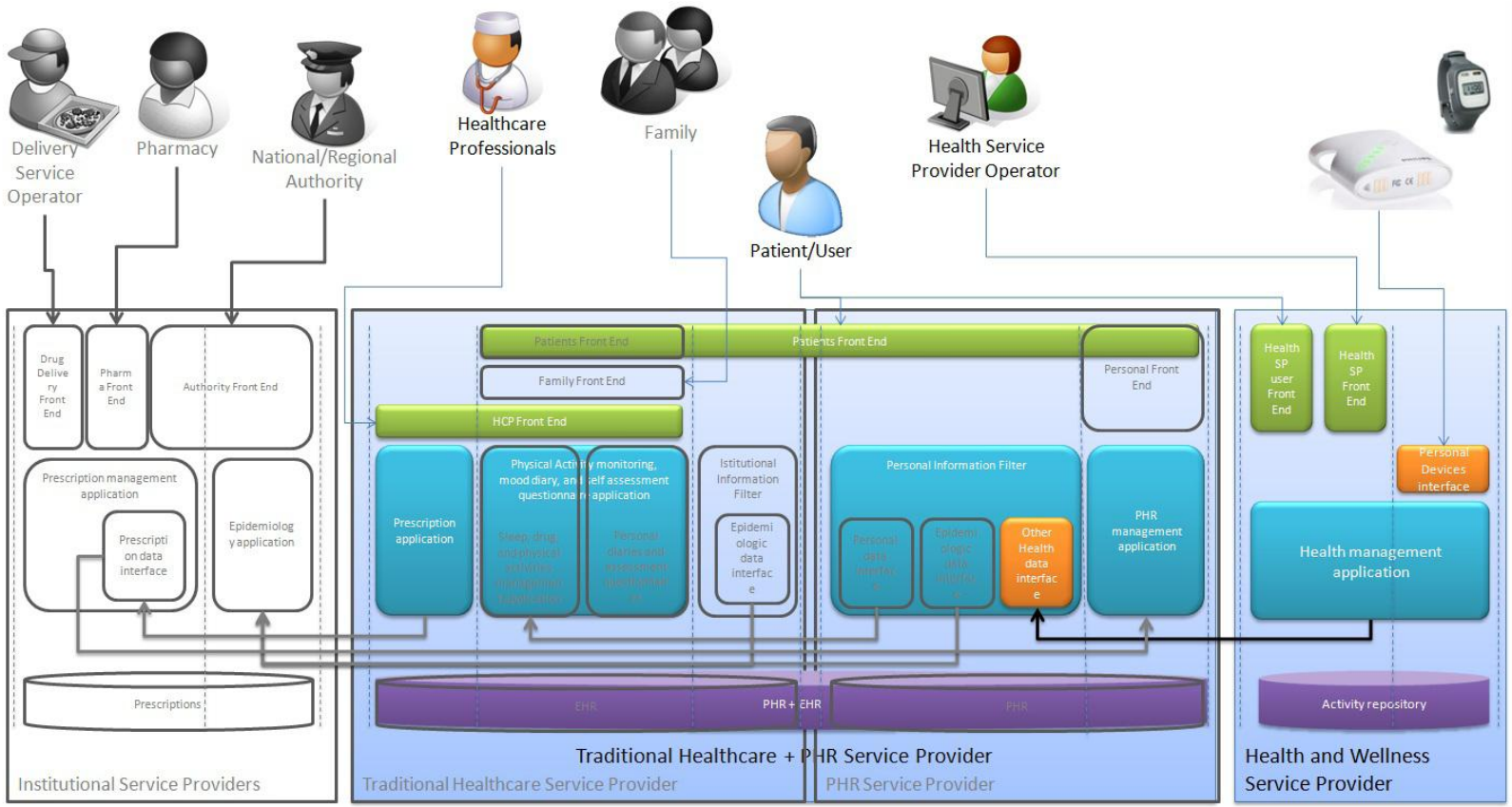

Figure 25 eHealth prototype of the first year architecture

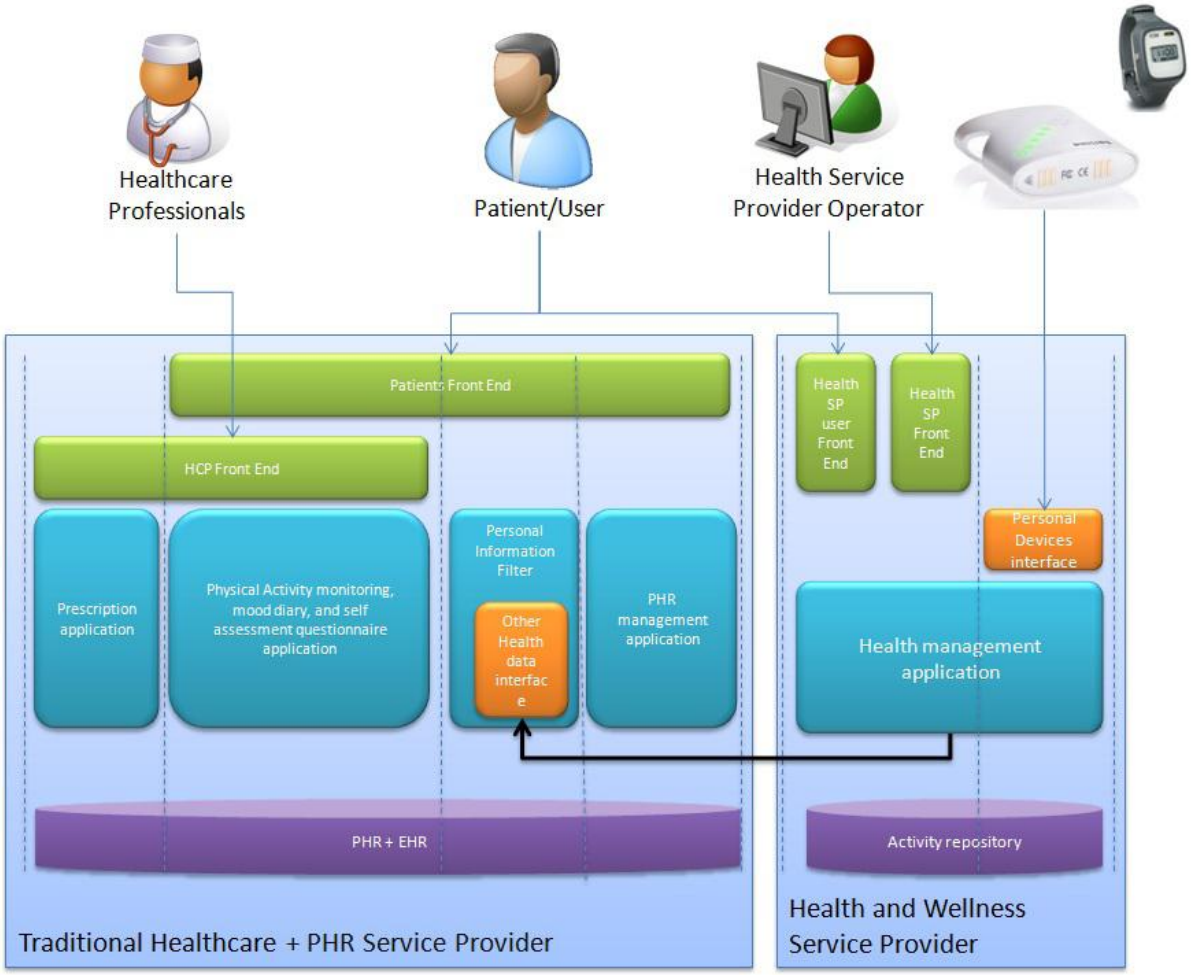

Figure 26 Cleaner version of the eHealth prototype of the first year architecture

Another difference with the original architecture is that the two applications of the Traditional Healthcare Service Provider have been merged in a unique application that will take care of physical activity monitoring, mood diaries and self assessment questionnaires. Furthermore all the components needed to make Traditional Healthcare Service Provider and PHR Service Provider have been of course removed, as they will be running on a unique provider in the prototype. Finally we can highlight the fact that, for the first year, Institutional Service Providers will not be included in the prototypes. Figure 27 shows a more detailed view of the possible realization of the mock-up:

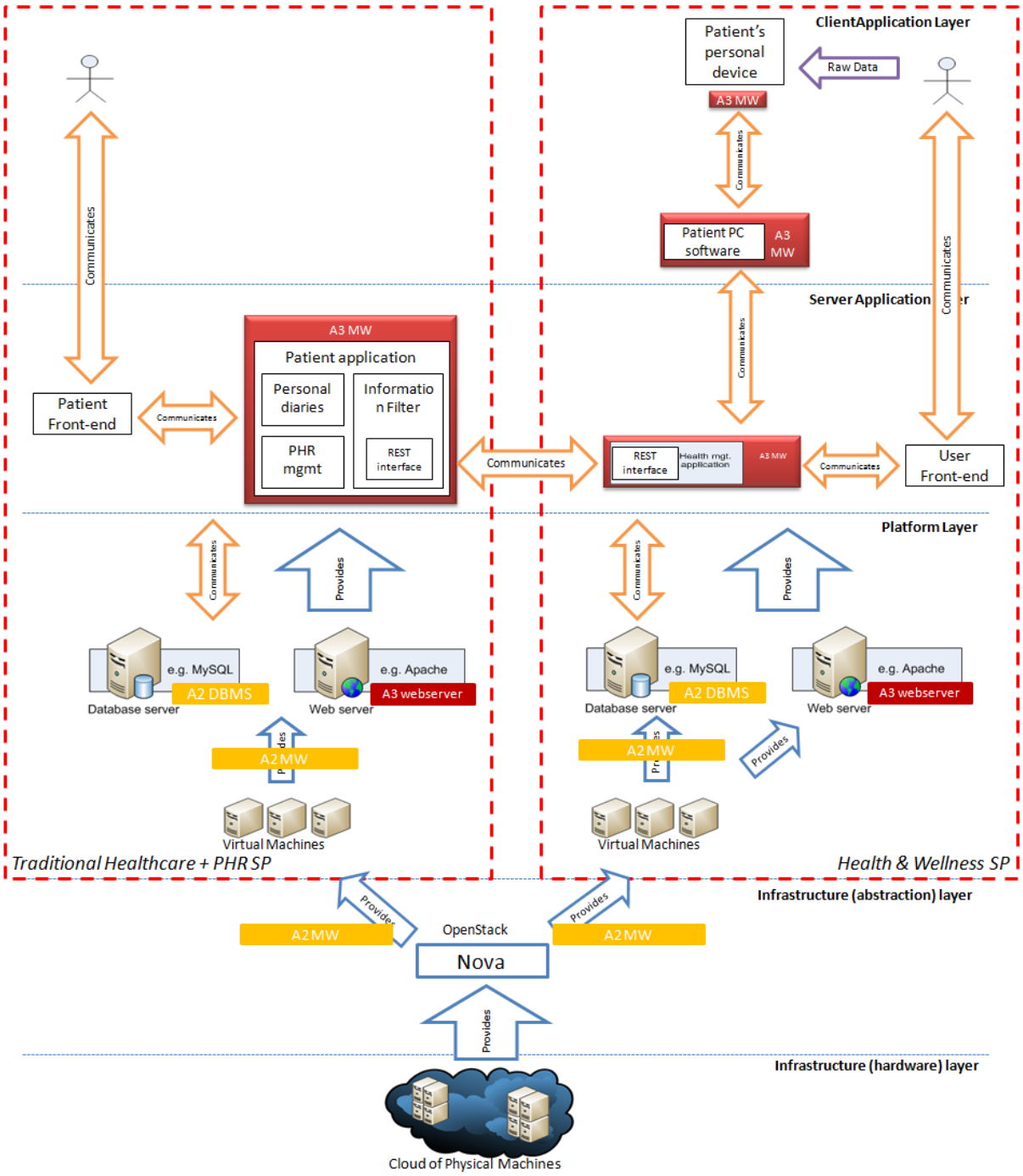

Figure 27 Architecture of the first year prototype and chosen technologies

The blue cloud below the picture represents the physical machines that provide the physical infrastructure for the cloud installation, while the layer above illustrate the commodity cloud that was selected by A2, namely OpenStack. For the first prototype we will use off-the-shelf cloud, or just with minimal integration of early results from A2. The red dashed box on the left, encapsulating all the other components, represents the Virtual Machine that will run the TH+PHR Service Provider application. In this case the Service Provider will be San Raffaele Hospital (HSR), and from now on we will refer to this SP as the Hospital, San Raffaele, or HSR. The OS will be the Linux system Ubuntu, most likely in the 10.04 version, with Java Run Time Environment.

In this machine, we will use TomCat as Application Server, and MySql as Database. This choice was suggested by A2, as from a preliminary analysis will ensure that the system is deterministic and thus it's very likely that it can be replicated.

The Patient application box represents the Java applications that will run on Tom-Cat. One particular application will be the back-end management application, in charge of the interface with the database. All the other applications will use the functionalities offered by this software layer to access the MySql database.

As far as concern the front ends, JSP technology supported by TomCat will be used. Two user interfaces will be realized: 1) a simple portal for the patient, to monitor her personal data, fill her mood diaries and self-assessment questionnaires, and manage her privacy settings; 2) a simple portal for the psychiatrist that will be able to monitor patient's data as well and to save simple prescriptions.

The application architecture follows the Model-View-Controller (MVC) paradigm, where Model is given by MySql Database and by the back-end management application, View is given by the Patient Front-end box and the Controllers are the boxes inside the Patient application. In orange we highlighted the communications, in particular the web services that will allow the Health and Wellness Service Provider to upload physical activity data to the PHR.

User Requirements

These user requirements are the functional requirements that this simplified proof-of-concept will adhere to.

1. The server-side programs must be run on a virtual machine running in a NOVA (Openstack 2011) computing cloud. To leverage the scalability offered by the cloud, the business processes should be decoupled as much as possible (i.e. allowing service logic to be ran on separate virtual machines).
2. The subscriber can log in and log out on the web portal with username/password.
3. An authenticated subscriber of the web portal can view his aggregated data.
4. An authenticated subscriber can register a device serial code on the web portal.
5. A subscriber can connect his personal device to his computer and set it up for synchronization.
6. A subscriber can connect his personal device to his PC and the data is synchronized from the device to his PC.
7. An authenticated subscriber with a registered device can upload data to the database for her account and her device.

Requirements 5 and 6 are requirements that will be implemented by the software that is delivered together with the personal device (see Section 0).

What specifically is *not* a requirement is the creation of accounts. This will be done manually via manipulation of the database. Due to the limited scale of this proof-of-concept, we will directly implement these user requirements and not do intermediate steps such as specifying detailed software requirements.

5.4.3 Health and Wellness Service Provider Instantiation

The sections below describe a trimmed down version of the proof-of-concept healthcare application for the domain of Health & Wellness Service Provider. It is a very simplified version of the proof-of-concept focusing only at the core functionality. This version is provisional, and it will be rewritten in later stages of this project.

Architecture and design decisions

The following design decisions (considerations) are the most prominent.

1. Client-server architecture is necessary. The server will aggregate the data, while the client must facilitate the upload of the data from the personal device to the server.
2. The server must facilitate two different communication channels. One for the personal device, and one for the subscriber's web browser to view aggregated data and register his or her device.
3. In order to facilitate the binding of a device to a subscriber, the demo will need to support multiple users.
4. We choose to make the web interface trivial. No in-depth functionality or customization, but a very functional front-end connecting to the data store. For rapid and secure development a python web framework will be used.
5. We choose to make the interface for the personal device in a formalized, structured way adhering to a Service Oriented Architecture design principle. In particular, we choose a SOAP solution for this.
6. Because of the SOA design, the web portal will also need to authenticate to the middle-tier and utilizes those services instead of directly querying the database.
7. We choose Python as cross-platform, free and high-level programming language to fulfil the tasks.
8. We use the Model-View-Controller paradigm to depict each of the three major entities.

Some notes on these design decisions follows. In design decision #5, the choice for a Service Oriented Architecture is stated. SOA is very akin to the distributed nature of cloud computing and TClouds in particular. By decoupling functionalities into services, it becomes possible to distribute business processes over multiple virtual machines, and thus leverage the scalability offered by the cloud computing.

SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services.

SOAP is the XML-based protocol that allows for web services message interchange and is the *de facto* choice when designing with SOA. It promotes interoperability, consistency and portability.

Concerning design decision #7, where many candidates exist, Python was chosen. For a thin client such as here, cross-platform is relatively easy and cheap compared to a fat client. Therefore using an open language (instead of for instance .NET) is an advantage. Furthermore, because this program has to interact intensively with a server using XML, a lightweight and high-level programming language like Python is preferred over a low-level language such as C/C++. Accessing the Universal Serial Bus (USB) is also possible from

within Python using libraries pyusb (PyUSB 2011), libusb (Libusb 2011) and libusb-win32 (LibWin 2011), if it would prove necessary to do so. The decision of Python over Java is slightly arbitrary, since both languages are equally well suited for the task. However, OpenStack is written entirely in Python, which is an advantage if our program would have to interact with OpenStack directly.

In design decision #4 we mention web2py as a python web framework to use. The advantages of using such a framework are some features which would normally take a lot of effort when done manually, for instance, escaping user input (against SQL injection and XSS), allowing easy templating (separation of design and presentation) and session management.

Finally, design decision #8 is a conventional one. The Model-View-Controller paradigm (Wikipedia 2011) allows us to abstract the interface from the data storage and is an established methodology. From Wikipedia (2011):

> The model manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller). In event-driven systems, the model notifies observers (usually views) when the information changes so that they can react.
>
> The view renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes. A viewport typically has a one to one correspondence with a display surface and knows how to render to it. The controller receives user input and initiates a response by making calls on model objects. A controller accepts input from the user and instructs the model and viewport to perform actions based on that input.

It will require some adaption to merge the MVC paradigm with the SOA architecture i.e. some components might be bare bone), but the essence of MVC remains visible: the model, view and controller component will be replicated on each of the three major entities (subscriber, portal and middle-tier).

Components

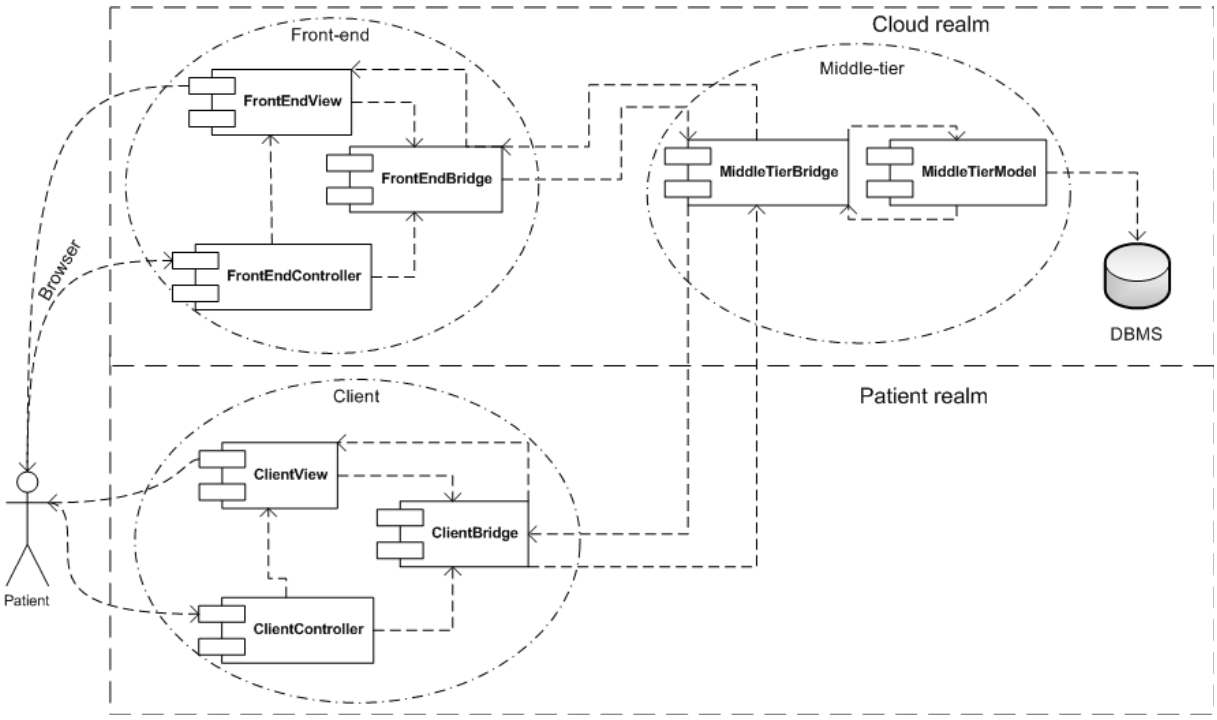

Figure 28 Components diagram

Figure 28 shows the relation between the components. As visible, the MVC, SOA and client-service architectures have been reflected in this design. Three 'bridges' have been introduced to merge MVC with SOA – at these points the SOAP interfaces play a definite role and entail the encapsulation and serialization steps.

Below we will discuss the three components in more detail. We will from now on refer to the front-end, middle-tier (back-end) and subscriber's PC-software as their component names, namely `FrontEnd`, `MiddleTier` and `Client`.

The decomposition of the three major component packages is briefly introduced.

- `FrontEndView`: generates the HTML for the subscriber's browser;
- `FrontEndController`: the events generated by the subscriber's browser (e.g. clicking links) are handled here;
- `FrontEndBridge`: a binder for the two previously mentioned components to be able to communicate with the model, and abstracts from the SOAP layer;
- `MiddleTierModel`: stores and queries the data from the DBMS;
- `MiddleTierBridge`: exposes the data model via its services;
- `ClientView, ClientController`: together form the GUI for the subscriber, with a simple synchronization daemon which for example runs in the system tray;
- `ClientBridge`: the data from the subscriber's device has to be transferred into the cloud, the bridge makes it possible with a binding

The middle-tier

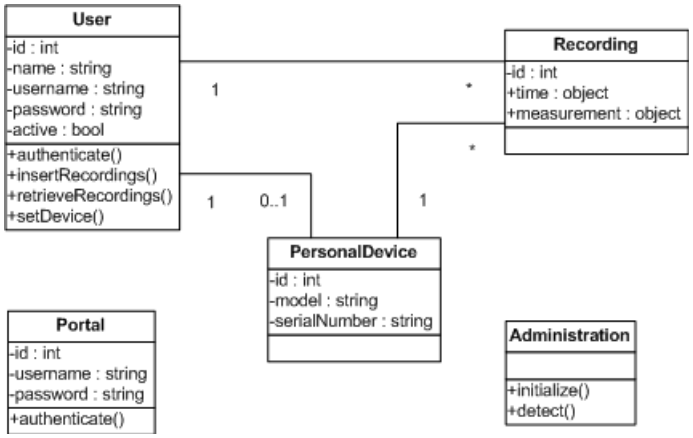

Figure 29 Middle-tier `MiddleTierModel` class diagram

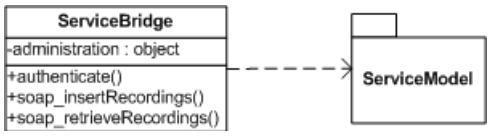

Figure 30 Middle-tier `MiddleTierBridge` class diagram

Figure 29 shows the class diagram for the `MiddleTierModel` middle-tier component.2 In this figure, linking both `PersonalDevice` and `Recording` to `User` allows for a flexible approach in which personal devices can be disconnected from a subscriber and be replaced. The `Portal` entity depicts consumers of the SOA service, e.g. the web portal. Although the web portal retrieves data on behalf of subscribers, the web portal has to be authenticated as well, not only for auditing, but also to execute privileged functions (such as querying subscriber data).

The `Administration` entity is a class which allows administrative operations to be executed. Whether this class finds its way into the final version is unsure. This class is not exposed to the client or the front-end, and it has methods to print the data in the tables to the screen and cleaning the tables and setting up dummy data.

As stated before, we will use a SOAP interface. Authentication in SOAP is generally done per message. This gives statelessness in between connections. This is however undesirable for front-ends, because otherwise the subscriber would have to supply his credentials in every web-page, so the web portal will abstract from that design in the front-end using a session system.

Figure 30 shows the bridge interface for the subscriber. In Python we can set up a SOAP handler which will wrap around the functions exposed in the `MiddleTierModel`.

The front-end

The front-end (i.e. portal) reroutes all requests to the middle-tier. Hence, it needs to generate markup for the subscriber's interface (a browser), and for convenience it must maintain sessions with the subscriber's browser.

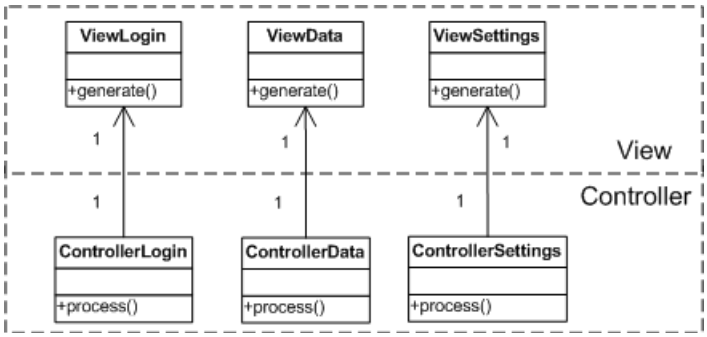

Figure 31 Front-end `FrontEndView` and `FrontEndController` class diagram

The diagram in Figure 31 contains both the View and Controller components. The reason is that these two components are intertwined. A subscriber sends his browser requests to the Controller, and gets his/her HTML from the View. The class names must be read as pages, viz.: `Login`-page, `Data`(-overview)-page and (user-)`Settings`-page.

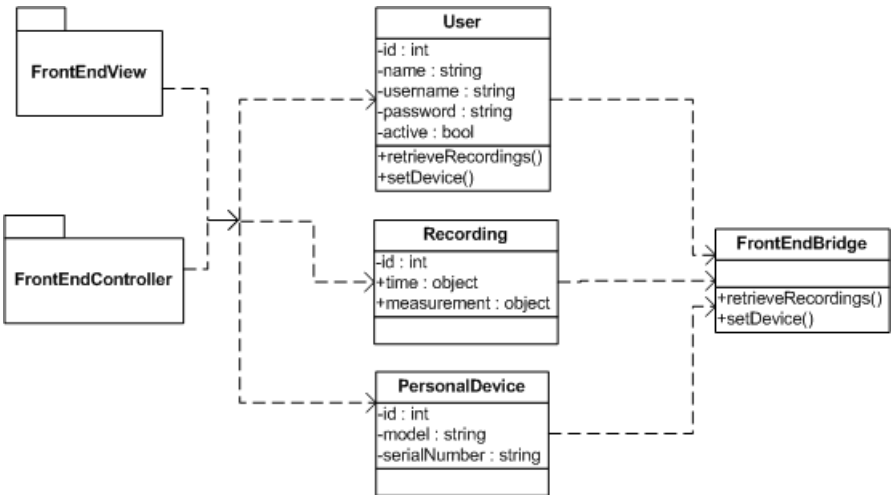

Figure 32 Front-end `FrontEndBridge` class diagram

Figure 32 shows the class diagram for the front-end bridge component. As mentioned above, the front-end relays all requests to the middle-tier. An important feature of this design is that the `FrontEndBridge` component has a derived copy of the model found in `MiddleTierModel`. This is because it is desired that front-end and client can operate on the data model as transparently as possible. The actual implementation of the methods found in the User entity differs from the one in the middle-tier. While in the middle-tier the methods use queries on the database, the exact same method here uses a Python SOAP binding via the `FrontEndBridge`.

The client

Philips Respironics (Philips 2011) is a company that offers personal devices to monitor the activity, rest and light-exposure patterns of patients with its ActiWatch (Figure 33) product line. The ActiWatch that is used in this proof-of-concept is the ActiWatch Spectrum (Philips 2009), which is the most functional device from the Philips Respironics ActiWatch family.

Figure 33 ActiWatch

We are restricted to the use of the Actiware software, this means that accessing data from the device has to be done via the Actiware software. We know that it is possible to export data from the Actiware software to a file. This functionality is vital for this mockup, because this will function as the bridge between the closed/blackbox workings of the Actiware software with our client-side application that pushes this data to the cloud.

The software manual describes the export processing:

> Once you have added intervals and analyzed your data, you can export the retrieved raw data or statistical results to a text file that can be easily loaded into Microsoft Excel, FAST, or a database application of your choice for additional processing. (Respironics 2009)

This export process is depicted with the screenshot in Figure 34.

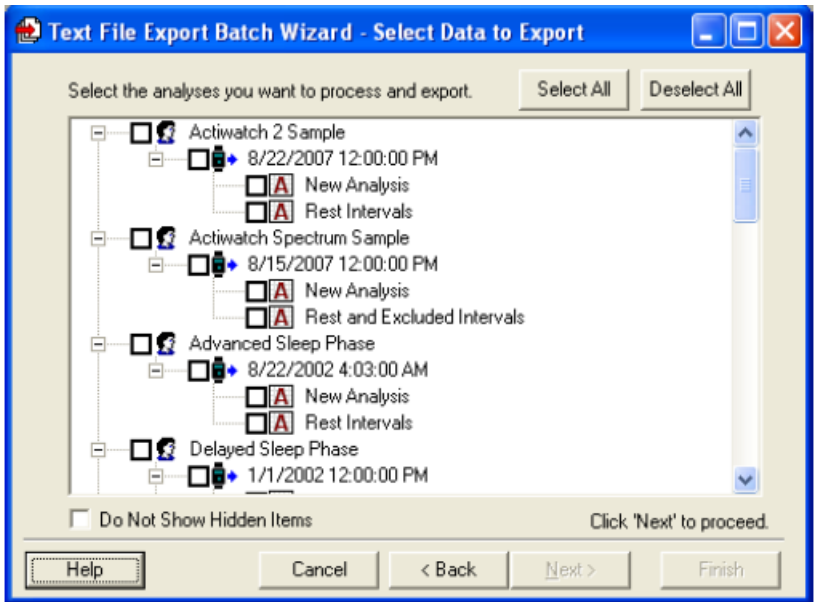

Figure 34 Export wizard

WSDL

While not a component on its own, the Web Services Description Language file plays an important role. The application is developed by first defining the .wsdl file, and from that file templates and boiler-plate code is generated for the Python middle-tier, front-end and client. Specifically, their bridges use this template code. This WSDL file allows discovery of our web

services and is essential for a proper SOA design. In the future, different languages and clients can access the middle-tier in addition to the set-up described above.

In our WSDL file we defined (for now) two methods: `insertRecordings` and `retrieveRecordings`. They both are of the type `recordings`, which are defined as in the following excerpt.

```
<!--custom types -->

<complexType name="recordings">

 <sequence>

      <element name="recording" type="finType:recording"
      minOccurs="0" maxOccurs="unbounded"/>

 </sequence>

</complexType>


<complexType name="recording">

 <sequence>

  <element name="id" type="xsd:integer" minOccurs="0"
 maxOccurs="1"/>

  <element name="time" type="xsd:dateTime" minOccurs="1"
 maxOccurs="1"/>

  <element name="measurement" type="xsd:float" minOccurs="1"

maxOccurs="1"/>

  <element name="PersonalDevice" type="xsd:integer"
  minOccurs="1" maxOccurs="1"/>

 </sequence>

</complexType>
```

Indeed, this is an array of recordings, and for each recording we defined some properties. This array is thus used both in inserting recordings as well as retrieving recordings.

5.4.4 TH+PHR Service Provider Instantiation

Architecture and design decisions

The following design decisions are relevant for the TH+PHR Service Provider:

1. Client-server architecture will be implemented, where the client will be represented by the users' browser, while the server will be a Virtual Machine (VM) running on NOVA (OpenStack).
2. In the server, we will use TomCat as application server and MySql as database.
3. The application will be written in Java, as cross-platform, free and high-level

programming language.

4. Spring was selected as Java Framework.
5. Overall the application will implement the Model-View-Controller paradigm.
6. The database and the middle-tier layer will run on the same VM.
7. The server will open three communication channels. One will be the web front-end toward the users (both patients and healthcare professionals). The second will be an input channel opened toward Health and Wellness Service Providers. The third will be a communication channel toward the user, but mediated by other service providers (e.g., SMS Service Provider, Email servers, etc.).
8. The interface toward the Health and Wellness Service provider will follow Service Oriented Architecture (SOA) design principles, and in particular the SOAP protocol will be used.

The choices #1, #2, #3 and #6 are dictated by the needs communicated by Acitvity2. In particular A2 requested for the first year implementation a very simple solution and set of services realized. The Cloud Computing framework selected by A2 is OpenStack, which computing component is called NOVA (design choice #1). Encapsulating both back-end and middle-tier (and of course front-end) in a single Virtual Machine (design choice #6) will probably create some problems to the scalability of the application. On the other hand, this will ensure something simple and easy to test, as requested by A2. Furthermore the first year implementation will be just a mock-up, which purpose is not to replicate a production system. Finally, also the choice #2 is the consequence of A2 guidelines, because TomCat and Derby are deterministic systems, and they will ensure replication of the VM, for reliability. Choice #3 is a consequence of choice #2, as Java is the most obvious language when using TomCat.

The decision #4 is taken to avoid spending efforts in the development of commodity features that may be already available in the OpenSource community, and to be able to concentrate the efforts only in the new developments relevant for TClouds and for the offered services in the home healthcare scenario. Decision #5 has already been motivated in the previous Section, and it adapts very well with the architecture designed in the previous Sections, in particular with the division in backend (model), middle-tier (controller) and front-end (view).

Components

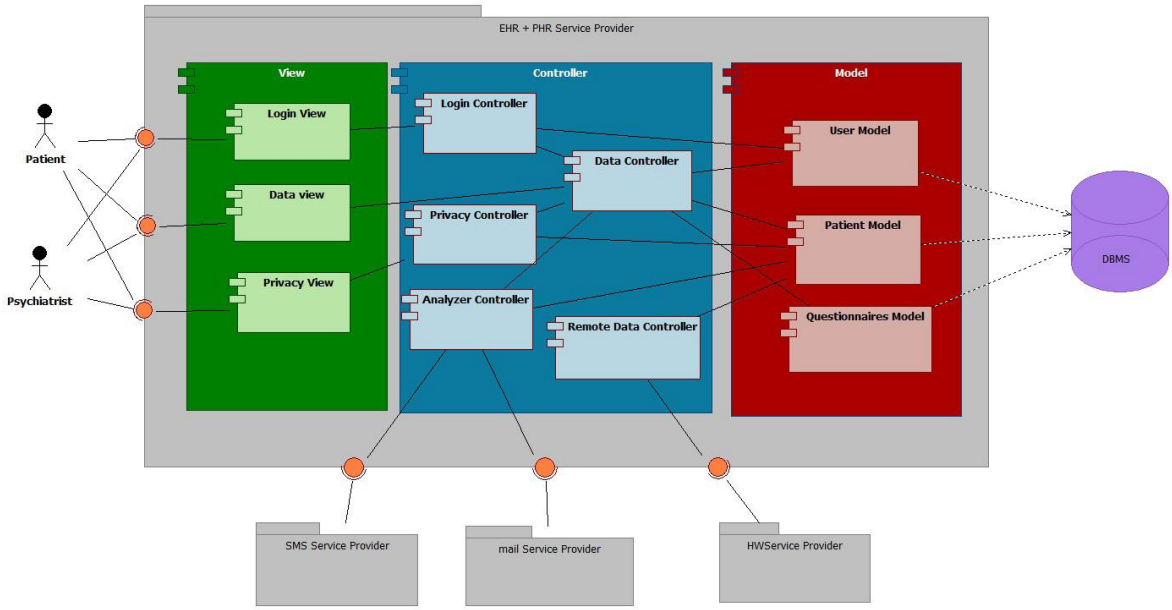

Figure 35 TH+PHR Service Provider component diagram

The component diagram (Figure 35) is designed according to the MVC pattern. In particular, it represents the connections from the TH+PHR Service Provider toward physical persons (like patients or psychiatrists) or Service Providers. The orange balls indicate the interfaces, through which it's possible to communicate, in both cases. In particular, there are three SP in the figure: two of them are used to send alarms with SMS or mails, and the third allows the HWSPs to insert data from the patient's devices. The red part, that represents the Model in the MVC paradigm, is connected to the DBMS and is responsible to receive and send data to it.

Below is short introduction for every component included in the TH+PHR Service Provider.

- `Login View`: creates the web page for the user's browser that permits the login to the application.
- `Data View`: through the specific interface, it allows the patient to manage her personal data or the psychiatrist to control his patients, generating the correct HTML code.
- `Privacy View`: consents to create the specific code to allow the patient to set the privacy permissions. This View communicates with the Login Controller that takes care of the procedure logic.
- `Data Controller`: handles all the events regarding the patient data. Furthermore it communicates with others controllers: the Login Controller, to receive the connection event, the Analyzer Controller, to add the alarms data to the Data View, and the Privacy Controller, to check privacy settings according to the events managed by the Data Controller.
- `Login Controller`: is used to manage the login events, with this component it is possible to control the access to the application and to the data (combining its functionality with the Data Controller).

- `Privacy Controller`: used to apply the privacy settings, in fact this controller is able to handle all the events concerning the personal data treatment.
- `Analyzer Controller`: is the handler for the alarm procedures. If there's an alert, the controller transmits the event to the SMS and/or mail Service Provider (using the specific interface). This component can communicate also with the Data Controller to send information about the alarms.
- `Remote Data Controller`: allows to manage data from external Service Providers, like Philips, communicating through the dedicated interface, and forwarding data to the patient model, that sends the information to the DBMS.
- `User Model`: used to communicate portal user's data to the DBMS. It's used by the Data Controller and it's also fundamental for the Login service (through the Login Controller).
- `Patient Model`: stores and queries information from the DBMS. This data regards the patient and the thresholds set by the doctor for the monitoring of the specific patient.
- `Questionnaires Model`: exchanges questionnaires information with the database.

Chapter 6

Preliminary Architecture of the application middleware

Chapter Authors:

Imad Abbadi (UOXF), Mina Deng (PHI), Marco Nalin, Ilaria Baroni (HSR)

6.1 Introduction

Cloud infrastructure is expected to support Internet scale critical applications (e.g. hospital systems, smart grid systems). Critical infrastructure will not outsource their IT on Cloud without strong assurance about its trustworthiness. Therefore, establishing trustworthy Cloud infrastructure is the key factor to move critical resources into the Cloud. In order to move in this direction for such a complex infrastructure, we virtually split Cloud infrastructure into layers.

1- **Physical Layer** --- This layer represents the main physical components and their interactions, which constitute Clouds' physical infrastructure. Example of these includes physical servers, storage, and network components. The physical layer resources are consolidated to serve the *Virtual Layer*.

2- **Virtual Layer** --- This layer represents the virtual resources, which are hosted by the *Physical Layer*. Cloud customers in IaaS Cloud type interact directly with the virtual layer, which hosts Clouds' customer applications. This layer consists of multiple sub-layers: Virtual Machine (VM), virtual network, and virtual storage.

3- **Application Layer** --- This layer has Clouds' customer applications, which are hosted using resources in the *Virtual Layer*.

Each layer relies on the services and resources provided by the layer directly underneath it, and each layer services relay on messages communicated with both the layer directly underneath it and above it. Each pair of adjacent layers has a specific middleware that provides self-managed services. For example, a *Virtual Layer Middleware* is needed between *Physical Layer* and *Virtual Layer* to provide infrastructure transparent services to virtual layer, and an *Application Layer Middleware* is needed between *Virtual Layer* and *Application Layer* to provide transparent management services to applications. The middleware services' implementations are based on the layer they serve. Different types of middleware services coordinate amongst themselves and exchange critical messages, which are paramount for providing trustworthy Cloud infrastructure.

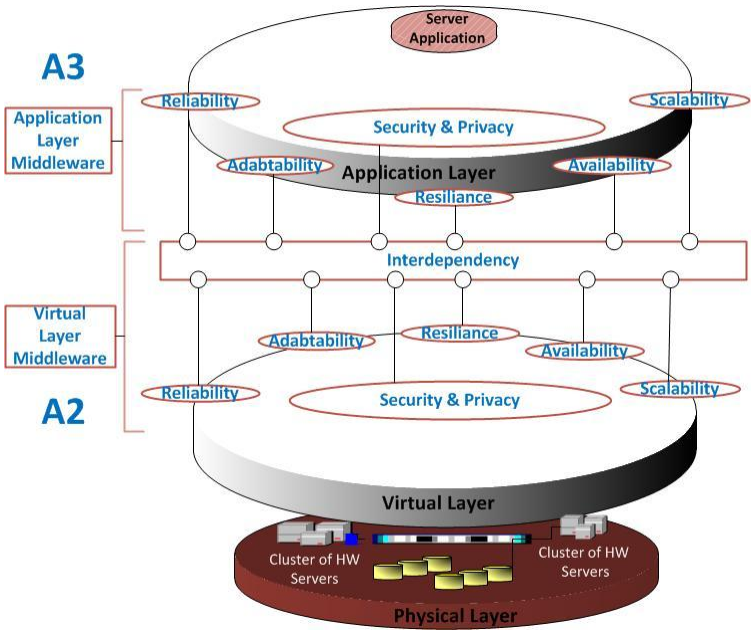

Figure 36 Self-Managed Services in TCloud

Figure 36 illustrates the fine line between A2 and A3 within TClouds project. In this Figure we see A2 will mainly be involved in providing trustworthy Virtual Layer Middleware services. A3, on the other hand, provides Application Layer Middleware services, which are application specific. The A2-A3 Interdependency is an API for shared trust functions between A2 and A3, which are mainly provided by A2 with help from A3.

6.2 Application Layer Middleware Self-Managed Services

Self-managed services for Application Layer are about providing Cloud Application Layer with exceptional capabilities enabling it to automatically manage all applications running on the Cloud, their interdependencies, and take appropriate actions on emergencies. Self-managed services are concerned about supporting availability, reliability, scalability, resilience, and adaptability of Clouds resources. These services must provide security and privacy by design. In this section we provide a set of conceptual models for these services. Section 7 discusses the main services which are of interest to home healthcare application for TClouds project.

6.2.1 Adaptability

Adaptability is the ability of the application to provide timely and efficient reactions on system changes and events. *Adaptability* should always consider the overall system properties are preserved (e.g. security, resilience, availability and reliability) when taking an action. The *Adaptability* service at application layer is concerned with ensuring that the system responds in time to changes and events in the end-to-end application environment. The service should automatically decide on an action plan and then manage it by coordinating with other services in the same layer or other layers.

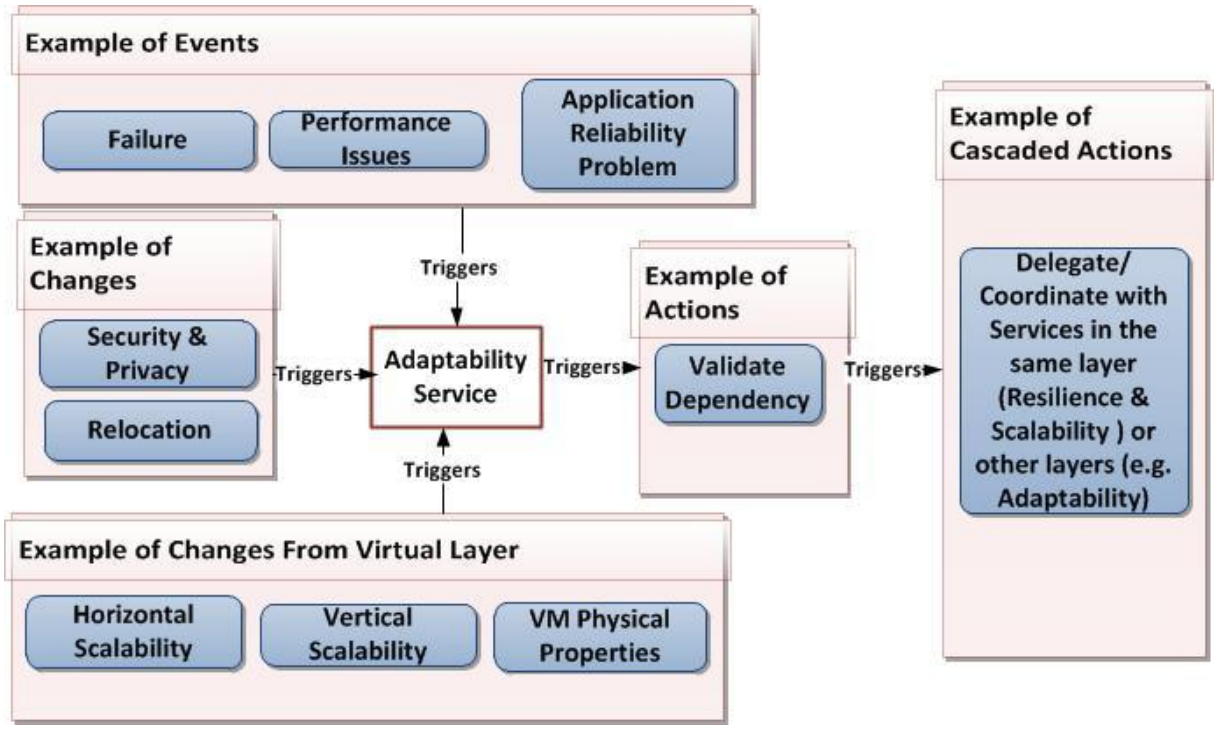

Figure 37 Adaptability Service

Figure 37 provides a conceptual model for *Adaptability* service's functions. This Figure provides examples of *Events* and *Changes*, which triggers the *Adaptability* service. The *Adaptability* service in turn takes *Actions* based on the *Events* or *Changes*. The *Actions* also generate *Cascaded Actions* to other services in both *Application Layer* and *Virtual Layer*. The *Adaptability Service* follows a set of rules defined by cloud authorised employees defining the *Actions* and *Cascaded Actions*.

6.2.2 Resilience

Resilience is the ability of a system to maintain its features (e.g. serviceability and security) despite a number of sub-system and components failures. High resilience can be achieved by providing redundancy together with careful design (eliminating single points of failure) and well planned procedures. *Resilient* design helps in achieving higher availability and end-to-end service reliability, as its design approach focuses on tolerating and surviving the inevitable failures rather than trying to reduce them. The *Resilience* service communicates with other services to collaborate in providing end-to-end resilient Cloud.

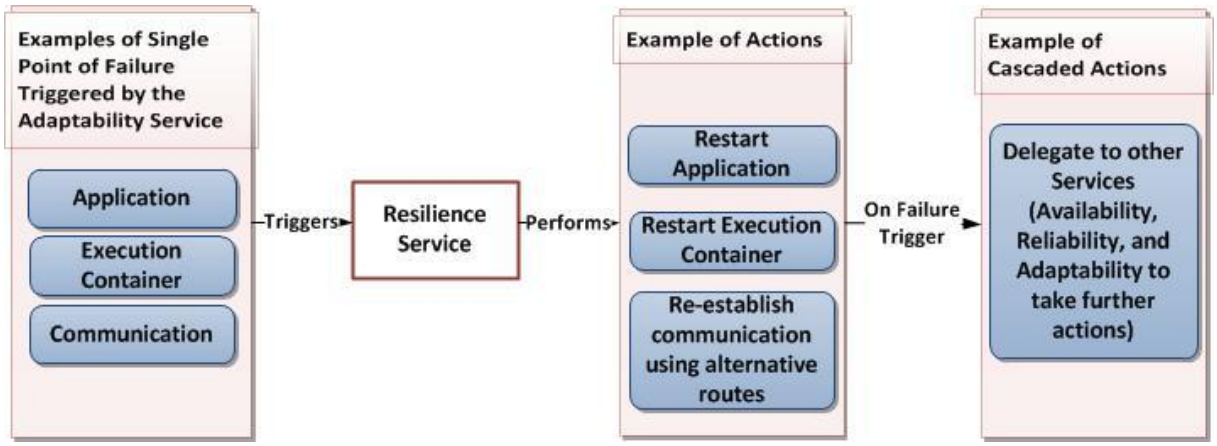

Figure 38 Resilience Service

Figure 38 provides a conceptual model for *Resilience* service functions that should be provided to maintain the overall end-to-end application resilience. This Figure provides examples of *Single Point of Failure*, which triggers the *Resilience* service. As we see in the Figure the adaptability service first receives a notification of *Single Point of Failure events,* and then it manages them. This management would include interacting with other services, as the resilience service.

The *Resilience* service in turn *Performs Actions* based on the *Single Point of Failure*. If the *Actions* failed to guarantee resilience the Figure provides examples of *Cascaded Actions* that are followed. Such *Actions* and *Cascaded Actions* follow a set of rules defined by Clouds' authorised employees.

6.2.3 Scalability

Scalability at the Application Layer is providing an application with capabilities to quickly and efficiently adapt to the addition and removal of virtual resources at virtual layer. For example, on peak periods the virtual layer scales resources up, and similarly on off-peak periods the virtual layer should release unneeded resources. These should be reflected at the application to support the addition and removal of virtual resources. Also, these should not affect fundamental system properties and should always represent user requirements (e.g. security and privacy). The *Adaptability* service at the Virtual Layer upon detecting a need for either adding resources (e.g. peak period) or removing resources it instructs the *Scalability* service at the Virtual Layer to do so. This service (i.e Virtual Layer Scalability) should trigger the *Adaptability* service at application layer to adapt to changes in the Virtual Layer. The Adaptability service at the Application Layer then triggers the Scalability service at the application layer to scale the application to adapt to such changes.

Scalability type at virtual layer can be: *Horizontal Scalability*, *Vertical Scalability*, or combination of both. Horizontal Scalability is about the amount of instances that would need to be added or removed to a system to satisfy increase or decrease in demand. Vertical Scalability is about increasing or decreasing the size of instances themselves to maintain increase or decrease in demand. Application layer scalability reacts differently to both types of scalability. For example, horizontal scalability means the application will be replicated at the newly created VMs; however, vertical scalability means the application needs to take advantages of the additional allocated resources (e.g. increase memory usage, spawn additional child processes). Also, in both cases the scalability process needs to notify the *Availability* and *Reliability* services.

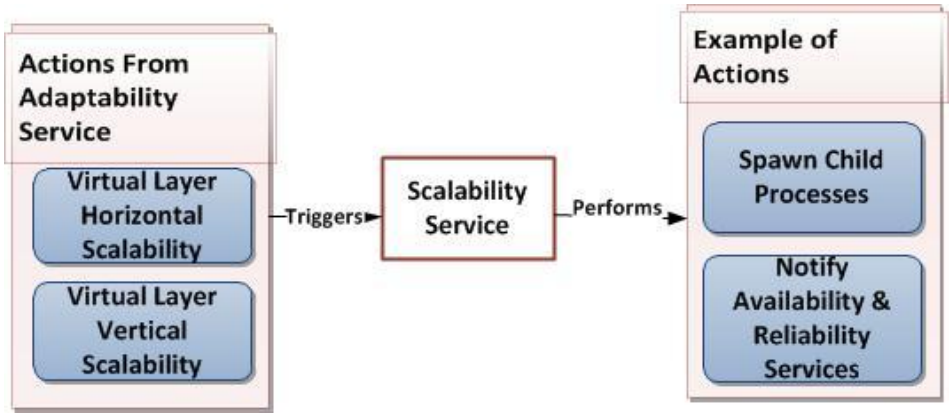

Figure 39 Scalability Service

Figure 39 provides a conceptual model for application scalability service. This Figure provides the *Actions* from *Adaptability* service that triggers the *Scalability* service. The *Scalability* service in turn takes appropriate *Actions*. As discussed above, the actions trigger other services in Application Layer.

6.2.4 Availability

Availability service represents the relative time a service provides its intended functions. High levels of availability are the result of excellent architect, which considers well crafted procedures, redundant services, and high service reliability; i.e. resilient design.

Availability service at application layer is in charge of distributing requests coming to an application across all redundant application resources based on their current load. If a resource is down or it is relatively overloaded, the *Availability* service should immediately stops diverting traffic to that resource, and re-diverts traffic to other active resources until the *Adaptability* service fixes the problem or until the overloaded resource returns to normal processing capacity.

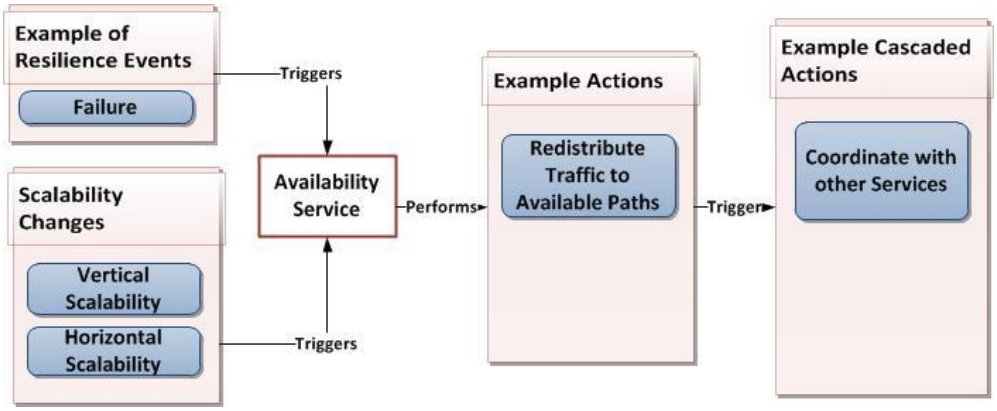

Figure 40 Availability Service

Figure 40 provides a conceptual model for application *Availability* service. This Figure provides examples of *Events* from *Resilience* and *Changes* from *Scalability* service, which

triggers the *Availability* service. The *Availability* service in turn takes *Actions* based on the *Events* and *Changes*. The *Actions* also generate *Cascaded Actions* to other services in both *Application Layer* and Virtual Layer.

6.2.5 Reliability

Reliability is related to the success in which a service functions. High end-to-end service reliability implies that a service always provides correct results and guarantees no data loss. Higher individual components reliability together with excellent architect and well defined management processes, help in supporting higher resilience. This in turn increases end-to-end service reliability and availability.

Reliability is of higher priority than Availability service. Most importantly it ensures that the end-to-end service integrity is maintained (i.e. no data loss and correct service execution). If service integrity is affected by anyway and cannot be immediately recovered, *Reliability* service then notifies the *Availability* service to immediately bring a service or part of a service down. This is to ensure that data integrity is always protected. Simultaneously, *Adaptability* and *Resilience* service should automatically attempt to recover the system and notifies system administrators in case of a decision cannot be automatically made (e.g. data corruption that requires manual intervention by an expert domain administrator).

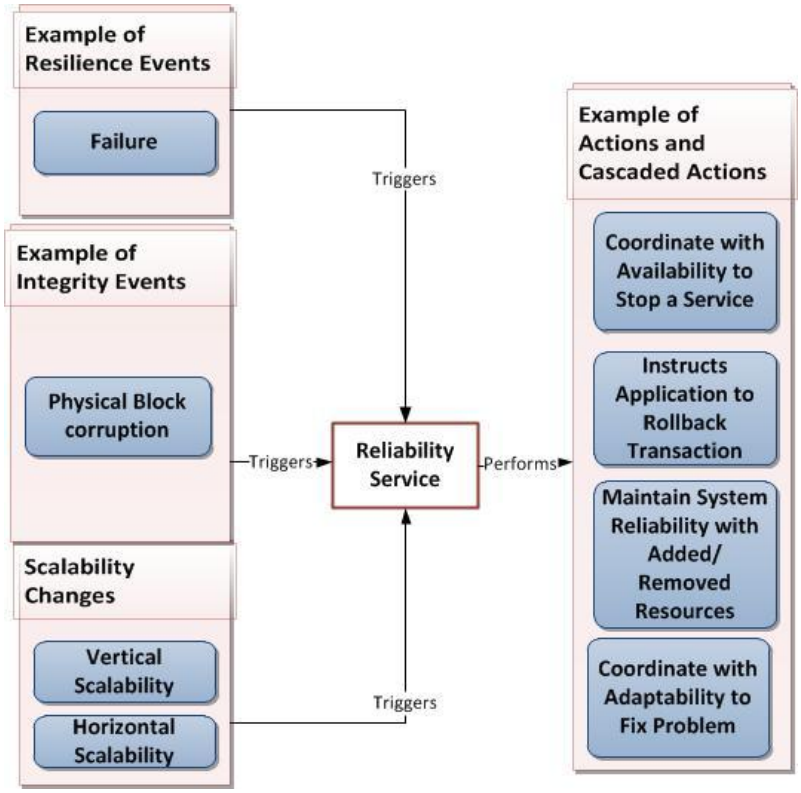

Figure 41 Reliability Service

Reliability Service provides a conceptual model for application *Reliability* service. This Figure provides examples of *Events* from *Resilience*, *Events* from Virtual Layer Services, and *Changes* from *Scalability* service, which triggers the *Reliability* service. The *Reliability* service in turn takes *Actions* based on the *Events* or *Changes*.

6.2.6 Security and Privacy

TClouds project is particularly interested in security and privacy “by design”. Security and privacy would need to be integrated with various services at different Cloud layers. In this report we focus on security and privacy at application layer, which is about ensuring Cloud user security and privacy requirements are maintained by the environment surrounding the application (It is important to re-stress that we are covering the middleware services supporting the application and not the application itself). This section provides an overview of these, and more details are provided in Chapter 7. Security and privacy a application layer, for example, include the following (a.) protecting Clouds user data whilst in transit (transferred to the Cloud and back to the client, and transferred between Cloud structural components), (b.) protecting data whilst being processed by application, (c.) protecting the data whilst being transferred across Cloud services, (d.) protecting data whilst in storage, and (e.) ensuring that the application runs at a pre-agreed geographical location and also data stored at pre-agreed geographical location. Security and privacy should be built into all other services by design.

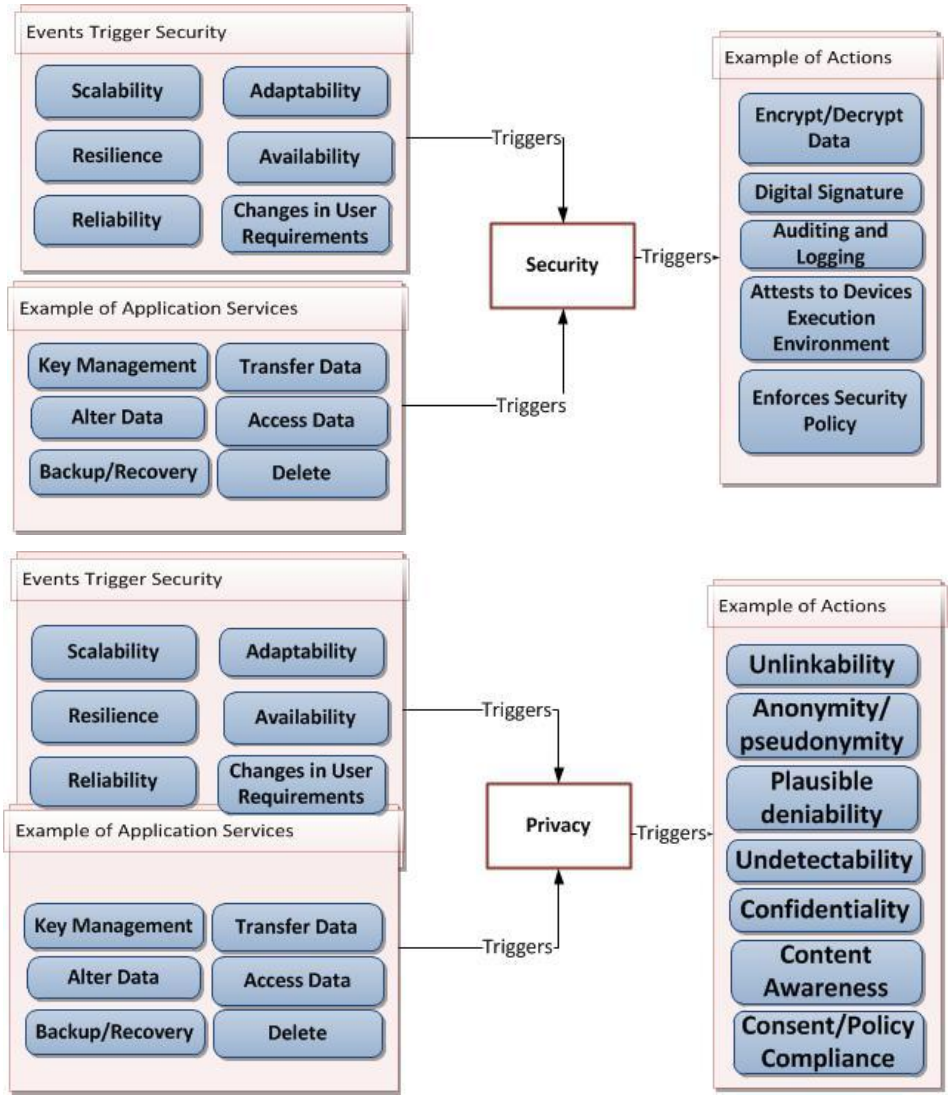

Figure 42 security and privacy service

Figure 42 provides a conceptual model of *Security and Privacy* at Application Layer. This Figure provides examples of *Events* and *Application Services*, which trigger the *Security and Privacy* service. The *Security and Privacy* service in turn takes *Actions* based on the *Events* or *Application Services*.

6.2.7 Services Interaction

Figure 43 provides a summary for the interaction amongst Application Layer middleware self-managed services, as we discuss throughout this section. This Figure provides a high level overview and it is meant not to cover deep details for clarity. In this Figure, the *Adaptability Service* acts as the hart of self-managed services. For example, it intercepts faults and changes in user requirements, manages these by generating action plans, and delegates action plans to other services. To be in a position to do this, the *Adaptability Service* communicates with *Resilience Service*, *Scalability Service*, and *Reliability Service*.

The *Resilience Service* requires having redundant resources, which is represented by relation *Maintains* on *Redundancy*. Excellent resilient design results in higher availability and reliability. This is indicated using *Supports* relation between *Resilience Service* with *Availability Service* and *Reliability Service*.

Scalability Service (based on *Triggers* received from *Adaptability Service*) instructs either *Adapt to Vertical Scaling* and/or *Adapt to Horizontal Scaling* processes. It also *Notifies*} *Availability Service* and *Reliability Service* once scaling is done.

The *Reliability Service* is linked with *Integrity* process using *Must Provide* relation. The outcome of the *Integrity* process is fed to the *Reliability Service*. If application integrity is affected by any way the *Reliability Service* sends an *Integrity Failure* message to both *Availability Service* and *Adaptability Service*.

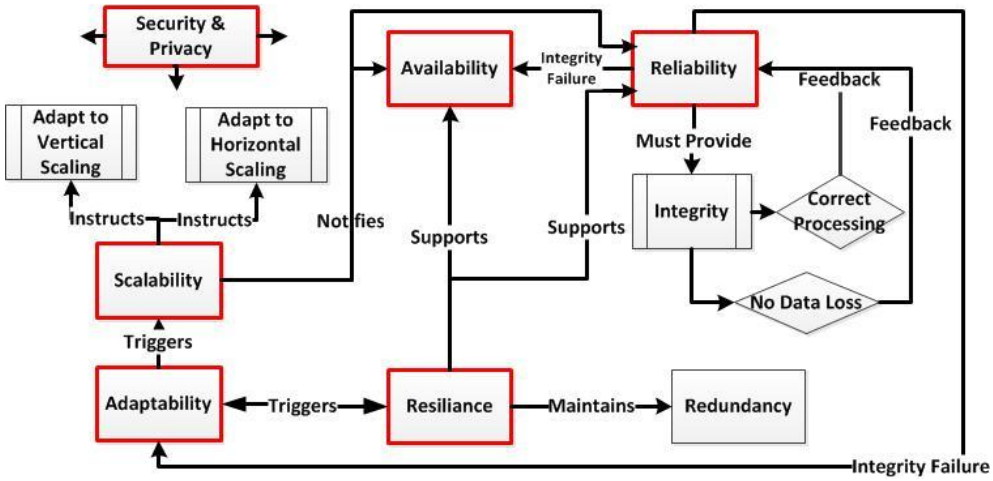

Figure 43 Application layer self managed services interaction

6.3 Services Interaction for Multi-tier Application in the Cloud

In this section we start by proposing a typical multi-tier application architect in Clouds. We then describe how it can be managed using the proposed services' conceptual models. We describe these in context of Home Healthcare application.

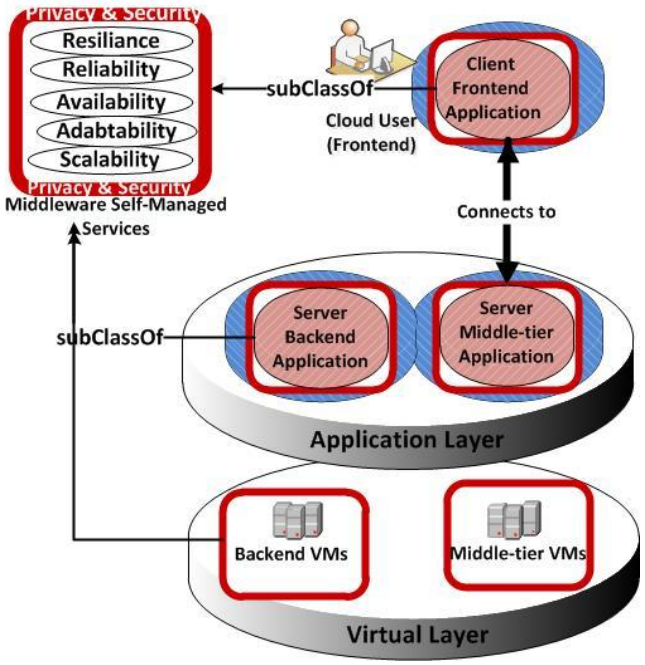

Figure 44 Typical multi-tier application architecture in clouds

6.3.1 Application Architecture in the Cloud

Figure 44 illustrates an architect of a multi-tier application in the Cloud. The Application Layer in a multi-tier architect would typical be composed of the following components.

1. Front-end Application – Generally speaking front-end application could be a combination of HTML, JavaScript, Java Applets, or even a standalone application that would need to communicate with the Cloud for special purposes (e.g. upload data on the Cloud for backup or be part of supply chain application). In home healthcare case, client front-end is split into two main types: (a.) combinations web pages (e.g. HTML/JavaScript) that are used by hospital staff, patients, family members, National/Regional authority, and delivery service operators; and (b) specific application related to patient monitoring devices that regularly uploads patients' monitoring data in the Cloud.
2. Middle-tier Application - Is in charge of running application service logic functions that interact with client front-end application. The middle-tier application runs in an appropriate container (e.g. Apache/Tomcat, Weblogic, and Oracle Application Server). Home health care system is composed of multiple applications as discussed in Chapter 4.
3. Backend Application - Is in charge of maintaining backend data files (database repository). The database repository is managed within a database management system (DBMS, e.g. Oracle, Microsoft SQL Server, and Derby). The application

architect would structure the data (e.g. as a relational database) and store it inside the backend database to be accessible to the application middle-tier using, for example, a structured query language (SQL). Also, the application architect would need to add maintenance scripts as part of the backend application for maintaining the data stored in the database. Generally speaking, backend application consists of two parts: i) a combination of scripts for maintaining the application data (application architect concerns), and ii) scripts for maintaining the database itself (DBA and system administrator concerns). Home Health care system is composed of many data repositories as discussed in Chapter 4, e.g. PHR and EHR repositories.

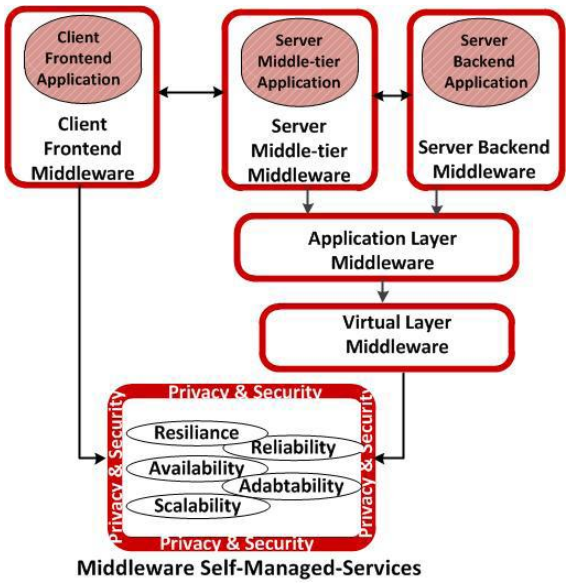

Figure 45 Middleware types for a multi-tier application in the cloud

The proposed multi-tier application architect requires a set of trustworthy middleware, as follows (see Figure 45).

1. *Virtual Layer Middleware* - This middleware intermediates the communication between the physical layer and other layers. It should provide transparent infrastructure management services to application layer via a set of self-managed services. *Application middleware* requires these services to support trustworthy and resilience application.
2. *Application Layer Middleware* - As discussed before this middleware should provide a transparent management services to server applications via a set of self-managed services. This middleware is conceptually composed of two parts:

   (a.) *Server Middle-tier Middleware* that supports Server Middle-tier Application

   (b.) *Server Backend Middleware* that supports Server Backend Application.

   These middleware should coordinate amongst each other to provide trustworthy and resilient service between server middle-tier applications to server backend application. They also need to coordinate with the other types of middleware to provide trustworthy and resilience service between Client Frontend Application to Virtual Layer.
3. *Client Frontend Middleware* - This middleware should provide transparent management services on Client Frontend Application via a set of self-managed services. The services' functions should coordinate with *Server Middle-tier*

Middleware in order to provide trustworthy service between client middle-tier middleware to *Server Middle-tier Middleware*.

6.3.2 Middleware Services Interaction

In this section we use the conceptual models proposed in Section 6.2 to discuss middleware services interaction when managing the multi-tier architect proposed earlier. Our discussion is based on providing several examples for the interaction amongst *Client Frontend Middleware*, *Server Middle-tier Middleware*, and *Server Backend Middleware* to self-manage the overall application. In this report we do not discuss *Virtual Layer Middleware except when absolutely necessary*.

Home healthcare application requirements are mainly focusing on security and privacy aspects. Our discussion in this section reflects these requirements.

6.3.3 Client Frontend Middleware

We start by Client Frontend Middleware when supporting Client Frontend Application (in this we do not discuss issues related to customer environment's self-managed services; for example, we do not discuss Availability and Scalability services for this specific case). This requires the following self-managed services.

1. *Adaptability* – This service is in charge of adapting client frontend application side to changes provided by Cloud provider (i.e. Server Middle-tier Middleware), e.g. changes in service location, degraded performance, and incidents. This would enable adaptability service at client side to take appropriate actions. Example of actions include (see Figure 8): (a.) on change of service location the Adaptability service of the Middle-tier Middleware updates changes of location with the Client Frontend Middleware, which enables the client to keep connected with the right server; (b.) on changes of performance due to emergency the client could reduce its requests to the minimal or even do offline processing and then upload the result on the Cloud; and (c.) on security incidents the client could temporarily follow an emergency plan. These are just sample examples, which would be based on the application context. It is important to re-stress at this point that the application is not necessarily a simple HTML, as it could be an interactive application that do processing at Clouds' customer location and then communicates with Cloud for follow up process. For example, ActiWatch (see Chapter 4) have their own application that communicates with the Cloud to upload data for patient monitoring.
2. *Resilience* - This service is about providing resilient service at client side when communicating with the Cloud (see Figure 38). The service, in this context, mainly attempts to re-establish failed communication with the Cloud (i.e. with Server Middle-tier Middleware).
3. *Reliability* - This service is concerned about maintaining service reliable for client frontend application when communicating with the Cloud (see Figure 41). The service, in this context, ensures reliability when data transferred/received to/from Cloud, and ensures reliability when data processed at client frontend application.
4. *Security and Privacy* --- Is related to providing security measures at Cloud customer side for client frontend application (see Figure 42). This, for example, includes (a.) protecting client's data when retrieved from the Cloud and stored or processed at client environment, and (b.) protecting data whilst being transferred to/from the Cloud.

6.3.4 Server Middle-tier Middleware

Server Middle-tier Middleware supports Server Middle-tier Application and requires the following self-managed services.

1. *Adaptability* - This service is in charge of supporting changes and events that might affect functions of Server Middle-tier Application, as illustrated in
2. Figure 37. Examples of these include the following.
   (a.) Problems in the Cloud, which require relocating the service to another location. The service communicates with the client frontend middleware's adaptability service to take an appropriate action.
   (b.) If Server Middle-tier Application cannot be restarted because of hardware related issues the adaptability service coordinates with the adaptability service at all other dependent middleware (e.g. virtual layer middleware and client frontend middleware).
   (c.) If application cannot be restarted because of dependency problem, the adaptability service manages this by finding dependent applications and re-validating their availability.
3. Resilience - This service covers the following examples (see Figure 38).
   (a.) Restart server middle-tier application on failure.
   (b.) If the application cannot be restarted because of an error (application, environment, or others) the service follows appropriate procedure based on the error nature (e.g. triggers the adaptability service).
4. Scalability – This service is mainly concerned about server middle-tier application adaptability issues when the hosting underneath resources scales up/down. This covers the following (see Figure 39).
   (a.) Scaling up resources allocated to a VM hosting server middle-tier application. This requires the application to follow a set of processes, e.g. spawn further child processes.
   (b.) Scaling up by adding a VM, which require the application to follow a different process, e.g. notifies the availability service to redistribute the incoming load to the newly created VM, and redistribute client sessions considering the new VM;
   (c.) Scaling down by removing additional resources allocated in (a.) or removing the additional VM allocated in (b.), each requires following a somehow a reverse process and notifies the availability service.
5. Availability - This service is in charge of distributing the load coming from client frontend application and server backend application evenly across server middle-tier application redundant resources. If a resource is down, the availability process immediately stops diverting traffic to that resource, and re-diverts the traffic to other active resources until the adaptability process fixes the problem. Also, when the hosting environment scales up/down the availability service re-considers incoming requests distribution based on the nature of the scaling. These are illustrated in Figure 40.
6. Reliability - This service is concerned about maintaining service reliable for server middle-tier application when communicating with both server backend application and client frontend application. Example of processes provided by this service includes the following (see also Figure 41).
   (a.) Verifying reliability when data transferred/received between applications.
   (b.) Verifying reliability whilst data is processed.
7. Security and Privacy - Is related to maintaining Cloud customer's security and privacy requirements which are related to application layer. This includes the following.

(a.) Protecting client's data when retrieved from the client frontend application.
(b.) Protecting data whilst being processed by server middle-tier application.
(c.) Protecting data when transferred to/from server backend application.
(d.) Protecting data on storage
(e.) Ensuring security and privacy is preserved for all other services (e.g. securing communication baths).

6.3.5 Server Backend Middleware

Server Backend Middleware, which is required to support *Server Backend Application*, requires same services that are required for *Server Middle-tier Middleware*. The main difference is that this middleware does not communicate with the client frontend middleware. It mainly protects the application that intermediates the communication between *Server Middle-tier Application* and backend storage, where data eventually stored. This in turn means this middleware services' implementation would require to provide additional functions and security features for managing database instance that interacts with the storage.

Chapter 7

Trust model

Chapter Authors:

Imad Abbadi (UOXF), Mina Deng (PHI), Marco Nalin, Ilaria Baroni (HSR)

7.1 Introduction

In this Section we specifically focus on identifying the functions of middleware services which are required for supporting home healthcare application requirements. Such middleware services are in charge of automatically managing the interaction between Cloud services and between Cloud services and clients. Having a trustworthy services helps in building Cloud trust model. We discuss the security threats that can and cannot be covered by middleware services. Additional mechanisms need to be provided in order to protect against some attacks, which are also discussed throughout this Section.

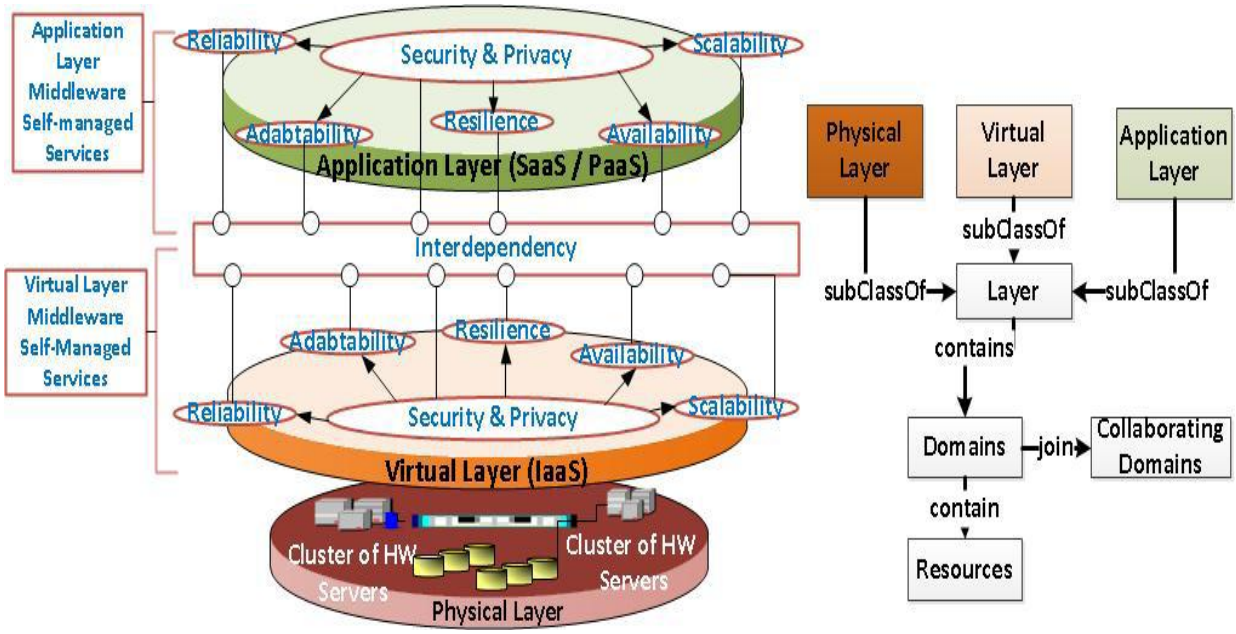

Figure 46 Cloud Taxonomy and Middleware Services

Understanding the way Cloud manages the infrastructure is a fundamental requirement for establishing trustworthy middleware services. This in turn helps in establishing trust in the Cloud. In this section we briefly outline Cloud taxonomy and middleware services, which has been discussed in further details in Deliverable D2.1.1. We start by splitting Cloud environment into following layers (see Figure 46).

Physical Layer - This layer represents the main physical components and their interactions, which constitute Clouds' physical infrastructure. Example of these includes physical servers, storage, and network components. The physical layer resources are consolidated to serve the Virtual Layer.

Virtual Layer - This layer represents the virtual resources, which are hosted by the Physical Layer. Cloud customers in IaaS Cloud type interact directly with the virtual layer, which hosts Clouds' customer applications. This layer includes Virtual Machine (VM), virtual network, and virtual storage.

Application Layer - This layer has Clouds' customer applications, which are hosted using resources in the Virtual Layer. Cloud customers for PaaS and SaaS interact with services at this layer.

Figure 46 provides a conceptual overview of Cloud layering. In the right part of this figure we identify an entity Layer as the parent of the three Cloud layers. Layers contain Resources which are conceptual entities that provide services to other entities. From an abstract level the Layer contains Domains; i.e. we have Physical Domain, Virtual Domain, and Application Domain. A Domain resembles a container, which consists of related resources. Domain's resources are managed following Domain defined policy. Domains that need to interact amongst themselves within a layer join a Collaborating Domain. A Collaborating Domain controls the interaction between Domains member in the Collaborating Domain using a defined policy. The nature of Resources, Domains, Collaborating Domains, and their policies are layer specific. Domain and Collaborating Domains concepts help in managing Cloud infrastructure, and managing resources distribution and coordination in normal operations and in incidents. Resources are spread across Cloud heterogeneous infrastructure and would need to cooperate, exchange critical messages and coordinate amongst themselves. Such coordination requires a set of trustworthy middleware, which glues Cloud entities together by providing a set of automated self-managed services (Abbadi,I.M.,2011,2 Abbadi,I.M.,2011,4), as illustrated in Figure 46.

These services support Cloud's resources resilience, availability, reliability, adaptability, and scalability properties that consider users' security and privacy requirements by design. The services should be transparent to Clouds' customers and require minimal human intervention, as defined by NIST(NIST.,2011). The implementation of self-managed services' functions in middleware would mainly depend on the middleware location within Cloud's layers. The left part of Figure 46 illustrates Cloud Layers' middleware, as follows: a Virtual Layer Middleware is needed between Physical Layer and Virtual Layer to provide infrastructure transparent services to virtual layer, and an Application Layer Middleware is needed between Virtual Layer and Application Layer to provide transparent management services to applications. Virtual Layer Middleware is discussed in detail in Deliverable D2.1.1 while application layer middleware is discussed in (Abbadi,I.M.,2011,2). In this Section we are mainly interested in security, privacy and resilient self-managed services. Specifically, we focus on identifying the services functions that are required to support home healthcare application as discussed in Chapter 4.

7.2 Functions of Middleware Services

In previous section we outline middleware services within the provided Cloud taxonomy. In this section we focus on three middleware services at application and virtual layers, namely, security, privacy and resilience. Specifically, we discuss the main functions that such services should provide to satisfy home healthcare requirements. In addition, we briefly discuss some of Cloud users (we call them clients) middleware services of interest to us. In order to move in this direction we first summarize the home healthcare requirements, and then discuss middleware services at each layer. As discussed in Chapter 3, home healthcare system needs the following security properties: confidentiality, integrity, availability, authentication, authorization, non-repudiation, and auditability. It also requires the following privacy properties: unlinkability, anonymity, confidentiality, content awareness, and consent/policy compliance. In addition, home healthcare system requires resilient architecture.

Figure 47 illustrates an example of a possible deployment of home heathcare application in the Cloud. The following illustrates the following main types of middleware:

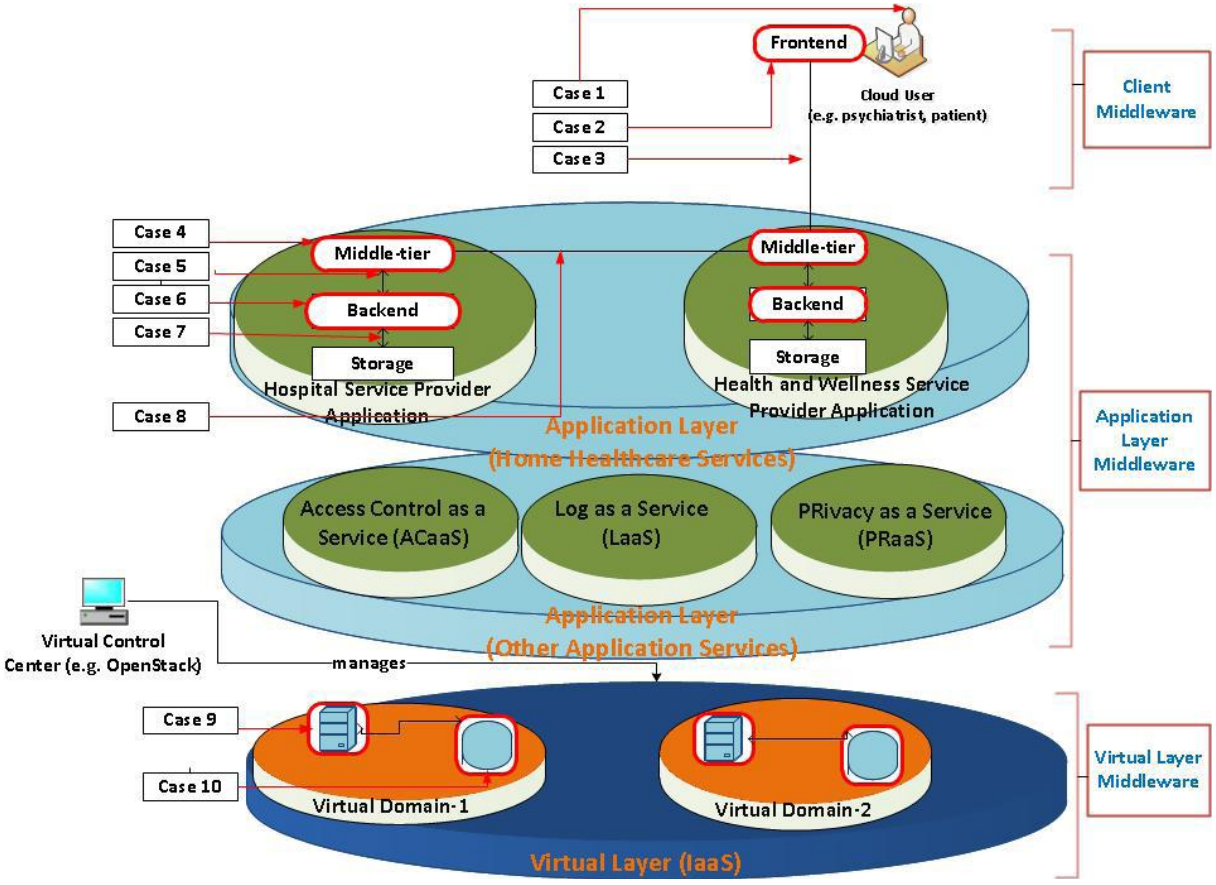

Figure 47 Mapping Home Healthcare system to Cloud Taxonomy

- **Client Middleware** - As we discussed earlier, a home healthcare application requires two types of frontend applications. Each application would require a supporting middleware. In this Section, for convenience we discuss those middleware as a united entity, which is illustrated in the upper part of Figure 47. The Client Middleware supports frontend applications and is associated with three cases (Case-1 - Case-3). Application Layer Middleware - As we discussed earlier, a home health care

application requires two application domains. Each application at the Cloud has two components: the first is middle-tier application which is supported by a middle-tier middleware, while the second is a backend application which is supported by a backend middleware. For simplicity we discuss the middleware services for a single application domain, which is equally applied to other application domains.

- **Application Layer Middleware** - As illustrated in Figure 47, is associated with five cases (Case-4 - Case-8). The middle part of the Figure 47 also illustrates a set of common application services which are required by all home healthcare applications.
- **Virtual Layer Middleware** - As illustrated in the lower part of Figure 47, the virtual layer has virtual domains. Each domain is application specific and has two types of virtual resources: virtual machines which are supported by virtual machine middleware, and virtual storages which are supported by virtual storage middleware. Virtual Layer Middleware is associated with two cases (Case-9 and Case-10). It is outside the scope of this report to discuss the details of virtual layer middleware functions. Instead, we mainly focus the functional services which are directly related to home healthcare applications.

The sequence numbers of the above cases represent the business process workflow. We use these in the subsequent three subsections to help discussing the contribution of the identified middleware towards achieving the overall home healthcare properties.

7.3 Client Side Middleware

As discussed above, we are mainly interested in the functions of client middleware services which are directly related to the identified home healthcare requirements at the Cloud. Specifically we focus on the security, privacy and resilient middleware services in the following cases.

Case-1 - The security properties which are required at case-1 include accountability, auditability and none-repudiation; i.e. a user cannot deny committing an action. Providing these services require the middleware to use strong authentication mechanism (e.g. combination of a smart card, biometric verification, and password or PIN). It also requires trustworthy logging mechanism which could be provided by the Cloud (see Log as a Service (LaaS) discussed in Section 4.4.2). Each transaction of data (e.g. access, upload, download, modification, or deletion) should be logged with sufficient details (e.g. user ID, time/date stamp, and details of transaction).

Case-2 - Middleware services should support frontend application as follows:

a. Resilience at frontend application. This can be achieved by different means; e.g. to automate the process of re-starting the application whenever it fails. In this case the middleware, whenever it restarts the application, should always validate the environment security and the identity of the application as discussed next.
b. Integrity and confidentiality (Environment Protection) - Frontend middleware at client should only allow authorized frontend application to communicate with the middle-tier application and access sensitive data. It should also verify that the client environment execution status is trusted as we discuss in case-3. This essentially prevents the application from being attacked by other applications in the environment.
c. Integrity and confidentiality (Patients Data Protection) - Frontend middleware acts as policy enforcement point (PEP) for data access rights which are retrieved from Access Control as a Service (ACaaS) (see Section 4.4.1). ACaaS acts as a policy decision point (PDP) which manages access rights and content protection key, as

discussed in Section 4.4.1. Frontend middleware provides transparent data encryption/decryption on behalf of middle-tier application.

Case-3 - Middleware services at client side should support the interaction between frontend application and middle-tier application, as follows:

a. Confidentiality and integrity of data whilst in transit. This service could be provided in different means, for example, by the use of symmetric encryption and MAC. Besides, entity mutual authentication of frontend middleware to middle-tier middleware and vice-versa is required. The provision of this service is implementation-dependent, and involves a protocol exchange between frontend middleware and middle-tier middleware. It is initiated when the frontend middleware and the middle-tier middleware mutually authenticate each other. This mutual authentication attests to both middleware execution status, and whether it is trusted.
b. Resilience of communication channel. This can be achieved in different means; e.g. automatic reestablishment of the communication path with the Cloud middle-tier middleware whenever it fails - point (a) above should always be validated.

7.4 Application Layer Middleware

The middle part of Figure 47, which is related to application layer middleware, is associated with cases 4-8. We are interested in discussing security, privacy and resilient middleware services at the following cases.

Case-4 - Middleware services should support middle-tier application as follows.

a. Resilience - This can be achieved by different means; e.g. automatic restarting the application whenever it fails. In this case the middleware, whenever it restarts the application, it should always validate the environment security and the identity of the application as discussed next.
b. Integrity and confidentiality (Environment Protection) - This follow similar discussion to those provided in case-2.
c. Integrity and confidentiality (Patients Data Protection) - This follow similar discussion to those provided in case-2.
d. Auditability property - Middle-tier middleware acts as a proxy which interacts with LaaS mechanism. LaaS provides secure logging mechanism (discussed in Section 4.4.2). The application uses APIs provided via the middleware to log application access details at LaaS.

Case-5 - The middle-tier middleware and the backend middleware should provide protected communication path for exchanging messages. This follows the same description provided in case-3 above.

Case-6 - Backend application middleware requires the same services as discussed in case-4 above.

Case-7 - The backend middleware should securely stores application data at backend storage. This can be achieved by different means. For example, using protected storage functions, or storing only encrypted data.

Case-8 - Integrity and confidentiality of cross-domain applications communication. Different home healthcare applications can only communicate via their middle-tier components. Protecting the secure communication path amongst such components, and attesting to components identity and execution status are provided using the middle-tier middleware exactly as discussed in case-3.

7.5 Virtual Layer Middleware

The virtual layer middleware (as illustrated in Figure 47) has two cases: *Case-9* - Virtual machine middleware and *Case-10* - Virtual storage middleware, which run as part of Virtual Control Center (VCC) (Abbadi,I.M.,2011,4). It is outside the scope of this report to discuss these cases. Further details about these can be found in Deliverables D2.1.1, D2.2.1, D2.3.1, and D2.4.1.

7.6 Application Layer Services

In this section we briefly outline the application components which, in addition to middleware functions, support home healthcare application properties. Specifically, we outline Access Control as a Service (ACaaS), Log as a Service (LaaS), and Privacy as a Service (PRaaS).

7.6.1 Access Control as a Service

ACaaS (discussed in Deliverable D2.4.1 – Section 9) is an Enterprise Rights Management (ERM) application customized to serve Cloud customers' needs. ERM schemes (see, for example, (Oracle,2008)) are proposed to protect enterprise's content when exchanged between devices in organizations. For example, when an employee creates a document and sends it to his/her colleagues, the sender wants to ensure that receivers will use content based on usage conditions. Such conditions (or fine grained access rights) are defined by content owner. To address such a requirement ERM schemes provide the ability to allow content owner to define access rights which are associated with content, in some way, and which are enforced at client devices. ACaaS follows similar approach to that used by ERM. However, in ACaaS access rights enforcement is required at two levels: (a.) as in the case of ERM, ACaaS is used to define access rights and enforce them wherever content is transferred and used, and (b.) unlike ERM, ACaaS should also enforce access rights at virtual and even physical layers (e.g. to impose geographical restrictions of storing and processing customers data). The latter is especially important in Cloud; in ERM case the source of organization content are physically stored and protected inside organizational premises and, in addition, organization employees have a direct contractual relationship with the organization. In Cloud case, on the other hand, organizational content storage and execution are no longer controlled by the organization. This raises security and privacy concerns by Cloud users. Therefore, we are planning to extend ERM schemes to cover this requirement; i.e. enforces access rights at virtual and physical layers as well as extending it to cover client side that fits with the Cloud context.

7.6.2 Log as a Service

LaaS (discussed in Deliverable D2.1.1 – Section 6, and D2.4.1 – Section 7.6) is a trustworthy service provided by the Cloud for logging purposes. Its main function is to store log events in trusted storage which provides log integrity protection and enables the search of events by authorized users. As discussed earlier LaaS is needed at all layers in the Cloud. For example, LaaS is used at application layer to manage all application logging activities; LaaS is also required at virtual layer to manage all logging activities related to accessing virtual resources; similarly it is required at physical layer to manage all physical resources logging activities. The component described in D2.1.1 and D2.4.1 shall provide a trustworthy LaaS mechanism.

7.6.3 Privacy as a Service

PRaaS is provided by the Cloud to protect patient privacy. It offers a series of services. First, personal data stored in the Cloud need to be protected. For instance, the European Data Protection Directive advises that personal data disclosures and retention are controlled in compliance with the data minimization principle (EU.,1996.) to be limited to what is directly relevant and necessary to accomplish a specified purpose. This implies that personal data need to be filtered or anonymized according to the corresponding privacy policy under certain conditions. In addition, transparency needs to be guaranteed that data subjects are aware of their privacy rights and able to specify and delegate the access control policy of their data. Moreover, accountability should be ensured such that data access and usage is securely logged.

Chapter 8

Conclusions

Chapter Authors:

Mina Deng (PHI), Marco Nalin, Ilaria Baroni (HSR), Eva Schlehahn (ULD), Imad Abbadi (UOXF)

This deliverable presents the TClouds healthcare use case that focuses on developing a cloud-supported home healthcare application to provide collaborated services across different health care providers.

First, the TClouds home healthcare use case scenario is described in Section 2. This use case aims to provide innovative services for the depressed patient's remote home monitoring to support patient's therapy. Patient's monitoring information is collected by a mobile monitoring device from patient's home. This information is then sent to the Cloud via a client side program and shared with other healthcare service providers such as hospitals.

Preliminary technical requirements are derived from the aforementioned application in Section 3. Two security and privacy requirements engineering strategies have been followed. One follows the service logic driven approach, and the other one follows the architecture driven approach. Both cloud generic and the healthcare specific use case technical requirements are identified with respect of security, privacy, and resilience.

Next, a preliminary overview regarding possible legal requirements and exemplary solutions for the TClouds healthcare use case are discussed in Section 4. In this use case, an elementary aspect is the collection, processing and storing of personal data of depressed patient's in a cloud computing environment. The discussion takes into account the current and ongoing developments in the context of the revision of the European data protection framework. This overview however is not intended as a complete analysis of the legal requirements concerning this scenario. Nevertheless, it already outlines roughly the arising problems for storing and processing medical data in a cloud computing environment. It also gives some first guidelines how the electronic patient file must be composed to comply with the general data protection framework on EU and national level. This preliminary overview needs to be refined to work out problem fields, explore open questions and present tangible results.

An overview of the reference architecture for the healthcare home monitoring scenario is then discussed in Section 5. The application services are detailed through the definition of the use cases, illustrating also the use cases dependencies and involved actors. The reference architecture is derived from the above mentioned use cases and scenario. A practical instantiation of the reference architecture is be illustrated that most likely will be implemented in the first year prototype.

Finally, preliminary middleware functionalities and trust issues of the TClouds healthcare use case are discussed in Section 6 and 7. Establishing trust in the Cloud is a big challenge that requires collaborative efforts from academia and industry, and it is a fundamental requirement especially for Cloud's potential future as an Internet scale critical infrastructure. It is not only beneficial to Cloud's users, but also to Cloud providers, collaborating Cloud-of-Clouds, and external auditors. For example, it can be used in computation of a trust value for a given Cloud and thus enable comparison between alternative Cloud providers.

This work forms one of the foundations for our planned future research in establishing trust in the Cloud. Our next objective is to extend this work further and deploy it on the TClouds proposed infrastructure. Specifically, we start by developing the functions proposed in Section 6 and 7. In parallel, we establish trust protocols based on the identified middleware functions. Once these are done we can deploy the home healthcare application on the TClouds platform architecture supported by the proposed middleware functions and other application services.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., and Zaharia, M. (2009). *Above the Clouds: A Berkeley View of Cloud Computing.*

Retrieved September 14, 2011, from http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf.

BridgeHead (2010). *Report: The BridgeHead Software International 2010 Data Management Healthcheck Survey*. Retrieved September 14, 2011, from http://www.bridgeheadsoftware.com/uploads/BH_Rpt_Data_management_survey_results_US_Letter.pdf

Deng, M. ,. (2010). A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering, vol. 16, no. 1* , 3–32.

ENISA. (2009 November). *Cloud computing benefits, risks and recommendations for information security .* The European Network and Information Security Agency (ENISA).

EU. (1996). *Directive 95/46/EC of the European parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.* Official Journal of the European Communities 281:31–50.

Libusb (2011). Libusb homepage, http://www.libusb.org/, retrieved March 2011.

LibWin (2011). Libusb win32, homepage, http://sourceforge.net/apps/trac/libusb-win32, retrieved March 2011.

Lipner, M. H. (2006). *The Security Development Lifecycle.* Microsoft Press.

NIST. (2011). *The NIST Definition of Cloud Computing (Draft) Recommenda-tions of the National Institute of Standards and Technology, The NIST Definition of Cloud Computing, NIST Special Publication 800-145.* NIST National Institute of Standards And Technology.

Opdahl, G. S. (2001). Templates for Misuse Case Description. *Proceedings of the 7 International* Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ 2001), (pp. 4-5).

OpenStack (2011). Nova homepage, http://nova.openstack.org/, retrieved March 2011.

Philips (2009). Respironics Philips, A trio of solutions, 2009, http://www.actiwatch.respironics.com/pdf/01-3095-PHILIPS-AWTechSheet%2007_02_09.pdf.

Philips (2011). Philips respironics homepage, http://www.healthcare.philips.com/main/homehealth/respironics.wpd, retrieved March 2011.

PyUSB (2011). PyUSB homepage, http://sourceforge.net/apps/mediawiki/pyusb, retrieved March 2011.

Respironics (2009). Actiware and actiware ct software and hardware manual, http://global.respironics.com/UserGuides/Actiwatch_SW_Instr_Manual.pdf.

Wikipedia (2011). Wikipedia, Model-view-controller, http://en.wikipedia.org/wiki/Model_view_controller, retrieved March 2011.

Abbadi,I.M. (2011)(1). Clouds' infrastructure taxonomy, properties, and management services. In CloudComp '11: in proceedings of the International workshop on Cloud Computing Architecture, Algorithms and Applications, Lecture Notes in Computer Science. Springer-Verlag, Berlin, July 2011.

Abbadi,I.M. (2011)(2). Middleware services at cloud application layer. In IWTMP2PS '11: in proceedings of the Second International Workshop on Trust Management in P2P Systems, Lecture Notes in Computer Science. Springer-Verlag, Berlin, July 2011.

Abbadi,I.M. (2011)(3). Operational trust in clouds' environment. In MoCS 2011: in proceedings of Workshop on Management of Cloud Systems. IEEE Computer Society, June 2011.

Abbadi,I.M. (2011)(4). Self-Managed Services Conceptual Model in Trustworthy Clouds' Infrastructure. In Workshop on Cryptography and Security in Clouds. IBM, Zurich, March 2011. http://www.zurich.ibm.com/ cca/csc2011/program.html.

Abbadi,I.M. and Lyle,J. (2011)(5). Challenges for provenance in cloud computing. In to appear in 3rd USENIX Workshop on the Theory and Practice of Provenance (TaPP '11). USENIX Association, 2011.

Deng,M Petkovi_c,M, Nalin,M and Baroni,I (2011). A home healthcare system in the cloud - addressing security and privacy challenges. In Proceedings of the 4th IEEE international conference on cloud computing 2011.

Oracle (2008). Information rights management | managing information everywhere it is stored and used, June 2008. http://www.oracle.com/technology/products/content management/irm/IRMtechnicalwhitepaper.pdf.

Ahmad-Reza Sadeghi, (2008). Trusted computing | special aspects and chal lenges. In V. Geffert et al., editor, SOFSEM, volume 4910 of Lecture Notes in Computer Science, pages 98{117. Springer-Verlag, Berlin, 2008.

Il Garante per la protezione dei dati personali, Linee guida in tema di Fasciolo sanitario electtronico (Fse) e di dossier sanitario (Guidelines on the Electronic Health Record and the Health File), 16 Guigno 2009,

Ninja Marnau, Eva Schlehahn, TClouds Deliverable D1.2.2 – Cloud Computing: Legal Analysis, 2011

Ronald Leenes, Jan Schallaböck, Marit Hansen, PRIME Privacy and Identity Management for Europe - White paper v3

Yale University Introduction to HIPAA, Break Glass Procedure: Granting Emergency Access to Critical ePHI Systems, URL: http://hipaa.yale.edu/security/breakglass.html

Melissa Chase and Kristin Lauter, Microsoft Research, *An Anonymous Health Care System*, URL: http://research.microsoft.com/en-us/um/people/melissac/healthsec.pdf.