

Software Architecture Design for HealthSuite System

Group 28 - Cluster D

TianHao Xu
2740825
Computer Science
t4.xu@student.vu.nl

Li Zhong
2688794
Computer Science
l3.zhong@student.vu.nl

Ruijia Lei
2733510
Computer Science
r.lei@student.vu.nl

Jingye Wang
2702383
Computer Science
j18.wang@student.vu.nl

Emmanouil Chalkidakis
2691664
Computer Science
e.chalkidakis@student.vu.nl

2021.12.17

Revision history

05/11/2021	W1 – Stakeholders and preliminary architecture	<ul style="list-style-type: none"> - Described the roles of stakeholders Healthcare Providers, Patients, Caregivers, Payers, Security Overwatchers in the <i>Stakeholder profiles</i> section - Included preliminary architecture design in the section <i>System overview</i>
12/11/2021	W2 – Stakeholders and improved architecture	<ul style="list-style-type: none"> - Described the business goals and architecturally significant requirements of stakeholders Healthcare Providers, Patients, Caregivers, Payers, Security Overwatchers in the <i>Stakeholder profiles</i> section - Included our improved architecture design in the section <i>System overview</i>
19/11/2021	W3 – Viewpoint Selection	<ul style="list-style-type: none"> - We have selected different viewpoints for our system and partially borrowed from the 4+1 schema, which includes logical viewpoint, process viewpoint, implementation viewpoint, and deployment viewpoint. We also included the security viewpoint to cover the interests of all stakeholders. - Describe the addressed concerns, rationale, and main modelling technology of our selected viewpoints.
26/11/2021	W4 – Preliminary Architecture design	<ul style="list-style-type: none"> - Reviewed and modified some mistakes and insufficient parts in the current delivery of architecture design, including the business goals, the ASRs and the Architecture sketch, etc. - We have improved the structure and layout of this document by adding a table of contents, captions for figures and tables, et cetera.
03/12/2021	W5 – View	<ul style="list-style-type: none"> - We finished the primary presentation of views for every viewpoint and described every element that appeared in the views. Also stated the design decisions and their rationales of views.
10/12/2021	W6 – Design Decisions & Rationale, Assessment	<ul style="list-style-type: none"> - Finished the design decision and rationale part - Selected at least two scenarios for each stakeholder, finished the corresponding utility tree, and analyzed architectural approaches
17/12/2021	W7 – Final Architecture design	<ul style="list-style-type: none"> - Reviewed and modified some mistakes and insufficient parts in the preliminary delivery of architecture design according to feedbacks from peer-reviews - Finished the final version of the architecture design of this HealthSuite System

Changelog

Some important feedback and corresponding modifications.

Feedback	modifier	Modification
The reason why Healthcare Providers are needed is missing	Li Zhong	The reasons why Healthcare Providers are needed are added, which can be seen in the description part of Healthcare Providers in stakeholder profile.
The quality attribute of [ASR-01-04] should not be performance	Li Zhong	The quality attribute of [ASR-01-04] is changed to Functional suitability.

The metamodel of deployment viewpoint is too simple to express the design.	Li Zhong	The metamodel of the deployment viewpoint is redesigned with more concrete components and notations.
Needs to expand more about stakeholder 4(SH-04)	Jingye Wang	Add more description about payers
Same problem as above. The stimulus should be an attack.	Jingye Wang	No need to change because, according to the lecture, “stimulus is an event arriving at the system (an input)”.
Logical Viewpoint need to add more description and short name	Jingye Wang	Add more description about viewpoint-03-01; Add short name, And reference; Adjust notation
The architecture overview did not explain very well	Jingye Wang	Refine the architecture overview
Stakeholder 5(SH-05) - they don't follow the structure of business goal scenarios	Emmanouil Chalkiadakis	Refine the structure according to chapter 16.3
Viewpoint 5 - Notation, metamodel and corresponding view	Emmanouil Chalkiadakis	The notation used in network diagrams is not strict. Any symbols are acceptable and are not limited. They are inserted on demand to express the network as closely as can be with natural language.
The metamodel of the implementation viewpoint	Ruijia Lei	Modified the meta-model from a detailed description diagram to the meta description(relation of component) of the view.
It seems that [ASR-03-01] is more likely to be about performance	Ruijia Lei	Agree with this suggestion because it is more about the limitation of the response time. I have changed the ASR-03-01 from Usability to Performance.
STK-04 Payers - please check the grammar	Ruijia Lei	I think this is an ambiguous suggestion, and no grammar mistake is made(Should be fewer suggestions like this).
STK-03 Expectations and demands - Must be more clarified even if they don't want to say exactly	Ruijia Lei	Modified the expression of the response performance and gave a more detailed time limitation as “in few seconds”.
Image clarity of architecture overview	Tianhao Xu	Re-uploaded the image and enlarged it to improve clarity.
The reason why this solution and its elements were chosen is not clear. Sometimes only description.	Tianhao Xu	Refined the description and solution of each element, explained more clearly.
Would be better if this point is not listed in the list of 'expectations and demands' but as references and examples.	Tianhao Xu	Deleted this point but used it as the reference into other points.

Revision history	2
Changelog	2
1. Stakeholder profiles	6
1.1 Stakeholder 1 - Healthcare Providers(SH-01)	6
Description:	6
Expectations and demands:	6
Business goals:	6
Architecturally Significant Requirements:	6
1.2 Stakeholder 2 - Patient(SH-02)	7
Description:	7
Expectations and demands:	7
Business goals:	8
Architecturally Significant Requirements:	8
1.3 Stakeholder 3 - Caregivers(SH-02)	9
Description:	9
Expectations and demands:	9
Business goals:	9
Architecturally Significant Requirements:	9
1.4 Stakeholder 4 - Payers(SH-04)	10
Description:	10
Expectations and demands:	10
Business goals:	11
Architecturally Significant Requirements:	11
1.5 Stakeholder 5 - Security Overwatchers (SH-05)	11
Description:	11
Expectations and demands:	11
Business goals:	12
Architecturally Significant Requirements:	12
2. Architecture Overview	13
3. Viewpoints	15
3.1 Viewpoint 1 - Logical Viewpoint	15
3.2 Viewpoint 2 - Process Viewpoint	16
3.3 Viewpoint 3 - Implementation Viewpoint	18
3.4 Viewpoint 4 - Deployment Viewpoint	19
3.5 Viewpoint 5 - Security Viewpoint	21

4. Views	23
4.1 View 1 - Logical View	23
Primary presentation	23
Element catalog	23
Design decisions and rationale (for this view)	25
4.2 View 2 - Process View	27
Primary presentation	27
Element catalog	27
Design decisions and rationale (for this view)	28
4.3 View 3 - Implementation View	30
Primary presentation	30
Element catalog	30
Design decisions and rationale (for this view)	32
4.4 View 4 - Deployment View	33
Primary presentation	33
Element catalog	34
Design decisions and rationale (for this view)	35
4.5 View 5 - Security View	37
Primary presentation	37
Element catalog	37
Design decisions and rationale (for this view)	39
5. Mapping between views	40
6. Design decisions and Rationale	42
6.1 API Gateway	42
6.2 Load balance	43
6.3 Third parties centre	43
6.4 Unified Access Controller	44
6.5 Security Privacy Centre	45
7. Assessment	46
7.1 Selected assessment scenarios	46
7.2 Utility tree	47
7.3 Analysis of architectural approaches	48
8. Glossary	50

1. Stakeholder profiles

1.1 Stakeholder 1 - Healthcare Providers(SH-01)

Description:

One group of stakeholders that are involved in this system are healthcare providers. Healthcare providers are responsible for providing services to monitor and take care of patients' health conditions. Patients adopt this system to keep track of their health, and the hospital uses the AI diagnosis prediction model to detect diagnosis. The monitoring tools and AI diagnosis detection models are provided by Healthcare Providers, thus they are important in this system.

Expectations and demands:

1. Records about various aspects of patients' body conditions, such as heart rate, blood oxygen concentration, etc., for periods, are needed to be stored efficiently.
2. Data needs to be integrated in advance and stored or transferred efficiently by other modules.
3. Data should be updated from time to time to make sure that the monitoring and prediction stay useful.
4. Enough computing resources and storage resources are also necessary for healthcare providers to perform the diagnosis prediction efficiently, considering that the volume of records about patients' conditions is huge and the AI diagnosis prediction model is computing-intensive.
5. IoT devices are managed and organised in an effective way to decrease the complexity of data management.

Business goals:

1. For HealthSuite, the Healthcare providers desire that the system improves the patient's experience and their satisfaction with the healthcare service and will be satisfied if statistical results show that the patient does notice their diseases and receive treatments earlier than before, such as the meantime for patients to receive treatments significantly decreases with p-value less than 0.05.
2. For HealthSuite, the Healthcare providers desire that the system helps the hospital to detect diagnosis of patients as soon as possible and will be satisfied if the accuracy of the AI prediction model obtains 96% and hospital feedback surveys show an increase of at least one level, such as from 6 to 7.

Architecturally Significant Requirements:

1. **ASR-01-01** Integration: Must integrate data about patients from different sources, such as different hospitals or caregivers, into a uniform format and store them in one data node on the cloud. Integrating data in advance can lift the speed of processing data and speed of responding to patients, which can improve the user experience with this system and promote users' satisfaction about this system.

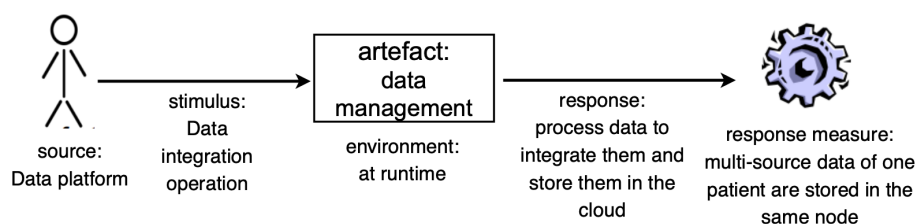


Figure 1.1 Integration attribute for Healthcare providers

2. **ASR-01-02** Performance: Must efficiently store data of patients to respond to every query executed by caregivers or patients within 2 seconds. Requiring the speed of data access response to patients to improve users' satisfaction about this system.

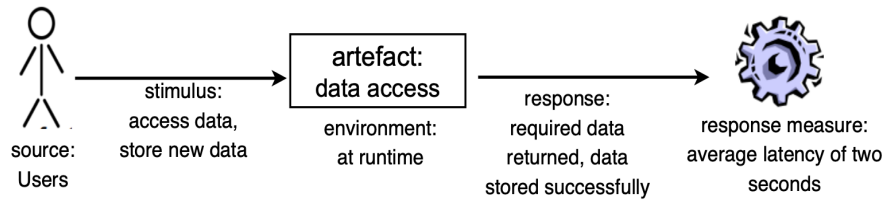


Figure 1.2 First Figure performance attribute for Healthcare providers

3. **ASR-01-03** Usability: Must process data of patients' health conditions with statistical tools in advance to extract valuable information and display information with clear and effective graphs and tables to make sure that patients or caregivers understand within a mean of 5 seconds. Clear and effective graphs and tables help patients and caregivers to understand the deep meaning of data about patients' health conditions to be able to discover anomalies or diseases earlier.

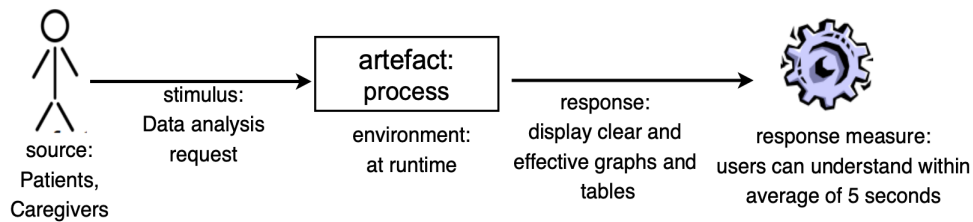


Figure 1.3 Second performance attribute for Healthcare providers

4. **ASR-01-04** Functional suitability: Must provide accurate results about patients' diagnosis predictions. The accuracy must be higher than the expectations of patients or caregivers or take 96% as the default standard. This requirement states the detailed content and the measurement for the second business goal.

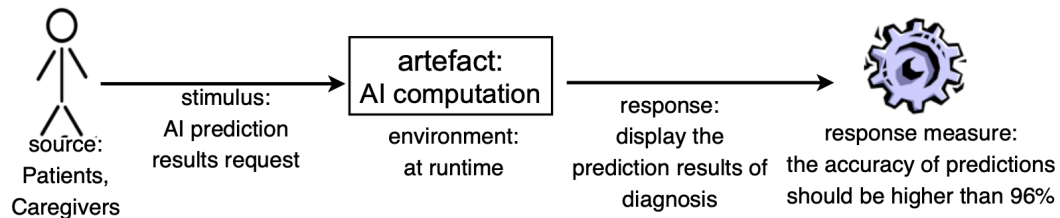


Figure 1.4 Third performance attribute for Healthcare providers

1.2 Stakeholder 2 - Patient(SH-02)

Description:

Patients are customers of Philips HealthSuite. They want to treat illnesses and maintain good health in general through Philips' professional healthcare services such as image-guided treatment, patient testing, management of health information, and home care.

Expectations and demands:

1. Patients want to be able to manage their health data efficiently and to be able to view the data of their body indicators.
2. Software to be able to advise them on some of the problems they have with their body. Serious conditions are provided with detailed solutions.
3. The software's data platform can include information from different hospitals, departments, and general practitioners so that the most authoritative diagnosis in each medical field can be obtained.
4. Patients hope to provide simple online consultation for minor ailments and specialist consultation in case of difficult cases.

5. Test the body at home without going to the hospital, for example, by manually recording some data through the software or by opening a wearable device to obtain health data on a regular basis so that patients can make a judgment. If there is a problem, a warning will be given.
6. This ensures the security of patient information, the non-disclosure of medical records, and the accuracy of data and treatment plans such as records.

Business goals:

1. For patients, they desire to get updated data information in time when they are using the system—synchronization of data between client and cloud database within a limit of time as 0.5s.
2. For the detection equipment, the accuracy rate can reach 99.8%, the daily data of patients can be recorded. It will immediately be warned when patients are in a dangerous situation.
3. The design and interaction of the user interface are very important; it needs to be easy for understanding and use. There is a simple user manual. It takes only 10 seconds for the user to be able to know how to use it, so it doesn't need to spend so much time learning how to use it.
4. For security, it is committed to almost 100% data security protection compared to other software.

Architecturally Significant Requirements:

1. **ASR-02-01** Usability: User-friendly user interface provides good user interaction. It enables the user to understand the operation of the software in a short time.

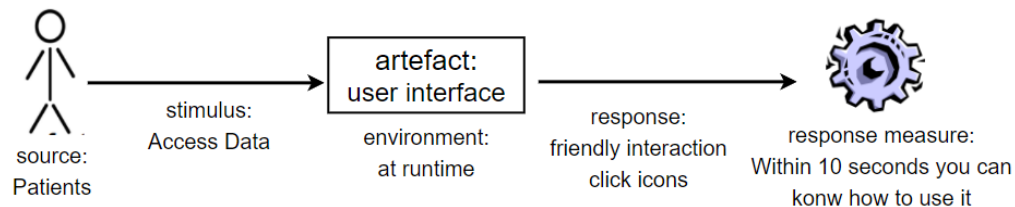


Figure 1.5 Usability attribute for patients

2. **ASR-02-02** Security: Ensure patients' data security when using the software.

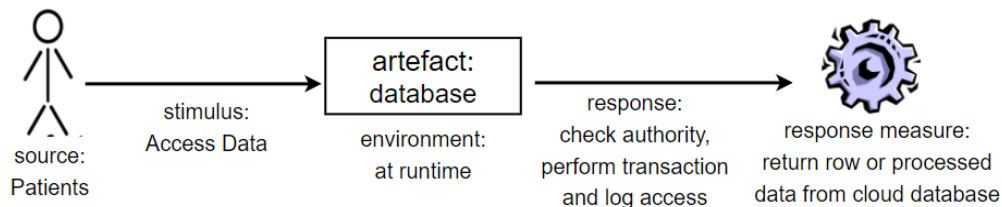


Figure 1.6 Security attribute for patients

3. **ASR-02-03** Availability/Performance: Providing real-time and reliable online communication between caregivers and patients. If a crash happens, it will be recovered in two minutes, and backup keeps data 100% complete.

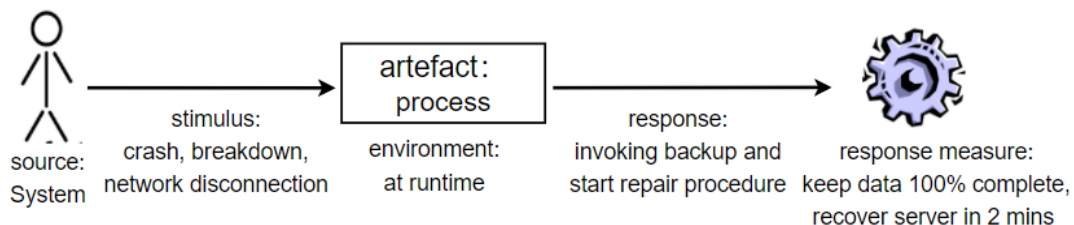


Figure 1.7 Availability/Performance attribute for system

1.3 Stakeholder 3 - Caregivers(SH-02)

Description:

Most Caregivers are doctors, technicians, scientists, and nursing staff. Can be the users of the software and to save important data for the system. They manipulate medical machines and record the health condition data to manually upload it to the cloud data warehouse. They also communicate with patients and research treatment solutions and estimate the costs for payment.

Expectations and demands:

1. Caregivers want to achieve more comprehensive disease surveillance, improved experience between patients and themselves.
2. The system needs to contain eye-catching signs and highlight important functional areas in order to be easy to operate for caregivers.
3. It contains functions for numerical tests, accurate control of the treatment lifetime cycle, and interacting/delivering data to the software.
4. Software needs to respond in a few seconds and help facilitate communication between the patient and the caregivers.
5. Caregivers are supposed to get processed and analysed data from cloud AI-Assistant, which helps improve disease diagnosis rates.
6. The system should provide an interface for accessing the medical data for caregivers and scientists.

Business goals:

1. For the system, caregivers and patients desire the system to control the response time as fast as possible, less than 50 ms. If a breakdown or crash happens, the server should be repaired in 1 minute. Real-time online communication should enhance caregivers' supervision of patient treatment and monitoring of conditions to obtain timely feedback.
2. For the system, more accurate data analysis and tracking are needed, the cloud-end AI assistants process diagnostic data (automatically or manually uploaded by instruments operated by caregivers like images and disease statistics). Caregivers desire the system will help increase the accuracy of disease detection from 96% to 98% by using more diverse methods.
3. For the system, patient groups desire the society to achieve higher disease detection and prevention, cloud servers conduct the collation and integration of medical data, analyze the disease problem of the whole society in the long term, and obtain timely feedback and processing results and will be satisfied if it enhances 10% of tracking or monitoring rate specific diseases.

Architecturally Significant Requirements:

1. **ASR-03-01** Performance: Wearable smart devices and medical instruments need to be linked to cloud databases in real-time for data logging and synchronization within at most 5 seconds. This requirement states the detailed content and the measurement for the second and the third business goals.

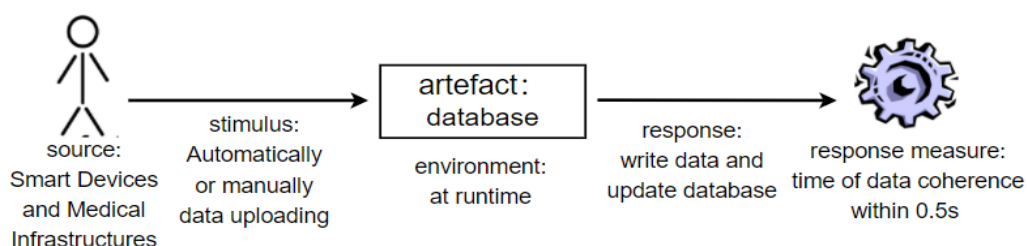


Figure 1.8 Instrumental data and database attribute for system

2. **ASR-03-02** Security: Access authority to medical device data needs to be provided to caregivers according to whether they have been validated by UZI-pass or not. This requirement states the detailed content and the measurement for the second and the third business goals.

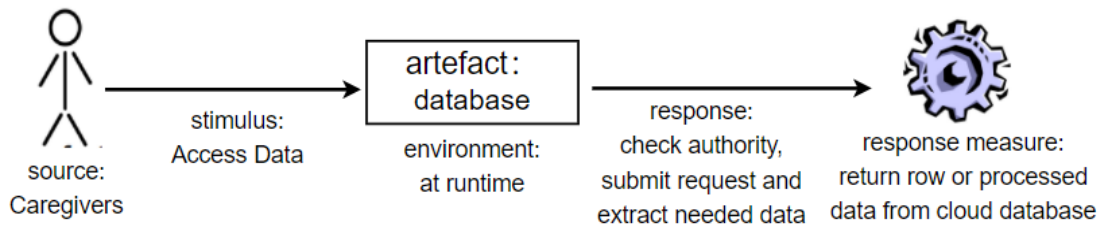


Figure 1.9 Authority-based security attribute for caregivers

3. **ASR-03-03** Availability: Providing real-time and reliable online communication between caregivers and patients. If a crash happens, it will be recovered in a very short time and backup keeps data 100% complete. This requirement states the detailed content and the measurement for the second and the first business goals.

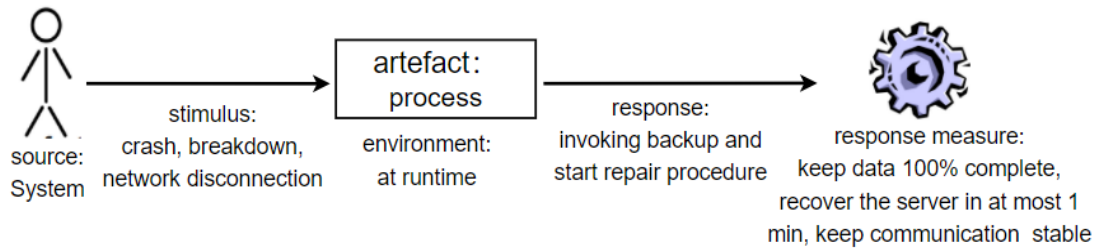


Figure 1.10 Availability for system

4. **ASR-03-04** Performance: Must process-specific medical data like instrumental images with computer vision, big data analysis with AI techniques, etc. This requirement states the detailed content and the measurement for the second business goal.

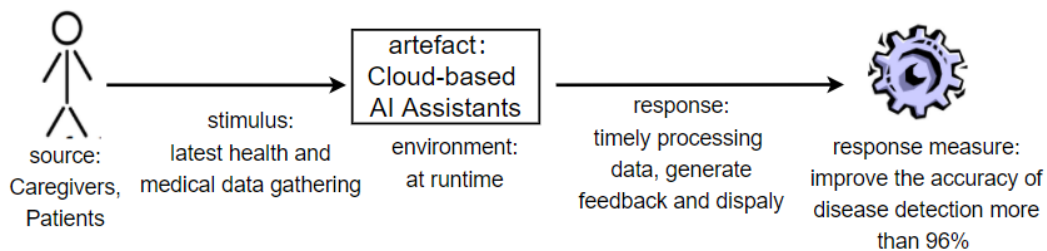


Figure 1.11 Performance for caregivers and patients

1.4 Stakeholder 4 - Payers(SH-04)

Description:

They are the ones who pay the bills. Their roles are various in the circumstances. They may be patients or hospitals because we identify many clients in our system. Patients can buy wearable medical devices and services to monitor their condition more efficiently. Hospitals can buy our software or services to help doctors improve the efficiency of the work. Payers can pay their bills through phone or web pages.

Expectations and demands:

1. The way of payment should be diverse. It should contain cash and include many other ways like transfer through a bank.
2. The system should have detailed information about each order, such as the product or service name, price, bill number, payment time.
3. When calculating the price, the system shall automatically subtract the reimbursable insurance part and provide the accurate price.

Business goals:

1. For using the system, payers desire that the system can improve the payment experience and will be satisfied if they learn how to use it within 30 seconds.
2. For using the system, payers desire that the system ensures secure payment and will be satisfied if it provides at least one plan to ensure their money is not lost because of hacking and provides insurance for the money lost.

Architecturally Significant Requirements:

1. **ASR-04-01** Security: for payers, security is the most important requirement. When payers pay their bills the system should secure all the connections and data transportation to achieve the second business goal.

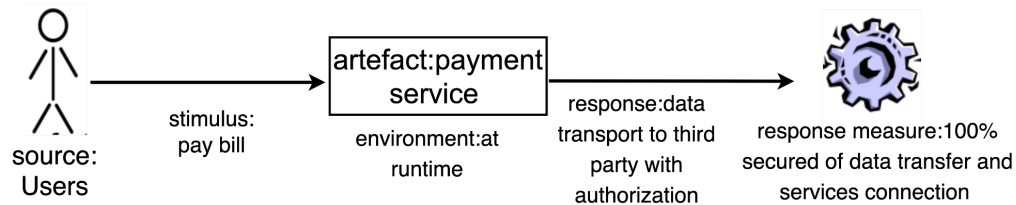


Figure 1.12 Security attribute for payers

2. **ASR-04-02** Usability: For achieving the first business goal, the system provides enough tips and labels to help users pay the bill and users can easily learn how to pay within 30 seconds.

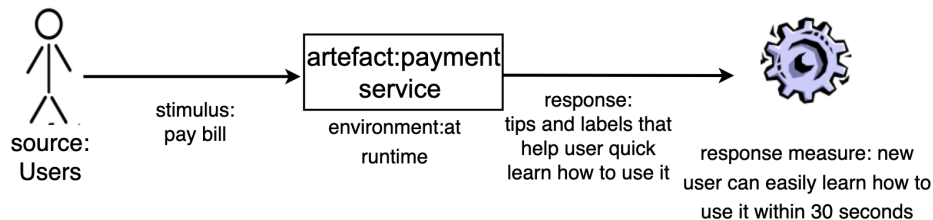


Figure 1.13 Second functional appropriateness attribute for payers

3. **ASR-04-03** Availability: After payers pay their bills, the system should send a payment message so that it can accomplish the first business goal.

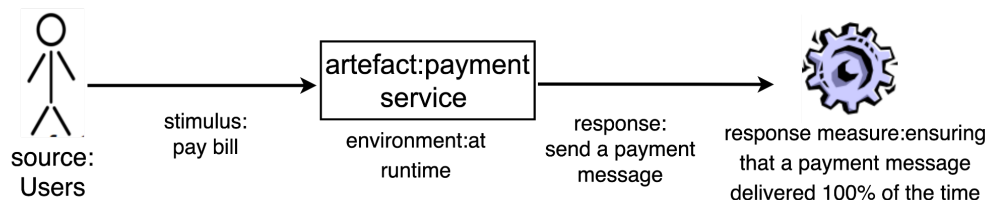


Figure 1.14 Interoperability attribute for payers

1.5 Stakeholder 5 - Security Overwatchers (SH-05)

Description:

This group of stakeholders is responsible for maintaining the confidentiality, integrity, and availability of information regarding the services provided. A leak in sensitive information impacts the reputation of the healthcare provider. An alteration of data might endanger the life of a patient. An unavailable system might cause a cascading disaster resulting in byzantine failures. Their role is to make sure systems work as expected.

Expectations and demands:

1. Sensitive information should not be disclosed or leaked.
2. Data in the system should not be altered in a way other than expected.
3. The system and data should always be usable and available.
4. Hired staff should be specialized, which increases the costs.
5. Specialized staff should be available 24/7 since problems might arise at any time, which increases the cost.

6. Regulations should be strictly followed to avoid penalisation.

Business goals:

1. For the system, security overwatchers desire that all components achieve low downtime in the context of availability and will be satisfied if they have at least 3 9's of availability.
2. For the system, security overwatchers desire that the system achieves no loss of data in the context of integrity and will be satisfied if there exist at least 2 backups for all data and extra off-site backup.
3. For the system, security overwatchers desire that the system achieves no leak of information in the context of confidentiality and will be satisfied if data is encrypted twice with different keys.
4. For the system, security overwatchers desire that specialized staff achieve system resourcefulness in the context of physical infrastructure and will be satisfied if inspection and maintenance take place weekly.

Architecturally Significant Requirements:

It is important to identify the critical points of the operation, so they are treated with extra care. They should be wrapped in an extra layer of security and constantly monitored. Overall, security is implicated and affects every aspect of the architecture.

The system must:

1. **ASR-05-01 Reliability:** For identification and data validation, authenticating any users and logging all access history with a 100% guarantee.

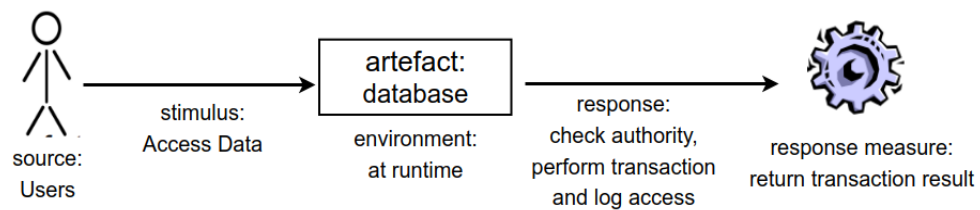


Figure 1.15 Identification and data validation requirements for Security Overwatchers

2. **ASR-05-02 Reliability:** For system reliability, having redundancy in resources and infrastructure.

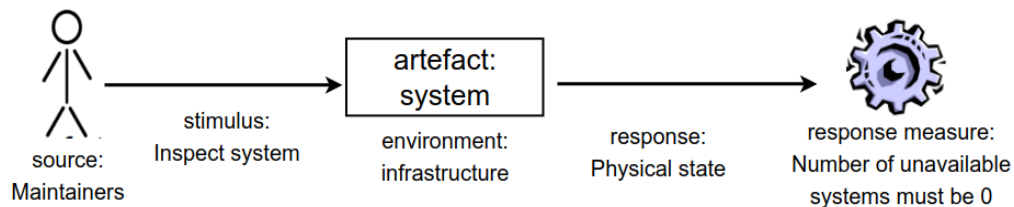


Figure 1.16 Reliability attribute for Security Overwatchers

3. **ASR-05-03 Security:** For data security, encrypting stored data and keeping every public-private key pair.

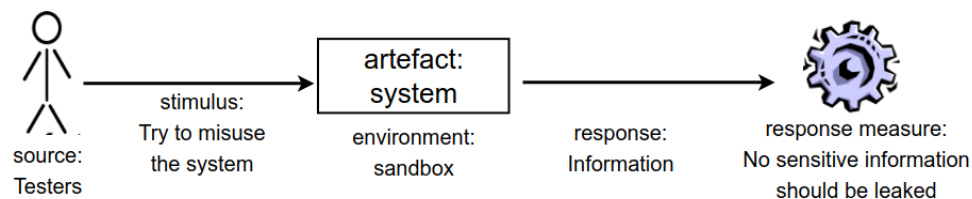


Figure 1.17 Data security attribute for Security Overwatchers

4. **ASR-05-04 Fault tolerance:** For fault tolerance, keeping backups to improve system reliability to prevent data errors and loss caused by sudden crashes, ensuring that the data re-reading response time is within 0.5s.

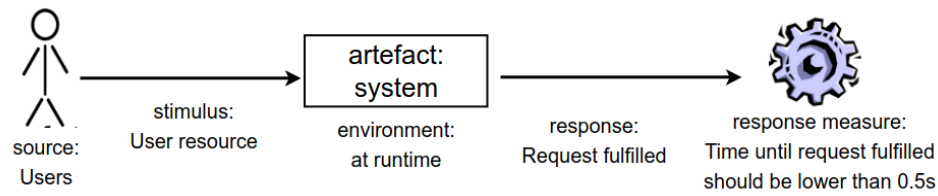


Figure 1.18 Fault tolerance attribute for Security Overwatchers

2. Architecture Overview

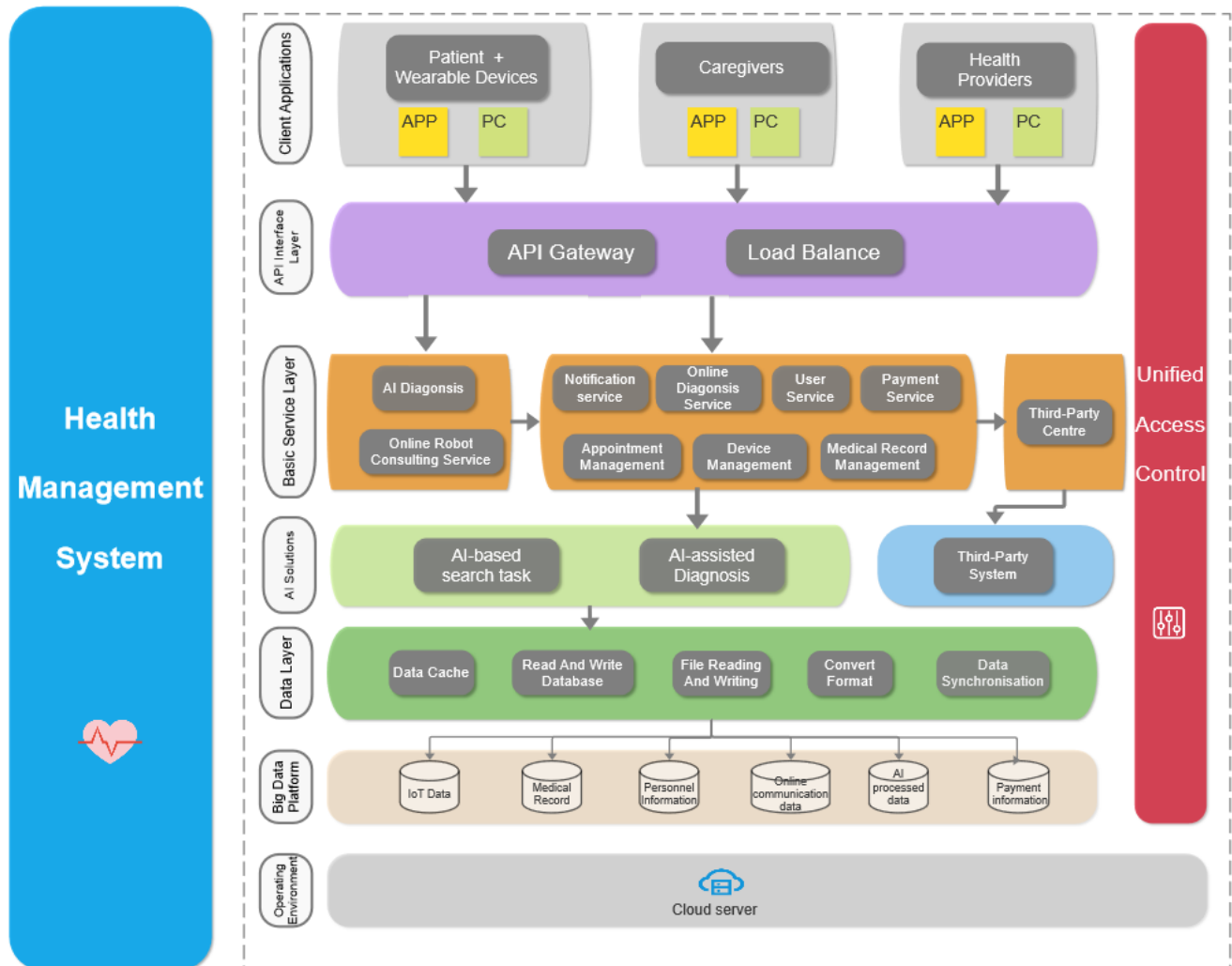


Figure 2.1 The Architecture overview of this system

Our system is named "Health Management System". Our target end-users are patients and caregivers(see AR-07). Our design's primary goal is to help doctors diagnose and improve their work efficiency. Furthermore, we hope through this system helping patients find their illness in an early stage. So in order to achieve these goals, we identify several functions like appointment management, AI Diagnosis and online Consulting service etc.(see AR-05). For the maintainability, availability and security, we also identify API gateway, Load balance and Unified Access Control(see AR-06 & AR-08). Our system uses the Cloud server to run(see AR-01). For the interface to end-users, we design two interfaces APP and webpage.

We have chosen a modular multi-layer architecture diagram, distinguished by different colors. Moreover, every color represents a layer(for example, AI Solution Layer) or an individual module(for example, Unified Access Control). This diagram has a total of eight layers. Aside from that, besides the last two layers, other layers contain several services like AI Diagnosis etc. Below is the description of each layer from the bottom to the top:

AR-01. Operating environment layer

Our system runs on the cloud server. It stores all the data and executes all the programmes. We use cloud servers because they are highly scalable, relevant communities are active and easy to implement and manage using Docker and Kubernetes.

AR-02. The big data platform layer

The big data platform integrates data from various data resources and types. Based on the functionality, we classified the data into IoT data, Medical Records, Personal Information, Online communication data, AI processed data and Payment information.

1. All IoT devices data will be stored in IoT data.
2. Medical Records are the patient's medical records obtained from healthcare providers(SH-01).
3. Personal Information contains user personal information like name, date of birth, phone number and email.
4. The online communication information only contains general information about the communication, like the sender and receiver's name, the rate of user about the communication, and the date of the communication.
5. AI processed data stored every time the AI analyzed results.
6. Payment information contains the bill number, the product that users buy, the payment date and the price.

AR-03. Data layer

The data layer guarantees the scalability of the system. It represents the database, cache and IO operation between the cloud server and the local application. The format conversion and data consistency which play a vital role in security and stability, are mostly considered in this layer. In this layer, we will use cache to increase the system performance. Moreover, we will use a read and write database and file reading and writing system to increase the system security and stability.

AR-04. AI solutions layer

There are two main functions provided here: "AI-Based search task" and "AI-assisted Diagnosis".

1. The AI-assisted Diagnosis will auto analyze the data from IoT data, find if the patient has some abnormal psychical conditions, and make a suggested diagnosis.
2. The AI-Based search task can automatically find a suitable treatment based on the diagnosis or automatically search the answers users requested.

Both are based on artificial intelligence techniques like the computer vision to help improve the diagnostic accuracy of medical images, natural language processing to accelerate the extraction of relation and core Information from text data to answer user questions or find related treatment.

AR-05. Basic service layer

This layer provides essential services.

The one kind is based on the AI solutions layer(AR-04): AI Diagnosis and Online Robot Consulting Service.

1. AI Diagnosis helps the doctor to diagnose and automatically find the abnormal physical conditions of the patient.
2. Online Robot Consulting Service automatically answers the general question that the patient asked.

Another kind is not related to the AI solutions layer(AR-04). They are Health Reminder, Online Diagnosis service, Payment Service, User Service, Appointment Management, Device Management, Medical Record Management and third-party centres. (The Third-Party system represents all kinds of the outside system).

1. The Notification Service notify users after their payment is completed or when they alter doctor and patient if any abnormal data is found through AI diagnosis service.
2. Online Diagnosis service is a channel that the doctor helps patients and gives patients diagnoses remotely.
3. Payment Service is related to payment, like checking bill information or calculating the price.
4. User Service is responsible for all kinds of user data like the role of the user, managing personal user data, authentication and identification. Every individual user has a unique registered account. The account can be linked with DigiD/BSN for being a senior one and assigned more permission. Moreover, caregivers are supposed to access more medical data, so they are required to submit the UZI-pass, which helps verify their medical identity.
5. Appointment Management is related to managing appointments between doctors and patients.
6. Device Management mainly manages all kinds of device data like wearable devices, MRI etc.
7. Medical Record Management manages the patient's medical record. Moreover, the patient can access their data, and the doctor can access their patient data with patient authentication.

8. The third-party centre integrates external services like payment, bank or SMS third-party.

AR-06. API Interface Layer

Two API interfaces are provided. One is 'API Gateway', the entry point for all requests for the services and finding the route of actual services. The second is 'Load balance', which improves the system stability and deals with high concurrency problems.

AR-07. Client Applications

We classified end-users into Patients, Health providers and Caregivers. They can use our software with mobile applications and PCs.

AR-08. Unified Access Control

This module mainly controls all the access. If the current user is not logged in, it will redirect them to the user service to log in or register. After login, the user will be assigned an encrypted token that contains expiration time and the role of this user. Every time, the request will be checked for the role's permission, and if it has no permission, Unified Access Control will deny the request.

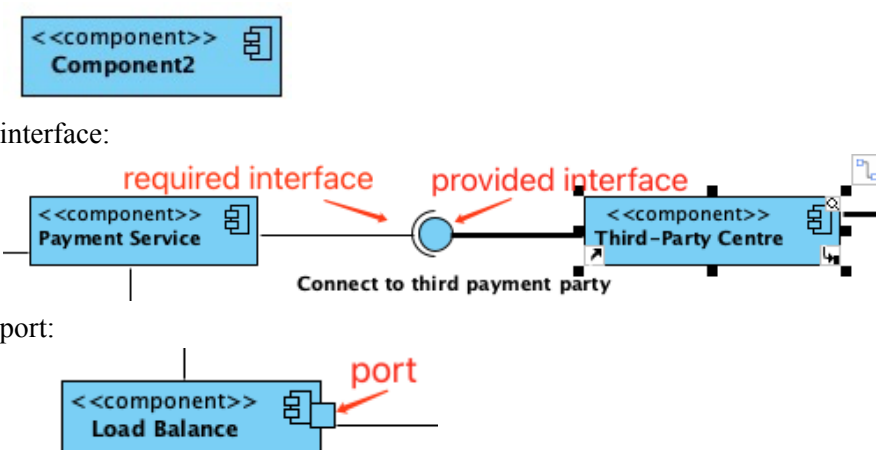
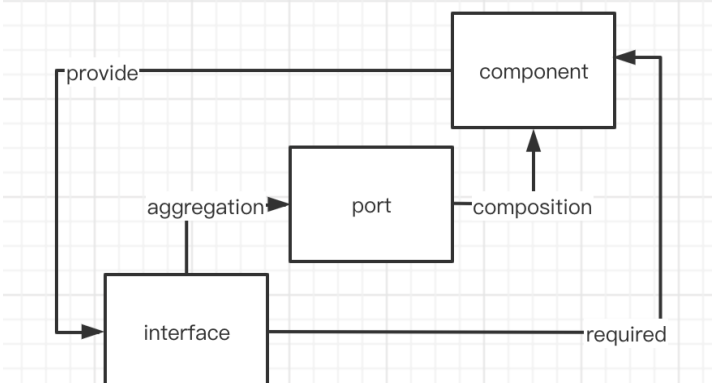
We have connected the blocks with arrows to show the request flow for better comprehension. We only defined the relation between each subsystem. Furthermore, by default, components like AI Diagnosis and Online Robot Consulting Service in the same subsystem can not communicate with each other. There are a few different flows in the basic service Layer. This is because the AI-based subsystem (includes AI Diagnosis and Online Robot Consulting Service) need to use the notification service (non-AI-based subsystem) to notify users(see AR-05). And payment or notification services need to request the third parties. The third-party centre will communicate with all the third-party systems. (NB: third party system is outside our system and it does not belong to any layer)

3. Viewpoints

3.1 Viewpoint 1 - Logical Viewpoint

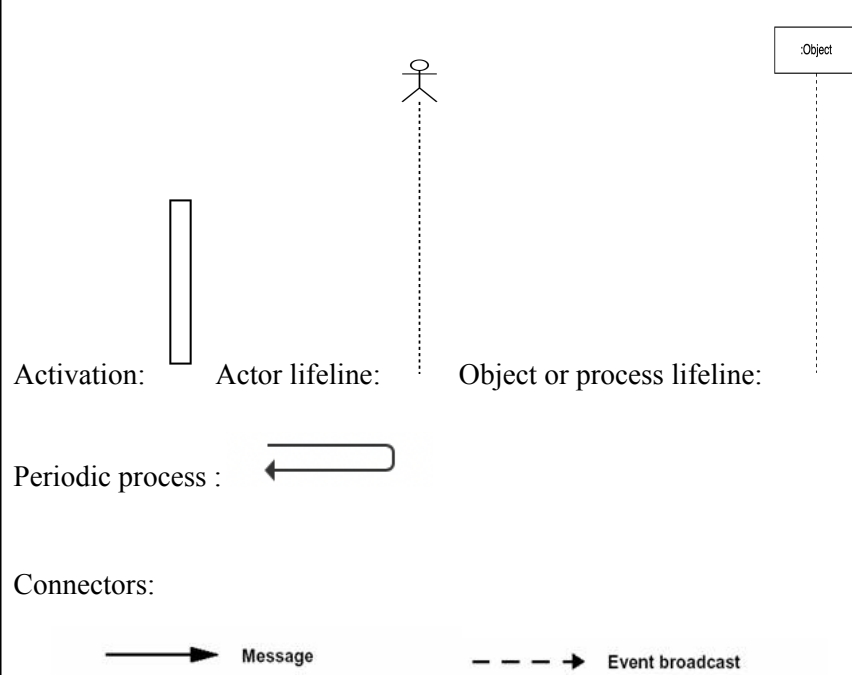
Table 3.1 Descriptions of Logical Viewpoint

Field	Description
Viewpoint Name	Logical Viewpoint
Short Name	viewpoint-03-01
Viewpoint Description	This viewpoint focuses on connecting with the functionality that the health suite system provides to end-users(patients, caregivers). And it also shows the subsystem how to communicate with each other. For example, It illustrates how the payment service connects to third parties.
Addressed concerns	<ol style="list-style-type: none">1. Healthcare Providers: Integrate data about patients from different sources, for instance, different hospitals, caregivers, etc and store them in one data cloud.2. Security Overwatchers: All the data like patients' health conditions and bill information etc should be accessed with authorization.3. Payers: The way of payment should be diverse.4. Healthcare Providers: All the devices should be centralized and accessible (proven identity of the user and authorized by the patients).5. Caregivers: Efficient and fast way to help doctor diagnosis.
Stakeholders	Healthcare Providers(SH-01), Security Overwatchers(SH-05), Payers(SH-04), Caregivers(SH-03), Patients(SH-02)
Modeling techniques	The modelling language for this view is captured as UML component diagrams which describe the decomposition of structural elements. For example, the components of the system and interactions between them.

<p>Notation</p>	<p>the view uses a UML component diagram component:</p>  <p>interface:</p> <p>port:</p> <p>Relationships:</p> <p>association: ————— composition: —————◆</p> <p>aggregation: —————◇</p>
<p>MetaModel</p>	 <p>the component provides the interface and the interface also may be required by the component. A port is used to provide a uniform port to help the interface communicate with the component.</p>
<p>Rationale</p>	<p>This view shows end-users how to interact with the system, which gives a clear idea of the system workflow and how the patient data can be managed and accessed with authorization. And it also describes how to integrate information about the patient from multiple sources. Aside from that, it explains how to interact with third parties like banks and demonstrates how to provide various payment methods. Furthermore, it shows how to manage all devices. And it presents the process of doctors managing patient data and interacting with AI framework to help them the diagnosis</p>

3.2 Viewpoint 2 - Process Viewpoint

Table 3.2 Descriptions of Process Viewpoint

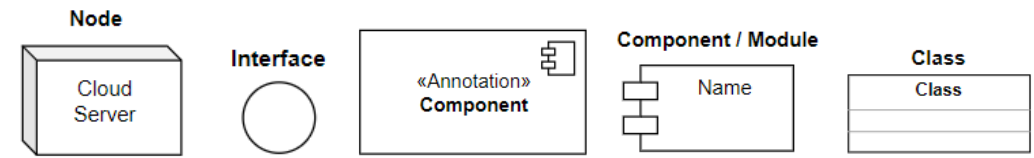
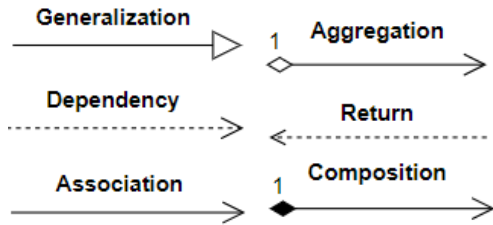
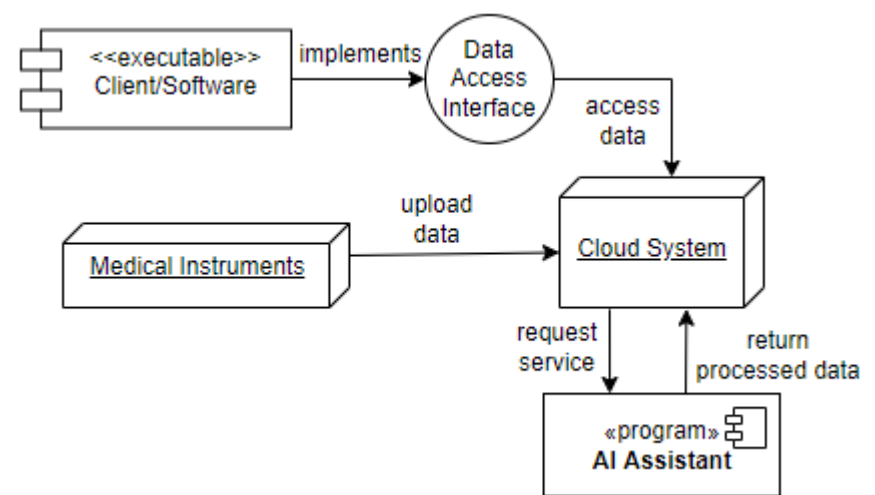
Field	Description
Viewpoint Name	Process Viewpoint
Short Name	viewpoint-03-02
Viewpoint Description	<p>The logical viewpoint, the development viewpoint and the deployment viewpoint, all describe static information about the system. Until now we have lacked a description of the dynamic behavior of the system, and the process viewpoint is used to describe the dynamic information in the system.</p> <p>The most common design tool for operational views is the UML sequence diagram.</p>
Addressed concerns	<p>1. Health providers should understand clearly the operational processes and can know non-functional requirements such as performance, system availability, etc.</p> <p>2. Security Overwatcher can have a certain knowledge of the safe deployment of the system through this viewpoint, understand the operational characteristics of users to improve robustness, security, and concurrency of the software.</p>
Stakeholders	Healthcare provider(SH-01), Security Overwatchers(SH-05)
Modeling techniques, structure/metamodel	<p>UML sequence diagram</p> <p>Process architectures can be described at the same level for different objects for different problems. When at the highest level, a process architecture can be seen as a logical network of independently executing communication programs.</p>
Notation	<p>Components:</p>  <p>The diagram shows three types of lifelines: a vertical rectangle for 'Activation', a stick figure for 'Actor lifeline', and a rectangle with a dashed line for 'Object or process lifeline'. Below these, a 'Periodic process' is shown as a horizontal line with a loop arrow. At the bottom, 'Connectors' are shown: a solid arrow for 'Message' and a dashed arrow for 'Event broadcast'.</p> <p>Activation: Actor lifeline: Object or process lifeline:</p> <p>Periodic process :</p> <p>Connectors:</p> <p>Message Event broadcast</p>

MetaModel	<pre> graph TD PP[Periodic process] -- "request (0..1 to 1..*)" --> PP PP -- "Message" --> PL[Process lifeline] PL -- "reply (1..1)" --> PP A[Activation] -- "isPartOf (1..*)" --> AL[Actor lifeline] PL -- "Message" --> AL AL -- "Event broadcast" --> PL </pre>
Rationale	<p>On the Healthcare provider’s side, using this viewpoint helped them to see the usability and feasibility of the software system so that they could make changes to suit the user's actions and experience.</p> <p>On the Security Overwatchers’ side, using this viewpoint helped them to understand the threads in which specific operations are performed, to ensure that user data is stored, to prevent information leakage and to improve the security and concurrency of the software.</p>

3.3 Viewpoint 3 - Implementation Viewpoint

Table 3.3 Descriptions of Implementation Viewpoint

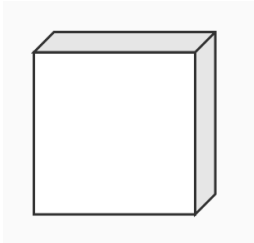
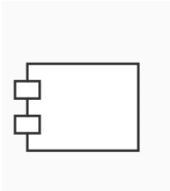
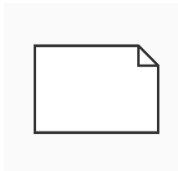

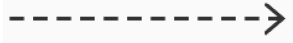

Field	Description
Viewpoint name	Implementation Viewpoint
Short Name	viewpoint-03-03
Viewpoint Description	The implementation viewpoint describes the subsystems which are responsible for different functionalities like the IoT system with smart wearable devices.
Addressed concerns	<ol style="list-style-type: none"> 1. Subsystem architects of the system are concerned with the general architecture of the IoT system behind the smart bracelet (provides data synchronization and interaction). Its structure and the distribution of components, and the topology by which they are interconnected; 2. Implementers concerned the structure of application platforms/software and how they relate to supporting technology; 3. Engineers should know the technical description of system components, including APIs, interfaces, protocols, behaviors and other properties; 4. Implementers and installation engineers should be familiar with an implementation map demonstration of the activities identified in the usage viewpoint to the functional components, and from functional components to the implementation components; 5. Developers speed up development by understanding an implementation map for the key system characteristics that shows how data be processed and how the client controls the components.
Stakeholders	Healthcare Providers(SH-01), Caregivers(SH-03), Patients(SH-02)
Modeling techniques, notation,	The modelling language for this view is UML(Unified Modeling Language) that describes the connection of sub. For example, the subsystems and modules which play

structure/metamodel	different roles of the system and control association relation between them.
Notation	<p>Components:</p>  <p>Relation connection:</p> 
MetaModel	
Rationale	<p>This view shows how different modules interact with each other, including the access and control relation between them. It shows some of the implementation details of the entire software architecture.</p> <p>On the Health Provider side, Edge Devices (with Sensor) Tier will provide smart wearable devices (electronic bracelets). It is an IoT subsystem, and the view describes the controls by APIs and interacts with cloud servers.</p> <p>On the Caregiver's side, manually recorded disease condition data and medical instruments data need to be uploaded by the caregivers to cloud-based server databases. Platform/Software Tier, online communication happens on both the Caregiver's side and the Patient's side, the platform gives automatic reminders and online booking services, etc.</p> <p>Icons representing different modules in view will have connection links with each other and the control relation between different stakeholders and functional components.</p>

3.4 Viewpoint 4 - Deployment Viewpoint

Table 3.4 Descriptions of Deployment Viewpoint

Field	Description
Viewpoint Name	Deployment Viewpoint



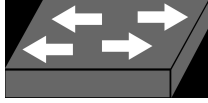





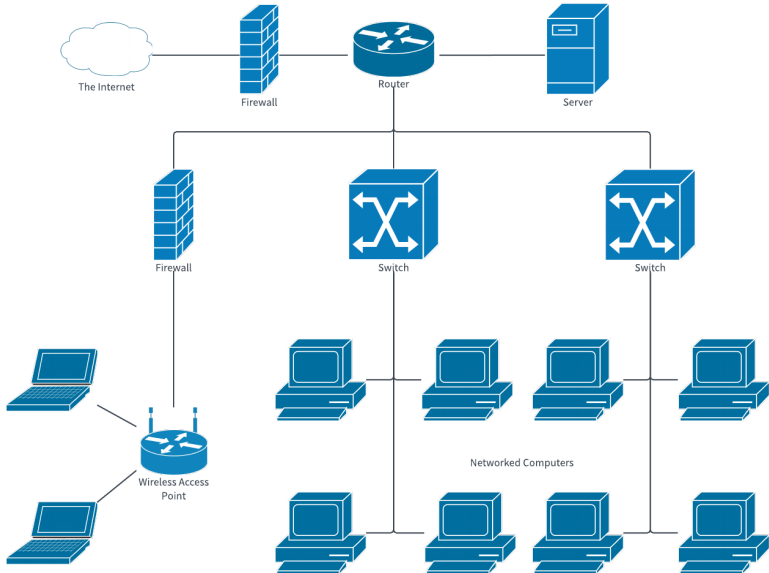
Short Name	viewpoint-03-04
Viewpoint Description	This viewpoint focuses on the transition from software elements to hardware in live operation. It defines the physical environment it will run in, such as processing nodes, network interconnections and disk storage. It also defines technical environment requirements for each processing node and the mapping of elements to the runtime environment that will execute them.
Addressed concerns	<ol style="list-style-type: none"> 1. High throughput of data transfer: the set of hardware nodes in runtime models, such as how nodes connect via interfaces and which software elements are on which hardware nodes should be organized in an efficient way to achieve the high throughput of data transfer for Healthcare Providers to process data quickly and patients or caregivers can access data in time. 2. Trade-offs between performance and security: find good dependencies among hardware and appropriate level of network layers to make quality trade-offs between performance and system security to provide the basis of implementation for Security Overwatchers. 3. High performance of AI computation: identify important client hardware requirements and add special-purpose hardware for each AI computing node, such as GPU for increasing the speed of AI computation to make Healthcare Providers complete the computation in time. 4. Fault-tolerance: adopt useful fault-tolerance measurements to achieve high reliability of this system, such as data backup in a regular manner to ensure the completion of data of patients and ensure the performance of analysis results of patients' health conditions.
Stakeholders	Healthcare Providers(SH-01), Security Overwatchers(SH-05), Patients(SH-02), Caregivers(SH-03)
Modeling techniques	The modeling language for this view is captured as a UML deployment diagram showing software elements, hardware nodes and networks.
Notation	<ul style="list-style-type: none"> • Basic Elements: <ul style="list-style-type: none"> ○ A Node:  A component:  An artifact:  An interface:  • Relationships <ul style="list-style-type: none"> ○ Dependency:  Association: 

Metamodel	<pre> graph TD SN[Server node] -- contains --> A[Artifacts that implement corresponding component] SN -- contains --> DI((data interface)) SN -- contains --> FC[Functional component] A -- depends on --> DI FC -- provides --> DI FC -- depends on --> SN </pre>
Rationale	<p>On the Healthcare Providers' side, they desire that data on the cloud can be processed and obtained in milliseconds. This view describes how the storage and transferring of data are organized in an efficient way. Besides, it also utilizes some special hardware, like GPU or TPU to iteratively retrain AI prediction models quickly. On the patients' and caregivers' side, it added data backup nodes to increase the reliability of this system. On the Security Overwatchers' side, this view describes how network access policies and users' rights assignment policies should be configured when deploying on hardware to lift up the security of this system.</p>

3.5 Viewpoint 5 - Security Viewpoint

Table 3.5 Descriptions of Security Viewpoint

Field	Description
Viewpoint Name	Security Viewpoint
Short Name	viewpoint-03-05
Viewpoint Description	<p>The Security Viewpoint focuses on ensuring that all parts are working as intended. It consists of micro and macro components defining the pre-actions and post-actions on security-related issues. Each component has to be analyzed, followed by the system as a whole to better address the concerns that arise from their interaction. More specifically, it focuses on the correct usage of the system. Unexpected requests should not be fulfilled, legitimate requests always satisfied and data should not be leaked.</p>
Addressed concerns	<ol style="list-style-type: none"> 1. Any payments made through the system must be secure and not intercepted. 2. Caregivers should be allowed to use the equipment without any issue that might endanger the lives of patients. 3. Patients should feel that their data is safe and private, along with receiving the best care. 4. Healthcare providers should collect data and analyze them to provide the best insights for caregivers. 5. Security Overwatchers are the ones who should enforce tactics to ensure the above are satisfied.
Stakeholders	Security Overwatchers(SH-05), Healthcare Providers(SH-01), Caregivers(SH-03), Patients(SH-02), Payers(SH-04)

Modeling techniques	Network diagram with custom modeling language. Does not have a specific standard to follow. Notation is based on context and is dynamic.
Notation	<ul style="list-style-type: none"> Database:  Router:  Switch:  Server:  Firewall:  Device:  Device:  Connection: 
Metamodel	
Rationale	<p>On the Healthcare Providers' side, the functionality of the infrastructure must be uninterrupted. How securely data is collected and analyzed to provide complete history and the best insights are described by this view.</p> <p>On the Caregivers' side, collected data and corresponding insights have to be readily available. It is depicted by this view how the machines must not suffer from denial of service attacks or other mischievous acts that hinder their job.</p> <p>On the Patients' side, privacy is a value respected a lot. This view describes how personal data can not be leaked or altered as the reputation and life of the patient are endangered.</p> <p>On the Payers' side, it is described how transactions are processed securely ensuring that the correct amount is always charged and received.</p> <p>On the Security Overwatchers's side, it is clear that the system functions as a whole, with processes being dependent on one another. In their view, it is described how to ensure that the intermediate parts allow the smooth operation of the entity.</p>

4. Views

4.1 View 1 - Logical View

Primary presentation

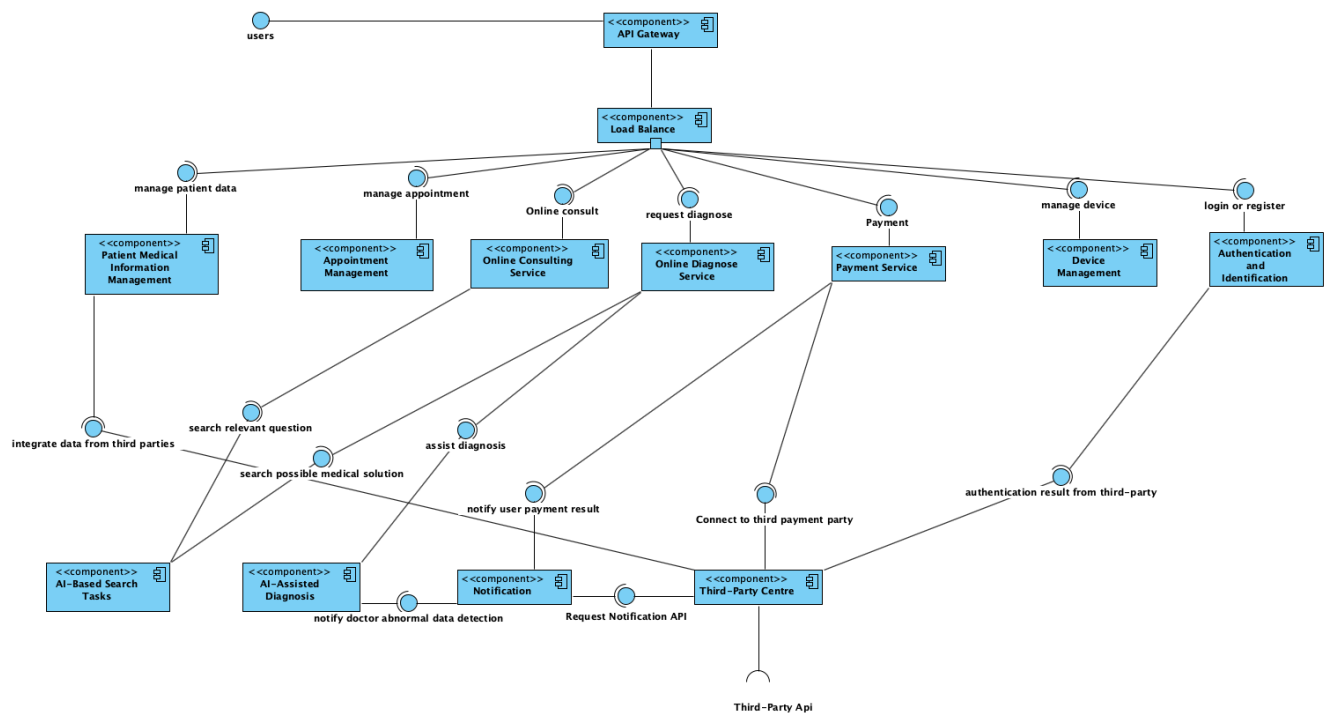


Figure 4.1 Logical View

This view based on the Logical Viewpoint(viewpoint-03-01) shows that all the requests of the user will go through the API gateway and Load Balance. Before using the system, users should log in by using the authentication and identification module. The request will be distributed to the relevant module by using Load Balance.

As for concerns in the viewpoint:

1. It has an interface called “integrate data from third-party” which is provided by the third-party centre and required by the Patient Medical Information Management. It is used to integrate all the patient data from third parties.
2. The authentication and identification module will control if the request needs authorization.
3. For payment, we use the third-party centre to communicate with banks or third-party payments like PayPal and the payment module will be responsible for providing diverse ways to pay.
4. The system has a device management model which is used to manage all kinds of devices.
5. The system has two AI modules to improve the efficiency of doctor diagnosis. They are “AI-Assisted Diagnosis” which helps the doctor find the abnormal data about the patient's physical condition and “AI-Based Search Tasks” which can search and provide possible treatment based on diagnosing.

Element catalog

Table 4.1 Catalog of Elements in the Logical View

Element	Description
Users	This element is the end-user of this system. They may be patients, doctors, insurers, etc.
Authentication and	This component is mainly responsible for the user login and identifying their role.

Identification	
log in or register interface	This interface provides the interface that the user login or registers to the system provided by Authentication and Identification
authentication result from the third-party interface	This interface will be called when the user login by using DigiD or UZI-pass provided by the Third-party Center.
API Gateway	This component is used to identify the domain to which the request belongs.
Load Balance	This component can help distribute the request and reduce the server's load.
Payment interface	This interface provides the interface related to payments. For example, check payment information, access bill information, calculate the suitable price or pay bills.
Payment Service	This component mainly provides the functions related to payment
Connect to third payment party interface	This interface provides the interface that calls the payment APIs provided by the third party center.
Third-party Center	This component is responsible for communicating with third parties such as PayPal, banks, SMS providers, etc.
Third-party API	This interface represents the third party's API, and the system uses those API to communicate with the third party.
notify user payment interface	This interface will be called when the user finishes their pay provided by the Notification component.
Notification	This component is responsible for the notification function. For example, before notifying the user, it will check this type of notification by which channel (app, email or SMS).
request diagnose interface	This interface provides an interface that assists users to diagnose and checks related diagnose results provided by the Online Diagnose Service component.
Online Diagnose Service	This component is responsible for providing the diagnosis relevant functions and helping doctors to diagnose. For instance, it can automatically highlight the abnormal index of a patient's test result and provide suggestions of possible disease and treatment.
assist diagnosis interface	This interface provides an interface that uses an AI diagnosis assistant provided by the AI-Assisted Diagnosis component.
AI-Assisted Diagnosis	This component is responsible for automatically analyzing the patient's current condition and test result. And if it detects abnormal data, it will alert the doctor.
notify doctor abnormal data detection interface	This interface provides an interface that AI-Assisted Diagnosis automatically sends alerts to doctors provided by the Notification component.
search possible medical solution interface	This interface provides an interface that uses the AI framework provided by the AI-Based Search Tasks component.
AI-Based Search Tasks	This component is responsible for automatically answering the user's question to reduce the workload of support staff. And it is also associated with AI-Assisted diagnosis, which searches and provides possible treatment based on diagnosing.

Online consult interface	This interface provides an interface that helps users manage appointments provided by the Online Consulting Service component.
Online Consulting Service	This component provides the function that helps patients consult a doctor remotely. And it also provides a robot that helps the doctor or medical staff answer some basic and repeated questions.
search relevant question interface	This interface provides an interface that helps users manage appointments provided by the AI-Based Search Tasks component
manage appointments interface	This interface provides an interface that uses the AI framework provided by the Appointment Management component.
Appointment Management	This component is responsible for managing appointments with the doctor. For example, it can help patients make or cancel an appointment.
manage patient data interface	This interface provides an interface that helps users manage patient data provided by the Patient Medical Information Management component
Patient Medical Information Management	This component is responsible for managing patient medical information. And the doctor can access their patient's medical record with patient authentication and the patient only can access their own medical record. Furthermore, patients cannot change or delete their records. And doctors only can add or append the record.
integrate data from third parties interface	This interface provides an interface that requests and integrates data about the patient from the third party provided by the Third-party Center.
manage device interface	This interface provides an interface that helps users manage all kinds of devices provided by Device Management.
Device Management	This component is responsible for managing devices. The patient can easily use its function to manage their own device like wearable devices etc. And the doctor or hospital staff can use its function to manage the device owned by the hospital with authentication such as MRI etc.

Because the relationship between two elements is alike and can be classified into two categories, the table below focuses on describing these three classes.

Table 4.2 Catalog of Relation between Element in the Logical View

Relation	Description	Source	Target
from port or component to interface	This type of relation represents a request that called from one component to another service by using the API provided by the interface	port or component	interface
between interface to port or component	This type of relation represents a request for the functionality that the target component provided.	interface	port or component

[Design decisions and rationale \(for this view\)](#)

1. Payment

Table 4.4 Design decisions and rationale for Logical View

ADD04-01-02: Payment	
Issue	For increasing users' experience, the payment methods need to be more diverse. It should be easy to add a new payment method into the system.
Decision	Using plugins to provide the payment method
Status	Decided
Group	structural
Assumptions	All the payers are able to use electronic payment like PayPal or online banks.
Constraints	It needs to meet the third-parties requirements to integrate them into the system.
Positions	Every time the system adds a new payment, it will require developing a new module.
Argument	By using plugins, it will save a lot of cost of development and is easy to reuse and maintain. If the programmer develops a new module to implement a new payment, it will introduce a lot of unknown errors then it may endanger the system.
Implications	It requires the developer to have the good abstract ability to implement it and it cannot be customized freely
Related decisions	ADD06-03: Third-party service & Third-party centre

4.2 View 2 - Process View

Primary presentation

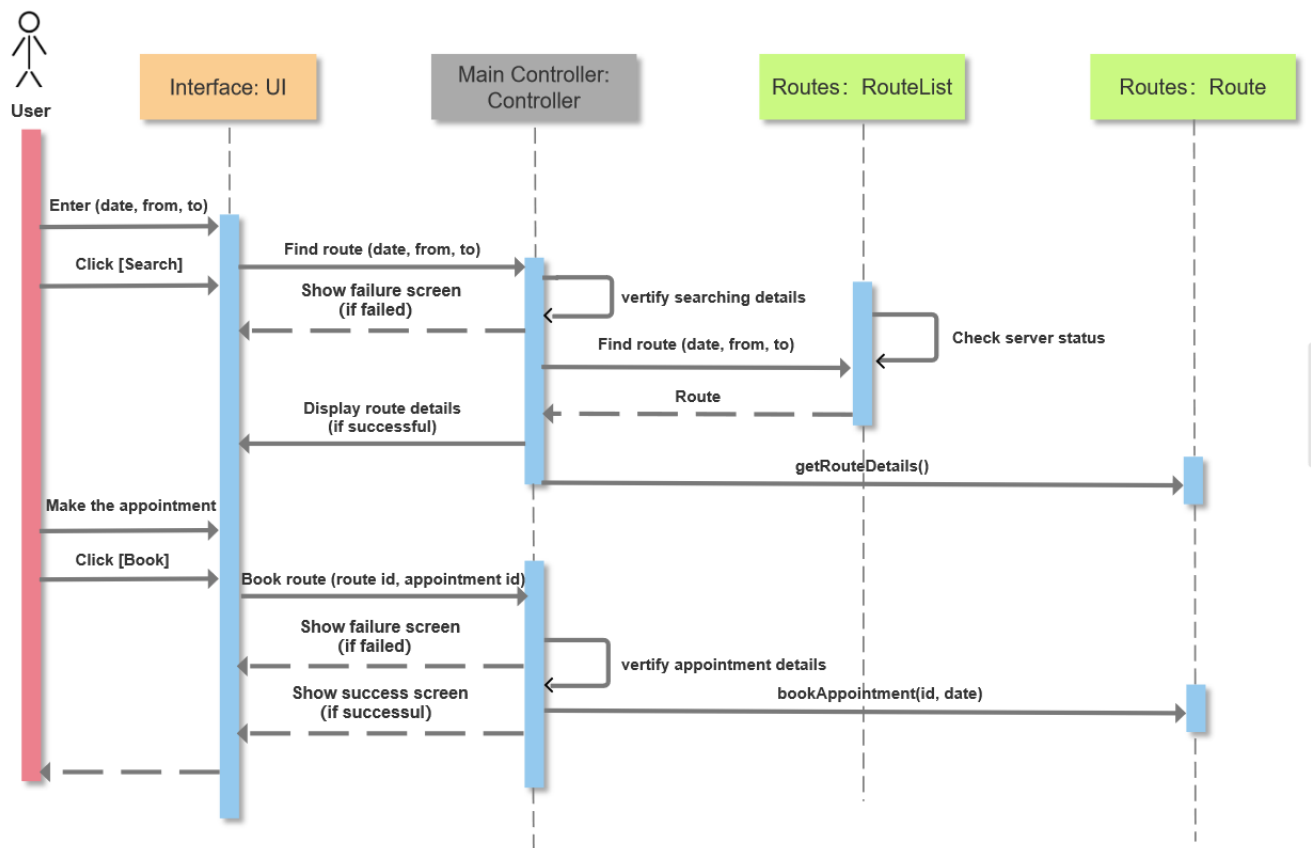


Figure 4.2 Process View

This view is based on the Process Viewpoint (Viewpoint-03-02) and is used to describe the dynamic information of the system. It uses the 'Actor Lifeline' to represent the user's icon. The vertical rectangle downwards represents 'Activation'. The layers connected to the user are 'Process Lifelines', with colours indicating the interface, main controllers and routes.

Some of the user's basic operating procedures are listed. For example, searching for data and making appointments to set up protection mechanisms to secure the data, and designing the user's operational processes through this process.

In particular, the process view features a 'Periodic Process' to check the status of the server so that if the current server is not available, it can return to the controller to call another server, improving the performance and availability of the system.

Element catalog

The elements appearing in the view are described in the element catalog.

Table 4.5 Catalog of Elements in the Process View

Element	Description
Interface: UI	The user interface of the user's terminal provides a range of interactive operations.
Controller	The main controller acts as an intermediate layer to send the request to the backend and return the reply to the UI interface.
RouteList	RouteList stores multiple routes in order to represent the relevant routes for input and output.

Route	The underlying implementation of the functional route.
User	Users of this system include patients, doctors, etc.

The relationship between two elements is alike and can be classified into two categories, the table below focuses on describing these three classes.

Table 4.6 Catalog of Relation between Element in the Process View

Relation	Description	Source	Target
From user to interface [search data]	This relation represents patients can search data when they input the data and data in the range.	User	Interface UI
Show failure screen	If the operation fails, it will show the screen of failure information.	Controller	Interface UI
From user to interface [make the appointment]	This relation represents patients who can make the appointment of patients by clicking the book button.	User	Interface UI
Route	Return the inquiry route.	RouteList	Controller
getRouteDetails	Display the progress of GetRoute.	Controller	Route
Verify searching details	Verification of operations whether successful or not.	Controller	Controller
Verify appointment details	Verification of operations whether successful or not.	Controller	Controller
Display route details	If the query is completed, it will show the results for users.	Controller	Interface UI
Show success screen	If the operation succeeds, it will show the screen of successful information.	Controller	Interface UI
Check server status	Check the usage status of the server	RouteList	RouteList

Design decisions and rationale (for this view)

1. Main controller

Table 4.7 Design decisions and rationale for Process View

ADD04-02-01: Main controller	
Issue	The system requires front-end user interface and back-end operation control.
Decision	Through the main controller, the system is able to transmit the user's requests to the back office and return the results to the user.
Status	Decided
Group	Structural

Assumptions	Set up the flow pipeline for the whole system to enable requests and replies.
Constraints	Communication is prone to failure when multiple tasks are running simultaneously.
Positions	When a new task user interface is transferred directly to the backend, the backend then transfers the results to the user interface.
Argument	Running multiple tasks without going through the controller can easily cause system crashes.
Implications	Ability to make simultaneous requests and replies in multi-tasking, increasing system stability and performance.
Related decisions	ADD06-01: API Gateway, ADD06-02: Load balance, ADD06-03: Third-party service & Third-party centre

2. Make the appointment

Table 4.8 Design decisions and rationale for Process View

ADD04-02-02: Make the appointment	
Issue	When a patient wants to make an appointment with a doctor, he/she can do so through the self-registration of the system.
Decision	Make the appointment by themselves and easily for them to guarantee the appointment
Status	Decided
Group	structural
Assumptions	Users can use the button 'book' to make an appointment.
Constraints	A doctor's office cannot be booked by two patients at the same time during a particular time slot.
Positions	Appointments are available at offline booking windows or booking machines
Argument	Online appointments are very convenient for patients without queuing and less pressure on staff.
Implications	It has improved the efficiency of staff and provided great convenience for patients.
Related decisions	ADD06-01: API Gateway, ADD06-02: Load balance

4.3 View 3 - Implementation View

Primary presentation

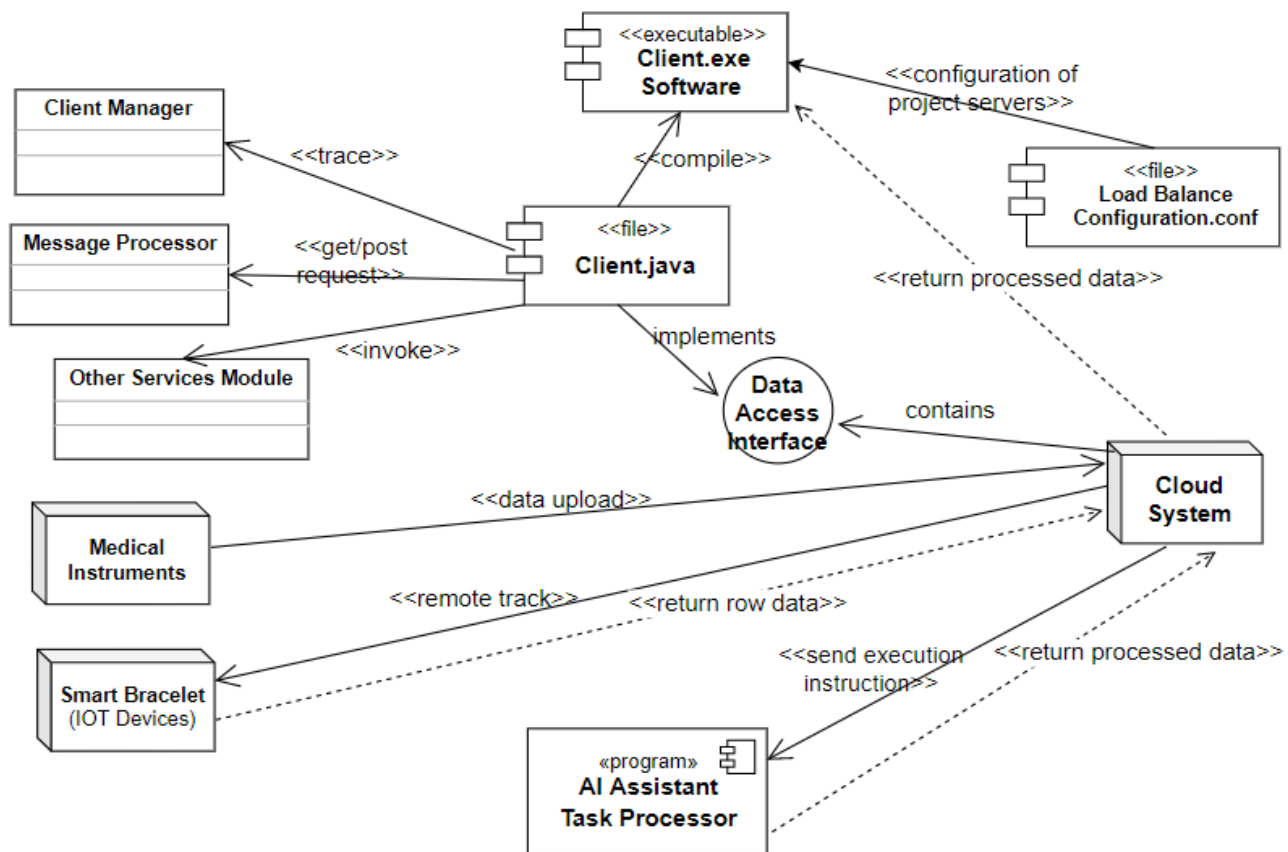


Figure 4.3 Implementation View

This view is based on the Implementation Viewpoint(viewpoint-03-03) and represents the overview of how the underlying implementation modules are related to each other. Source files are included, which is the underlying level implementation of each module, they can be integrated into client.java and then be compiled to be an executable application. Cloud system is responsible for the data storage so it keeps interacting with medical instruments no matter if they are local or remote. The AI assistant is built-in on the cloud, when it is necessary, it will be invoked to help process data and achieve some goals like disease prediction. Client Manager, Message Processor and the other service modules are the components that are the underlying functional partition. The Data Access API defines the data authentication and identity validation function and any other functional definition. The solid arrows between components demonstrate the interactive relationship, dotted arrows mean there are some returns given back to the source component.

Element catalog

Table 4.9 Catalog of Elements in the Implementation View

Element	Description
Client.java	This element is the source file that contains all invocations of libraries and other dependent files to be compiled to generate the client program(user-end, the entire software).
Client.exe	This component is the final executable program for the user-end. It can be seen as the software which integrated all of the functions of the system and it is the gateway where data interaction occurs.
Data Access Interface	This component is the interface that defines some of the necessary methods to access the data. For example, data uploading and downloading, data transfer, analysis permission application.

Cloud System	This component is the entire cloud server system which contains the database and AI task processor to help with medical recorded data processing(big data analysis, image processing, disease prediction etc.).
AI Assistant Task Processor	This cloud-based processor consists of classified programs which use artificial intelligence techniques for processing various row data and returns with processed data for further research.
Smart Bracelet	This is the remote smart device that is provided by the health provider and to monitor the patient's underlying physiology. It can upload data in real-time as an individual IoT system.
Medical Instrument	This is the physical device that is located in medical institutions such as hospitals. It is generally operated by the doctors(technicians) and gives feedback as outputting the results of patients. Then the data will be recorded and uploaded into the database for possible further processing.
Client Manager	This component is responsible for the back-end system that contains the security module, data interaction and client-side stability functionality module.
Message Processor	This element implements and guarantees the function of online real-time communication and maintains its stability.
Other Function Module	Such as doctor appointment service, medication reminder service, etc. Besides, it Includes other services that will be deployed.
Load Balance Configuration.conf	The configuration file defines the scheduling of server resources, node weights, polling policies etc. for the load balance section for the entire system.

Table 4.10 Catalog of Relation between Elements in the Implementation View

Relation	Description	Source	Target
Functional service invocation	This type of relation represents that the Client integrates all the functions and the functional module is the underlying implementation of the various services provided by the system.	Client	All functional modules(Client Manager, Message Processor, Other Service Modules)
Data access control	This type of relation represents that the cloud server needs to invoke the data access API to implement the complete control of users who have different identities.	Cloud Server	Data Access API
Data consistency of various devices	This relation is about the synchronization of the cloud data from different devices(Medical equipment in the hospital and IoT).	Devices	Cloud server
Configuration of servers for load balance	This relation aims to provide the configuration which controls and displays all of the servers to guarantee the load balance of the system.	Configuration Files	Client

Cloud-based AI service	This relation is all happening on the cloud server, the AI task is pre-set by building underlying code. The cloud server has cooperative work with the AI assistant and the request instruction is sent by the client.	Client	AI Assistant Task Processor
------------------------	--	--------	-----------------------------

Design decisions and rationale (for this view)

1. Client

Table 4.11 Design decisions and rationale for Implementation View

ADD04-03-01: Client(ultimate executable software)	
Issue	Final integration of all functions. The final layer for all components.
Decision	The most important component contains the front-end and the back-end. It is a collection of all the functionalities, a platform for various services and user operations. The ultimate executable software system for all development files.
Status	Decided
Group	Structural
Assumptions	All users operate on this platform and all data interactions generated by the service are through this client.
Constraints	It needs to maintain data synchronization and server stability.
Positions	Website system, entirely online.
Argument	All operations and data interactions are gone through the Client.
Implications	Try to avoid involving the user in too many decisions on issues and finding the software too cumbersome, developers should consider this part well. Safeguard the legal rights of each user and do a good job of data security, as health data is often very sensitive
Related decisions	ADD06-01: API Gateway, ADD06-02: Load balance, ADD06-03: Third-party service & Third-party centre, ADD06-04: Unified Access Controller

2. Cloud System

Table 4.12 Design decisions and rationale for Implementation View

ADD04-03-02: Cloud System	
Issue	The most important data interaction centre, the main execution site for data processing.
Decision	This component provides data-based services for both patients and medical professionals. It can be seen as a data warehouse and the machine learning task processor for providing processed data for disease prediction, social medical data analysis etc.

Status	Decided
Group	Structural
Assumptions	All kinds of data interaction (from different devices) happen here, data will be stored and re-organized, then if necessary, AI assistants will be involved in data processing tasks. Cloud servers will return processed data back or just respond when direct data access is achieved. All users will be assigned different permission of accessing data by their identities, the cloud system needs to validate and correctly return data.
Constraints	It needs to maintain data security like some encryptions. Data synchronization may take a little more time and it should guarantee the stability of the server.
Positions	Local based data storage system.
Argument	Data consistency, timeliness and security are top priorities. There may be a variety of contingencies that can cause server stability and data to be compromised. Secondly, ancillary functions like cloud-based AI prediction etc. are considered.
Implications	Low response time, high consistency of data. Good load balancing distribution, high disaster recovery capacity of cloud servers to handle high concurrency, etc. To guarantee the whole system is safe, fast and available.
Related decisions	ADD06-01: API Gateway, ADD06-02: Load balance, ADD06-04: Unified Access Controller

4.4 View 4 - Deployment View

Primary presentation

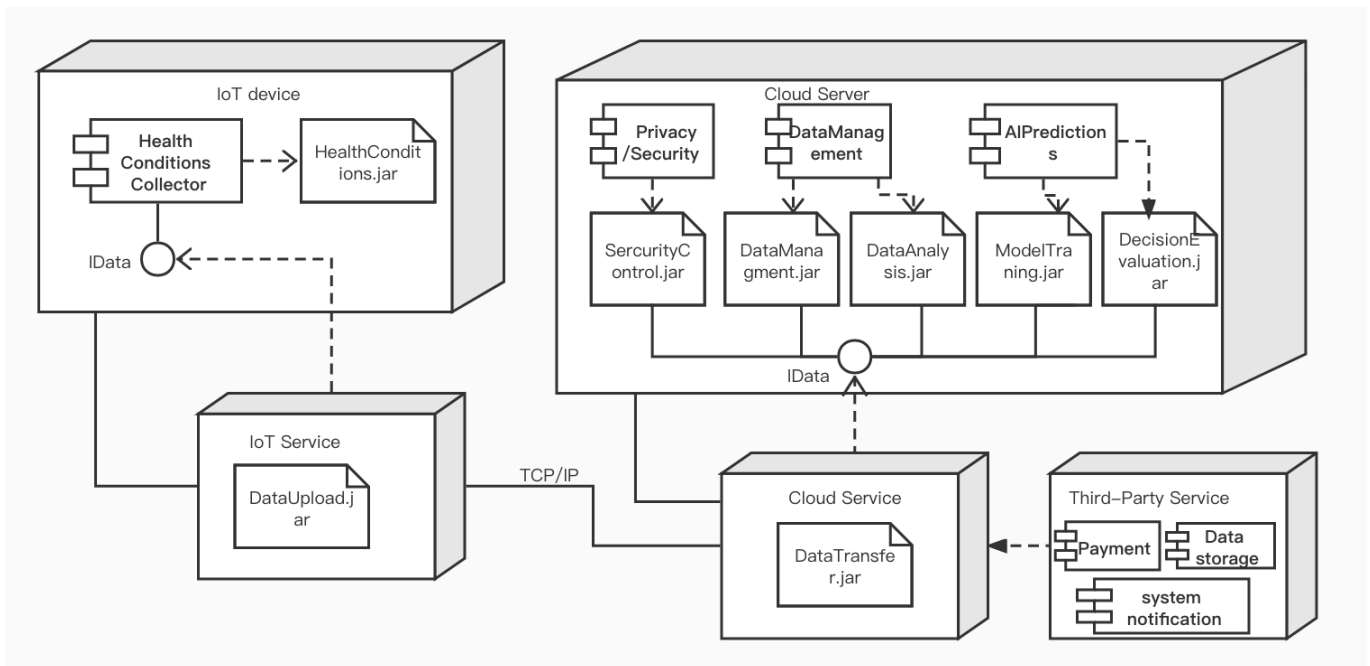


Figure 4.4 Deployment View

This view based on the Deployment Viewpoint(viewpoint-03-04) shows that this system mainly consists of two servers, the IoT device server and the cloud server. The IoT device server is responsible for uploading data in IoT devices to the cloud through the data interfaces. The cloud server is responsible for all other services, including the data management and AI diagnosis prediction models. There are two kinds of data interfaces in this system, where

one is provided by the IoT service and another is provided by the Cloud Service. The third-party service is responsible for payment-related transactions, system notifications and data storage.

Element catalog

Table 4.13 Catalog of Elements in the Deployment View

Element	Description
IoT device	This element is the container of elements needed to collect the health conditions of patients and to transfer data to the cloud through the IoT service.
Health Conditions Collector	This element is the component that implements the function of collecting the information about the health conditions of patients and storing them in the IoT device server.
HealthConditions.jar	The packed java file implements the functions required by the component Health Conditions Collector.
IData	It's the interface for data transferring between server containers and service containers.
IoT Service	It's the service container that is responsible for uploading health condition data in local IoT devices to the clouds.
DataUpload.jar	The packed java file implements the functions required by the container IoT service.
Cloud Server	This element is the container of elements on the clouds, which are needed to ensure the security of this system, manage data and perform AI predictions.
Privacy/Security	The component that implements the function of lifting the security of this system, including user authorization, payment protection, etc.
SecurityControl.jar	The packed java file implements the functions required by the component Privacy/Security.
Data Management	The component that implements the management of data about the health conditions of patients, including the storage and queries of data and the analysis of data.
DataManagement.jar & DataAnalysis.jar	The packed java file implements the functions required by the component Data Management.
API Prediction	The component that implements the AI diagnosis predictions for patients, including the pre-training, decision making and accuracy evaluation of decisions.
ModelTraining.jar & DecisionEvaluation.jar	The packed java file implements the functions required by the component API Prediction.
Cloud Service	It's the service container that is responsible for transferring health condition data between local IoT devices and the clouds and payment-related data between the clouds and third-party services.
DataTransfer.jar	The packed java file implements the functions required by the component Cloud Service.
Third-Party Service	It's the service container that is provided by a third party, which is responsible for payment-related transactions, system notifications and data storage.
Payment	The component provided by the third-party that implements the payment-related

	transactions.
Data storage	The component provided by the third-party that is responsible for temporary data storage.
System notifications	The component provided by the third-party that implements the system notifications according to the instructions from the cloud server.

Table 4.14 Catalog of Relation between Elements in the Deployment View

Relation	Description	Source	Target
IoT data encapsulation	This type of relation represents that IoT data on IoT devices are encapsulated by IoT service	IoT device server	IoT service
IoT data transferring	This type of relation represents that IoT data are transferred to the cloud service by IoT service	IoT service	Cloud service
Cloud data transferring	This type of relation represents that data stored in the cloud are transferred the third-party by Cloud service	Cloud service	Cloud server
Payment and notification implementation	This relation represents that payment transactions and system notifications in this system are handled by the third-party centre according to the instructions from the cloud service	Third-party service	Cloud service
Data transferring and exchange	This relation represents that data transferring between different functional components in these two servers are implemented by the data interfaces	Functional components in two servers	Data interface

Design decisions and rationale (for this view)

1. Two main separate servers:

Table 4.15 Design decisions and rationale for Deployment View

ADD05-04-01: Two main separate servers	
Issue	Deployment of main servers.
Decision	There are two main containers in deployment, including the IoT device server, which is responsible for local data collection, and the cloud server, which is responsible for all other services.
Status	Decided
Group	Structural
Assumptions	Every patient is equipped with one IoT device to be able to use all functions provided by this system.
Constraints	It needs to maintain data synchronization between these two servers.

Positions	When users cannot access IoT devices, equipment in the hospital can also be used to collect data about patients' health conditions.
Argument	IoT devices can collect much larger amounts of data than equipment in the hospital and are more convenient to use, thus an IoT device server is the first choice in this system. All other functions are implemented in the cloud server to increase the maintainability of the system.
Implications	The maintainability of this system may increase due to a large number of IoT devices.
Related decisions	ADD06-01: API Gateway, ADD06-04: Unified Access Controller

2. Data interfaces:

Table 4.16 Design decisions and rationale for Implementation View

ADD04-04-02: Data interface	
Issue	Data transferring needs to be implemented through protected data interfaces, so that sensitive data about patients' health conditions or payments need to be protected.
Decision	The data transferring communications between different servers in this system are encapsulated and implemented through data interfaces provided by server containers and corresponding data service containers.
Status	Decided
Group	Security
Assumptions	All users operate on this platform and all data interactions generated by the service are through this client.
Constraints	An IoT device server and the cloud server both need to encapsulate data and provide data interfaces.
Positions	IoT data can also be transferred and stored directly to third parties with users' consent.
Argument	Encapsulated Data communications and data interfaces implemented by service containers, i.e. IoT service and Cloud service, can help to increase the speed of data processing and lift up the security of data when transferring on the network.
Implications	The maintainability of this system may increase due to the encapsulation of data and management of data interfaces.
Related decisions	ADD06-01: API Gateway, ADD06-03: Third-party service

4.5 View 5 - Security View

Primary presentation

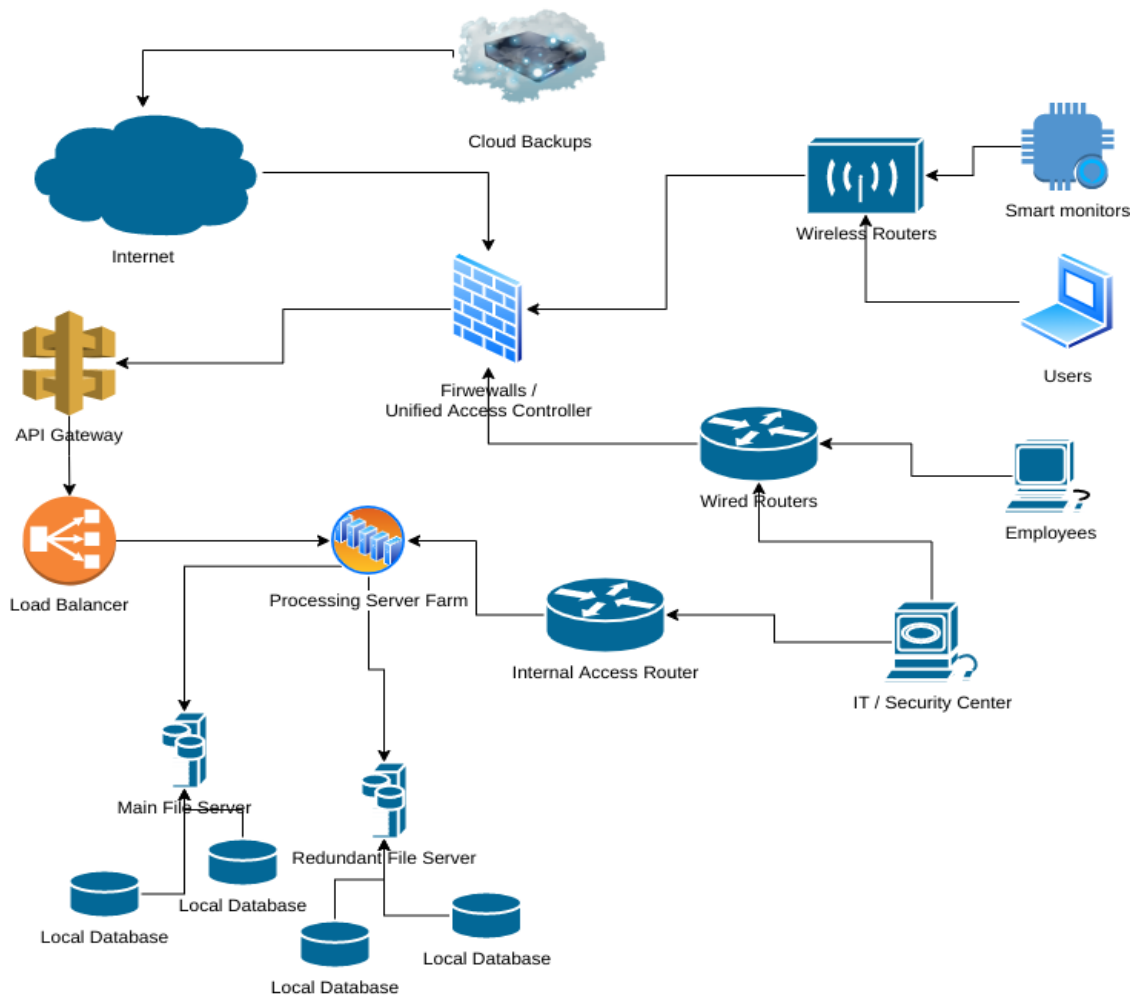


Figure 4.5 Security View

This view is based on the Security Viewpoint(viewpoint-03-05) and shows the layout of the network. All accesses to the infrastructure pass through Firewalls and the Unified Access Controller, which check whether the requests should proceed to their destinations. To access the internal infrastructure, the API Gateway is available which forwards the requests to the Load Balancer. This in turn chooses the Server in the Server Farm based on some policy, like the lowest load. The Server Farm consists of multiple types of servers, like AI processors, Web Servers, Proxies, Authentication Systems, Payment Systems and Telephone Systems. Directly connected to the Server Farm are the main File Server and the Redundant File Server, both of which consist of multiple Databases. All data is stored there, with sensitive ones being encrypted. There are 4 endpoints accessing the internal infrastructure. The Internet, where the cloud provider keeps another off-shore backup of data. The Wireless Routers, where Smart Monitors and generic Users are connected to. The Wired Routers, which provide access throughout the site and are used by Employees and the IT/Security Center. The IT also has access to the internal infrastructure through an Internal Access Router, for management purposes.

Element catalog

Table 4.17 Catalog of Elements in the Security View

Element	Description
Firewalls/Unified Access Controller	This element is placed centrally, so all accesses go through it, allowing them to be completed or not.

API Gateway	This element is used to identify the domain to which the request belongs.
Load Balancer	This element can help distribute the request and reduce the server's load.
Server Farm	This component consists of multiple Servers, where each has a specific purpose.
AI Processor	This element is responsible for processing AI-related computations.
Web Server	This element is responsible for Web-related processing and hosting.
Proxy	This element is responsible for proxying requests to other Servers and log accesses.
Authentication System	This element is responsible for keeping the access permissions and informing other components like Firewalls of the rules.
Payment System	This element is responsible for payment processing.
Telephone System	This element is responsible for the internal telephone infrastructure of the site.
File Server	This component is used to attach physical disks to the infrastructure.
Local Database	This element is used to store data and is found in the File Servers.
Internet	This element represents the Internet and WAN connection.
Cloud Backup	This component is responsible for keeping an off-shore backup of data for redundancy.
Wireless Router	This element provides wireless access to the infrastructure. Used by Smart Monitors and generic Users.
Wired Router	This element provides wired access to the infrastructure. Used by Employees.
Internal Access Router	This element provides internal access to the IT/Security Center, for management purposes and higher privilege. Should be physically secured and inaccessible by others.
Smart Monitor	This component is used to get data from patients by IoT devices.
User	This element represents the generic User, like patients and visitors.
Employee	This element represents the staff with access to workstations, like Doctors, Accounting, Management and IT.
IT/Security Center	This element represents the team responsible for maintaining the infrastructure and tending to the security needs. Has more access to the infrastructure than anyone else.

Table 4.18 Catalog of Relation between Elements in the Security View

Relation	Description	Source	Target
from endpoints to servers	This relation represents a request that was made from an endpoint such as User or Employee to some processing component.	endpoints	servers

from file server to cloud backup	This relation represents the data that is sent from the file server to the off-site backup.	file server	cloud backup
from authentication system to firewalls	This relation represents the rules sent to a firewall to dynamically update permissions	authentication system	firewall

Design decisions and rationale (for this view)

1. Data

Table 4.19 Design decisions and rationale for Security View

ADD04-05-01: Data	
Issue	The system needs to avoid the loss of data at all costs. Data should be available at all times. Sensitive data is confidential.
Decision	At least 2 copies of data should be kept locally and 1 off-site.
Status	Decided
Group	Security
Assumptions	Disks eventually crash, automated attacks are unavoidable.
Constraints	Data should be encrypted before being saved locally or off-site.
Positions	Not keeping backups or not encrypting data.
Argument	Keeping multiple copies of data ensures that in case of disk failure or corruption, immediate recovery is guaranteed.
Implications	Loss of data in multiple positions, more than what redundancy can recover or before recovery has been handled.

2. Infrastructure Separation

Table 4.20 Design decisions and rationale for Security View

ADD04-05-02: Infrastructure Separation	
Issue	Anyone accessing the infrastructure should have the least possible privileges.
Decision	The strict separation between server room and accessing endpoints.
Status	Decided
Group	Security
Assumptions	Malicious entities will try to access services and data they should not.
Constraints	Legitimate requests should be successfully served. IT has full access to infrastructure.
Positions	Having a flat network, where servers are directly accessible by users.
Argument	Firewalls should be placed centrally, intercepting any requests made from endpoints. These firewalls are updated from the Authentication System to set their rules

	dynamically. Firewalls can have integrated intrusion prevention/detection mechanisms. IT has an IoT device router for accessing and maintaining the infrastructure, which is physically protected.
Implications	Social engineering can allow people to be in places they should not be, altering the infrastructure to allow privileged remote access.
Related decisions	ADD06-01: API Gateway, ADD06-02: Load balance, ADD06-03: Third-party service & Third-party centre, ADD06-04: Unified Access Controller

5. Mapping between views

Table 5.1 mapping between view 1 and view 2

View1	View2	Correspondence
API Gateway	controller	Is equal to
Load Balance	RouteList	Is supertype of
Load Balance	Route	Is supertype of

Table 5.2 mapping between view 1 and view 3

View1	View3	Correspondence
Authentication and Identification	Client Manager	Is subtype of
Online Consulting Service	Message Processor	Is equal to
AI-Assisted Diagnosis	AI Assistant Task Processor	Is subtype of
AI-Based Search Tasks	AI Assistant Task Processor	Is subtype of
Device Management	Smart Bracelet	Is supertype of
Device Management	Medical Instrument	Is supertype of
API Gateway	Client Manager	Is subtype of
Load Balance	Load Balance Configuration	Is supertype of
Payment Service & Third-party Center & Notification & Online Diagnose Service & Appointment Management & Patient Medical Information Management	Other Function Module	Is subtype of

Table 5.3 mapping between view 1 and view 4

View1	View4	Correspondence
AI-Assisted Diagnosis	AIPredictions	Is subtype of
AI-Based Search Tasks	AIPredictions	Is subtype of

Third-Party Center	Third-Party Service	Is equal to
Patient Medical Information Management	IoT device Server	Is subtype of
Device Management	IoT device Server	Is subtype of
Payment service	Third-Party Service	Is subtype of
Integrate data from third parties	Third-Party Service	Is subtype of

Table 5.4 mapping between view 1 and view 5

View1	View5	Correspondence
API Gateway	API Gateway	Is equal to
Load Balance	Load Balancer	Is equal to
Users	User	Is equal to

Table 5.5 mapping between view 2 and view 3

View2	View3	Correspondence
UI	Client(executable)	Is component/part of
Main Controller	Client(executable)	Is component of

Table 5.6 mapping between view 2 and view 4

View2	View4	Correspondence
Click [search]	DataManagement	Is subtype of
Display route details	DataManagement	Is subtype of

Table 5.7 mapping between view 2 and view 5

View2	View5	Correspondence
Main controller	Security center	Is superclass of
Interface UI	API Gateway	Is subtype of

Table 5.8 mapping between view 3 and view 4

View3	View4	Correspondence
Smart Bracelet	IoT Devices	Is equal to
Cloud System	Cloud Server	Is equal to

Other Module	IoT + Cloud + Third-party Service	Is superclass of
Task Processor/AI Assistant	AI Prediction	Is superclass of
Load Balance Configuration.conf	Cloud Service	Is the deployment strategy of

Table 5.9 mapping between view 3 and view 5

View3	View5	Correspondence
Load Balance Configuration.conf	Load Balancer	Is component of
Smart Bracelet	Smart Monitors	Is equal to
Cloud Server	Cloud Backup + Internet	Is superclass of

Table 5.10 mapping between view 4 and view 5

View4	View5	Correspondence
Privacy/Security	IT/security center	Is subtype of
DataManagement	Main File Server & Redundant File Server	Is subtype of

6. Design decisions and Rationale

6.1 API Gateway

ADD06-01: API Gateway	
Issue	Since the system uses Micro Service Architecture, every request needs a common gateway to distribute it to the related service.
Decision	Using API gateway to receive requests and distribute to the relevant service.
Status	Decided
Group	Structural
Assumptions	All requests are followed by Restful style. And all the services can be identified by URL. For example, the request URL which contains “/appointment/” means it needs to use the appointment service.
Constraints	It needs to be scalable to deal with the increase of QPS. And after user login, it should be able to identify the request of the user so that the user does not need to log in every time he or she requests to the server.
Positions	Every request will directly request the server without using an API gateway.
Argument	API gateway provides a unified entrance, which is easy to maintain. API gateway not only can help monitor the system condition by using indexes like QPS, response time etc, but also is easy to apply service degradation when the QPS is too high and the system cannot hold all the requests. Without using it, it will be hard to maintain and implement service degradation etc.

Implications	All the requests need to be redirected to the API gateway first.
Related decisions	ADD06-02: Load balance, ADD06-04: Unified Access controller

6.2 Load balance

ADD06-02: Load Balance	
Issue	The system will face multiple requests from numerous applications. It will need a large number of load balancers to guarantee the stability and load balance a large number of concurrent requests to the server.
Decision	Load balancing is a good option when processing requests/data that needs to be spread evenly across multiple operating units for execution.
Status	Decided
Group	Structural
Assumptions	The device uses the application program through the gateway to ensure that multiple users use our server at the same time, including business service and basic service.
Constraints	Load balancers are integrated into the switching equipment and placed between the server and the Internet link. Generally speaking, hardware load balancing outperforms software methods in terms of functionality and performance, but is very expensive.
Positions	Ribbon is an open-source project published by Netflix, which also can be used to provide complex balancing algorithms and service calls on the client-side. Its client-side components provide a series of sophisticated configuration items such as timeouts, retries, etc.
Argument	Hardware load balance is better for the system. All requests from the client are passed to load balance, which then forwards the requests. It is achieved by the server-side.
Implications	It optimises link selection to improve access experience and system performance for simultaneous use by multiple devices.
Related decisions	ADD06-01: API Gateway, ADD06-03: Third-party service & Third-party centre, ADD06-04: Unified Access controller

6.3 Third parties centre

ADD06-03: Third-party service & Third-party centre	
Issue	Transactions related to payments are provided by a third-party centre so that the financial data of patients and hospitals will not be visited directly by this system and can be adequately protected. Besides, system notifications are implemented by third parties to separate different services and lift up the maintainability. Third-parties can be bank or SMS services.
Decision	This system provides a third-party centre to handle payment-related communications and notifications with the third-party system.
Status	Decided
Group	Structural

Assumptions	<ol style="list-style-type: none"> 1. All the third-party' systems provide adequate documents and API to help integrate them into the third party centre. 2. System notifications are generated automatically by the cloud server and are sent by third parties.
Constraints	Every time the third parties are integrated into the system, the developer needs to follow rules defined by both the cloud server and the third party.
Positions	Develop a whole new system to implement the functionalities.
Argument	<ol style="list-style-type: none"> 1. Insurances are the most common method in the Netherlands to pay the bills related to illness, thus it's the default choice for payers. To enable all customers to use this system, it's important to provide an alternative payment method by using third party services. 2. Adding the third party centre is a much simpler and efficient way to add new features. 3. By using the third party centre, it can protect the system in the largest way. Because if something goes wrong with third-party service, it will only affect the third party centre.
Implications	<ol style="list-style-type: none"> 1. All patients need to have their insurers or their own payment methods to use this system to receive treatments. 2. Every time the third parties are integrated into the system, the developer should add a new feature to the third party centre.
Related decisions	ADD06-04: Unified Access controller

6.4 Unified Access Controller

ADD06-04: Unified Access Controller	
Issue	Users for this system are mainly divided into two groups, patient group and professional medical group. They need to be identified as the role which matches the status. Data access authentication will be assigned to different characters according to their identity.
Decision	For patients, they are supposed to track their own health status information and the value of physical indicators. Generally, they are not allowed access to other medical data. For medical professionals (e.g. health providers/caregivers), they are not only authenticated for tracking the data of the patients they diagnosed, but also accessing the overall medical database where the processed data(by AI assistant/ big data processor, etc.) is stored.
Status	Decided
Group	Structural & Security
Assumptions	Every individual user has a unique registered account. The account can be linked with DigiD/BSN for being a senior one and assigned more permission. Moreover, caregivers are required to submit the UZI-pass which helps verify their medical identity.
Constraints	Obey the rules as a legal citizen. If there is a change of identity, it must be submitted to the system in time for identity and security verification to be completed on a regular basis.
Positions	Ask for assistance from the system administrator, after a special validation the user should be given temporary permission of accessing data that match his identity.

Argument	For health insurance at a rational and legal level, we need DigiD and BSN. UZI-pass is a strong guarantee of verification of medical professional status. The system will automatically assign different data access permission and services provided by API Gateway according to their verified identities.
Implications	All individuals need to keep DigiD and BSN, for safeguarding legal rights and interests. Health providers and caregivers should apply for a UZI-pass and other medical qualifications in different specialist areas.
Related decisions	ADD06-01: API Gateway

7. Assessment

7.1 Selected assessment scenarios

Stakeholder1 - Healthcare Providers:

S1. Performance[**ASR-01-02**]: Must store data of patients in an efficient way to respond to every query executed by caregivers or patients within 2 seconds. Requiring the speed of data access response to patients to improve users' satisfaction about this system. (H, H)

S2. Usability[**ASR-01-03**]: Must process data of patients' health conditions with statistical tools in advance to extract valuable information and display information with clear and effective graphs and tables to make sure that patients or caregivers understand within a mean of 5 seconds. (M, M)

Reasons: One business goal of Healthcare providers is to improve patients' experience and their satisfaction with the healthcare service by providing accurate diagnosis detection, thus the high performance of AI prediction is important. Besides, in order to enable patients to monitor their health conditions through records of their body indicators, it's important to display clear and effective graphs and tables to give patients a deep understanding of data about their health conditions.

Stakeholder2 - Patients:

S3. Usability[**ASR-02-01**]: User-friendly user interface provides good user interaction. It enables the user to understand the operation of the software within 10 seconds. (M, L)

S4. Security[**ASR-02-02**]: Ensure patients' data security when using the software. There should be no leakage of sensitive data with a good firewall to protect the data in the operating environment of the system. (H, H)

Reasons: The most important business goal of the medical system is to provide users with the best services, such as updating users' data in a timely manner, synchronizing them on different servers, and performing data backups to ensure the safety of users' data, without causing privacy leakage, and providing timely online Communication and consultation, etc. At the same time, it is necessary to focus on the corresponding part of the architecture. The architecture of business service and basic service should revolve around users.

Stakeholder3 - Caregivers:

S5. Security[**ASR-03-02**]: Access authority to medical device data needs to be provided to caregivers according to whether they have been validated by UZI-pass or not. (H, M)

S6. Performance[**ASR-03-01**]: Wearable smart devices and medical instruments need to be linked to cloud databases in real-time for data logging and synchronization within at most 5 seconds. (H, M)

Stakeholder4 - Payers:

S7. Security[**ASR-04-01**]: When payers pay their bills the system should secure all the connections and data transportation. (H, M)

S8. Usability[**ASR-04-02**]: The system provides enough tips and labels to help users pay the bill and users can easily learn how to pay within 30 seconds. (M, M)

Reasons: the payers want to have a good payment experience and a secured payment environment. And security definitely has the highest priority for them. And for a better experience, usability plays a crucial role in improving it.

Stakeholder5 - Security Overwathcers:

S9. Security[ASR-05-01]: For identification and data validation, authenticating **any** users and logging **all** access history with a 100% guarantee. (M, M)

S10. Security[ASR-05-03]: For data security, encrypting stored data and keeping **every** public-private key pair. (H, M)

Reasons: security is important for this system, considering that it includes sensitive data about patients' health conditions. In order to lift up the security of this system, user identification is the basis for managing patients. Besides, it's important to encrypt stored data at the level of data storage, because it will make it easier to maintain the system from the perspective of security.

7.2 Utility tree

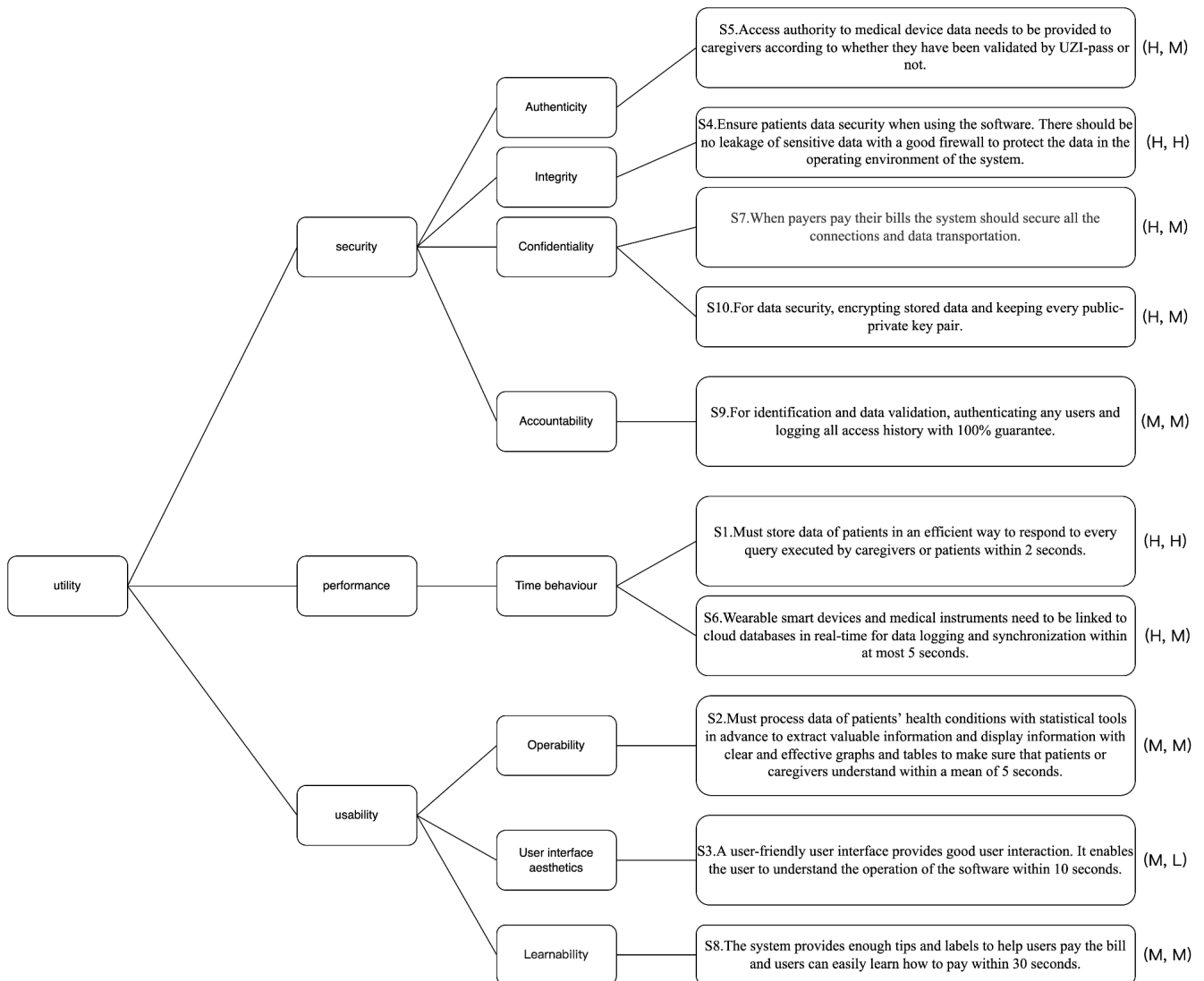


Figure 7.1 Utility Tree

The entire utility tree focuses on three quality attributes, which are Performance, Usability, and Security.

The security attributes are graded as a high priority, considering that authentication is one of the most necessary components of the entire system, through which the user can be assigned to different permissions to access data according to his or her identity. The encryption and transferring of payment data are also at the highest level of business goals and affect the architecture to some extent, thus they should be assigned to a high priority.

Accountability of data validation and user identification somewhat impacts the architecture, thus it is assigned to the medium level of priority.

The performance attribute considers more about the time of responding or processing. For data consistency of the cloud-end and all of the medical instruments like IoT equipment(smart bracelet) etc., data processing should be limited to a short time unit. Besides, data search services should provide relatively short response times. The business value and the impact on the entire system of performance attributes should be regarded at a moderate level.

The usability attribute is relatively less important for the architecture than the above two attributes. However, products that lack usability will also require more time and energy for users to learn to use. In order to improve the user-friendly features of this system, well-defined user interfaces and adequate useful hints should be provided. Improved user-friendly features will help to build a better architecture and achieve usability-related business goals.

7.3 Analysis of architectural approaches

Table 7.1 Analysis for Scenario #S4

Scenario #: S4	Scenario: The patient's health conditions should be protected to ensure that the data transferring is secure and will not have data leakage in the operating environment of the system.			
Attribute(s)	Security			
Environment	Runtime and normal operation			
Stimulus	Users want to access data via data interfaces			
Response	Data will be monitored in real-time			
Architectural decisions	Sensitivity	Trade-off	Risk	Non-risk
Data security [ADD05-04-02]	Data security is important for patient stakeholders[SH-02]. Loss of data or errors in data transmission can have a serious impact on the privacy of patients.	High level of security → more security, less maintainability	During the process of data transferring, it is easy to have data loss or data leakages.	None

Table 7.2 Analysis for Scenario #S6

Scenario #: S6	Scenario: Real-time and reliable online communication between nurses and patients			
Attribute(s)	Performance			
Environment	Runtime and normal operation			
Stimulus	Timely inquiries online			
Response	Data logging and synchronization within at most 5 seconds.			
Architectural decisions	Sensitivity	Trade-off	Risk	Non-risk

Client-side [ADD05-03-01]	The client-side is able to take into account the user's interactive experience and provides a reliable communication window, thus the performance of communication is sensitive to the client-side.	Client kindly use → more performance, less resource-efficient	None	Providing video and chat functionalities in the chat window to facilitate timely communication between nurses and patients.
Data transferring [ADD05-04-02]	Faster and more secure data transmission ensures real-time and reliable online consultations, thus the performance of communication is sensitive to the mutual transfer of information.	Transfer data faster → more performance, less resource-efficient	Slow information transmission will cause delays and problems will not be solved in time, thus the user experience will decrease dramatically.	None

Table 7.3 Analysis for Scenario #S7

Scenario #: S7	Scenario: Data in IoT devices are logged and synchronized to cloud databases in a real-time style within at most 5 seconds			
Attribute(s)	Performance			
Environment	Runtime and normal operation			
Stimulus	Automatically or manually upload IoT data			
Response	Write data and update the database			
Architectural decisions	Sensitivity	Trade-off	Risk	Non-risk
IoT device servers transfer data to the cloud through data interfaces[ADD05-04-02]	The frequency of data transferring is sensitive to the implementation of data interfaces	Uploading data more often → more performance, more complexity for processing	Data are not up-to-date: Less recent data arrive later than more recent data and replace more recent data	None
Load balance modules[ADD07-02]	A load of this system is sensitive to the frequency of data transferring	Uploading data more often → more performance, more complexity for load balancing	None	The high frequency of data synchronization increase the complexity of load balancing

Table 7.4 Analysis for Scenario #S8

Scenario #: S8	Scenario: Confidentiality of payment
-----------------------	---

Attribute(s)	Security			
Environment	Runtime and normal operation			
Stimulus	Patients trigger the pay bill function			
Response	Data about payments are transferred to the third-party with authorization and payments are handled by the third-party			
Architectural decisions	Sensitivity	Trade-off	Risk	Non-risk
Third-party center[ADD07-03]	Confidentiality of payment is sensitive to the confidentiality of the third-party and connection between payment service and the third-party	High level of security → more security, less maintainability	Successful transactions are not recorded: Connection between payers and the third-party centre crashes after payer transfers the money and the third-party does not record the successful transaction	None
Authentication and identification[ADD 05-01-01]	Confidentiality of payment is sensitive to the user authentication and identification	High level of security → more security, less maintainability	Failure in user identification and fake users being authenticated may leak sensitive information about patients' payments.	None

8. Glossary

Table 8.1 Glossary table

AI	Artificial Intelligence
BSN	Citizen Service Number
DigiD	Digital Identification
IoT	Internet of Things
QPS	Queries per second
UML	Unified Modeling Language
UZI-pass	Unique Healthcare Provider Identification