Software Architecture Design Project

Group 2 - Cluster A (on campus)

Qiuyang Fu
(2721961)
CS: SEG
q.fu@student.vu.nl

Eoan O'Dea
(2732791)
CS: SEG
e.odea@student.vu.nl

Nina van Liebergen
(2741646)
CS: SEG
n.m.van.liebergen@student.vu.nl

Abdellah Lahnaoui
(2699876)
CS: SEG
a.lahnaoui@student.vu.nl

Ruben Horn
(2702944)
CS: SEG
r.horn@student.vu.nl

August 26, 2022

Contents

1 Documentation Roadmap **6**

2 Stakeholder profiles **8**

3 Architecture overview **16**

4 Viewpoints **19**

5 Views **24**

6 Mapping between views **30**

7 Design decisions and Rationale **31**

8 Assessment **33**

9 Glossary **46**

Revision history

Date	Short description	Changes
2021-12-17	Final + week 7	Assessment & Documentation Roadmap & Final revision
2021-12-10	Week 6	Revisions (see Changelog below and blue text) & Assessment (WIP[1])
2021-12-03	Week 5	Views & mappings between views & design decisions and rationale (WIP)
2021-11-26	Assignment 1a	Viewpoints & revision of completed sections
2021-11-19	Week 2 + week 3	Updated week 2 & added utility tree & viewpoints (WIP)
2021-11-12	Week 1 + week 2	Updated week 1, business goals & ASRs[2]
2021-11-05	Initial + week 1	Set up document, added Stakeholder profiles and added Architecture sketch

Changelog

Section	Responsible	Changes
Viewpoints	Ruben	• Updated addressed concerns for each viewpoint to be less of a description of the capabilities of the view. • Did not update item "names" of the concerns in the list, since they merely refer to the requirements that the concerns are linked to. • (After review of concerns in feedback) Did not change Metamodels or notations, since the notation is adequate and the meta models are distinct from the corresponding view. • Updated rational of viewpoints to give a more thorough explanation for why the viewpoint was selected.

[1] Work in Progress
[2] Architecturally Significant Requirements

Structure, Layout & Document Quality	Eoan	• Added description of HSSE acronym within Stakeholder profiles. • Added captions to all tables (apart from Revision history and Changelog) • Addressed various spelling & grammatical issues. • Revised overall consistency of structure and layout. • Did not include sub-sections in the ToC, because some of them contain LaTeX macros which cannot be used in the ToC. The consistent hyperlink references to stakeholders, quality requirements, etc. help with navigating the document.
Stakeholder Profiles & Business Goals for ST1-ST4	Qiuyang	• ST1 Health Suite Provider: Specified demands, added more details in ASRs and BGs to reduce confusion and ambiguity. Did not replace ST1 with other stakeholder because the current one is closely relevant to this architecture • ST2 Patient: Specified demands and description, revised vocabulary and grammar. Added more details to BG5 and QAS4. • ST3 Solution Developer: Reworked "Expectations and demands", and moved some of the content to "Description". Revised BGs and QAS 6 to be more specific. • ST4 Government: Added demands. Did not change the name of this stakeholder because it does not cause negative consequences. Did not delete BG 15 because it is part of the architecture.

QA's and Business Goals & Healthcare provider	Nina	• In the Stakeholder profile, I did not distinguish between medical doctors (one that use data) and medical technicians (one that collect), because some medical doctors also collect medical data. • Refined the demand of the Healthcare provider by adding the sentence: The Healthcare provider demands to be informed about every critical change in the patients' data. • Made the measurement of more specific of BG16. Also revised BG17, BG18 and BG19. • Did not change the QAS12 and QAS13. A reviewer thought that these QAS were contradicting each other, but we do not agree. However, a explanation is added by Ruben to the QAS to make the difference between them more clear. • Did not rephrased all of QAS14. In our opinion the QAS is very important and is specifically phrased that way that it defines the quality attribute scenario. Only the response measure of this QAS is edited, because first it was phrased as a response. • Edit QAS15. Refined the different aspects of the scenario so that the stimulus, response and response measure are related to each other.
Architecture sketch	Abdellah	• Added annotations for all the relationships. • Added new items to the legend. • Provided some comparison with our decision for a "serverless" architecture. • Fixed sketch formatting in relation with the text (no longer in the middle of a paragraph). • Added explanation of public/private APIs. • Provided inspiration for considering blockchain as an alternative. • Did not change the format of the sketch as we believe that there is no one template for it.

1 Documentation Roadmap

This section contains general information about the contents of this document.

Scope and summary

The goal of the project is the development of a system that facilitates a variety of digital health services (system of engagements) and integrates existing or future digital storage of health data (systems of records). The composition of both of these parts forms the "Health Suite" which corresponds to an innovative solution by Philips of the same name.[3]

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system. The target audience for this document are the key stakeholders or their management as well as any technical leads working on implementation and operation of the system that will be developed in this project.

Organization

The first section of this documents contains the stakeholder profiles which includes a description, expectations, the goals, derived quality requirements and corresponding scenarios for each of the five most important stakeholders that are investigated.
In the subsequent section we give an overview the system's architecture and describe alternatives that we have considered.
This is followed up by two viewpoint descriptions that address important concerns of the stakeholders and the corresponding views with auxiliary explanations on how they correspond to each other.
Finally we motivate some of the most important design decisions and finally assess the scenarios relating to our architectural decisions.

View overview

In addition to an overview over the architecture (see Figure 1), the document also contains two different views. Both are described by a corresponding viewpoint including a Meta-model as a (UML)[4] class diagram.

V1 describes the interfaces that are used to interact with the system. It uses a custom notation to describe the "HTTP RESTful[5]"-like interface and adjacent resources that constitute relevant background information for the interface users.

V2 describes the authentication, authorization and logging processes within the system. This view uses a custom notation, includes the end-users and the third party applications that interact with the system.

Using the documentation

In the following paragraphs, we describe how different readers might use this document and what information they can take away from it.

Lead architects communicate key concerns of stakeholders and identify important requirements that impact the architectural design of the software. They also use this document to communicate potential impact that their decisions have on the stakeholder concerns.

[3] `https://www.philips.nl/healthcare/innovatie/healthsuite-digital-platform`
[4] Unified Modeling Language
[5] Representational state transfer

Technical service providers and third party developers Use the document to define the interfaces (and through that by extension the possible interactions) between the system and other external resources that serve an integral purpose (like user identity management) or integrate the system into other digital products.

End-users are not the primary target readers of this document, however, this document may be helpful to provide transparency about how potential end-user concerns such as privacy or availability are addressed and also which other stakeholders are involved.

2 Stakeholder profiles

When extracting quality requirements we try to adhere to the standardized quality requirements [1] as much as possible for the sake of consistency. The characteristics of ISO 25010 provide consistent terminology for specifying, measuring and evaluating software quality.
The following stakeholders have been identified:

- Health Suite Provider
- Patient
- Solution Developer
- Government
- Healthcare Provider

ST1 Health Suite Provider

Description

The Health Suite Provider is a stakeholder that gets the order of the customer for developing the Health Suite System Of Engagement (also referred to as "the system" or "the HSSE"). The Health Suite Provider develops, operates and maintains the system. The Health Suite Provider is dependent of the demands of the other stakeholders and is responsible to implement these demands in the system. Therefore, the Health Suite Provider needs to be in close contact with all the stakeholders, for delivering an appropriate end result.

Expectations and demands

The Health Suite Provider expects a well defined software architecture following standard patterns and practices. They demand the solution to be implemented correctly within the defined project time-frame, since operation and maintenance are their responsibilities as well. They also emphasise technical and business sustainability.

Business goals

BG1 For the HSSE, the Health Suite Provider desires that the HSSE can accommodate all their customers' and partners' end-users, which will be satisfied if the system's resources scale up according to demand without any bottlenecks. For example, we want to ensure that there will always be enough storage available for end-users.

BG2 For the HSSE, the Health Suite Provider desires that the HSSE meets the responsibility to all shareholders, which will be satisfied if the system has a significant, positive Return on Investment (ROI).

BG3 For the HSSE, the Health Suite Provider desires that the HSSE ensures its quality and reputation, which will be satisfied if the HSSE is rated by their users as an above average system (improvement over most and at least on a par with all established solutions).

BG4 For the HSSE, the Health Suite Provider desires that the HSSE contributes to the growth and continuity of the organization, which will be satisfied if the number of users of the HSSE never decreases.

Architecturally Significant Requirements

QA1 Scalability The system should meet the demanded load at any time in order to avoid a service outage and serve all potential users and customers. This relates to BG1.

QAS1 (Horizontal scaling on demand)	
Stimulus	Resource capacity nearly exceeded due to rapid user increase
Source of stimulus	System resource
Response	Provision additional resources and balance load (horizontal scaling)
Response measure	Available capacity of all resources is at least 10% higher than the provisioned capacity two minutes since the last increase in load
Environment	Operation
Artifact	Compute/Network/Storage

Table 1: Horizontal scaling on demand

QA2 Adaptability If the Health Suite Provider wants to offer on-premise solutions, the sensor data processing pipeline should work independent to the rest of the system without additional development effort. This relates to BG2.

QAS2 (On-premise solutions on demand)	
Stimulus	New on-premise solutions for receiving and processing Patient data streams has been ordered by a Healthcare Provider
Source of stimulus	Healthcare Provider
Response	The corresponding components of the HSSE is fully operational in the new deployment
Response measure	No stubs need to be implemented and an "off the shelf" deployment solution can be utilized (like container orchestration using Docker Compose) NEW: For faster deployment, existing solutions will be utilized (e.g. container orchestration using Docker Compose) within a reasonable time frame to avoid spending too much time on implementing stubs
Environment	Installation
Artifact	System components subset

Table 2: On-premise solutions on demand

QA3 Modifiability Adding an external Electronic Patient Record (EPR) should be possible without "re-developing" other components of the system of engagement. This relates to BG2 and BG4.

QAS3 (Integrating existing EPR)	
Stimulus	Request to integrate existing EPR
Source of stimulus	Healthcare Provider
Response	The "new" EPR as well as all existing EPRs can be accessed within the HSSE
Response measure	Only the corresponding system of records (and not any components of the system of engagements) needs to be extended with the connection logic.
Environment	Integration
Artifact	The system

Table 3: Integrating existing EPR

ST2 Patient

Description

The Patients are an end-user stakeholder in our scenario. They use smart wearable devices and other applications that are developed and integrated by Solution Developer and Health Suite Provider. The

system can then be used to monitor and track their vitals/health indicators in order to maintain or strive for a healthy lifestyle. Patient's information will be protected by the regulations that are established by the Government, and treatment will be given to Patient by the Healthcare Provider.

Expectations and demands

The Patientexpects an accessible and easy to use system regardless of their technological knowledge. Since there is a lot of medical data tracked and transmitted, privacy is a big concern for them. They also expect all recorded data to be processed in the correct order to derive correct insights over time to prevent altered information after a chain of data processing. The Patient demands their medical data to be confidential and effective treatment or diagnosis will be given by the Healthcare Provider

Business goals

BG5 For the HSSE, the Patient desires that their personal data remains confidential, which will be satisfied if nobody can access their data without being authorized by the Patient.

BG6 For the HSSE, the Patient desires that in an emergency they receive optimal treatment, which will be satisfied if the hospital is ready to treat them as soon as they arrive.

Architecturally Significant Requirements

QA4 Confidentiality Patient data should be confidential and only accessible to authorized users to ensure Patient privacy. This relates to BG5.

QAS4 (Confidentiality of Patient data)	
Stimulus	Patient data is requested by sending Patient a document that requires a signature
Source of stimulus	Healthcare Provider
Response	Gain access to the Patient data by signing the document
Response measure	Only access data which the Patient explicitly gave them permission to access
Environment	Operation
Artifact	Application Programming Interface (API)

Table 4: Confidentiality of Patient data

QA5 Transaction processing time Notifications about Patient sensor data events should be received timely (within a certain time that is dependent on the distance to the Healthcare Provider) so that the Patient can be treated immediately. This relates to BG6.

QAS5 (Transaction processing time)	
Stimulus	Critical health measurement is detected
Source of stimulus	Patient's device
Response	The notification is received by the corresponding healthcare provider(s)
Response measure	The maximum acceptable time, for the notification to arrive, decreases by 5min for every 10km closer to a minimum of 5min at $\leq$ 10km.
Environment	Operation
Artifact	API

Table 5: Transaction processing time

ST3 Solution Developer

Description

The Solution Developer is a manufacturer of wearable consumer devices that is also responsible for developing consumer medical applications. They integrate their product with the described system to provide additional value to the end-user through their product. To achieve this, a defined API is used to communicate with the Health Suite Provider.

Expectations and demands

The Solution Developer expects that the hardware and software utilize the APIs, because it is the essential component which is used to provide the ability for the device to submit health data. They also expect to integrate new functionalities of the APIs as per the end-users requests to support the usefulness of their product. The Solution Developerdemands that the quality of the hardware and software are managed to an excellent standard so other stakeholders will be satisfied.

Business goals

BG7 For the HSSE, the Solution Developer desires that the APIs facilitate their solution, which will be satisfied if the applications and the APIs are integrated with devices so the complete products can be received by the HSSE, and a reasonable percentage of data can be correctly exchanged from any internet-connected device/platform that supports TLS[6] 1.2.

BG8 For HSSE, the Solution Developer desires that financial objectives are met, which will be satisfied if the integration of their products with the system increases the ROI.

BG9 For HSSE, the Solution Developer desires that the integrity of users' data remain protected all the time, which will be satisfied if backup solutions are implemented with the system to prevent any sort of data loss in malfunctions of components.

Architecturally Significant Requirements

QA6 Interoperability The system should support data exchange using standard, public technologies. This relates to BG7.

QAS6 (Interoperability with consumer products)	
Stimulus	Interaction of third-party solution within the system developed
Source of stimulus	Solution Developer
Response	Successful integration and data exchanges between the device and the system
Response measure	Data can be submitted from any internet-connected device/platform that supports asymmetric encryption in transit (like TLS 1.2) NEW: 99.99% of the data exchanges are correctly processed from any internet-connected device/platform that supports asymmetric encryption in transit (like TLS 1.2)
Environment	Development (of third-party solution)
Artifact	API

Table 6: Interoperability with consumer products

QA7 Availability The system has received sensor data within one hour of the data being recorded. This relates to BG9.

[6]Transport Layer Security

QAS7 (Availability to receive measurements)	
Stimulus	Device cannot upload sensor data due to a system outage
Source of stimulus	Solution Developer
Response	Retry uploading the data
Response measure	Within one hour of when the data has been recorded it has been successfully received by the system for processing
Environment	Interrupted operation (outage/peak load)
Artifact	APIs

Table 7: Availability to receive measurements

ST4 Government

Description

The Government lays out the rules and regulations regarding medical information services to ensure a high quality of treatment, protect the privacy of Patients and maintain trust in the national health system as well as ensuring the overall health of all citizens. It provides electronic identification services for its citizens (DigiD) and also for healthcare providers (UZI)[7].

Expectations and demands

The Government expects all medical data to be treated confidentially and all access to such data to have a valid reason, be constraint to the required scope and be traceable to the user of such systems. The Government also demands the Health Suite Provider to establish security protocols to protect Patients' data, it should also be possible to monitor Health Suite Provider 's data usage.

Business goals

BG10 For systems processing personal data, the Government desires that personal data remains confidential, which will be satisfied if only entities that have been authorized by the corresponding person can access that data.

BG11 For systems processing personal data, the Government desires that access to personal data is transparent, which will be satisfied if the purpose and exact time of accessing the data is known.

BG12 For health data processing systems, the Government desires that health data can unambiguously be attributed to a person as their personal data, which will be satisfied if this person can be indisputably identified (for example through DigiD).

BG13 For healthcare provider applications, the Government desires that healthcare providers make use of existing digital public infrastructure for the sake of authenticity of Healthcare Providers, which will be satisfied if a centralized repository of Healthcare Providers (like UZI) is used to authenticate them.

BG14 For the healthcare system, the Government desires that responsibilities to society (lay out regulations to help improve treatment quality) are met, which will be satisfied if the health of every citizen can be ensured.

BG15 For the healthcare system, the Government desires that responsibilities to the state are met, which will be satisfied if the cost of healthcare and the general health of the population will not compromise the ability of the Government to enact its policies.

[7] Unieke Zorgverlener Identificatie (Unique Healthcare Provider Identification)

Architecturally Significant Requirements

QA8 Accountability It must be possible to trace who had access to Patient data precisely at which minute in time and to which document specifically. This relates to to BG11.

QAS8 (Accountability time precision)	
Stimulus	Patient data is requested
Source of stimulus	Healthcare Provider
Response	Traceable provide data
Response measure	Whenever this request has been received by the HSSE, it is knowable that this specific Healthcare Provider did access the data, accurate to the minute.
Environment	Operation
Artifact	API

Table 8: Accountability time precision

QAS9 (Accountability scope precision)	
Stimulus	Patient data is requested
Source of stimulus	Healthcare Provider
Response	Traceable provide data
Response measure	Whenever a request has been received by the HSSE, it is knowable exactly what data (which Patient and document) specifically the Healthcare Provider had accessed
Environment	Operation
Artifact	API

Table 9: Accountability scope precision

QA9 Authenticity All users of the system should be identifiable correctly and unambiguously. This relates to BG12 and BG13.

QAS10 (Authenticity of Patients)	
Stimulus	Patient interacts with a consumer application which submits to or requests data from the system
Source of stimulus	Consumer health application (or device)
Response	Allow authenticated Patients to interact with the system
Response measure	The Patient can only use the consumer application that interacts with the system, if they can prove their identity using the trusted identity provider integrated with the system (for example DigiD)
Environment	Operation
Artifact	API

Table 10: Authenticity of Patients

QAS11 (Authenticity of Healthcare Providers)	
Stimulus	Healthcare Provider interacts with a professional application integrates with the system
Source of stimulus	Consumer health application (or Internet of Things(IoT) device)
Response	Allow authenticated Healthcare Providers to interact with the system
Response measure	The Healthcare Provider can only interact with the system, if they can prove their identity in a central healthcare provider index (for example UZI)
Environment	Operation
Artifact	API

Table 11: Authenticity of Healthcare Providers

ST5 Healthcare Provider

Description

The Healthcare Provider treats Patients based on their condition. In order to provide optimal care, accurate, complete and recent data is required. Based on their treatment or diagnosis, the Healthcare Provider generates new Patient data that may be used by other Healthcare Providers. This data is either provided manually or by specialised equipment. They are another end-user stakeholder in this scenario.

Expectations and demands

The Healthcare Provider expects a smooth information-flow of the Patients data. The Healthcare Provider needs to have an accurate overview of the medical data and needs to be able to easily look up this information. When there is a critical development with a Patient captured by the system, Healthcare Provider expects to be informed in order to be prepared to provide the appropriate treatment or diagnosis. Thus, the Healthcare provider demands to be informed about every critical change in the patients' data.

Business goals

BG16 For HSSE, the Healthcare Provider desires that they are quickly informed and prepared for Patient emergencies, which will be satisfied if they are made aware of the patient's emergency within 15 minutes after the emergency is signaled.

BG17 For HSSE, the Healthcare Provider desires that the system achieves high availability, which will be satisfied if the uptime is 99.999% .

BG18 For HSSE, the Healthcare Provider desires that the Patient data is reliable, which will be satisfied if no measurements received from the HSSE are corrupted.

BG19 For HSSE, the Healthcare Provider desires that they can easily adopt this system, which will be satisfied if there are no more than 10 questions each year at the help-desk about adopting the system.

Architecturally Significant Requirements

Transaction processing time (QA5) This quality requirement is shared with the stakeholder Patient.

QA10 Availability The services offered to the Healthcare Provider must have adequate availability to be relied upon in all supported healthcare processes. This relates to BG17. [8]

[8] This requires two measurements as will become obvious from the following example: Consider a system is unavailable for 30 minutes every day. While it can at most be unavailable for a short 30 minutes at a time, it should obviously not be considered highly available. Also, consider a system that is only unavailable for 5 hours on one day per year. Even though it has a lower downtime than the other system it can also not be considered highly available, since the outage is very long.

QAS12 (Minutes of unavailability per month)	
Stimulus	Crash
Source of stimulus	Infrastructure
Response	Recover from crash
Response measure	The system may only be unavailable for Healthcare Providers for 15min per month on average
Environment	Operation
Artifact	API

Table 12: Minutes of unavailability per month

QAS13 (Maximum unavailability at a time)	
Stimulus	Crash
Source of stimulus	Infrastructure
Response	Recover from crash
Response measure	The system may only be unavailable for Healthcare Providers for a maximum of one hour at a time
Environment	Operation
Artifact	API

Table 13: Maximum unavailability at a time

QA11 Fault Tolerance The services offered to the Healthcare Provider must not be compromised by any issues or outages related to the consumer facing aspects of the system. This relates to BG18.

QAS14 (Fault tolerance across sub-systems for different users)	
Stimulus	Device sensor calibration resets and submits wrong measurements
Source of stimulus	IoT Device
Response	"Faulty" data is detected and not stored
Response measure	No data points outside of the possible value range have been stored. (For example: negative or kilohertz heart rate measurements.
Environment	Communication
Artifact	API

Table 14: Fault tolerance across sub-systems for different users

QA12 Replaceability The HSSE must be a full replacement of interactions with any previously used EPR system. This relates to BG19.

QAS15 (Replaceability of legacy systems)	
Stimulus	Replacement of the interface interacting with the EPR system
Source of stimulus	Implementing the system
Response	Healthcare Provider interacts with a single system
Response measure	All data from existing EPRs can be accessed through the single system.
Environment	Operation
Artifact	Process

Table 15: Replaceability of legacy systems

3 Architecture overview

In this section we give an overview of the functionality that is provided by the system and describe the corresponding architecture, as well as highlight a few alternatives that were considered and why they were disqualified.

Description of the functionality

- (Latent) Patients use wearable devices and applications which measure and report health indicators. The latter may also allow the Patient to view or generate an analysis from this data.
- Data is aggregated centrally and provided to healthcare providers. Their applications and IoT devices may download Patient aggregated data. Upon significant developments regarding the data of their Patients, they are notified about this event.
- Additionally, healthcare providers may access the EPR. This combines the centralized record with other non-centralized records by other healthcare providers.
- The access of consumer solutions to the system is strictly separated from the access of healthcare provider solutions. These systems are not part of the Health Suite and may be managed by the other organizations.
- External "Know Your Customer"(KYC)/identity providers are integrated to authenticate users (DigiD, UZI).
- Applications for consumer or professional connected devices or medical applications are provided by a third party (Solution Developer) and integrate with the HSSE.

Selected architecture

The HSSE uses a "serverless" architecture using separate resources that are connected through the network. This allows for very high flexibility when deploying the system, because it is hardware agnostic. These resources include dedicated storage nodes, data streams, compute units (may be persistent or functions that are executed on demand) and API gateways. Their composition within the system is depicted in Figure 1. Access from outside the system is only possible through the APIs which requires a token, so resources inside the system that require authentication do not need to be coupled to any identity providers.

Because there is much more data being submitted by consumer devices and applications at the same time than by devices and applications of Healthcare Providers, streams and aggregations are used before persistently storing the data in order to avoid overloading the corresponding resource.

Access to existing EPRs is facilitated through a cache that forwards requests, replicates frequently accessed documents and stores new documents. This decouples the API from the individual EPRs.

The systems resources are further split by the user. Data that is produced and processed by consumer applications is processed and stored separately from data that is produced and processed by professional applications by the Healthcare Providers. Only access to the storage of consumer application generated data is shared.

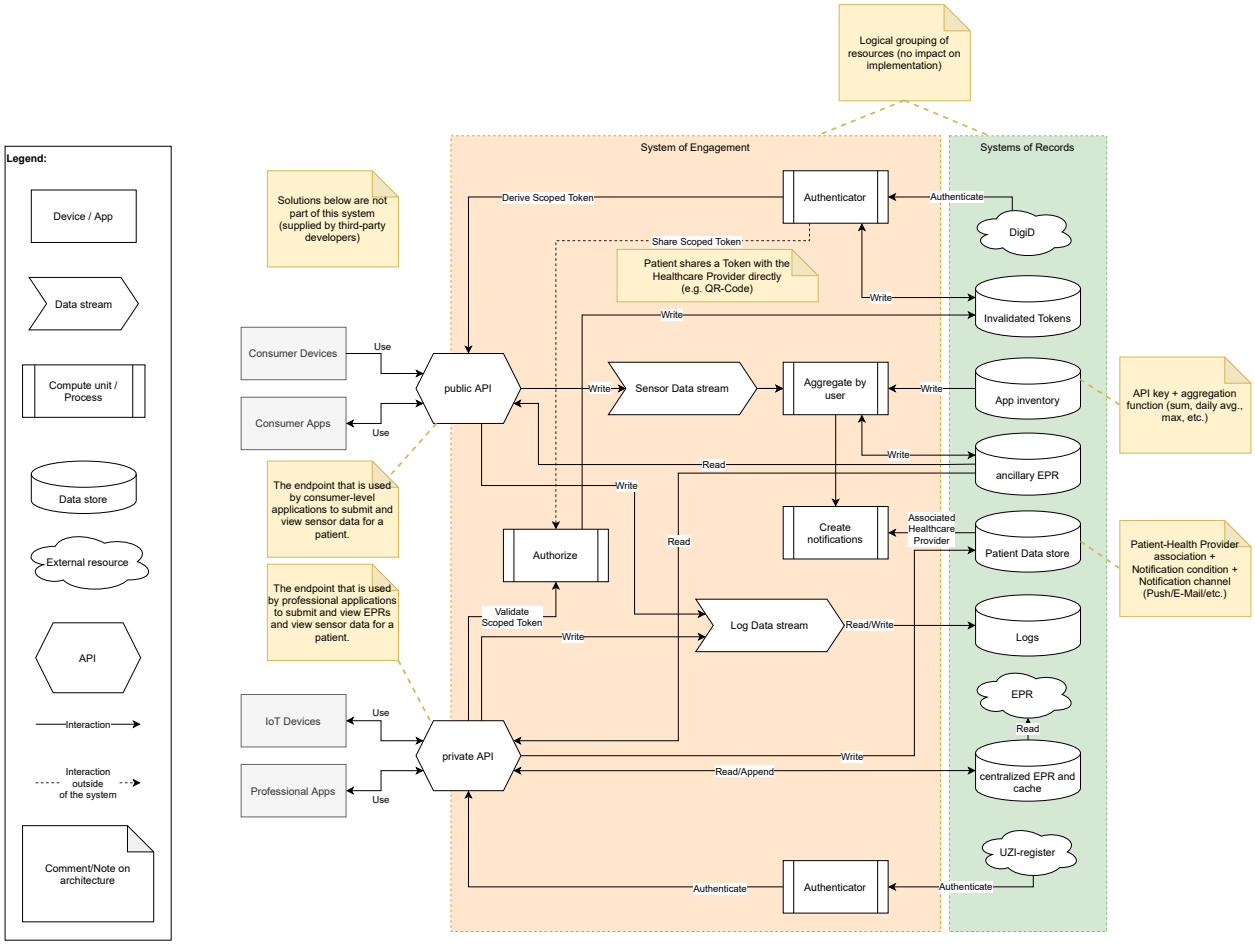


Figure 1: Architecture sketch

Alternatives considered

(Multi-instance) monolith Developing the system as a "traditional" monolithic client-server back-end would have resulted in a much simpler layered architecture (API/business logic/database access). Such a system could then be deployed at a limited scale. For scaling to arbitrary workloads however, a more complex deployment using a shared database and load balancing between multiple instances would be required, negating the advantage of simplicity of this approach. This is also the case for the selected architecture, however, through the use of hardware-agnostic technologies like containers, the management of physical and virtual resources is largely separated, simplifying the administrative overhead. Additionally, by using managed, specialized resources like event queues (data stream) instead of big single monolith applications, specific parts of the system can be scaled without also scaling everything else to solve a single bottleneck. Sharing data between the instances would not be possible "out of the box" with this approach.

Blockchain This idea has been inspired by articles on a potential future of blockchain in healthcare.[9] Using their private key, users submit encrypted health data to the blockchain. Validating nodes are operated by the Health Suite Provider which identify themselves using private keys. Unlike for Bitcoin, a complex consensus mechanism like PoW (Proof of Work)[10] is not required. Because nodes do not compete, the order in which they may append to the chain is agreed upon between all peers. Data on the blockchain can be accessed either by using a non-validating node or through an API offered by another node. The Health Suite Provider adds new nodes to the network depending on the number of requests to the data. Data is guaranteed to be correct by validating nodes and the replication across the network protects against loss of data. Using smart contracts, insights can be derived from the data on the chain. This approach is not suitable however, because public access even to encrypted health data does not comply with most regulation and data is replicated unnecessarily on every node. There is no advantage to using a private blockchain aside from the guarantee of immutability. In this scenario, an (HTTP-)API and the required infrastructure would still need to be provided so that end-users could interact with the system.

[9] IBM offers healthcare solutions that utilize the blockchain (`https://www.ibm.com/blockchain/industries/healthcare`) and the IOTA foundation outlines potential use-cases in healthcare for their lightweight distributed ledger (`https://www.iota.org/solutions/ehealth`) that cover scenarios that are relevant to the HSSE

[10] Proof of Work is a form of cryptographic proof in which one party proves to others that a certain amount of a specific computational effort has been expended. Verifiers can subsequently confirm this expenditure with minimal effort on their part.

4 Viewpoints

We define the following viewpoints according to *Systems and software engineering — Architecture description* [2], which outlines the following components as part of each viewpoint: Governing viewpoint (**description**), Framed **concerns**, Concern **stakeholders**, and Model kind (**Modeling techniques** and **Rationale**).

VP1 Integration Viewpoint

Description

This viewpoint describes the system with a focus on interface to consumer or professional solutions that interact with the system in detail and how it addresses technical concerns. The main goals of this viewpoint are to design and document the system at the point of interaction between the Health Suite Provider and Solution Developers providing additional products that integrate with the HSSE. This extends beyond the interfaces themselves to explain how the requirements that the Solution Developer has towards the behaviour of the interface by also showing how adjacent parts of the system support it.

Addressed concerns

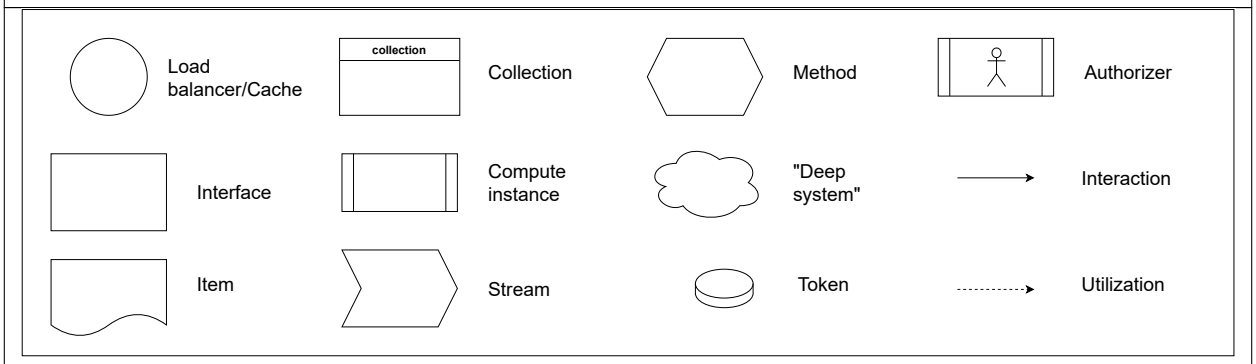
- **Interoperability (QA6)** The product of the Solution Developer must be able to interact with the HSSE.
- **Availability (QA7)** Interface adjacent components must ensure that data is received and processed by the system, even if the "deep system" (which consists of the systems of record and adjacent resources) is under high load.
- **Adaptability (QA2)** Separation between interfaces must be in place to ensure that distinct parts of the system can be operated independently.
- **Scalability (QA1)** The system must be able to scale if the load increases (higher number of incoming requests) by provisioning additional resources.

Stakeholders

- Health Suite Provider
- Solution Developer

Modeling techniques

Notation (View)



- **Load balancer/Cache** Distribute incoming requests between different resources/Memorize frequent requests
- **Interface** Access point to the system for 3rd party applications/devices
- **Item** A document that is submitted to or returned from the interface
- **Collection** Logical set of items
- **Compute instance** Persistent or on-demand logical resource for handling performing computation
- **Stream** Buffer for events/data
- **Method** Create/Read/Update/Delete
- **Deep System** Abstract representation of parts of the system or external resources that are non-adjacent to the interface
- **Token** Cryptographically signed proof of identity or authorization issued by a trusted party
- **Authorizer** Compute instance that is responsible for validating the credentials contained in requests to the interface
- **Interaction** Invocation/Flow of information
- **Utilization** Used/Processed by

Notation (Meta-model)

The meta-model uses standard UML class diagram notation [3].

Meta-model

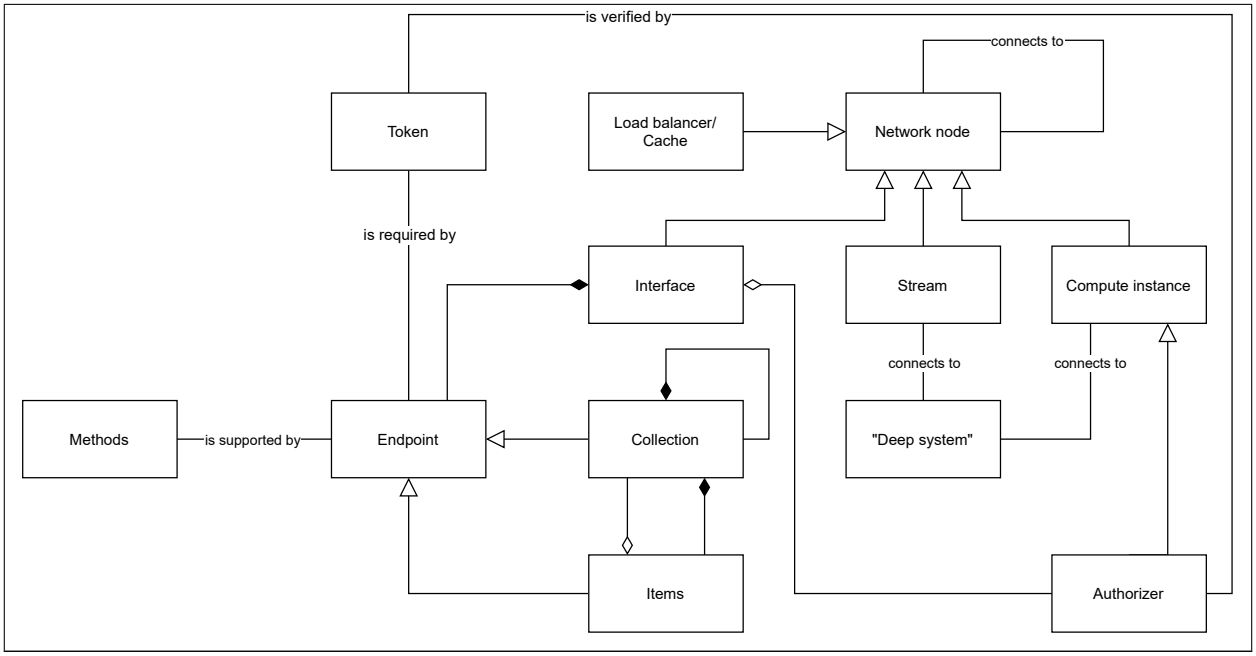


Table 16- The meta-model uses standard UML class diagram notation [3].

Rationale (for viewpoint selection)

This viewpoint covers the concerns of the "technical stakeholders". It is essential that both the Solution Developer and the Health Suite Provider can have a good understanding of the interface connecting both of their products. The internal structure of their systems however is not relevant to the shared concerns, so this viewpoint focuses on the interface adjacent parts of the system. As such it will produce a "macro" view of these parts. The selected viewpoint shows (using the parts from the notation that compose the interfaces) what data the Solution Developer can submit or access. The inclusion of the interaction and load balancers will show how the system can remain available and scalable by distributing the load across multiple resources that are not within the scope of this viewpoint.

VP2 User Identity Viewpoint

Description

This viewpoint describes the internal processes surrounding the identities of the users and their access to the data and is primarily intended for the "non-technical stakeholders" who are concerned with the data (access) separation and the required authentication/authorization mechanisms within the system. This includes both access control during the the request as well as access tracing at a later point.

Addressed concerns

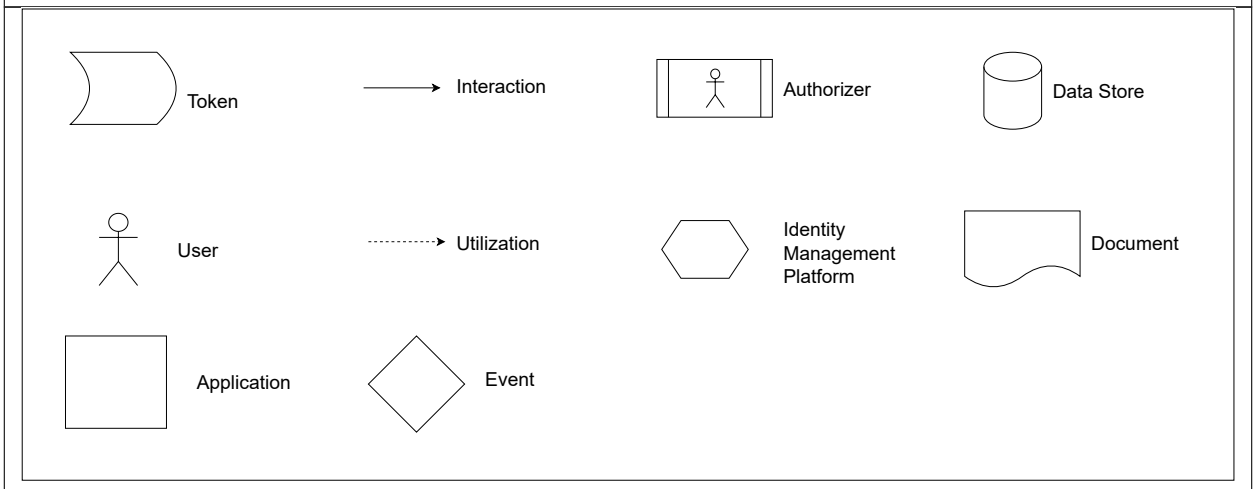
- **Confidentiality (QA4)** Patient data must only be accessible to authorized users to protect the privacy of the Patient and comply with regulations by the Government.
- **Authenticity (QA9)** All the users of the system must be identifiable when interacting with the system.
- **Accountability (QA8)** Information about the time and action that users perform need to be recorded.

Stakeholders

- Government
- Patient
- Healthcare Provider

Modeling techniques

Notation (View)



- **Token** The access token received from the identity management platform for giving the actor access. It includes information about the subject (which user does it authenticate) and scope (which privileges does it grant this user). A potential implementation would be using JSON Web Token (JWT).
- **User** Person that wants to access data from the system of records using an application that integrates with the HSSE. This can be either a Patient or a Healthcare Provider.
- **Interaction** Invocation/Flow of information.
- **Utilization** Used/Processed by
- **Data Store** Database system for storing various types of data. (Systems of records)
- **Authorizer** Compute instance that is responsible for validating the credentials contained in requests resulting from the user interacting with the application.
- **Identity Management Platform** Institution/Service that verifies the identity of the actor and returns a matching token (credentials for the authorizer).
- **Document** Data managed by the system of records.
- **Application** Software solution (client software and API of the HSSE) for the user.
- **Event** Passed between entities to exchange information and trigger processes.

Notation (Meta-model)
The meta-model uses standard UML class diagram notation [3].

Meta-model

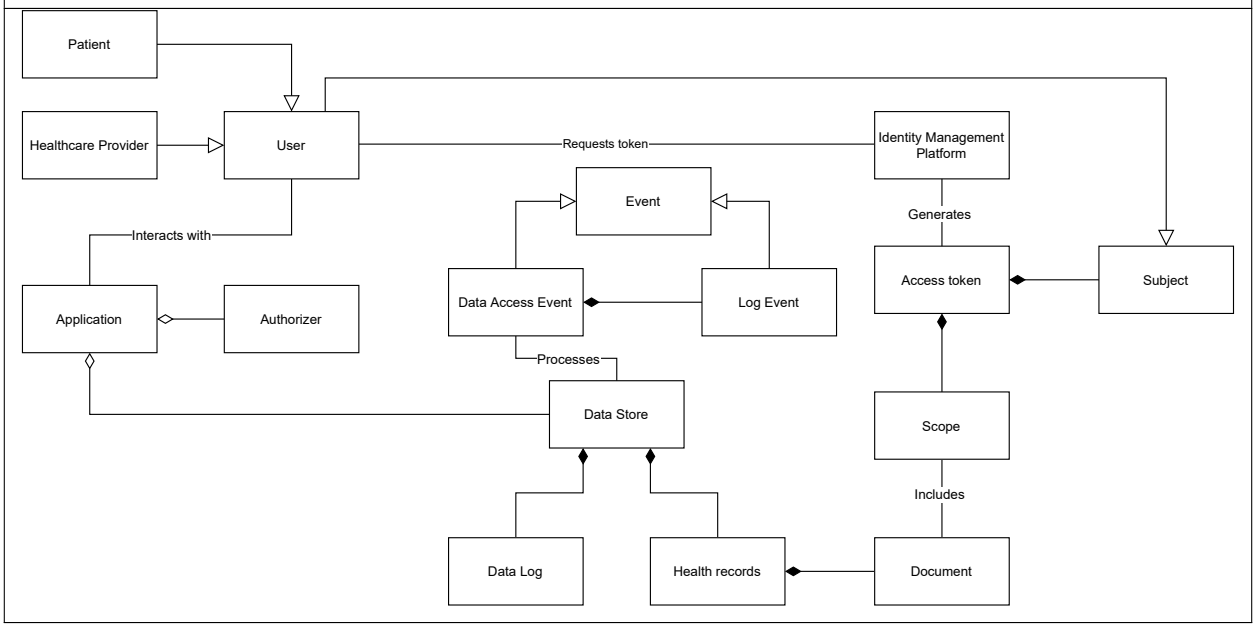


Table 17- The meta-model uses standard UML class diagram notation [3].

Rationale (for viewpoint selection)

This user identity viewpoint works well for the concerns of the "end-users" (Patient and Healthcare Provider) as well as the Government as all of these stakeholders are concerned about or affected by the handling of data and access. It is essential to the relevant stakeholders that they have a good understanding of how information is accessed and distributed which can be easily interpreted from the accompanying meta-model and further explained and simplified by utilizing the aforementioned notation. For these stakeholders, it is important to ensure that the correct identities of the users are captured and their behaviors are monitored. For them, it is less important how the information retrieval is realized or how their requests are fulfilled under different loads. The corresponding view to this viewpoint will show tracability of user interactions actions within the system, due to the relation between "Data Access Events", "Logging Events" and also how the external providers of user identities are included into the applications that access Patient data through the system.

5 Views

This section contains the views that correspond to the viewpoints that were described in the previous section.

V1 Integration View

The integration view in Figure 2 shows the two APIs of the system and how requests to these interfaces are processed. It provides information for the Solution Developer about which operations are supported to integrate their product with the system and how to authenticate when using the interface. At the same time the resources processing the requests indicate how the Health Suite Provider can ensure that the requests to the APIs are processed and that the resources can scale to meet increased demand. The left-hand side of the visualization can directly be used to design the software that integrates with the HSSE (and also documents the expectation towards the Health Suite Provider), while the rest of the view describes how the system may respond to requests, such as the fact, that reports can only have eventual consistency due to the non-immediate write operations to the system of records in the "Deeps system", due to the preceding queue in the pipeline.

Element catalogue

The following table contains detailed explanations of the elements found in Figure 2.

Element	**Description**
"Deep system"	Abstraction of systems of record which are not relevant within the scope of this view.
Patient (Token)	Access token for a Patient. (Most likely implemented as a JWT) This token may identify a user, in which case it can be used to interact with the system as that user. It may also be limited in the scope of which data of that user can be accessed. In this case the token may be transferred by any method such as barcode to a Healthcare Provider to access the data on behalf of that user. This token is limited to one authorized user only who is identified by their token so it cannot be shared further.
Healthcare Provider (Token)	The access token for Healthcare Providers to interact with the system. This type of token may be the subject of a token that authorizes access to the data of a Patient that is shared with the Healthcare Provider.
Public API	The endpoint that is used by consumer-level applications to submit and view sensor data for a Patient.
Measurements	Collection of measurements (document) in the public API. A corresponding method exists to create a new measurement and another to read the collection of measurements (set/sub-set). These correspond to the HTTP-REST methods POST and GET on resources.
Reports	Collection of reports (aggregated measurements) in the public API. A corresponding method exists to read the collection of reports (list) and to read a specific document inside that collection. These correspond to the HTTP-REST method GET on resources.
Private API	The endpoint that is used by professional applications to submit and view EPRs and view sensor data for a Patient.

Element	Description
reports	Collection of reports (aggregated measurements) in the private API. A corresponding method exists to read the collection of reports (list) and to read a specific document inside that collection. These correspond to the HTTP-REST method GET on resources.
Medical categories	Collection of medical categories/domains in the private API. A corresponding method exists to read the collection (list categories) and to create a new one inside that collection. The latter is used when, for example, a Patient visits a specialist in a particular field for the first time. These correspond to the HTTP-REST methods GET and POST on resources.
Medical documents	Inside each category there is another collection which contains the EPRs. This collection can be read (list documents) and individual documents can be read or modified, but not deleted. These correspond to the HTTP-REST methods GET, POST and PUT on resources.
Patient authentication	Verification of a Patient token.
Healthcare Provider authentication	Verification of a Health Suite Provider token.
Load balancer/Cache L1	Handles requests coming from the public API and authenticates them using the Patient authentication. Incoming sensor data is forwarded to one of the pipeline instances using a shard id that is derived from the hash of the Patient that the data is associated with. This ensures even distribution of the workload. This resource also forwards the reports requested through the API.
Shard $\#n$	Message broker instance $\#n$. This resource decouples producers from consumers and ensures First in First out (FIFO) queuing of data being sent to it. Additionally, if the delivery of an event in the queue to a consumer was not successful, a retry strategy is implemented.
Aggregator $\#n$	Consumer of measurements that computes and stores the aggregate of sensor measurements in a data store and triggers the Healthcare Provider notification (both part of the "Deep system").
Load balancer/Cache L2	Since reports for the same Patientmight be viewed multiple times before they change, this resource is used to access them efficiently from the "Deep system" using the APIs.
Load balancer/Cache L3	This resource authenticates the Healthcare Provider and verifies that they may access reports of a Patient via the Load balancer (Cache) through the private API.
Load balancer/Cache L4	This resource provides access the EPRs for the Healthcare Provider using the private API given the authorization through the corresponding Patient. This resource combines the different systems of records in the "Deep system" into a single access point. Changes in the systems of records invalidate the cache.

Table 18: Element Catalogue for VP1

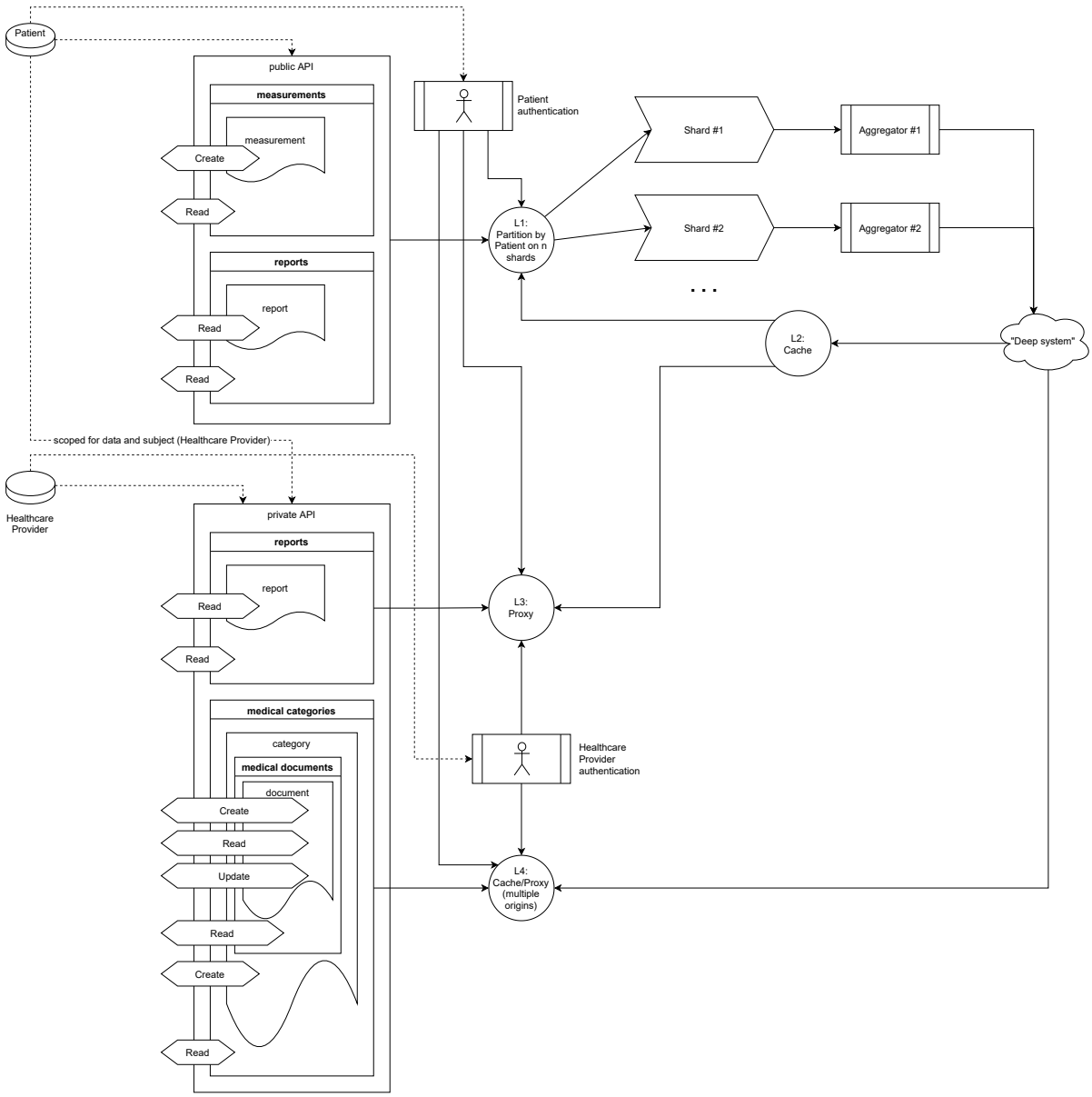


Figure 2: View corresponding to VP1

V2 User Identity View

The user identity view in Figure 3 shows how users accessing the HSSE of record are authenticated using identity providers, how a Healthcare Provider has to obtain authorization outside of the system to access Patient data and that all interactions with the system are logged.

The lower part of this view describes the flow that end-users go through in order to comply with the authentication and authorization scheme and how the external identity provider is utilized to verify their identity in the HSSE. Additionally, the view shows that every data access or modification event in the system always causes a corresponding logging event, so full traceability of interactions is ensured for the Government to investigate potential privacy violations.

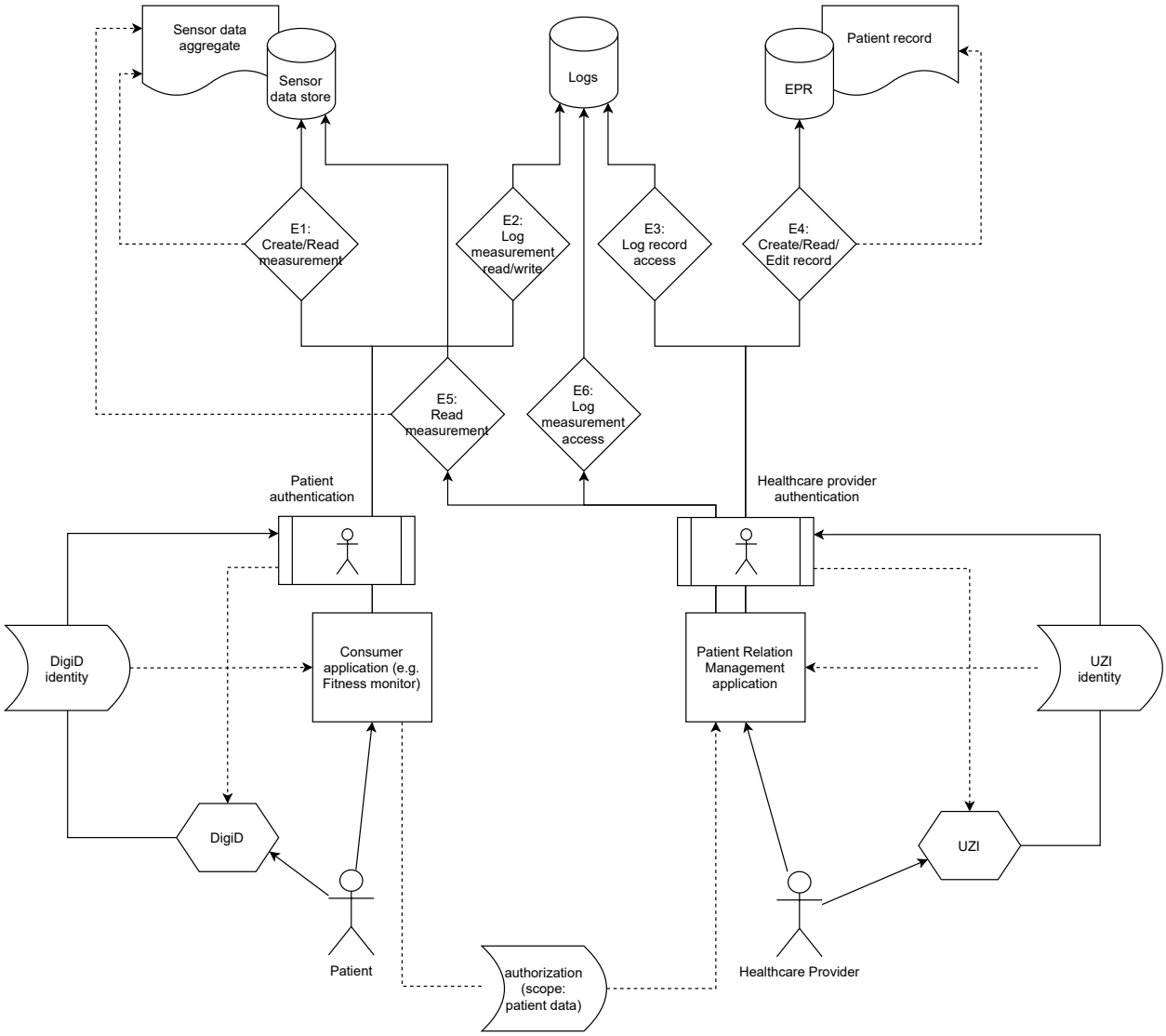


Figure 3: View corresponding to VP2

Element catalogue

The following table contains detailed explanations of the elements found in Figure 3.

Element	**Description**
Patient	(Latent) Patient and user of consumer applications that integrate into the HSSE.
Healthcare Provider	Healthcare Provider and user of professional applications that integrate with the HSSE.
Patient authentication	Verification of a Patient token.
Healthcare Provider authentication	Verification of a Healthcare Provider token.
Consumer application	Health/wellness application or smart wearable device for recording health indicators or reviewing accomplishments/trends.
Patient relation management application	Professional healthcare application for interacting with EPRs. This may be anything from a diagnosis platform to an IoT hospital bed.
DigiD	An identity management platform which can be used to verify the identity of Dutch residents on the Internet.
DigiD identity	Proof of identity issued by the official DigiD infrastructure to an app integrating it as an identity platform, most likely in the form of a JWT.
UZI	A Dutch identity register which which can be used to authenticate registered Healthcare Providers.
UZI identity	Proof of identity issued by the official UZI infrastructure to an app integrating it as an identity platform, most likely in the form of a JWT.
Authorization (scope: patient data)	Access token that is transferred by the Patient to the Healthcare Provider and authorizes them to access the Patients data though the HSSE.
Sensor data store	Data store containing Patient-submitted health indicator values originating from smart wearable devices or non-professional medical software. This data store does <u>not</u> contain a full EPR.
Sensor data aggregate	Document entity within the Sensor data store that contains the processed sensor data, such as, for example, the maximum heart-rate within the last week or a trend in body weight.
EPR	Any one of the systems of records which store digital Patient data and are accessed through the HSSE.
Patient record	Document, such as an X-ray scan for example, that is created, updated or viewed by the Healthcare Provider through the HSSE and is stored in the EPR.
Logs	Data store for log messages that are generated by user interaction with the system. This central repository serves the purpose of providing traceability.
E1: Create/Read measurement	Event corresponding to a Patient accessing or creating sensor data stored in the system of records.
E2: Log measurement read/write	Log event that is emitted when “E1” is emitted.
E5: Read measurement	Event corresponding to a Healthcare Provider accessing sensor data stored in the system of records.

Element	**Description**
E6: Log measurement access	Log event that is emitted when "E5" is emitted.
E4: Create/Read/Edit record	Event corresponding to a Healthcare Provider accessing or modifying Patient health data in the EPR.
E3: Log record access	Log event that is emitted when "E4" is emitted.

Table 19: Element Catalogue for VP2

6 Mapping between views

Below we describe how the views we created for the stakeholders of the system relate to each other. We focus on which parts are described by two views and the different aspects that the views focus on.

Mapping between V1 and V2

V1	**V2**	**Explanation**
"Physical backend"	"Logical backend"	Both views include accessing data in the "backend" of the system, the systems of record. V1 shows how the access to the systems of record is realized using different system resources in the deployment, but not the systems of record themselves, while V2 shows in more detail how data is separated by type over different systems of record.
Authenticators and tokens	Authenticators and tokens	Both views include the authenticators for Patients and Healthcare Providers and the corresponding tokens that are used in the authentication process. One notable difference is that V2 distinguishes between "private" and "shared" tokens for the Patient. This is not relevant for the interface described by V1, however it is important, since the latter does not allow a user to interact with the system as the subject (Patient) of this token, but only to access corresponding data.
APIs	Applications	The applications depicted in V2 makes use of the corresponding APIs from V1. Each application only uses the API for the appropriate scope (public for consumer applications, private for professional Healthcare Provider applications)

Table 20: Mapping between V1 and V2

7 Design decisions and Rationale

This section explains why the architecture is the way it is. In here, we focus on the most impact full architecture design decisions we made during the project. We show the design decisions and their rationale. We try to make clear made why we chose to document precisely those properties and why we have considered those properties.

Token based authorization

We use token based authorization to grant the Healthcare Providers access to Patient data. Because of the concerns of accountability, privacy and authorization, this design method is chosen over not having authorization for the "trusted" users that are Healthcare Providers. Despite the fact that the design would be more convenient for the users, we require patients to generate an access token based on their identity token which can be transferred to the Healthcare Provider (for example by scanning a matrix barcode during an appointment). Since this code uses asymmetric cryptography, the Patient is not required to be online to transmit the authorization or for any access to their data by the Healthcare Provider. This method also allows to limit access in scope, expiration and Healthcare Provider by encrypting this information in the access token (similar to personal and seating information printed on a venue ticket). By using this method instead of storing the authorization in a separate data store, a "man in the middle" attack is not possible.

Direct access to database for healthcare providers

The Healthcare Providers need to access the database immediately and therefore need to see the most up to date data. Therefore direct access for writing and reading data from the database is preferred over a data stream which offers eventual consistency. Despite the fact that the use of a data stream results in higher performance, it is necessary that the Healthcare Providers can reflect on the most up to date data immediately. This is necessary, because the they need to make healthcare choices based on accurate data. The system will be able to deal with this, since access to the system of records by the healthcare providers is low-frequency and low-volume compared to the incoming sensor data from consumer IoT-devices. Sharding on a per-user or per-specialty basis is still used to balance the load between the individual EPRs.

Use of seperate systems of record

We use several systems of records for different types of data that are stored by the system. This is done for several reasons: Different types of data are stored in different formats[11] and physically separate data for security reasons. This makes it unlikely that someone who has access to the data store containing configurations for the Solution Developer ("App inventory") can escalate his access privilege to include access to medical data of Patients.

Use of identity management platforms

We decided to use two government provided identity management platforms, DigiD and UZI. The DigiD consists of a username and password that are linked to each person's personal public service number (BSN). It allows the users to log in anywhere easily and securely. It also ensures that their personal data is always protected. Meanwhile, UZI is a platform that enables unique identification of healthcare providers and care assessment agencies.
Having access to these two platforms simplifies authentication of users and improves the interoperability between the HSSE and other 3rd party healthcare solutions. Using existing identity providers also eliminates the need to on-board Healthcare Providers and Patients to yet another platform with new credentials which should improve the rate of adoption for this solution.

[11] Logs could be stored in a NoSQL database like MongoDB that can handle large amounts of records whereas Elasticsearch might be used to efficiently find diagnosis documents in the EPR.

Use of data streams

A data stream consists of an event queue that can process a series of data elements ordered in time. The data represents an "event" or a change in state that has occurred and mostly consists of sensor measurements to be processed in the HSSE to aggregate and analyze, often in real-time. Another advantage of data streams is that they can be used to provide insight into large amounts of data over time. This works very well with the Healthcare Providers who want to access Patients' data and infer any insightful discoveries or detect anomalies in real time and receive notifications when something requires their attention instead of having to fetch the data periodically. In order to allow this implementation to scale to millions of users, several pipelines using such streams can be deployed in parallel (shards) over which the Patients are distributed, since all measurements only correspond to a single user.

Use of two RESTful APIs

The system utilises two separate RESTful API endpoints. This is done for several reasons: routing traffic between two endpoints can reduce the chances of a single endpoint becoming congested, and each endpoint performs different business logic. The first (public) API, is used with consumer devices and applications, and is authenticated using the DigiD identity management platform. The second (private) API, is used for IoT Devices and Professional Apps. It is used by Healthcare Providers to access Patient data, and is authenticated using the UZI-register. Separating these two access points also adds an additional layer of security, since the most sensitive information cannot be accessed or modified through the public interface.

Use of "serverless" functions

The system utilises a serverless architecture using separate resources that are connected through a network. There are several reasons for this, it helps with scalability and enables flexible deployment. Since the system is modular it can be maintained at ease. The source of issues can be spotted easily and modular deployments can be made without affecting the entire system. The use of serverless functions also enables elastic scaling, allowing the system to handle increasing amounts of traffic, while preventing congestion.

8 Assessment

In this section, a number of scenarios is prepared for assessing our architecture. The scenarios are chosen by analysing the stakeholder profiles and they are expressed in Quality Attribute Scenario's. First, all the scenarios are shown in the utility tree, see figure 5 and 6 and explained, see table 21. Second, the most high-ranked scenario's are analyzed and sensitivity points, trade-off points, risk and non-risk and risk theme's are identified.

Utility tree

Figure 5 and Figure 6 contain the utility tree. (The root notes on the very left side of both figures are the same and are merged when re-combining the figures into the original utility tree.) The colors indicate the corresponding stakeholders for each quality attribute scenario as described in Figure 4. To the right of the quality attribute scenario we have ranked them in terms of business value and architectural impact as either high (H), medium (M) or low (L). In table 21, the general ranking criteria are explained. Scenarios selected for further assessment are marked with a diamond containing a short identifier.

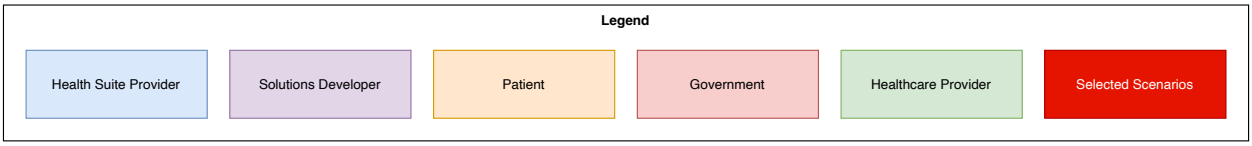


Figure 4: Legend for the utility tree

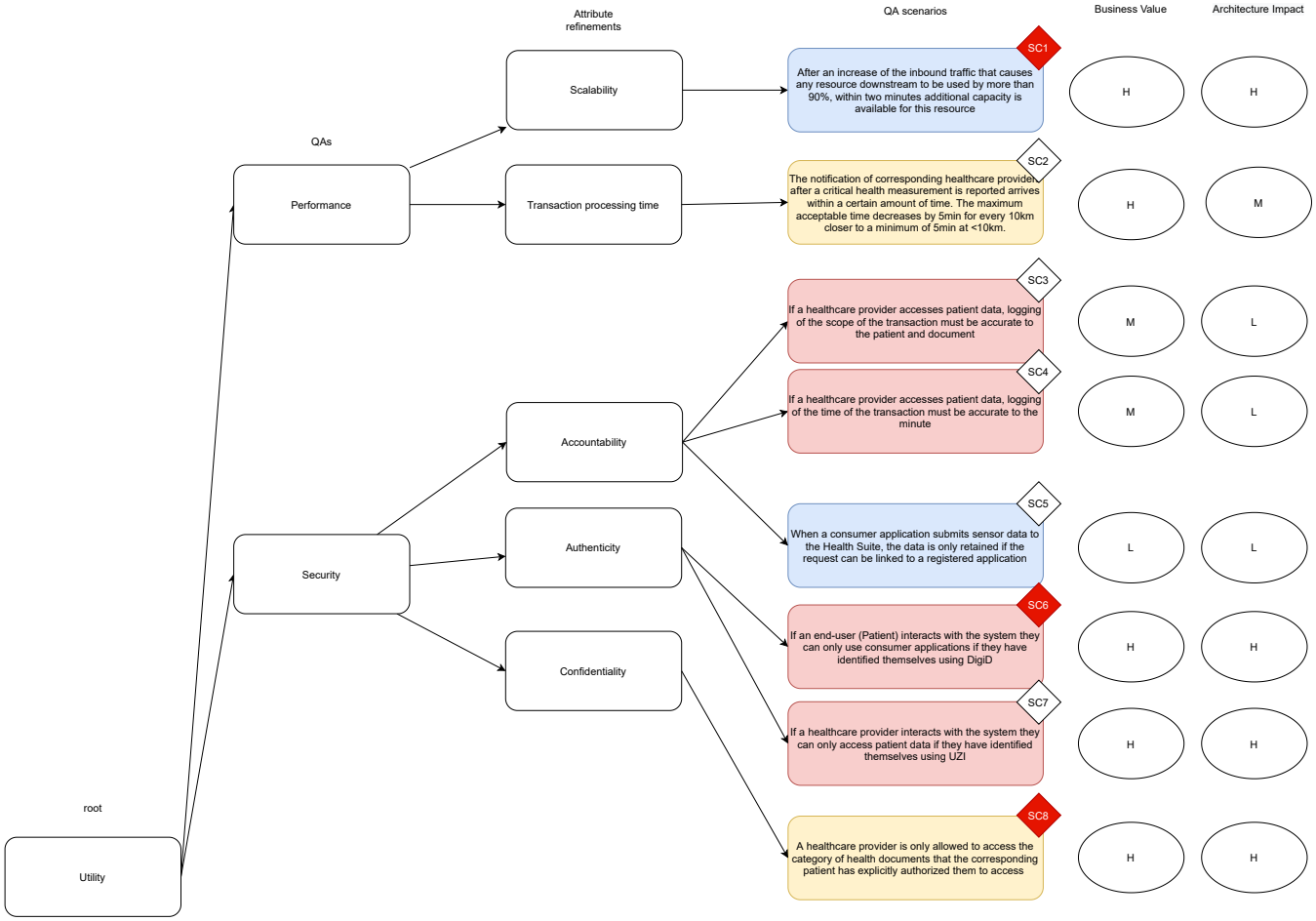


Figure 5: Utility tree (1/2)

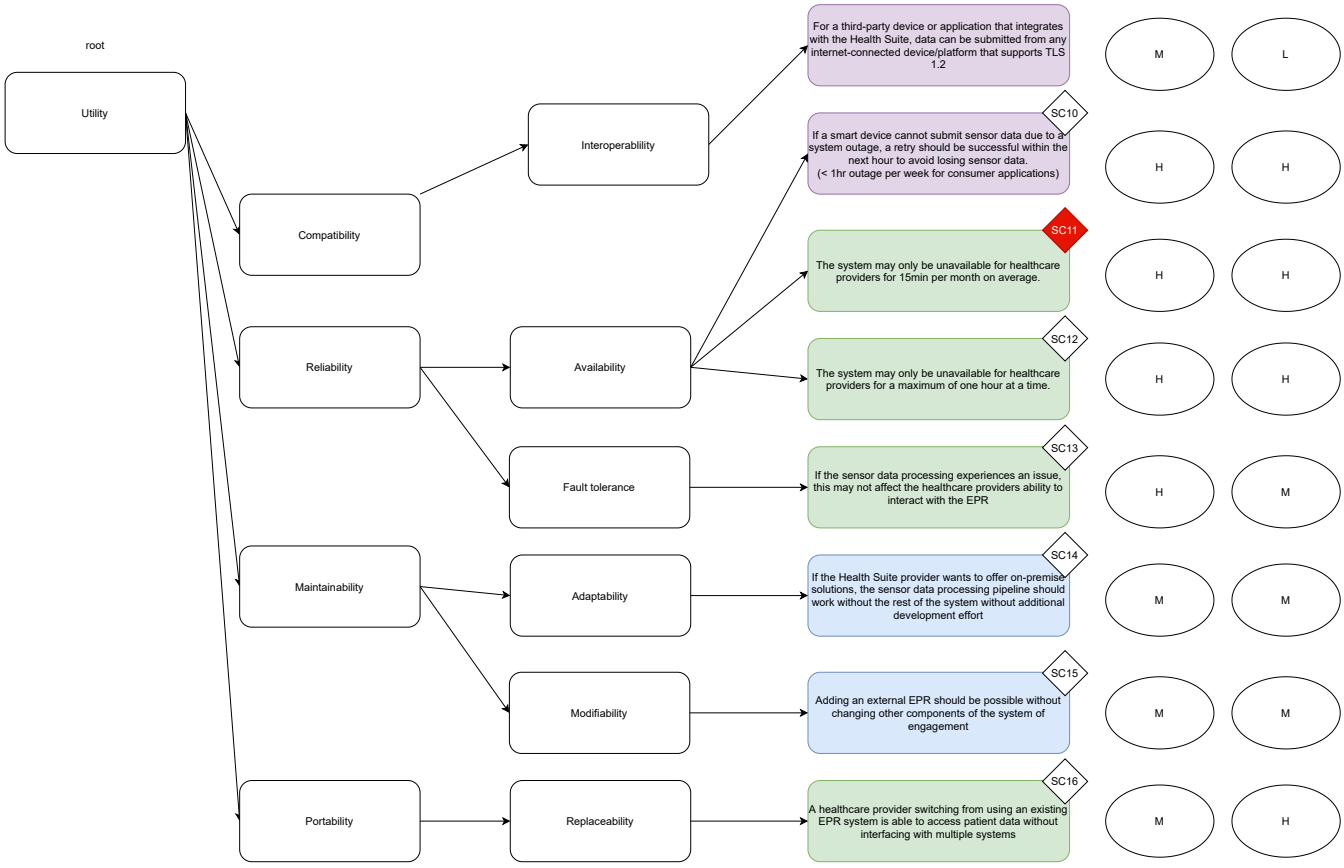


Figure 6: Utility tree (2/2)

General ranking criteria

The following table (Table 26) elaborates on the ranking of the scenarios found in the utility tree in Figure 5 and Figure 6.

Scenario	Business Value	Architecture Impact	Related BGs
SC1	**H:** the number of Healthcare Providers and Patients using the system can change abruptly. Furthermore, the COVID-19 situation proved that Healthcare Providers are working near full capacity. So, we need the system to be adaptable and scale up when needed.	**H:** We would need to to have multiple "additional" resources ready for deployment. Additionally, we would need proxies to balance the load between the resources. We also have to consider data partitioning and fragmentation.	BG1

SC2	**H:** This way, (almost) direct patient healthcare can be ensured after a critical health measurement. This addresses one of the most important requirements of the patient	**M:** We need to have a fast information flow	BG6
SC3	**M:** Logging access data is important to trace the purpose of every transaction (data access). However, omitting it would not lead to project failure.	**L:** Logging access data would not change how the architecture is designed	BG11
SC4	**M:** Logging access data is important to trace the time of every transaction (data access). However, omitting it would not lead to project failure.	**L:** Similar to **SC3**	BG11
SC5	**L:** We consider the ability to weed out spam requests from non-registered applications as a nice bonus, however it is better to use resources for other scenarios.	**L:** This barely impacts the architecture, as we only to compare incoming data to a known list of registered applications.	BG10
SC6	**H:** Having end-users (Patients) to be authenticated is a strict requirement because we need users' identities to be verified so we are able to provide more functionalities as they are identity-sensitive	**H:** Only authenticated users have full access of the system, further application operations will only be possible after authentication process is successfully completed	BG12
SC7	**H:** Similar to **SC6**; however, the target end-user is Healthcare Provider	**H:** Similar to **SC6**; however, the target end-user is Healthcare Provider	BG13
SC8	**H:** Data confidentiality is very important both for the Governmentand the Patient. Our project would not be complete or even considered functional without fulfilling this scenario.	**H:** The system has to integrate two 3rd party platforms as well as have a procedure for sharing an authorization token between them.	BG5
SC9	**M:** TLS provides both high performance and improved reliability compared to previous versions. However, it is not crucial for the overall functionality of the system.	**L:** Supporting TLS 1.2 will have little effect on the architecture	BG7
SC10	**H:** Sensor data availability is crucial in order to avoid Patients being stranded if their health reaches critical health conditions at the same time as a system outage. Therefore, avoiding	**H:** The architecture has to be adjusted to account for data redundancy as well as introducing event queues. This limits some of the possible options	BG9

SC11	**H:** This is a must-have for the business because the customers expect that they (almost) always are able to interact with the system and that the system is not longer unavailable than 1 hour.	**H:** This has high impact on the architecture, because there are multiple elements in the architecture that could cause for unavailability of the system. Therefore, all these components need to have high certainty.	BG17
SC12	**H:** This is a must-have for the business because the Healthcare Providerneeds to connect to the system at least within an hour. Otherwise, the system is not useful.	**H:** This has high impact on the architecture, because there are multiple elements in the architecture that could cause unavailability of the system. Therefore, all these components need to have high certainty.	BG17
SC13	**H:** This is an must-have for the business because the customers expect that they (almost) always are able to interact with the EPR	**M:** this somewhat affects the architecture: there need to be an decoupling between the sensor data processor and the connection between the Healthcare Provider(user) and the EPR.	BG18
SC14	**M:** This is an important scenario because the on-premise solutions have a high impact on the system, but will not lead to failures of the project if it is not implemented. Then, there just need to be additional development effort.	**M:** This is a one time implementation and will not affect the architecture on a high level.	BG2
SC15	**M:** the number of customers and users will increase when they can easily add an external EPR	**M:** A simple but decent way for connecting an EPR has to be included in the architecture.	BG2, BG4
SC16	**M:** There can be a lot of different customers with all their own existing EPR systems. The user experience will form a higher quality when the Healthcare Provider does not need to switch between multiple systems.	**H:** There need to be a well-defined overarching system that connects all the EPR system data. Because there can be multiple customers with different technical infrastructures, it has a high architecture impact.	BG19

Table 21: Business value and Architecture impact ranking criteria

Analysis of architectural approaches

There are 7 scenarios with a high business value as well as a high architecture impact. They are: **SC1, SC6, SC7, SC8, SC10, SC11, SC12**. For us, we decided that **SC1, SC6, SC10, SC11** are challenging the architecture the most as well as providing a varied perspective on different attributes. See also table 21 for more explanation.

Scalability

Related to the most important Business Goal of the Healthcare Provider, BG1, this scenario has high impact on the business value and the architecture. The scenario will lead to project failure if it is omitted. Therefore, we examine the sensitivity points, trade-offs, risks and non-risks. See table 22.

Scenario	**SC1: Horizontal scaling on demand**
Attribute(s)	Scalability
Environment	Operation
Stimulus	Resource capacity nearly exceeded due to rapid user increase
Response	Provision additional resources and balance load (horizontal scaling):Available capacity of all resources is at least 10% higher than the provisioned capacity two minutes since the last increase in load

Architectural decision	**Sensitivity**	**Trade-off**	**Risk**	**Non-risk**
D1: Sharding sensor data submission by user	S1: Distribute load across dynamic number of resources	T1: Deployment complexity	R1: Harder to debug (user-specific errors)	N1: Increased deployment & operational complexity
D2: Using multiple (existing) EPRs through a proxy reduces the load on any single one of them. New systems of records can be added at any time to deal with increased storage requirements.	S2: Distribute load across multiple resources	T2: Fragmentation of patient data across multiple systems of record.	R2: Accidental inconsistency due to data duplication caused by the "re-creation" of a document or document-category	N2: Increased deployment & operational complexity managing the integration of multiple EPR systems

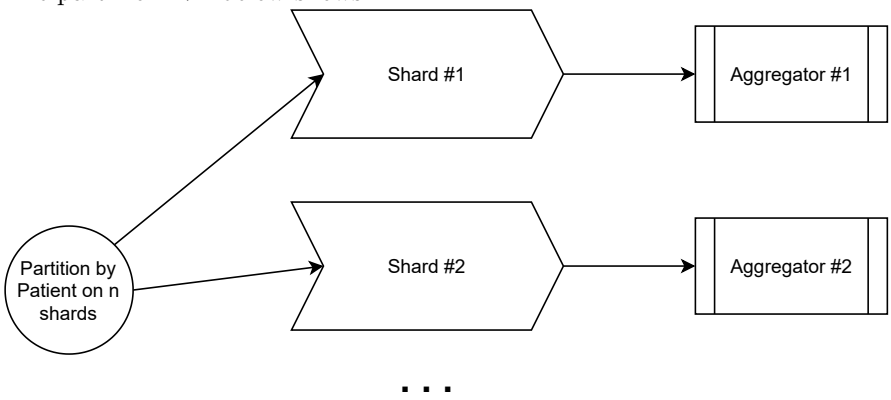
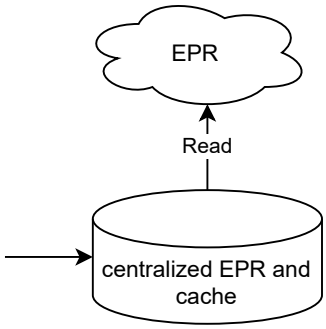
Reasoning	The goal of these architectural decisions was to address the load on the resources that are expected to experience the most amount of load. (S1, S2) Incoming sensor data can be handled independently from each user, so this pipeline can be instantiated multiple times. Since different systems of records for storing Patient data already exist, they can be leveraged to distribute the requests from the Healthcare Providers. The resulting increased complexity (N1, N2) is not important, since the HSSE is already a complex system and requires skilled, dedicated operations personal. However, the distribution of the pipeline between multiple groups of users introduces the risk of "local" errors which only apply to a subset off the user-base and are thus hard to debug and lead to an inconsistent experience for different users. However, this is an acceptable risk for ensuring that this service is available to as many users as possible. T2 is an acceptable trade-off, since it allows the re-use of existing EPRs. R2 may occur if a Healthcare Providers attempts to create a document in the EPRs that already exists and the new document is stored in a different EPR system. This can be avoided however by following a "pull-before-push" process when implementing client applications that interact with the HSSE.
Architecture Diagram	The part from V1 below shows D2:  The part from the architecture overview (Figure 1) below shows D2. "EPR" represents an arbitrary number of such external systems which store the Patient data: 

Table 22: SC1 Horizontal scaling on demand

Authenticity

Related to the Business Goal of the Government, BG12, this scenario has high impact on the business value and the architecture. The scenario will lead to project failure if it is omitted. Therefore, we examine the sensitivity points, trade-offs, risks and non-risks. See table 23.

Scenario	**SC6: Patient attempts to access consumer applications without authenticating themselves with DigiD**
Attribute(s)	Security
Environment	Operation
Stimulus	Patient's authentication status is verified after log-in
Response	Restrict access to consumer applications and prompt the Patient to authenticate with DigiD

Architectural decision	**Sensitivity**	**Trade-off**	**Risk**	**Non-risk**
D1: Patient is able to use limited applications if authenticated using other ID documents, full access is granted after authenticated with DigiD	S1: Since the database has data for both DigiD-verified Patients and non-DigiD Patients, the sensitivity point is the authenticity of two different groups of Patients	T1: Less reliability for authenticity, more usability for Patient, but more flexibility to use other identity providers that comply with the standard identity source interface.[12]	R1: Hard to fully verify Patient's real identity without DigiD (dependency on third party system)	N1: Increases the complexity of Patients' IDs
D2: Interactions with the system are not allowed for unauthenticated Patient	S2: The security of the Patients' identities by enforcing authentication	T2: Maximum reliability for authenticity, no access for unauthenticated Patients	R2: Significantly affects the usability of the system, because authentication process is required for unauthenticated Patients	N2: Increased maintenance effort for a more strict DigiD authentication process

[12]This opens up the possibility to expand into other markets and also to integrate users outside of the chosen identity ecosystem such as unregistered people.

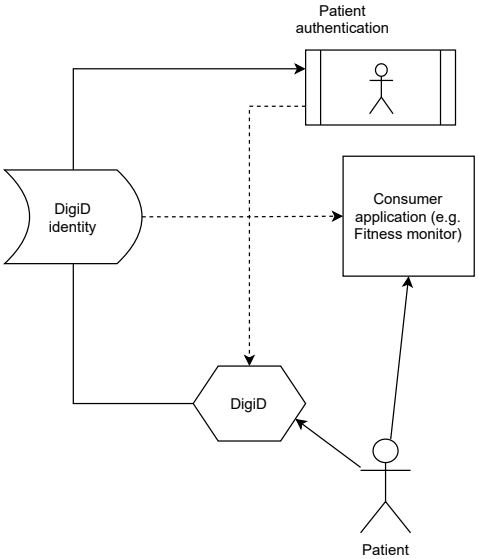
Reasoning	The goal of the authentication process is to increase the security of the system and avoid having unauthenticated users from accessing the system. For D1, the decision focuses more on the user experience (usability) while providing a moderate level of security (T1). But because allowing authenticating using documents other than DigiD may increase the risk of having Patients with fake IDs, it becomes harder to verify (R1), this is not an optimal trade-off. As for D2, it is set that Patients will not be able to use any of the consumer applications if they have not been successfully authenticated with DigiD. It strictly forbids unauthenticated Patients' ability to access applications and submit inputs, which maximizes the security of the system (T2). The downside is that only the DigiD-verified Patients are able to enjoy the functionalities of the system (R2) which may turn potential users away from the system.
Architecture Diagram	The part from V2 below shows D2: 

Table 23: SC2 Patient attempts to access consumer applications without authenticating themselves with DigiD

Confidentiality

One of the Business Goals of the Patientis the BG5. This scenario has high impact on the business value and the architecture. The scenario will lead to project failure if it is omitted. Therefore, we examine the sensitivity points, trade-offs, risks and non-risks. See table 24.

Scenario	**SC8: Healthcare providers only accessing records authorized by the patients**
Attribute(s)	Security
Environment	Operation
Stimulus	Patient data is requested
Response	**Only** gain access to the authorized Patientdata

Architectural decision	**Sensitivity**	**Trade-off**	**Risk**	**Non-risk**
D1: Patient shares a token with the Healthcare Provider directly which includes the scoped access authorization[13]	S1: Access of the Healthcare Provideris limited to what is specified by the token. This is transparent to the Patient, since they are the one to issue the token to authorize their Healthcare Provider.	T1: This decision improves confidentiality, but it also negatively affects usability, since patients need to be able to generate and share tokens with healthcare providers, which might not be an intuitive process for everyone.[14]	R1: By moving authorization to a token-based flow, the responsibility of revoking a token (Patient) and holding on to a token (Healthcare Provider) lies with the end-users. This might lead to situations, where due to the loss of the token the privacy or health of the Patient may be compromised, if they cannot revoke their token or their Healthcare Provider cannot access their files.	The use of tokens for authorization requires a form of ensuring the trustworthiness of tokens which may be compromised when self-signed tokens are accepted by the system, allowing an attacker to forge then and obtain access to sensitive data. However, only two keys (one for each identity provider) need to be kept up to date in the system. This constitutes an acceptable operational security challenge.

[13]This can be compared to, for example, the seat and flight number that a ticket issued by an airline is restricted to. The agent at the gate only needs to verify the authenticity of the ticket and the identity of the passenger.

[14]With the rise of digital health certificates, such as COVID-vaccination passports, this is an acceptable trade-off.

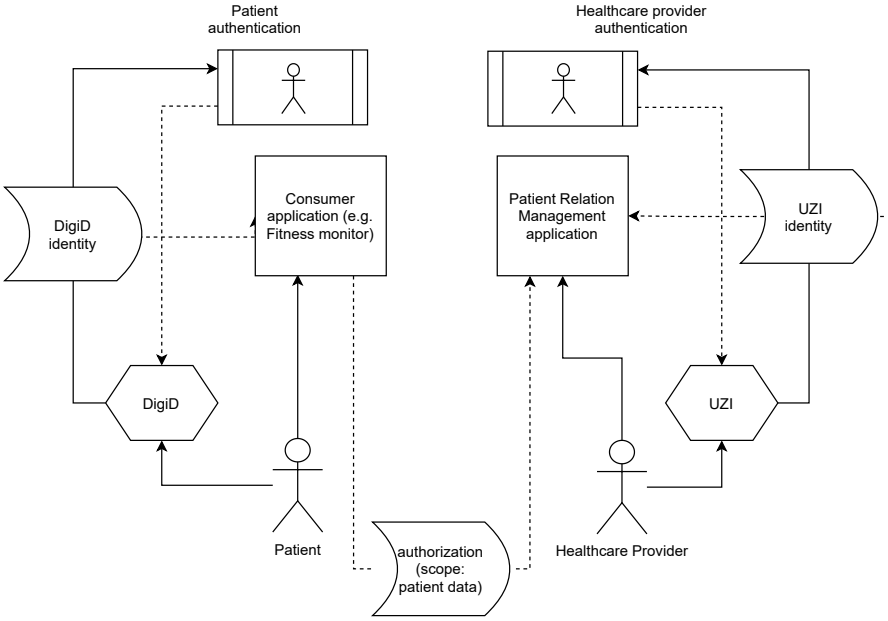
Reasoning	Patients (Patient) share a scoped token with their healthcare provider, which allows the latter to access a specified set of confidential patient records (S1). the patient has full control over the scope of the available records available for access. However, this negatively affects system usability (T1) as the token sharing adds an extra process which might not always be convenient and intuitive for all patient groups. This also puts a lot of trust in the end users to make sure their tokens are always secure and available to them. Otherwise, the patients might compromise their confidentiality (R1) or the healthcare providers can be locked out of viewing their patients' data (R1). Finally, there exists the possibility to forge self-signed tokens which leads to a breach of security. However, this is considered as a non-risk (N1) as it can be dealt with by keeping one of the keys (for each identity provider) up to date.
Architecture Diagram	See an excerpt from V2 User Identity which shows D1: 

Table 24: SC3 Healthcare providers only accessing records authorized by the patients

Availability

For the Healthcare Provider, the system is valid when it has high availability, see QA17. Thus, this scenario has high impact on the business value, but also profoundly affects on the architecture. Therefore, we analyse this scenario, see table 25.

Scenario	**SC11: he system may only be unavailable for healthcare providers for 15min per month on average**
Attribute(s)	Reliability
Environment	Normal Operations
Stimulus	Device cannot upload sensor data due to a system outage
Response	99.9997% availability of the databases

Architectural decision	**Sensitivity**	**Trade-off**	**Risk**	**Non-risk**
D1: Redundancy/ "Back-up" of databases through centralized cache	S1: Having the data stored in multiple databases will affect the reliability of the system positively. When one of the databases fails, not all the information will be unavailable, only a part of it	T1: Having multiple data stores affects maintainability.	N/A	N1: However, this is considered as a non-risk (N1), since the HSSE operational staff will have experience with multiple data stores and therefore "know the ins and outs" of these systems
D2: Event producers and consumers are decoupled through the event queue	S2: This decision ensures that a failure of the databases does not affect the information gathering. When a database fails, new information can still be produced	T2: This architecture decisions will affect the performance negatively, because the data will not directly enter the database	N/A	N2: The staff has also have experience with decoupling the producers and consumers, so this high rate of complexity is marked as a non-risk.
D3: Event queue implements buffering and retry logic	S3: This decision makes sure that the insertion of the information is not failing immediately. It stretches the time by buffering and retrying	T3: This architecture decisions will affect the performance negatively, because the insertion of information will be slowed down	R1: This is a risk, because this can create an accumulation of information flows.	N/A

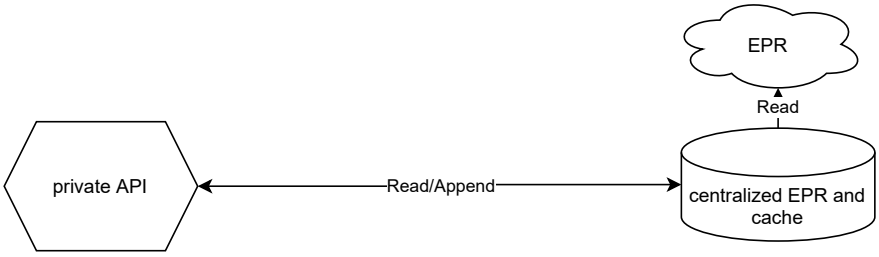
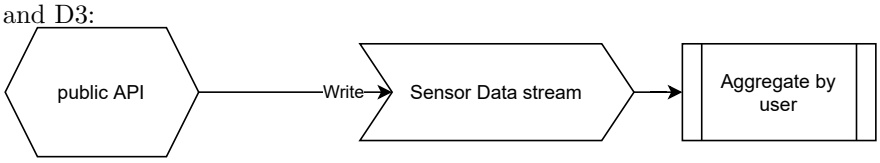
Reasoning	The first decision affect the reliability of the system positively (S1). When one of the databases fails, not all the information will be unavailable, only a part of it. However, the second decision ensures that a failure of the databases does not affect the information gathering. When a database fails, new information can still be produced. (S2) The last decision makes sure that the insertion of the information is not failing immediately. It stretches the time by buffering and retrying. (S3). Having multiple databases affects maintainability. (T1) However, this is considered as a non-risk (N1), since the staff will have experience with multiple databases and therefore know the ins and outs. The staff has also have experience with decoupling the producers and consumers, so this high rate of complexity is marked as a non-risk (N2). The other two architecture decisions will affect the performance negatively, because the data will not directly enter the database (T2) and the insertion of information will be slowed down (T3). The last is seen as a risk (R1), because this can create an accumulation of information flows.
Architecture Diagrams	The following part of the architecture overview (see Figure 1) visualizes D1. "EPR" represents an arbitrary number of such external systems which store the Patient data:  The following part of the architecture overview (see Figure 1) visualizes D2 and D3: 

Table 25: SC4 Detection and recovering from the system's outage

9 Glossary

API Application Programming Interface (Most commonly an HTTP-RESTful interface that defines how other programs may interact with a given system and offers the corresponding functionality to be invoked programmatically.). 5, 10–18, 22, 24, 25, 30, 32

ASR Architecturally Significant Requirement. 3

BSN Burgerservicenummer (Dutch Citizen Service Number). 31

CS Computer Science. 1

EPR Electronic Patient Record (Electronic medical data for a single patient including, but not limited to images, diagnoses, measurement readouts, prescriptions that are stored in a system of records). 9, 15, 16, 24, 25, 28, 29, 31, 37–39, 45

FIFO First In, First Out. 25

HSSE Health Suite System of Engagement (subject of this document). 4, 8–11, 13, 15, 16, 18, 19, 22, 24, 27, 28, 31, 32, 39, 44

HTTP Hypertext Transfer Protocol (Text based protocol for transmitting documents on the web). 6, 18, 24, 25, 46

IoT Internet of Things (Internet connected devices that are not primarily computing devices). 14–16, 28, 31, 32

JWT JSON Web Token (A standard for implementing cryptography signed tokens that replace credentials following initial authentication in order to facilitate secure and scoped transactions in authenticated systems). 22, 24, 28

KYC Know Your Customer (Protocols and principles of customer identity verification before the initiation of business). 16

N/A Not Applicable. 44

PoW Proof of Work (A distributed consensus mechanism that is used in mining Bitcoins and email spam prevention). 18

REST Representational state transfer (see `https://restfulapi.net/`). 6, 24, 25, 32, 46

ROI Return on Investment. 8, 11

SEG Software Engineering & Green IT. 1

TLS Transport Layer Security (Standard security protocol for the internet that uses "public-private key"-cryptography). 11

ToC Table of Contents. 4

UML Unified Modeling Language. 6, 20, 21, 23

UZI Unieke Zorgverlener Identificatie (Dutch Unique Healthcare Provider Identification). 12, 14, 16, 28, 31, 32

WIP Work in Progress. 3

References

[1] "Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models," International Organization for Standardization, Geneva, CH, Standard, Mar. 2011.

[2] "Systems and software engineering — Architecture description," International Organization for Standardization, Geneva, CH, Standard, Dec. 2011.

[3] "Information technology — Object Management Group Unified Modeling Language (OMG UML) — Part 1: Infrastructure," International Organization for Standardization, Geneva, CH, Standard, Apr. 2012.