Tab 1

9 Apr 2025

Smart home system

Architectural analysis - Final report

Team Eternal Architects

Al-Tuwaijari Ahmed
Kianiangolafshani Sepehr
Mareš David
Porthan Richard
Robin Asaduzzaman
Seek Anakin
Virolainen Julius

Table of Contents

Introduction

Our startup, Eternal Architects, is developing a software system to help customers optimize their electricity by using a mobile app or a web page to view information and control smart devices. Our proposed solution is a hardware device that allows users to monitor smart devices in their household, collect data on electricity consumption and production habits, and provide electricity usage recommendations and automated smart device control to optimise electricity usage. We are mainly targeting single detached homes and housing companies. The solution is intended to be deployed at a large scale, with one instance per household. One instance may support multiple users.

Our team consists of computer science experts with diverse specializations. As a startup, we aim to leverage modern technologies and achieve a potential for high profitability in the future, by focusing on rapid and efficient scaling. For scalability, the solution needs to be easily adoptable by users, catering to consumers with varying levels of technical knowledge. We are also facing certain near term cost pressures due to our limited funding.

The high-level solution involves three main parts: gathering data to understand electricity consumption, using this data to optimize consumption, and interfacing with smart systems to provide usage recommendations and possible automation. The main complexity will doubtlessly lie in how to integrate various smart appliances, and we need to explore existing standards and standardization drives in the smart home and energy sectors (if there are any).

The software project, while conceptually simple, involves significant complexity in execution. The system's reliability is critical, as any failure could lead to suboptimal electricity consumption and higher costs for consumers. With no existing baseline architecture, and no relevant organisational praxis, we are free to explore novel technologies and designs. On the other hand the electrical energy markets are highly standardized and heavily regulated, so we do not expect significant changes in the external context.

Data will most likely be managed by us, since households would otherwise need some form of private server to manage the data, and this requires robust data protection and security measures. Additionally, we must ensure compliance with relevant regulations and standards in the electrical energy sector.

In summary, our project aims to create a scalable, innovative solution that optimizes electricity consumption costs for consumers. By addressing the complexities of smart-appliance integration, ensuring reliability, and defining a clear business model, we can position our startup for success in this highly regulated and standardized market.

Stakeholder analysis

In this section, key stakeholders of the system are presented and their needs and concerns are identified.

The table below presents the key stakeholders of the system and why they were included into consideration. They are ordered according to how “close” their relationship to the system is, from closest to farthest, with the exception of the dev team, which is ordered last. Note that the ordering *does not imply* any prioritisation of stakeholders.

Key stakeholders		
Id	Stakeholder	Who and why?
USR	Users	● Includes dwellers, owners and housing companies. ● The main customer of the product, commercial success depends on whether we can provide users with value
MAN	Smart appliance manufacturers	● Companies that produce smart appliances ● An essential part of the smart grid ecosystem, their devices are the devices our system aims to interface with
DAT	Data consumers	● Organizations and individuals who want to utilize data produced by the app. ● An essential part of the smart grid ecosystem
GOV	Local governments and the EU	● Provide the legal framework within which we must work in
DEV	Dev team	● The people working to make the product ● Their skills etc. shape the product including its architectures

The following table identifies and describes the needs and concerns of the stakeholders presented above. We define a *stakeholder need* as something that said stakeholder would want to have or something that they would want to happen, and *concern* as a demand or restriction on how their needs are addressed.

Each need/concern has an id consisting of either ND-NN (for needs) or CRN-NN (for concerns) and is prefixed with the stakeholder it belongs to. They are presented in order of the stakeholder which they concern, with stakeholders ordered as in the table above.

Again, note that no prioritisation is taking place. Also note that DEV and GOV stakeholders do not have any listed needs, and concerns, since they are not expected to directly interact with the functions of the system. They do, however, impose some general constraints on the architecture, which are discussed in the next section.

Needs and concerns			
Id	Stakeholder	Type	Description
USR-ND-01	USR	Need	Users want to save money
USR-ND-02	USR	Need	Users desire visibility of their electricity consumption and production
USR-ND-03	USR	Need	Users desire *real time* visibility of their electricity consumption and production
USR-ND-04	USR	Need	Housing companies want to monitor/manage multiple apartments at once
USR-CRN-01	USR	Concern	Users want to do minimal work to save money
USR-CRN-02	USR	Concern	Users do not want to learn to use complicated or cumbersome systems
USR-CRN-03	USR	Concern	Users want to protect their privacy
USR-CRN-04	USR	Concern	Users want to protect their data
USR-CRN-05	USR	Concern	Users do not want to get hacked
MAN-ND-01	MAN	Need	Integration capabilities with the system (e.g., APIs, protocols)
MAN-ND-02	MAN	Need	The system to allow users to customize appliance settings for energy efficiency
MAN-CRN-01	MAN	Concern	How much control the system will have over their devices
DAT-ND-01	DAT	Need	Access to accurate energy usage data
DAT-ND-02	DAT	Need	Clear guidelines on data ownership and privacy regulations

External Constraints

This section presents significant external constraints that arise from the context that the system exists in. The constraints discussed below are expected to influence the system architecture in some way.

Constraints have been identified to come from three main sources:

- **Project constraints** include time, budget, and resources such as personnel and infrastructure. These factors naturally affect what architectural decisions are viable. Notably, they are also the most dynamic, as risks and opportunities arise throughout the project.
- **Regulatory constraints** define the legal framework within which we must operate. A key requirement is compliance with data protection regulations, such as the GDPR. The specific regulations depend on the countries in which the product will be deployed.
- **Industry standards** impose additional constraints, particularly given the product's role in a smart grid ecosystem. For example, standards set by the International Electrotechnical Commission (IEC) ensure compatibility in energy management by defining common data exchange formats. There are also critical cybersecurity and safety standards that must be adhered to.

These external constraints and their relationship to the system are displayed in the table below. Each constraint has been given an id, one or more types, and a description. Relevant stakeholders are also mentioned.

External Constraints			
Id	Related stakeholder	Type	Description
EXT-01	DAT	Regulatory	Access restrictions or legal constraints on data sharing
EXT-02	GOV	Regulatory	Ensure the system complies with legal requirements, including GDPR and consumer rights
EXT-03	GOV	Regulatory	Compliance with monopoly and fair market regulations
EXT-04	DEV	Project, Industry	That the system is implementable, particularly the interface for smart devices
EXT-05	DEV	Project	Ensuring scalability and maintainability of the system

Architectural principles

For this project our team has decided to adopt the following architectural principles which we believe will help us make the best use of our resources while also helping us focus on customer value and personal development:

Adopted principles		
Id	Principle	Rationale
AP-1	Consumers can access real-time information about their energy consumption through the system.	This is a basic functionality expected by consumers as they want to understand their consumption better
AP-2	All smart energy appliances can be controlled remotely from the system	The system should enable users to make active changes to their electricity consumption
AP-3	The system has to use machine learning models to recommend configurations to minimize electricity costs	The system needs to reduce the knowledge and effort required from the consumer when using the application for monitoring and controlling smart energy devices
AP-4	Application has to not require too many manual configurations	The system needs to reduce the knowledge and effort required from the consumer when using the application for monitoring and controlling smart energy devices
AP-6	The application has to be compatible with diverse smart devices from different manufacturers	We want to reduce the knowledge and effort required from the consumer when selecting smart appliances for their homes.
AP-7	The application should always be operational and available	As the system manages electrical devices for consumers in real-time, a breakdown in the system may cause electricity costs for consumers to spike or even deny them access to energy.
AP-8	The system should be interoperable with most existing smart energy systems	We do not want interfacing with our application to be a tedious process for smart appliance manufacturers. We want to maximize the number of devices our application can control to maximize efficiency for users.

Architecturally significant requirements

This section lists the *architecturally significant requirements* (ASRs) that the proposed system should fulfill in order for it to be considered successful. For brevity, *architecturally significant requirements* and *requirements* are used interchangeably in this document. Also, unless mentioned otherwise, *electricity data* refers to all electricity related data, that is, consumption, production, and costs. The requirements listed here are derived from stakeholder needs and concerns, constraints, and architectural principles.

The tables below presents the requirements of the system that have been identified. Each requirement has an id, a high-level description, a trace to needs, concerns, related constraints and principles, and optionally a link to a concrete scenario. Requirements without scenarios might already be considered scenarios, be very general and thus difficult to concretise, or are otherwise difficult to concretise. The scenarios are presented further below.

Functional requirements (FRs)			
Id	Description	Trace	Scenario
FR-01	The system collects and can provide electricity data on demand, at least six months back	USR-ND-02, USR-ND-03, USR-ND-04, AP-1	SCN-FR-01
FR-02	The system can automatically optimise electricity costs by controlling smart devices connected to the system	USR-ND-01, USR-CRN-01, USR-CRN-02, MAN-ND-01, AP-2	SCN-FR-02
FR-03	The system can provide electricity consumption and production recommendations for the coming week	USR-ND-01, USR-ND-04, USR-CRN-01, USR-CRN-02, AP-3	SCN-FR-03
FR-04	The system can detect abnormal electricity consumption/production/prices in real-time and report them to the User	USR-ND-01, USR-ND-03, USR-ND-04, USR-CRN-01	SCN-FR-04

Non-functional requirements (NFRs)			
Id	Description	Trace	Scenario
NFR-01	The system should be secure	USR-CRN-03, USR-CRN-04, USR-CRN-05	SCN-NFR-01
NFR-02	The application should have a simple user interface by abstracting complex configurations into high-level controls	USR-CRN-02, AP-4	SCN-NFR-02
NFR-03	The system can interface with smart devices using commonly used WPAN (e.g. Zigbee, Bluetooth), WLAN (e.g. Wi-Fi 802.11 standards), and WWAN (e.g. LTE, 5G, NB-IoT) technologies	MAN-ND-01, USR-CRN-02, EXT-05, AP-6, AP-8	SCN-NFR-03
NFR-04	The system must recover automatically in case of failure, in a reasonable time	USR-ND-03, USR-CRN-04	SCN-NFR-04
NFR-05	System data must not be traceable to the system owner	USR-CRN-03, USR-CRN-04, DAT-ND-01, DAT-ND-02, EXT-01, EXT-02	SCN-NFR-05
NFR-06	The system must comply with IEC 61850 and IEC 62351	MAN-ND-01, EXT-04	NULL
NFR-07	Consumption and production recommendations should provide cost savings	USR-ND-01	SCN-NFR-07

Scenarios		
Id	ASR	Description
SCN-FR-01	FR-01	When a user or data consumer requests electricity data for a month within the previous six months, the system provides the data with hourly precision
SCN-FR-02	FR-02	When the system detects that electricity prices are high, the system commands connected electricity producers to dump electricity to the external grid.
SCN-FR-03	FR-03	Every Sunday, the system drafts recommendations for when to use smart devices and when to sell electricity based on previous consumption and actual/predicted future electricity prices
SCN-FR-04	FR-04	When a smart device has used/produced 10% more/less electricity in the previous minute than the average of the last five minutes, the system notifies the user of the event
SCN-NFR-01	NFR-01	When an unauthorized user tries to access a feature of the system, the system denies the request
SCN-NFR-02	NFR-02	When a user accesses the user interface, core functionality should be accessible with a maximum of two clicks of a computer mouse, smartphone touchscreen or two button presses on a keyboard or similar interface
SCN-NFR-03	NFR-03	When a device connects to the same WLAN that the system is connected to, the system acknowledges the new device and offers to the user to start managing it
SCN-NFR-04	NFR-04	When the system loses and regains power, the system, automatically resumes what it was last doing within a couple minutes (if it is reasonably possible)
SCN-NFR-05	NFR-05	When data leaves the system, either authorized by the system owner or not, the data should not be traceable to the system owner (i.e. the system should either anonymize all data or not store *personally identifiable information* at all)
SCN-NFR-07	NFR-07	When the usage patterns recommended by the system are implemented, the user should end up having saved

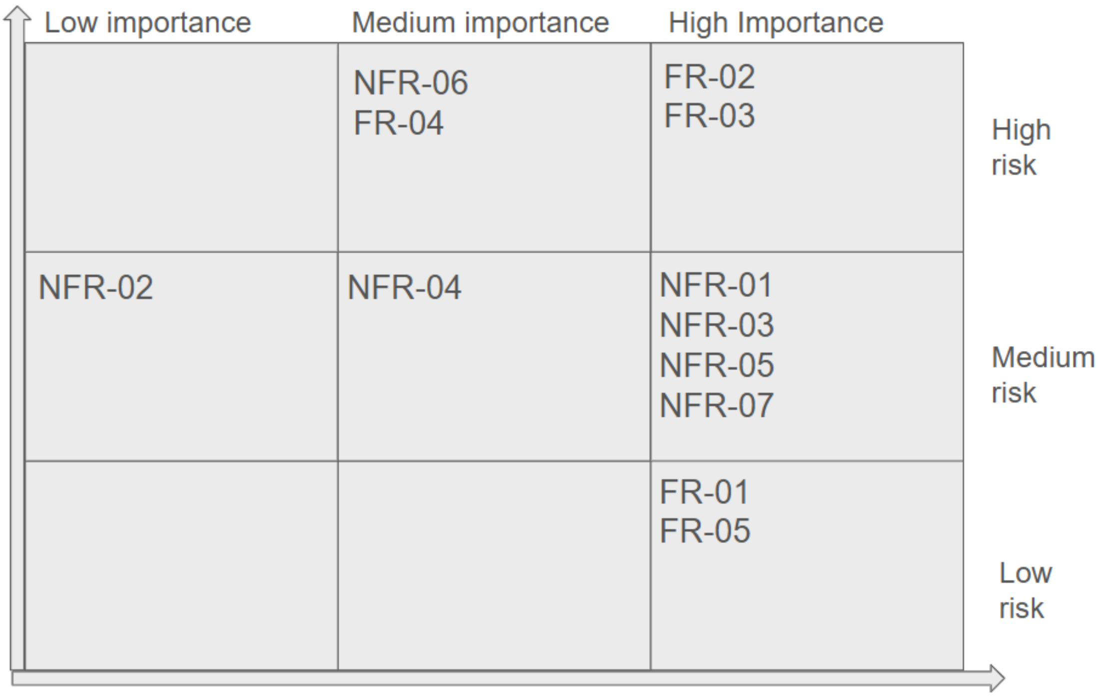
Scenarios		
Id	ASR	Description
		money at the end of the week compared to random usage

Prioritization

This section ranks each ASR relative to each other based on *importance* and *risk*.

Importance can be a deceptive term, since all ASRs listed above are important to the success of the system and thus influence the nature of the system in some important and meaningful way. However, it is also clear that some requirements are more fundamental than others, for instance, one requirement might depend on another. This quality is captured in the importance metric of a requirement. On the other hand, the risk of implementing an ASR should be considered as the relative difficulty of implementation. This might be due to the requirement being ephemeral and vague, or obviously difficult to implement. The risk comes from the potential bad consequences if the system fails to fulfil said requirement.

It is worth noting that the definitions of importance and risk used here are impossible to completely detach from each other, since fundamentally, an important requirement is more risky than a less important one and vice versa. Nevertheless, the relative rankings of each ASR are presented below.



In terms of priority, it is obvious that the *high-importance, high-risk* requirements be most carefully considered, and as such, they will receive the most attention in the coming sections. After that, we consider the *high-importance, medium-risk* requirements as the next-most important, and third, the *medium-importance, high-risk* requirements. The relative prioritization of the latter two categories is largely arbitrary.

Architectural Design

Having analyzed the architecturally significant requirements in the previous sections, the following sections describe Eternal Architects' proposed system for addressing the requirements. The system is described in four views: the context, functional, deployment, and information view, in that order.

Context view

This section describes the context view of the system. The context view describes the relationship between the system and the outside world by identifying and describing the main interactions between the external and the system. In the following, *external systems* refer to key stakeholders and other systems that interact directly with the system and are outside the proposed system.

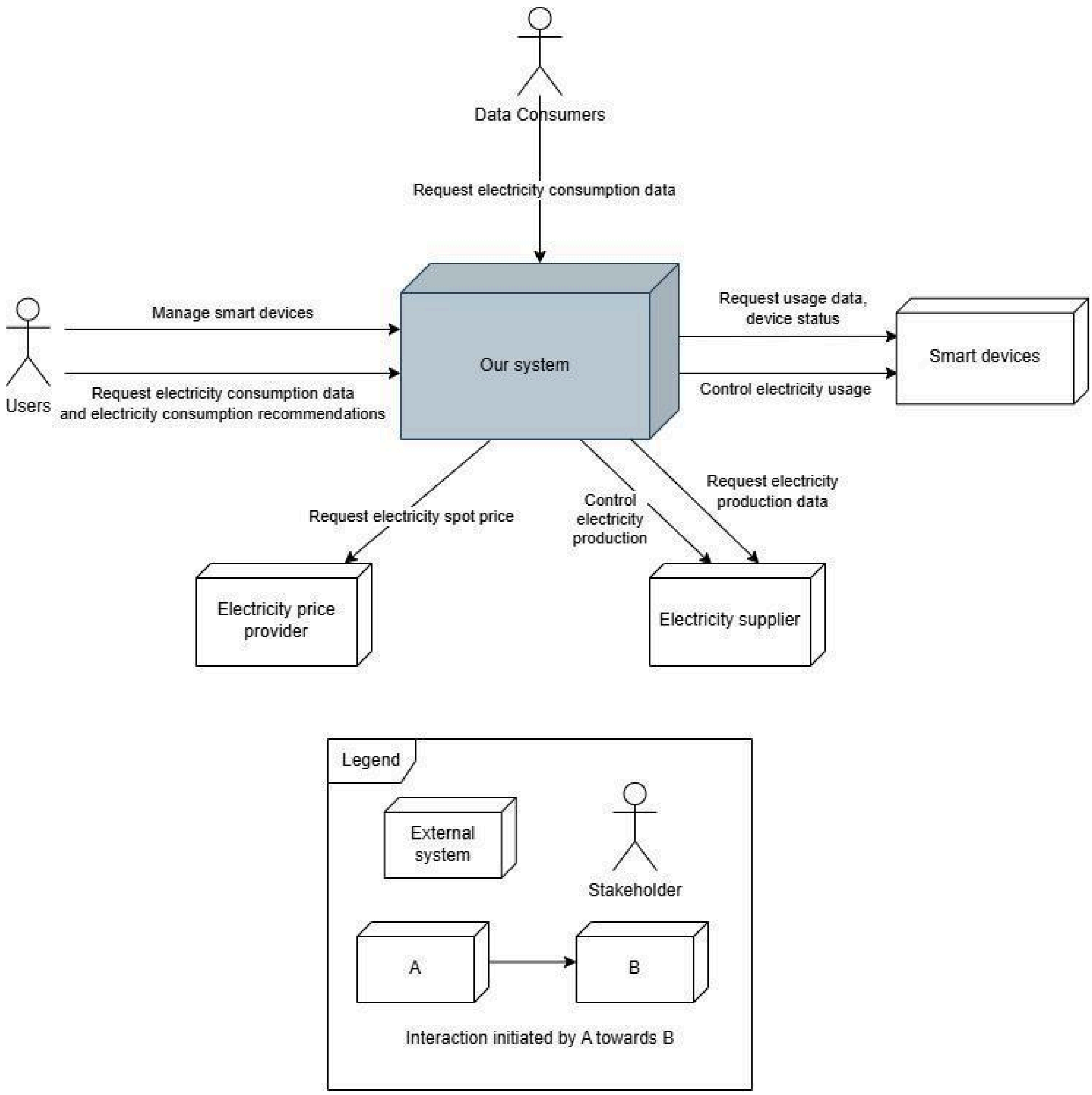
In particular, the following external systems have been identified: users, data consumers, smart devices, electricity price providers, and energy suppliers. In the list below, the relationship of each external system and their main interactions with the system are identified and described. Each interaction addresses one or more functional requirements. Non-functional requirements are largely ignored at this time.

The main external systems and their relationship to the system are the following:

- Users interact directly with the system, and represent the USR stakeholder. They interact with the system in a couple core ways. First, they use the system to gain visibility of their past and projected electricity consumption, production, and total electricity cost of the smart devices connected to the system (FR-01, FR-03). Second, they leverage the provided analytics to save costs by either manually optimising their electricity consumption and production (FR-03), or letting the system do it automatically (FR-02).
- Electricity Price providers are the external authorities responsible for setting electricity prices. These electricity prices are assumed to be provided always one day ahead, and with an accuracy of price per hour. This data is critical to determining optimal usage patterns of devices (all FRs).
- Electricity suppliers refer to the production devices prosumers use to provide electricity. These can refer to solar panels, batteries, etc. Importantly, the system can direct the array of electricity suppliers to store or provide electricity depending on the current and projected price (FR-02).
- Smart Devices are devices connected to the system that consume electricity and, importantly, can provide centralised data about the usage of the device. Common aspects are also remote control capability and scheduling. Similarly to electricity suppliers, the system collects data provided by these devices and provides a centralised platform for manual or automatic device control (all FRs).

- Data consumers can use the data provided by the system to more efficiently manage the larger-scale smart electricity grid, provide aggregated industry analytics etc (FR-01). They represent the DAT stakeholder.

See the diagram below for a graphical reiteration of the above:



Functional view

This section covers the functional view of the proposed system. The functional view divides the system into independent *functional elements* (elements) that interact together to provide the required functionality.

As mentioned, the system has four functional requirements. These requirements are have already been addressed in the context view, albeit not necessarily sufficiently, and therefore will be addressed more in-depth now. In addition, the non-functional requirements, which have not been addressed yet at all, impose certain constraints on the system, and will now be addressed.

In this section and future sections, unless mentioned otherwise, *device* is used to refer to smart devices and electricity suppliers interchangably.

Beginning with FR-01, it is clear that the system needs a way of collecting electricity data and providing it to the user. To accomplish this, the system must be able to: interface with smart devices and electricity suppliers, get electricity prices from external authorities, be able to understand the data they provide, have someplace to store the data, and finally have some way of providing the data to the accessor. It is worth noting that this data will also be provided as anonymized data (NFR-05) to DAT stakeholders. The responsibilities mentioned above are split up into the following elements:

- The Price Collection and Electricity Data Collection services (here, data refers to consumption/production only) are responsible for collecting and managing electricity price and electricity consumption/production data, respectively. Each service is also given it's own database. They are split into separate services since the Data Collection service needs to poll smart devices continuously for data, while their Price Collection service polls external APIs much more infrequently. Additionally, the split provides a clear distinction of what data is external and what is internal.
- The Data Transformation service is responsible for interfacing with and acting as an interpreter between the system and external devices. The service provides support for relevant hardware interfaces and protocols (NFR-03), while abstracting the errant functions of the devices into generic functions like "turn on/off," facilitating ease of use (NFR-02).
- Finally, an interface to these functions is provided by the User Interaction service, which acts as an abstraction on top of which to implement user interfaces.

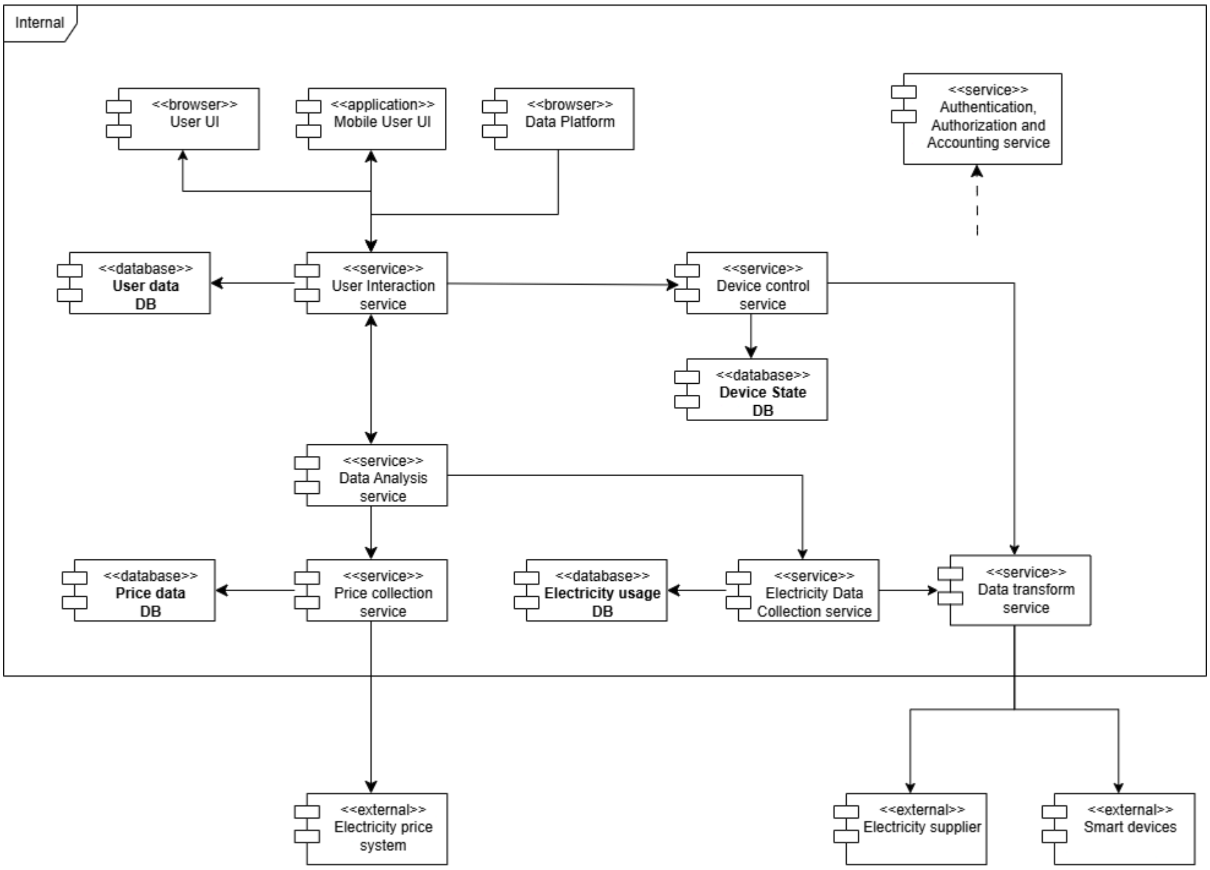
Next, FR-03 calls for an analytics service that could predict electricity production, prices and usage habits, and provide recommendations on usage habits to optimize one's electricity cost. To account for this, the Data Analysis service was introduced. The service uses historical data provided by the data collection services to extrapolate and predict future data points. This data is provided to the interaction service on demand. It is of critical importance that the recommendations provided by the service indeed end up being effective (NFR-07), however this is rather impossible to evaluate in the abstract. See the sections Evaluation of SCN-FR-03 and Prototype and Minimum Viable Product on how this uncertainty is managed. Somewhat similarly to FR-03, FR-04 calls for the continuous monitoring of the electricity data in order to detect potential abnormalities. It makes sense to also have this handled by the Data Analysis service. The Data Analysis service is provided with the ability to alert the User Interaction service in case abnormalities arise.

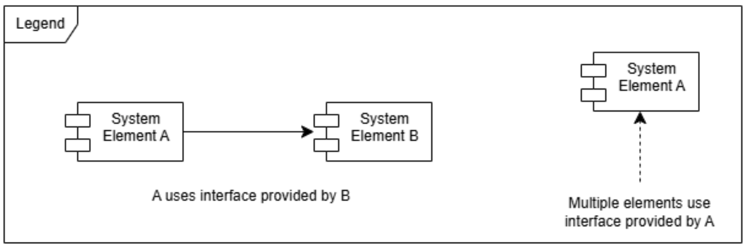
The account for the final functional requirement, FR-02, the Device Control service is introduced. The service monitors the state of each device connected to the system, and controls the devices through the functions exposed by the Data Transform service. The Device Control service does this by acting upon usage parameters provided to it by the user interaction service, either manually provided by the user (FR-03) or automatically (FR-02), which it stores in its own database, the Device State DB, for ease of access. The automatically provided usage parameters naturally come from the Data Analysis service, i.e. the User Interaction service acts as a middle-man between the Data Analysis and Device Control services.

Finally, there are some NFRs that haven't been addressed yet:

- In order for the system to be secure (NFR-01), the errant services must reject service requests from unauthorized users. In turn, for this to be possible, the system must be aware of which users are authorized and which are not. To account for this, a Authentication, Authorization and Accounting service (AAA) is included in the system. The service is responsible for keeping track of the authorized users of the system, and is manageable by the owner of the system. The service then uses common cryptographic methods in order to secure the service interfaces.
- The system should also be able to recover automatically in case of e.g. a temporary power outage (NFR-04). How the system addresses this requirement will not be addressed at depth, suffice to say, all relevant system settings and service parameters are stored in respective databases, and such the services have all the data they require to resume operation in case of a sudden system shutdown. In order to account for the users personal system preferences and such, a User data database is included. Regardless of what the name might imply, this service does not need to and will not store any personally identifiable information (NFR-05).

See the diagram below for a visual reiteration of the above. See the element catalogue further below for more details on the depicted services.


Internal
<<browser>> User UI
<<application>> Mobile User UI
<<browser>> Data Platform
<<service>> Authentication, Authorization and Accounting service
<<database>> User data DB
<<service>> User Interaction service
<<service>> Device control service
<<database>> Device State DB
<<service>> Data Analysis service
<<database>> Price data DB
<<service>> Price collection service
<<database>> Electricity usage DB
<<service>> Electricity Data Collection service
<<service>> Data transform service
<<external>> Electricity price system
<<external>> Electricity supplier
<<external>> Smart devices


Legend
System Element A
System Element B
A uses interface provided by B
System Element A
Multiple elements use interface provided by A

Element catalogue			
Name	Description	Technology	Why Technology?
User Data DB	Contains Past 6 months of User Consumption Data, Current mode of the application, Smart Devices connected to the application and their protocols, saved preferences/settings, names for devices/homes, etc.	Redis	Redis is suitable for lightweight databases and application caches as it is an in-memory data store. Redis can support different data types and structures, which makes it suitable for User Data DB which contains many different types of data.
User Interaction Service	Backend logic for user interaction	Node.js	Comes with support for both mobile and browser (React and React Native). Relatively light-weight and flexible
User UI	Frontend for the browser interface	React	Compatible with Node.js
Mobile User UI	Frontend for the mobile app interface. Besides sending requests to the User interaction service, this element receives notification requests from the service.	React Native	Compatible with Node.js
Price data DB	Contains current and past electricity prices	PostgreSQL	Used to handle structured data It is widely used and compatible with many third-party tools. Easy to integrate with the *Recommendation Engine*
Authentication, Authorization and Accounting service	Backend logic for authentication Used by *Data for consumers service* and *User Interaction Service*	Node.js Passport.js	Ensures consistency with *User Interaction Service.* Node.js libraries allow for integration with MSSQL servers Able to integrate an API with the authenticator, allowing *Data for consumer service* to use the service.
Price Collection Service	Collect price data from external electricity price systems	Node.js	Ensures consistency with the rest of the stack Able to handle API calls and HTTP requests
Data Transform Service	Process raw data from smart devices into structured data which can be used to train our models.	Python Pandas	Extensive libraries are available to perform complex data transformations. Python also allows the system to

Element catalogue			
	Processes commands from *Device control service* to the correct format to control Smart devices.		use existing libraries/APIs to interface with smart devices.
Electricity Data Collection Service	Collect and collate data from multiple sources to be fed into *Data Analysis service* and saved into the *Electricity usage DB.*	Python Pandas Numpy	Ensures consistency with the rest of the data pipeline. Extensive libraries are available to aggregate and manage data.
Data Analysis Service	Calculates settings recommendations based on prices and usage data. The algorithm is to be retrained periodically with Electricity usage data and Price data. Specifically, it will be updated with usage data every minute and with price data daily. Finds anomalies in collected data and raises alerts to *User Interaction service*	Python Numpy Pandas Scikit-learn	Extensive libraries are available for training and implementing ML models. Allows for efficient training and updating of the ML model.
Electricity usage DB	Use a time-series database optimized for handling continuous, timestamped data.	TimescaleDB	Allows for structured time-series data storage.
Device control Service	Controls smart devices using API	Node.js	Allows for usage of API. Ensures consistency with *User Interaction Service.*
Device state DB	Stores the current state of smart devices.	MongoDB	For the flexibility related to handling unstructured state relevant to each different smart device.
Data Platform	Serves as an interface for external data consumers to access *Electrical usage*	Node.js	Ensures consistency with the rest of the tech stack. Allows for handling multiple API requests

Deployment view

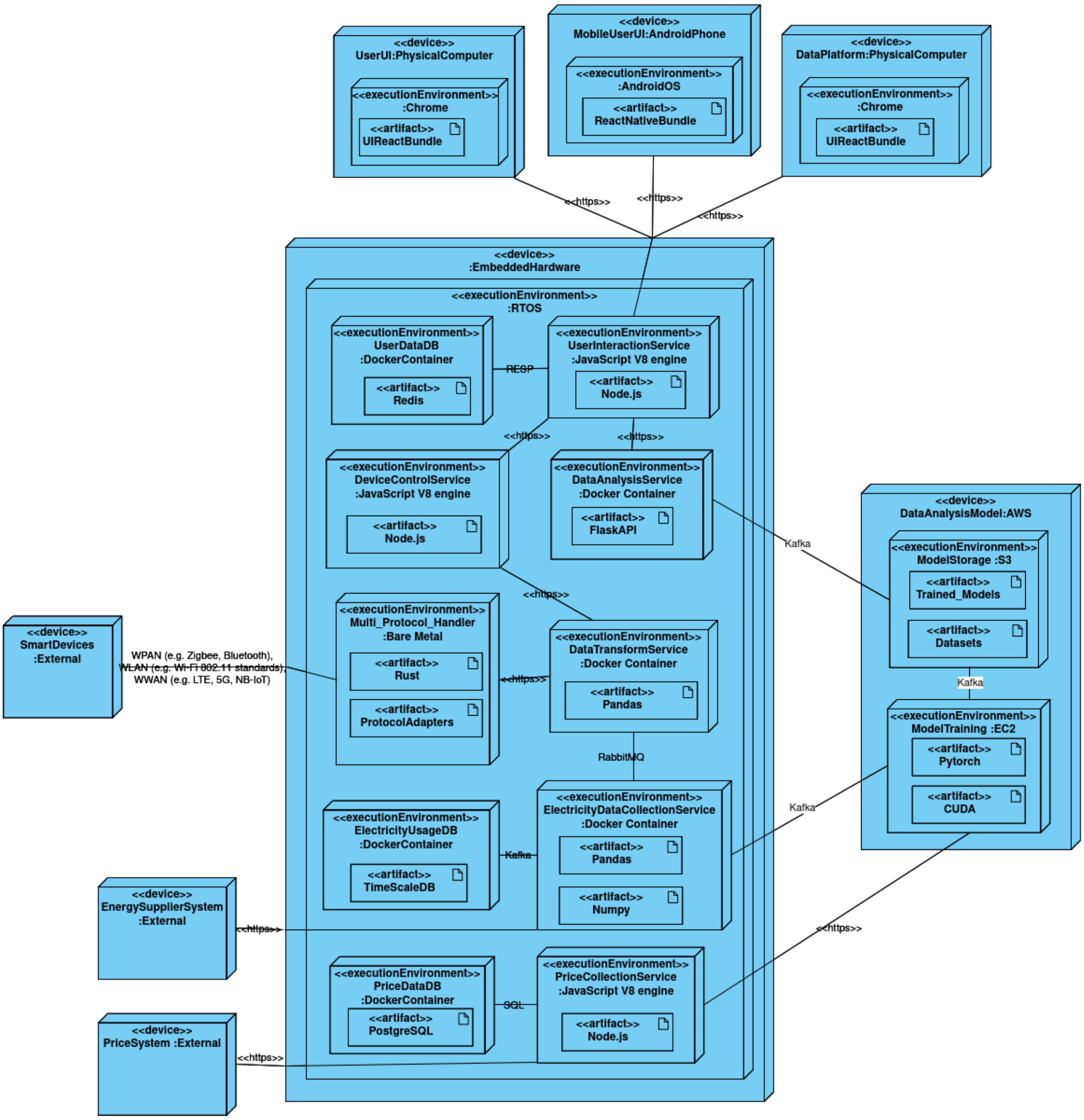
So far, the physical nature of the system has largely been ignored, and it is noted especially that the explanation for how the system interfaces with devices provided so far is unsatisfactory to many. For this reason, a deployment view is provided in this section.

A key consideration is interfacing with devices, which may use different communication protocols such as WLAN, Bluetooth, or other wireless technologies (NFR-03). Physical limitations, such as distance, affect the system's smart device interface. For example, supporting Bluetooth requires proximity. In the functional view, this interface is called the Data Transform Service. Given the physical constraints, it was decided that this service would be hosted on a physical computer running a standard Linux operating system, positioned within or near the household environment. This decision serves as the foundation for the rest of the deployment view.

To simplify the architecture, most of the system is hosted on the same device as the Data Transform Service. While data storage requirements are a concern, modern physical storage solutions should be sufficient. Local hosting also enhances user data privacy.

The exception is the Data Analysis Service, which is hosted in the cloud. This service processes and trains on large datasets, requiring significant computational power. Additionally, as it aggregates data from multiple sources, such as apartment complexes, cloud deployment is the most viable approach.

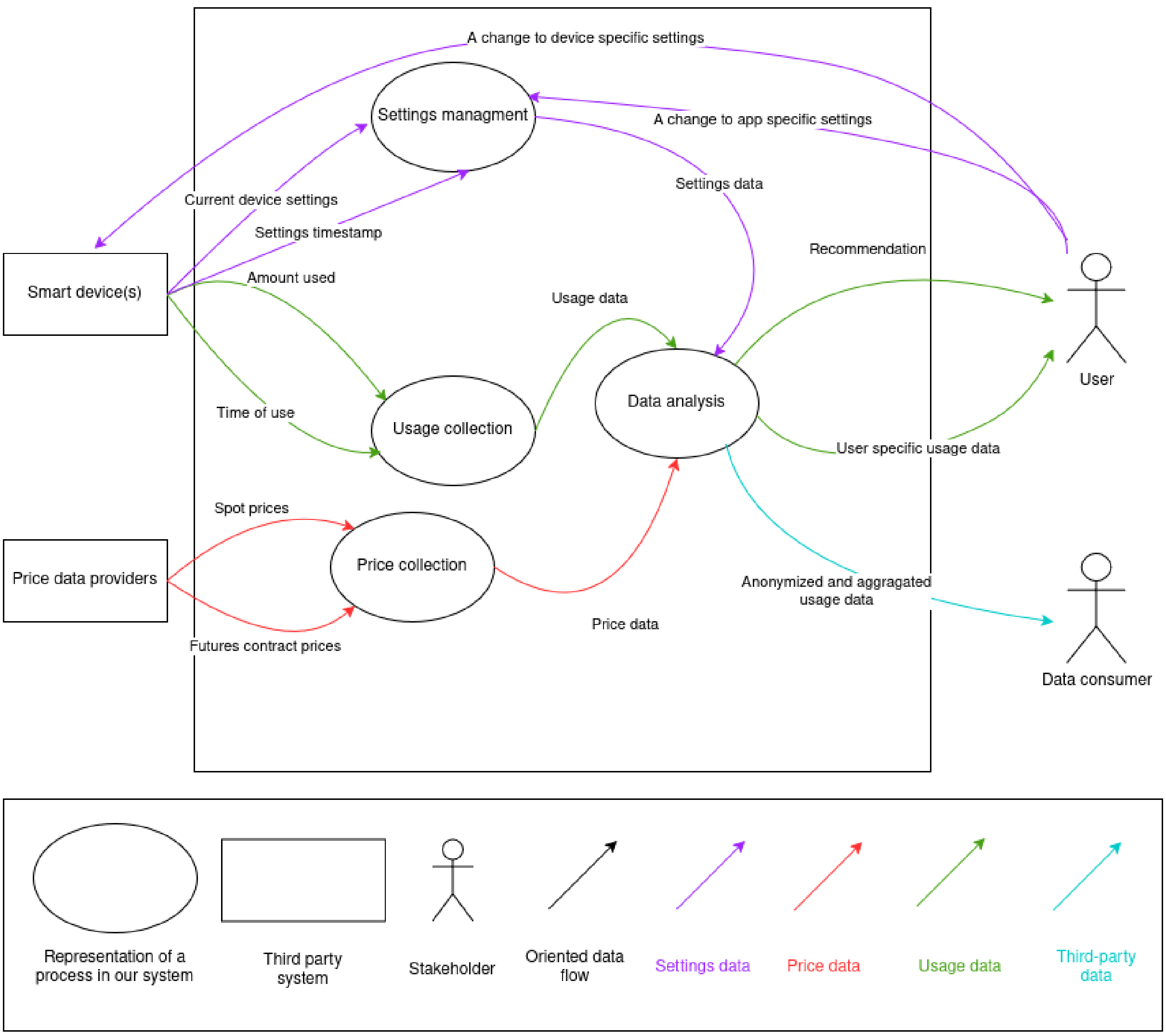
See the diagram below for details.



<<device>>
UserUI:PhysicalComputer
<<executionEnvironment>>
:Chrome
<<artifact>>
UIReactBundle
<<device>>
MobileUserUI:AndroidPhone
<<executionEnvironment>>
:AndroidOS
<<artifact>>
ReactNativeBundle
<<device>>
DataPlatform:PhysicalComputer
<<executionEnvironment>>
:Chrome
<<artifact>>
UIReactBundle
<<https>>
<<https>>
<<https>>
<<device>>
:EmbeddedHardware
<<executionEnvironment>>
:RTOS
<<executionEnvironment>>
UserDataDB
:DockerContainer
<<artifact>>
Redis
RESP
<<executionEnvironment>>
UserInteractionService
:JavaScript V8 engine
<<artifact>>
Node.js
<<https>>
<<https>>
<<executionEnvironment>>
DeviceControlService
:JavaScript V8 engine
<<artifact>>
Node.js
<<executionEnvironment>>
DataAnalysisService
:Docker Container
<<artifact>>
FlaskAPI
Kafka
<<device>>
DataAnalysisModel:AWS
<<executionEnvironment>>
ModelStorage :S3
<<artifact>>
Trained_Models
<<artifact>>
Datasets
Kafka
<<executionEnvironment>>
ModelTraining :EC2
<<artifact>>
Pytorch
<<artifact>>
CUDA
<<https>>
<<device>>
SmartDevices
:External
WPAN (e.g. Zigbee, Bluetooth),
WLAN (e.g. Wi-Fi 802.11 standards),
WWAN (e.g. LTE, 5G, NB-IoT)
<<executionEnvironment>>
Multi_Protocol_Handler
:Bare Metal
<<artifact>>
Rust
<<artifact>>
ProtocolAdapters
<<https>>
<<executionEnvironment>>
DataTransformService
:Docker Container
<<artifact>>
Pandas
RabbitMQ
<<executionEnvironment>>
ElectricityUsageDB
:DockerContainer
<<artifact>>
TimeScaleDB
Kafka
<<executionEnvironment>>
ElectricityDataCollectionService
:Docker Container
<<artifact>>
Pandas
<<artifact>>
Numpy
Kafka
<<device>>
EnergySupplierSystem
:External
<<https>>
<<https>>
<<executionEnvironment>>
PriceDataDB
:DockerContainer
<<artifact>>
PostgreSQL
SQL
<<executionEnvironment>>
PriceCollectionService
:JavaScript V8 engine
<<artifact>>
Node.js
<<device>>
PriceSystem :External
<<https>>

Information view

The information view describes how the system collects, processes, and uses information and data. The view illustrates the data flow within system processes, stakeholders, and third-party systems. This view clarifies data management and movement within the system, ensuring alignment with the relevant ASRs, namely NFR-03 (interoperability with devices), NFR-05 (protection of consumer data), FR-02 (controlling devices), and FR-01 (collecting consumption data).



The viewpoint clearly highlights how the system complies with regulations and privacy concerns. This is achieved through the anonymization and aggregation of data provided to “Data consumers”.

Access to different classes of data is restricted even internally within the system. The following table provides an overview of how modules from the functional view interact with different kinds of data.

Data	User DB	Electricity Usage DB	Price Data DB	Data Analysis Service	User	Data User	Price Collection Service	Electricity Data Collection Service	Data Transform Service
Settings data	Owner	None	None	None	Creator	None	None	None	Updater
Electricity Usage Data	None	Owner	None	Reader	Reader	Reader	None	Creator	Creator
User-specific Usage Data	Owner	None	None	Reader	Reader	None	None	Creator	None
Raw Price Data	None	None	None	None	None	None	Reader	None	None
Price Data	None	None	Owner	Reader	None	None	Creator	None	None
Recommendation	Owner	None	None	Creator	Reader	None	None	None	None

Below is a catalogue of the various types of information to be used and stored in the system.

Data	Description
Settings Data	Data regarding the smart device settings, such as its current state and the type of device.
Electricity Usage Data	Aggregated data regarding electricity consumed by smart devices of all users. This includes quantity and other metadata such as type of device and timestamps. The raw electricity data is aggregated at the collection services and then stored in the Electricity Usage DB.
User-specific Usage Data	Data regarding electricity consumed by the smart devices owned by the user. Collection services sort this data before forwarding it to the User DB. This data should only be stored for 3 months to avoid overloading the User DB.
Raw Price Data	This refers to electricity price data extracted from external APIs. Data consists of spot prices as well as futures contract prices. Data here is aggregated to Price Data, which is then used by the other system modules..
Price Data	Electricity price data for training and optimising the recommendation model. Data is stored in Price DB and is retrieved by Data Analysis Service when needed.
Recommendation	The output for the recommendation model. It consists of changes the system recommends to the user. This is generated by the Data Analysis Service using Price Data and Usage Data. This data is forwarded to User Interaction Service then stored in User DB for users to access.

Evaluation

To ensure that the system complies with the identified ASRs, it is evaluated through scenario walkthroughs in the following section. The scenarios are the ones defined in the Scenarios table. For brevity, only the highest priority scenarios are evaluated, namely, SCN-FR-02, SCN-FR-03, SCN-FR-04 and SCN-NFR-03, to analyze their respective requirements.

Evaluation of SCN-FR-02

FR-02 is concerned with abstracting complex controls into a simple user interface to reduce the knowledge requirement to use the application. SCN-FR-02 evaluates a specific function from this requirement, requiring that “when the system detects that electricity prices are high, the system commands connected electricity producers to dump electricity to the external grid.”

The spike in electricity prices will be detected by the Data Analysis service. The service will receive price data every day and will have an algorithm to perform trend analysis on the electricity prices to determine if the current prices are the “high point”. The service will send a command to the User Interaction service, which will send a notification to the user through the UI services and automatically send a command to the Device Control service.

The Device Control service will then execute its standard functions of changing the settings of the smart energy producer devices. This involves sending a command to the Data Transform service, which will translate the command to one that is understood by the smart devices. The smart device will then receive the instruction to switch its output to the external grid rather than consuming it locally.

One concern made apparent from this scenario is security. Specifically, the scenario involves the system automatically making changes to physical smart devices in the user’s system without the user’s explicit permission. Switching the flow of electricity may cause a denial of service in the internal devices in the network, something which has to be monitored by the system. This monitoring would be managed by the Device Control service, as it would have the most information as it is connected to the Device State DB.

Evaluation of SCN-FR-03

SCN-FR-03 states that “Every Sunday, the system drafts recommendations for when to use smart devices and when to sell electricity based on previous consumption and actual/predicted future electricity prices”.

To begin our analysis, we will assume that the user has past consumption data stored in the system and the user has requested settings recommendations from the system.

The Functional view shows that all interactions between the user and the system pass through the User Interaction service. When the user requests recommendations from the system, the User Interaction service will send a request to the Data Analysis service. The Data Analysis service then requests the required data from the Price Collection and Electricity Data Collection services, respectively.

The Price Collection service retrieves electricity prices from the past week using external APIs, then processes the data for training if needed. The Price Collection service will then store this processed data in Price DB. It will then extract all needed data from Price DB before forwarding the data to the Data Analysis service for training.

The Electricity Data Collection service will request data from the Data Transform service, which will, in turn, query all Smart devices in the network for the latest consumption data. The Electricity Data Collection service will also query any energy supplier systems connected to the network for production data. This data is stored in the Electricity Usage DB. The Electricity Data Collection service then extracts all required data from the database before forwarding it to the Data Analysis service.

The storage process is done so the data pre-processing does not need to be repeated every week, as all data from previous weeks have already been processed. This ensures that the process is reasonably efficient.

Thus, the Data Analysis service has access to relevant data and can perform statistical analysis to provide predictions.

Although the walkthrough above is conceptually simple, there are some concerns. As stated by NFR-07, the recommended settings should save the users money overall. However, this quality can only be examined once the true data has been collected.

Another concern stems from the availability of data. Creating a properly general model of electricity consumption, production, and price requires a vast amount of data, which would not be available at the start of deployment. There is also the more general concern of whether or not the electricity consumption, production, and price data is sufficient for accurate predictions, let alone relevant to the predictions at all (i.e. there might be other influential data that we are unaware of).

Naturally, this will lead to the recommendation algorithm failing the quality requirements defined in NFR-07 at the start of deployment or when significant changes are made to the system, as the algorithm would lack sufficient data to perform well.

There is also a concern with data storage. The vast amounts of data required may overload the physical systems depicted in the deployment view, which might make the implementation cumbersome.

Additionally, the implementation of the Data Transform service would require constant monitoring of smart device development to keep the service compatible. Naturally, standardisation drives should be monitored.

Evaluation of SCN-FR-04

The focus of FR-04 is on the system's ability to detect anomalies and inform the user accordingly. SCN-FR-04 requires the system to notify the user when a smart device has used/produced 10% more/less electricity in the past minute than the average of the last five minutes.

This anomaly detection should be done quickly to minimize any possible damage to the system.

Anomalies are detected and raised by the Data Analysis service. Data is forwarded from the Electricity Data collection every minute, and the model in the Data Analysis service is constantly retrained and calibrated to detect anomalies.

When an anomaly is detected, the Data Analysis service will forward the detection to the User Interaction service. The User Interaction service then sends the detection as a notification to the user either through User UI or Mobile User UI.

This sequence of events is simple to implement and should be deployable without significant risks. However, it is difficult to determine the data needed to develop models to detect anomalies, and it is even more challenging to determine the nature of the anomaly that will be detected.

The Data Transform service processes data from the smart devices for the Device Control service, which forwards the data to the Data Analysis service. There are two main concerns related to the services.

The first concern is the pre-processing of data. As the Data Transform service is expected to work with all the Smart Devices, which operate using different technologies and protocols, it is entirely possible that some data is processed incorrectly. Such a mistake may trigger an anomaly detection in the system. These detections are not actionable from the user's perspective and should be separated from more typical anomalies.

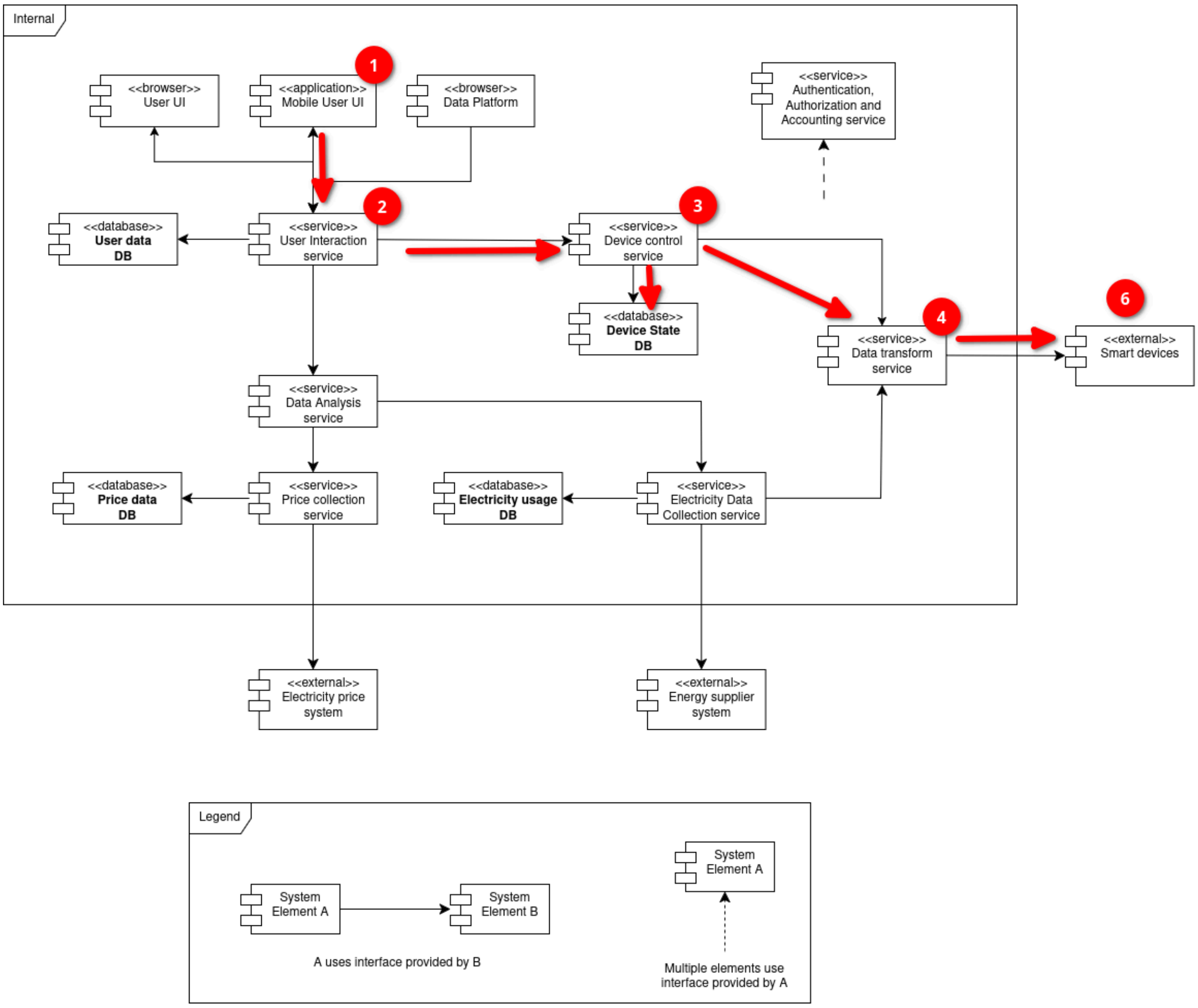
The second concern is smart device malfunction. These incidents are critical and should be sent as alerts to the user for immediate action.

The different urgencies and types of anomalies hints at the need for the anomaly detection algorithm to be able to differentiate different types of anomalies, with different actions being taken for each type of anomaly, However, such anomalies can only be detected and implemented with actual data and is beyond the scope of the system architecture.

Evaluation of SCN-NFR-03

The NFR-3 is concerned with interoperability and states. The scenario to evaluate NFR-3 is defined as *"When a device connects to the same WLAN that the system is connected to, the system acknowledges the new device and offers to the user to start managing it"*

For the performed walkthrough, we will assume that the user is using the mobile platform, however, the scenario is identical to if the user was using the browser interface instead. The stimulus in this walkthrough starts using a Mobile User UI and, for clarity described as well using an annotated functional view.



As mentioned, the stimulus comes from a user, in this case, a Mobile User UI. The request is forwarded by User Interaction service to a separate Device control service, which processes the request and, if needed, uses a Device State DB to find out the current state or setting of the device. To then apply the changes, it has to use the correct technology and protocol to communicate with the device, which is the interoperability described by NFR-3. For those purposes, there is a Data transform service which is used to communicate with respective devices. This service transforms the data into a request supported by said smart device and forwards it. Lastly, there is a confirmation if supported by the device and the new state is stored in the database. In case of any problems, the request should be stopped and an error message shown to the user.

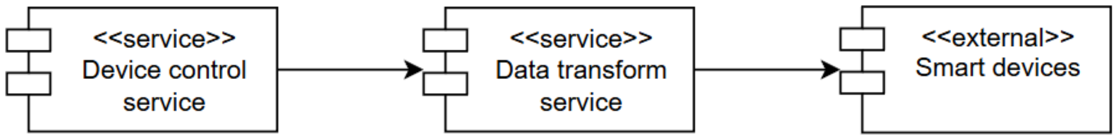
Prototype and Minimum Viable Product

This section presents how prototyping could be utilised to give more confidence in the system. Also, a *minimum viable product* (MVP) is proposed.

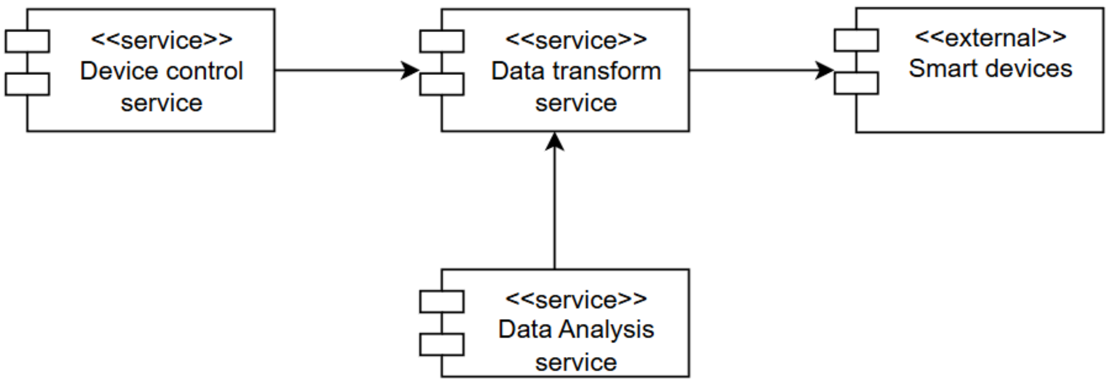
It is clear from constraint EXT-04 and most of the functional requirements that the ability of the system to interface with external devices is of big concern. This is due to the aforementioned lack in standardisation in how and what data smart devices provide to users. In particular, EXT-05 , together with NFR-03 and NFR-06 states that the system should be able to comfortably grow to support more different interfaces as time goes by. Since this is the main purpose of the data transform service in the functional view, To alleviate these concerns, it is proposed that a prototype of said service is developed. Development should begin with implementing support for a couple of different devices to gain a better understanding of how to interface with devices in general and so as not to overwhelm the development team. Success in aggregating data from and providing common controls to a couple of different devices would give the development team much-needed trust in project success. Approaches to increasing support for different devices could also be explored and evaluated. Naturally, the prototype service could be expanded upon to finally be usable in the complete system.

For much the same reasons as above, how the system could be implemented to control the connected devices, whether automatically or through providing a centralized interface to the user, is of great concern. Therefore, it is suggested that a prototype of the device control service be developed in conjunction with the data transform service. From the perspective of the device control service, it is not too big of a difference whether its commands come from the user interaction service or if it works by its own accord. The most important aspect of the service is that it can implement the recommended usage patterns provided by the data analysis service through the user interaction service. Since the recommended usage patterns are ephemeral and finally dependent on the usage context, they should be mocked. The mock data could then be used to develop tests for the device control service. Notably, since this mock data would not need to be stored, the various data collection and storage services would not be needed in the prototype.

In summary, it is suggested that a first slice prototype consisting of the device control service and the data transform service be developed. This prototype would be developed to interface with a select few external devices. Ignoring the multitude of other existing devices, it could be used to prove adherence to FR-01, FR-02, and FR-03 early. Additionally, the prototype would be used to investigate how to develop an efficient platform for providing interoperability to a multitude of devices in the future, alleviating some of the biggest implementational concerns. Also, what controls the device control service provides would be determined. Slice prototype 1 is depicted below. The diagram uses the same notation as the functional view:



After implementing slice prototype 1, it could be expanded upon relatively easily to also implement FR-04 by introducing the data analysis service into the slice. Again, the data analysis service would be tested on mock data. Slice prototype 2, addressing functional requirements FR-01, FR-02, FR-03, and FR-04, could be considered the minimum viable product, and is depicted below. Note that the electricity data collection service is not included in the slice, since for small datasets, it could reasonably be included in the data analysis service.



Appendix A - Architectural Decision Log

<template>	In the context of <ASR a>, facing concern <concern c> we decided for <option o> to achieve <quality q>, (optional) accepting <downside d>. (optional)
ADR-01	In the context of NFR-1, facing the concern that the system should be secure, we decided to include an authentication service in the system, to ensure that the system cannot be compromised by adversaries.
ADR-02	In the context of NFR-2, Facing the concern that the system should achieve usability, We decided to employ the client-server pattern, To allow us to easily create multiple different interfaces with the same functionality.
ADR-03	In the context of NFR-3, facing the concern that the system should be compatible with many different smart devices, We decided to introduce an element responsible for providing a common interface for different smart devices, i.e. the DataTransformService, To achieve compatibility.
ADR-04	In the context of NFR-4, Facing the concern that a database failure will cause a denial of service, We decided that all central databases should be replicated and updated alongside the original databases, To achieve enhanced reliability, Accepting the increase in required infrastructure and network load.
ADR-05	In the context of NFR-6, Facing the concern that IEC 62351 mandates that power system control applications have immediate detection of cyber attacks and are secure by design[3], We decided to create a separate class for purposes of authentication and authorisation so, To achieve flexibility in how we enforce security in the system recognising the dynamic nature of cyber attacks to achieve regulatory compliance with IEC 62351.
ADR-06	In the context of NFR-7, Facing the concern that the consumption predictions of the system should not achieve more than 10% difference, We decided to create a single coherent element,

	DeviceControlService. To achieve less deviations.
ADR-08	In the context of NFR-8, Facing the concern that the prediction model needs to accurately predict electricity price on top of electricity consumption, We decided to create services to monitor and store electricity price data (both current and historical).
ADR-09	In the context of FR-01, facing the concern that multiple queries to a central database will cause lag or denial of service, We decided to store electricity consumption data from the past 6 months for each user in the application cache, To remove the potential bottleneck of queries at the central electrical usage database, accepting that this will increase the complexity of data management and increase the attack surface for potential cyber attacks.
ADR-10	In the context of FR-03, facing the concern that an accurate ML model will require a vast quantity and diversity of data, We decided that the energy consumption data of all users should be forwarded to and stored in one central ElectricalConsumptionDB.
ADR-11	In the context of FR-04, facing the concern that detecting anomalies in data is not a computationally trivial task, We decided to create an ML model service to create a baseline of user consumption for anomaly detection, To achieve accuracy in detecting potential electrical faults, Accepting that this will increase the computational load of the system.
ADR-12	In the context of FR-02, facing the concern that the application needs to be able to remotely control the smart devices, We decided to create a single element to interface with the smart devices, i.e. DeviceControlService, To achieve modularity and extensibility.
ADR-13	In the context of NFR-05, facing the concern that data consumers and other electricity consumers might act in bad faith and attempt to steal personal data from electrical consumers, We decided that access to the central ElectricalUsageDB should be managed by an API and no system should access it without the API.
ADR-14	In the context of NFR-05, facing the concern that data consumers and other electricity consumers might act in bad faith and attempt to steal personal data from electrical consumers, We decided that data consumers should be accessing a separate database from the internal system, and should only be accessing

	aggregated usage data.
ADR-15	In the context of NFR-03, facing concern interoperability, we decided to host the DataTransformService on a physical device INSTEAD OF in the cloud, so that the system is able to interface with smart devices using distance-limited interfaces such as physical cables and short-distance wireless such as bluetooth.
ADR-16	In the context of FR-02, facing the concern that there may be problems with information desynchronization when it comes to settings, We decided to use the smart devices as the main reference when considering device specific settings to ensure that our settings data matches what is actually being used.
ADR-17	In the context of the technical constraints, facing concern physical hardware limitations, We decided to put all parts of the system, other than parts related to the DataAnalysisService, in the same device node, to achieve a minimum number of physical hardware.
ADR-18	In the context of the NFR-05, facing data anonymity, We decided to choose embedded hardware as the whole system's device node
ADR-19	In the context of the ASR-1, facing concern big data analysis limitations, we decided to host the Data Analysis Service in AWS, to achieve the necessary performance and reliability.
ADR-20	In the context of the technical constraints, facing concern infrastructure choices, We decided to chose AWS S3 and EC2 as the execution environments of ModelStorage and ModelTraining, respectively, to achieve reliability due to the common usage
ADR-23	In the context of the technical constraints, facing concern of infrastructure choices, we decided to use Kafka protocol for the use-cases that work with Big Data streaming, to achieve performance
ADR-24	In the context of the technical constraints, facing concern of infrastructure choices, we decided to use RabbitMQ protocol for the use-cases that contain fast and reliable message queuing purposes, e.g. between DataTransformService and ElectricityDataCollectionService, to achieve performance

ADR-25	In the context of the NFR-05 and NFR-06, facing data anonymity and regulation compliance, We decided all data provided to Data Consumers is anonymized properly in the DataAnalysisService.
ADR-26	In the context of FR-01, FR-02, FR-03, It was decided to make the interaction between the system and the smart devices and the system and the electricity suppliers one way (system to external device), since otherwise there could arise concurrency issues between the three.
ADR-27	In the context of FR-02 and NFR-04, It was decided to introduce a Device state database in the functional view, In order to be able to save relevant device settings
ADR-28	In the context of FR-04, It was decided to allow make the interaction between the user interaction service and the data analysis service two-way, Such that the user interaction service would not have to regularly poll the data analysis service for detected anomalies.
ADR-29	In the context of NFR-03, In the functional view, it was decided to move the Energy supplier system external to interface with the data transform service, Such that it can provide abstract interfaces to those types of devices too.

Appendix B - Changelog

Id	Description	Date & Time
CHG-1	Removed open-ended architectural principles that were too general and thus did not imply any architectural decisions	26 Feb 2025
CHG-2	Combined multiple ASRs that were describing similar features/qualities	12 Mar 2025
CHG-3	Revised ASRs to reduce ambiguity and increase testability	12 Mar 2025
CHG-4	Created User Stories to combine functional and non-functional requirements into concrete goals	12 Mar 2025
CHG-5	Adopted new architectural principles to account for business goals Added Deployment view to the architectural design	12 Mar 2025
CHG-6	Added Information view	12 Mar 2025
CHG-7	Added Information Ownership Table	12 Mar 2025
CHG-8	Changed ADR 14 to requiring the separation of user-specific usage data	12 Mar 2025
CHG-9	Updated context view to explicitly specify external systems	12 Mar 2025
CHG-10	Enlarged appendix C to make it more readable	12 Mar 2025
CHG-11	Added NFR-03 and NFR-07 to the requirements addressed by ASR-01	24 Mar 2025 14:00
CHG-12	Changed NFR-07 to address accuracy of predictions in general, not just electricity consumption	24 Mar 2025 14:30
CHG-13	Start tracking owner and timestamp in changelog, add id, update table style	24 Mar 2025 18:05
CHG-14	Added short description of what the system physically is in the introduction	24 Mar 2025 18:06

Id	Tт Description	Date & Time
CHG-15	Slightly expanded description of Smart appliance manufacturers	24 Mar 2025 19:13
CHG-16	Rename “Key stakeholders” section to “Stakeholder analysis” and add metatext to the section	24 Mar 2025 19:20
CHG-17	Split old stakeholders tables into one table identifying and describing the stakeholders and give them id’s . . .	24 Mar 2025 19:30
CHG-18	. . . and another table listing needs and concerns. Also, clarify distinction between needs and concerns.	25 Mar 2025 15:42
CHG-19	Updated Architecturally Significant Requirements section metatext	25 Mar 2025 16:26
CHG-20	Updated information view description	25 Mar 2025 17:23
CHG-21	Added colored data flows to information view diagram	25 Mar 2025 17:29
CHG-22	Trimmed needs and concerns that didn’t have a tangible connection to the use of the system	25 Mar 2025 18:17
CHG-23	Reworded and made user needs and concerns more general and granular	25 Mar 2025 18:29
CHG-24	Rewrote section on external constraints, added more metadata to constraints	25 Mar 2025 19:25
CHG-25	Changed FR-05 to “Anonymized electricity data collected by the system should be accessible to data consumers and updated regularly”	25 Mar 2025 19:39
CHG-26	Removed NFR-08 due to being implied by NFR-07	25 Mar 2025 20:05
CHG-27	Updated the context view diagram, clarified interactions and their directions	25 Mar 2025 20:58
CHG-28	Rewrote context view section to clearly identify what the external	26 Mar 2025 09:45

Id	Tт Description	Date & Time
	systems/stakeholders are, why they are considered, and what their main interactions with the system are	
CHG-29	Rewrote the functional view section to resolve problems received from the feedback.	Mar 26, 2025 11:15 AM
CHG-30	Added section on prototyping and minimum viable product	26 Mar 2025 12:33
CHG-31	Removed FR-05, since it was largely accounted for by NFR-05. Removed quality column from NFR table. Created a scenario for almost all ASRs.	8 Apr 2025 13:26
CHG-32	Rewrote prioritization section metatext. Made definitions of importance and risk a bit more explicit.	8 Apr 2025 16:42
CHG-33	Rewrote context view to show more explicitly how ASRs are addressed	8 Apr 2025 17:27
CHG-34	Rewrote the evaluation section to match the newly defined scenarios	9 Apr 2025 00:00
CHG-35	Rewrote functional view to more explicitly address all ASRs	9 Apr 2025 12:23
CHG-36	Changed title to “Smart home system”	9 Apr 2025 13:05
CHG-37	Removed old appendix C since it was outdated	9 Apr 2025 13:22
CHG-38	Removed bibliography section, since it was outdated	9 Apr 2025 13:24