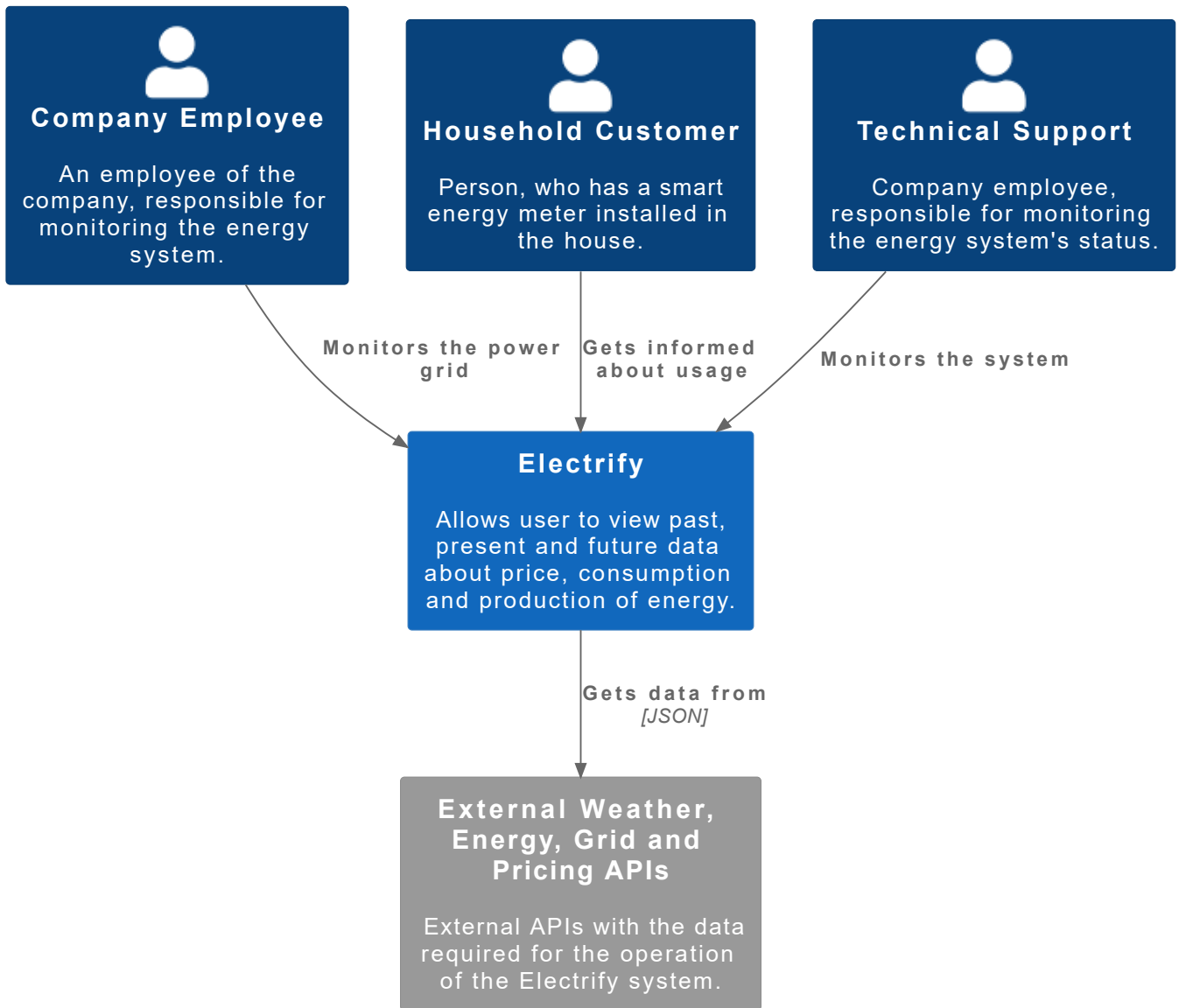


Electrify-Architecture

Overview



Legend	
person	person
system	system
external person	external person
external system	external system

Level 1: System Context diagram

Scope: A single software system.

Primary elements: The software system in scope. Supporting elements: People (e.g. users, actors, roles, or personas) and software systems (external dependencies) that are directly connected to the software system in scope.

Intended audience: Everybody, both technical and non-technical people, inside and outside of the software development team.

Description

A system context diagram is the highest-level diagram, and it shows the scope of the software system and its intended audience. It also shows the relationship between the main systems and the external APIs that will be used for the project.

Target User: Company Employee: The company employees will use *Electrify* to monitor the production, distribution and usage of power.

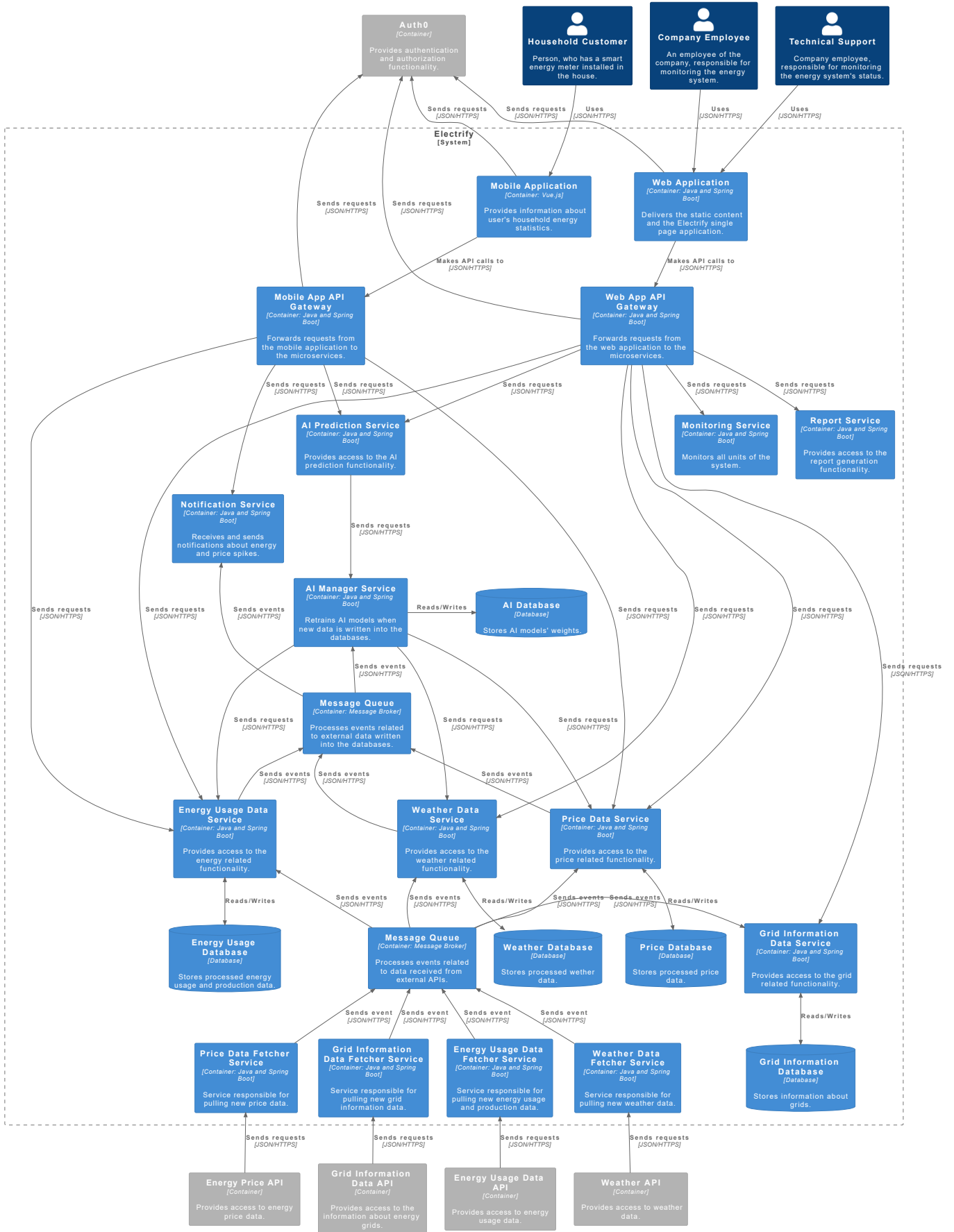
Target User: Household Customer: The household customer will use *Electrify* to get acquainted with their power usage in their household.

Target User: Technical Support: The Technical Supports will use *Electrify* system to oversee that everything is working properly.

System: Electrify: The System that will be developed to serve the needs of the intended audience.

External System: External APIs: The External APIs will be used to fill *Electrify* with data about energy prices, the weather, the grids and energy data.

Electrify System



Level 2: Container diagram

Scope: Electrify system.

Primary elements: Containers within the Electrify system. Supporting elements: People and software systems directly connected to the containers.

Intended audience: Technical people inside and outside of the software development team; including software architects, developers and operations/support staff.

Notes: This diagram says nothing about deployment scenarios, clustering, replication, failover, etc.

Auth0

External security service which provides authentication and authorization functionality to the system. Additionally, users and their permissions are also managed by this service.

Mobile Application

Progressive web application (PWA) which provides users with various energy-related information based on the monitoring sensor located in their house.

Web Application

Single-page web application which gives companies the energy-related information of south part of the Netherlands.

Mobile App API Gateway

Service responsible for routing requests from mobile application to other microservices.

Web App API Gateway

Service responsible for routing requests from web application to other microservices.

Notification Service

Service, related to the mobile application. The main responsibility is sending alerts about energy consumption or prices spikes. Service is autonomous.

AI Manager Service

Autonomous service responsible for training new AI models once in a while and interacting with them.

AI Database

Database which stores weights of the AI models or models itself which later can be used to quickly reinitialize model in the application and start interacting with it.

AI Prediction Service

Service responsible for getting predictions for various data types. To get the predictions, it interacts with the AI Manager Service.

Report Service

Service which main goal is to generate PDF reports based on the input parameters.

Monitoring service

Service responsible for monitoring the stability of the system. Monitoring is done for all internal components of the system.

Message Queues

RabbitMQ message broker which enables asynchronous behavior in the system.

Price/Energy Usage/Weather/Grid Information Data Service

Services responsible for the CRUD functionality and preprocessing of the respective data.

Price/Energy Usage/Weather/Grid Information Data Fetchers

Autonomous services responsible for pulling data from the external APIs once in a certain timeframe. Pulled data is then transferred asynchronously via message queues.