

Chapter 1

System Architecture

1.1 Introduction

In the ?? and ??, the logical layering and the responsibilities of the components were introduced but how they fit together and work in harmony is still a mystery. In this chapter, I will put the pieces of the puzzle together by discussing of the actual architecture of the system and its behaviours with increasing levels of detail.

1.2 Top-level architecture

The top-level architecture is a simplified architecture where many low-level components are being grouped together and treated as a singular composite component. It is shown in the figure 1.1. The following components in the top-level architecture are composite component:

- Remote Sensing (See section 1.3)
- Backend (See section 1.4)
- Frontend (See section 1.5)
- Real-time (See section 1.6)
- Governor (See section 1.7)

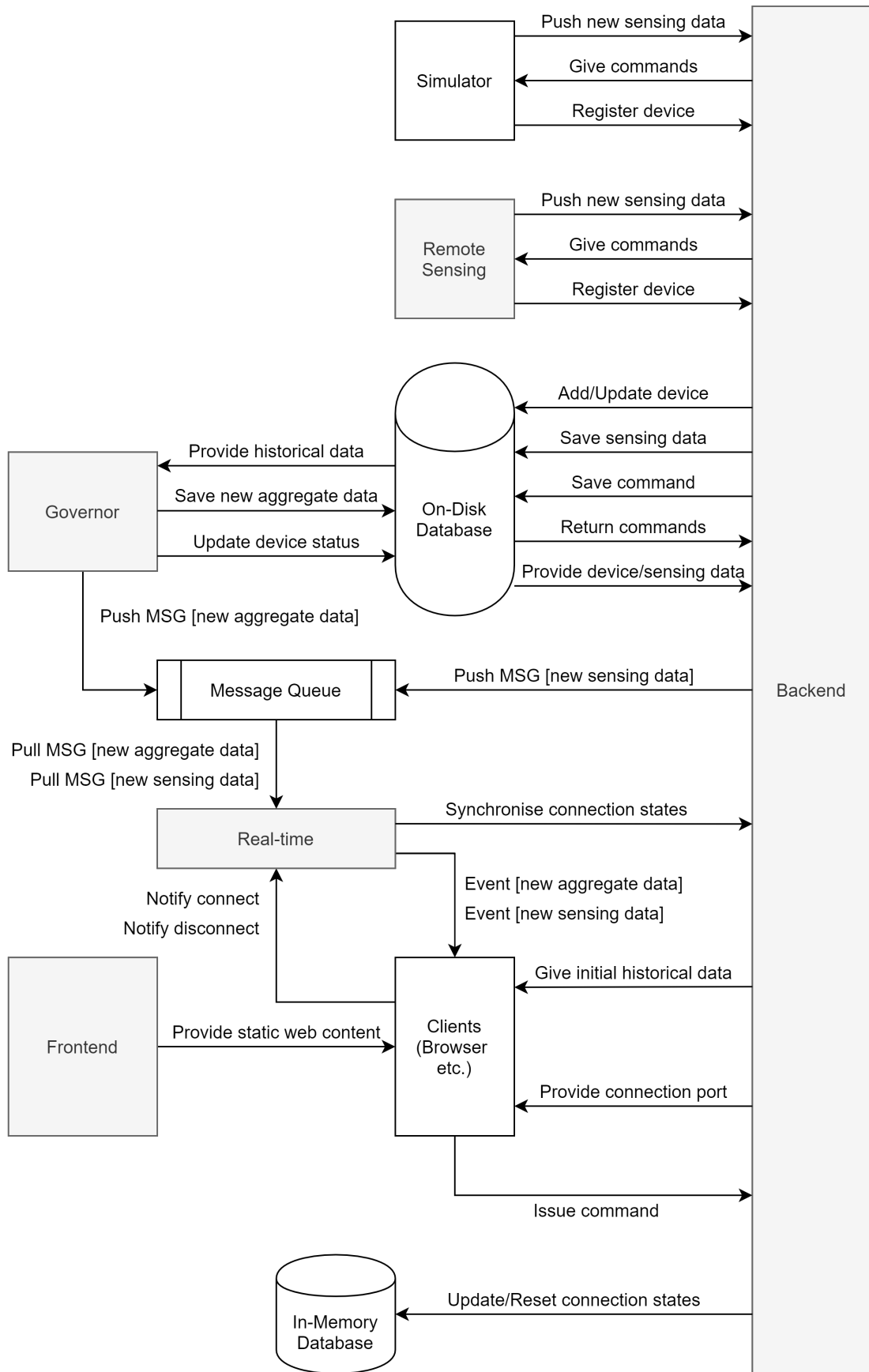


Figure 1.1: Top-level architecture of the proposed system.

1.2.1 Initialisation

The initialisation process is referring to a device in the remote sensing composite component such as a microcontroller attempting to connect with the server before it can send data and receive commands. When a microcontroller is initialising, it sends a register device request to the backend and then wait for the reply from the backend. Once the request is received by the backend, it tries to find a matching device in the database. If found, then it updates the device status to online. Otherwise, it generates a new device ID, creates a representation of the new device in the database, and set the device status to online. Finally, the backend notifies the microcontroller by returning a response to the initial device registration request, and the intialisation sequence is complete. The figure 1.2 shows the flow of the initialisation process and the relevant function calls shown in the top-level architecture.

The initialisation process achieves two goals, the first goal is letting the backend knows the device is online and ready to transmit data, and the second goal is getting a unique ID so that it can be identified during the data transmission phase. Once the initialisation process is finished, the microcontroller is ready to transmit data.

1.2.2 Data recording and device controlling

The data recording process aims to record data from the sensors and store them into the database. The process begins at the microcontroller where it reads the sensing data from the sensors and then sends it to the backend using the *push new sensing data* request. Once the backend received the sensing data, it begins to execte two sequence of instructions simultaneously. The first sequence is used for data recording, where the data is added to the database and then sends a message to clients notifying a new sensing data is available. The second sequence is used for device controlling, where it retrieves a list of commands cumulated in the database, and sends them to the microcontroller through the response to the *push new sensing data* request. As soon as the microcontroller received the commands, it would perform each command in order and wait for 1 second before reading and sending data again. The process is shown in the figure 1.3.

The reason why data recording and device controlling happen at the same time is improving communication efficiency. The microcontroller doesn't have to send two different requests to push data and retrieve commands, which reduce the communication time by half and reduces the server load by reducing the number of access to the backend by half.

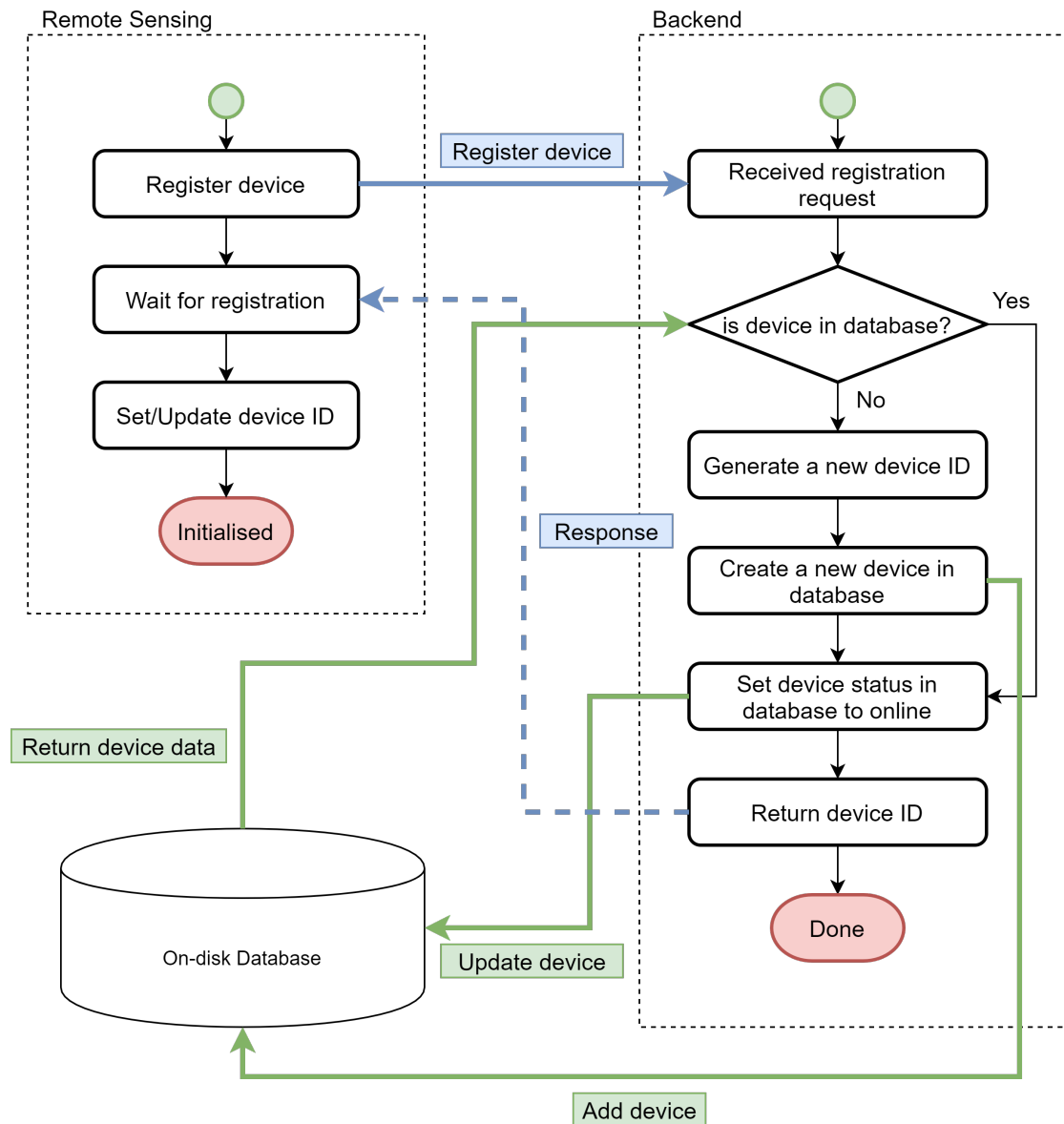


Figure 1.2: The initialisation process.

1.2.3 Governing

1.2.4 Context-aware load balancing

1.2.5 Data monitoring

1.2.6 Issuing command

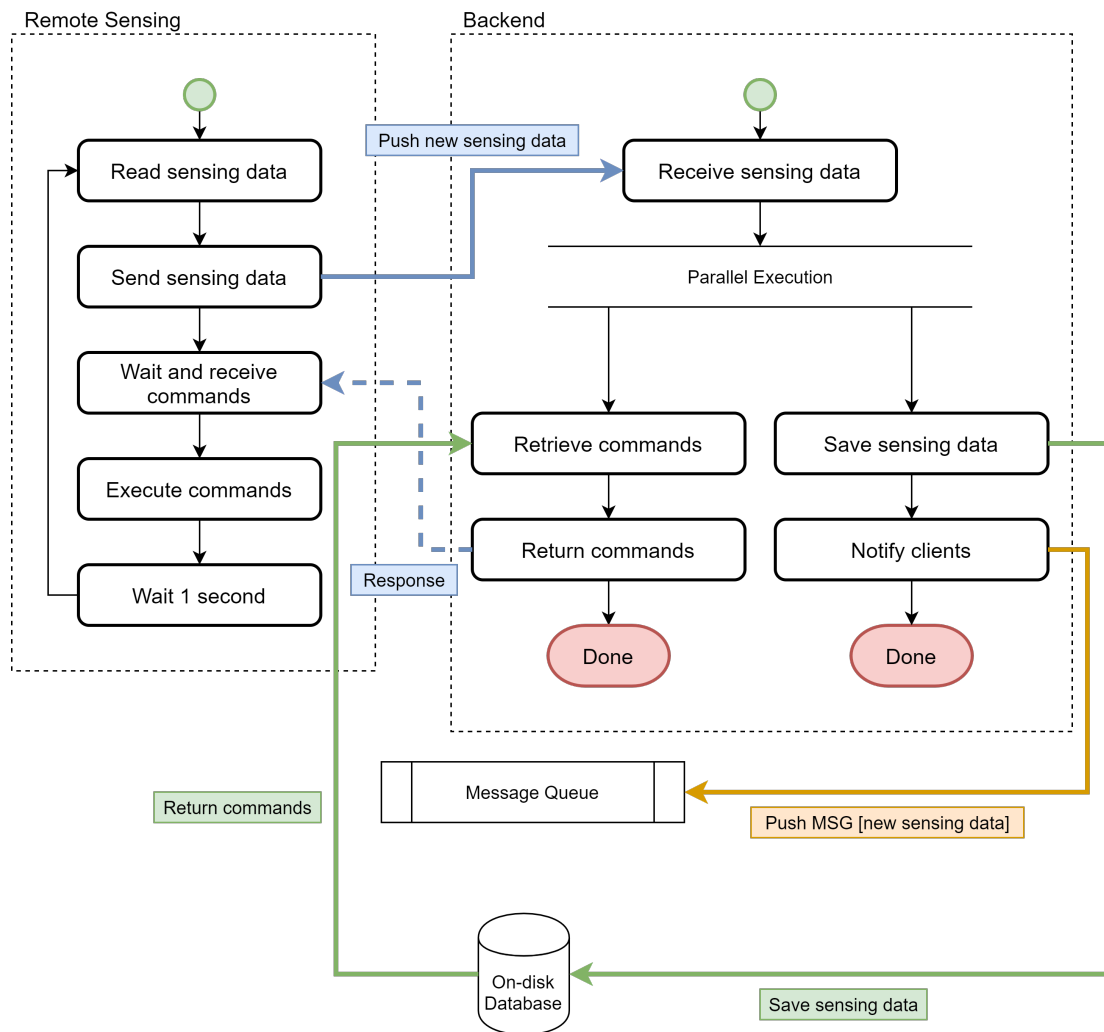


Figure 1.3: The data recording process.

1.3 Remote sensing

sdfsdf

1.4 Backend

sdfsdf

1.5 Frontend

sdfsdf

1.6 Real-time

sdfsdf

1.7 Governor

sdfsdf

1.8 Simulator

sdfsdf

1.9 Client