

Abstract

The Quantified Student (QS for short) project focuses on helping students with their development and optimising their performance with the help of collected data. The collected data will be shown in a dashboard where the student can see it. After which, the student can conclude where and how to improve their workflow. For example, the system can show when it is the best time to work for the student.

Table of Contents

[1 Introduction](#_gjdgxs) **5**

[2 Model](#_30j0zll) **6**

[2.1 Context](#_1fob9te) 6

[2.2 Containers](#_39g3v58i2oiw) 7

[**Conclusion**](#_jlm3rhbk2ww1) **8**

Table of Figures

[Figure 1 C1 - Context diagram 6](#_3znysh7)

[Figure 2 C2 - Containers diagram 7](#_tyjcwt)

Revision History

| **Revision** | **Date** | **Author(s)** | **Description** |
| --- | --- | --- | --- |
| 0.1 | April 26, 2022 | J. Maas | Initial document setup and write conclusion |
| 0.2 | May 10, 2022 | J. Maas | Write introduction and elaboration for the C4 model |
| 0.3 | May 23, 2022 | J. Maas | Update context and container diagrams |
| 0.4 | May 30, 2022 | J. Maas | Update context diagram to show extendibility of data sources |
| 0.5 | June 9, 2022 | J. Maas | Update diagram |

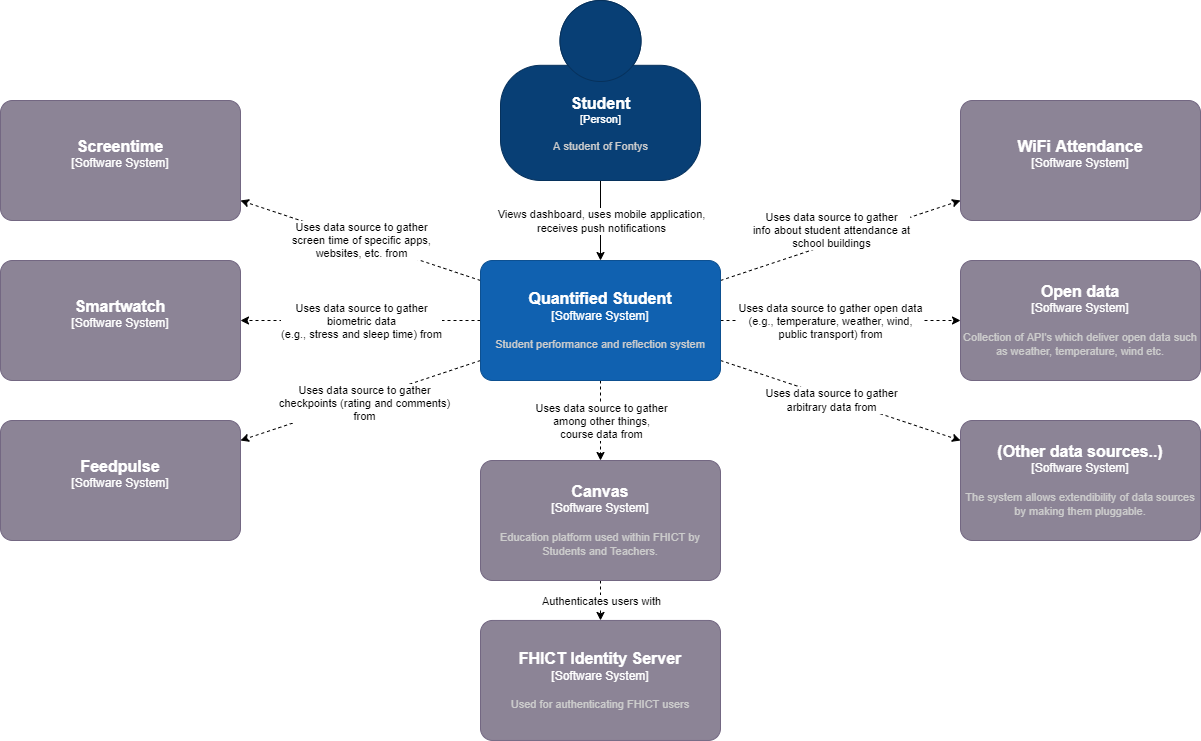
# 1 Introduction

This document contains and elaborates upon the first two levels of the software architecture for the Quantified Student (also referred to as QS) software system. The C4 model outlines the required containers and components for the QS system. Note that this model will always be subject to change when the parameters of the project change over time.

# 2 Model

The C4 model is an "abstraction-first" approach to diagramming software architecture, based upon abstractions that reflect how software architects and developers think about and build software. The small set of abstractions and diagram types makes the C4 model easy to learn and use. For more information, please refer to [https://c4model.com/.](https://c4model.com/)

## 2.1 Context



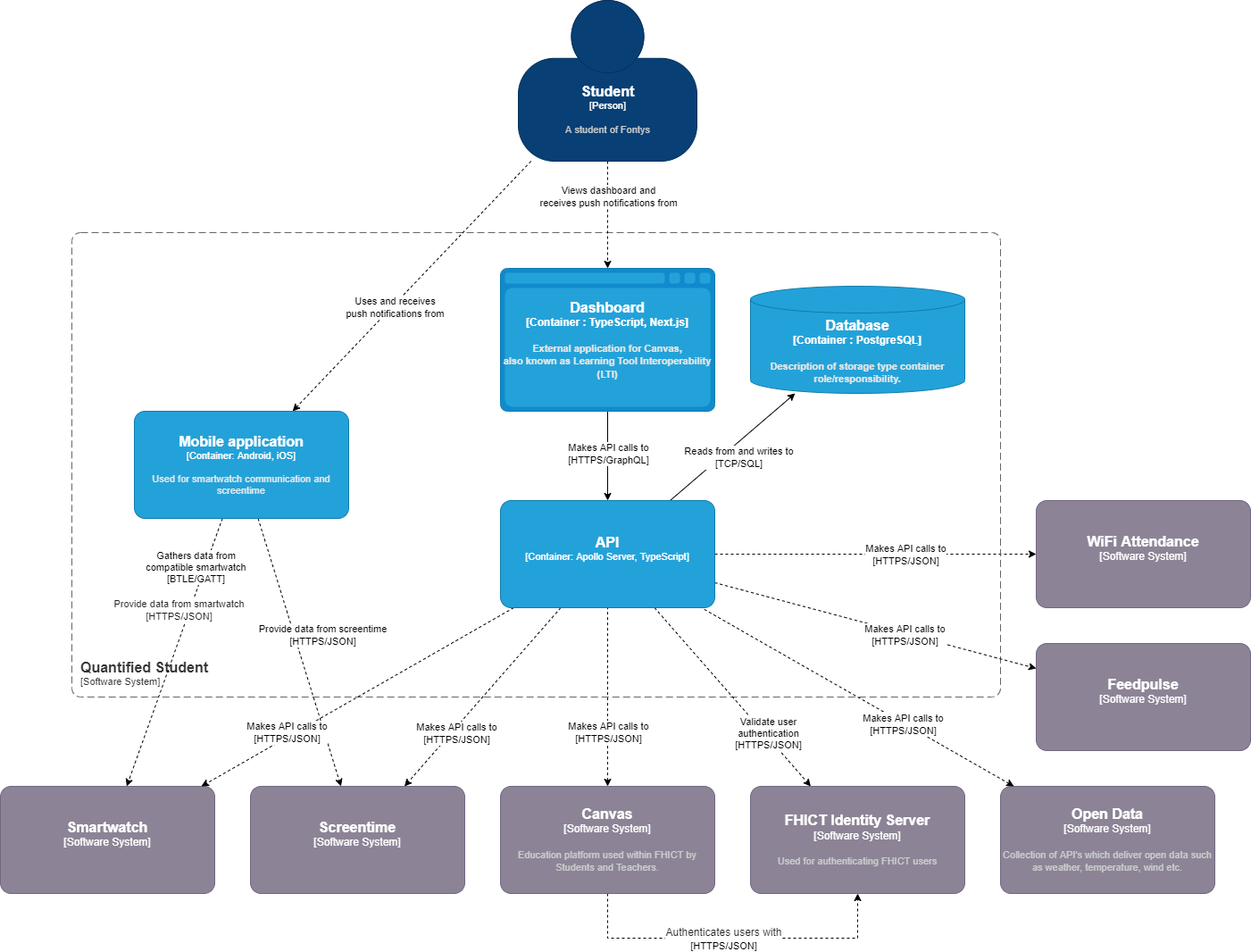
*Figure 1 C1 - Context diagram*

The Quantified Student software system has one main role, the student, who is and must remain the sole user of the platform. This means that the student has full ownership of their data and no one else can access it without permission, especially not teachers. The Quantified Student software system can interact with many different data sources, and aren’t limited to the ones shown above.

| **Data source** | **Description** |
| --- | --- |
| Screentime | Gathering screen time of specific apps, websites, etc. |
| Smartwatch | Gathering biometric data (e.g., stress and sleep time) |
| FeedPulse | Gathering checkpoints (rating and comments) |
| Canvas | Gathering among other things, course data |
| “Open data” | Gathering open data (e.g., temperature, weather, wind, public transport) |
| Wi-Fi Attendance | Gathering info about student attendance at school buildings |

## 

## 2.2 Containers



*Figure 2 C2 - Container diagram*

The Quantified Student software system consists of four components, namely the dashboard, mobile application, API and the database. The student interacts directly with the dashboard and optionally with the mobile application. These two interfaces make use of the API, which is used as a central data access interface, and implements caching, validation of authentication and authorization.

The mobile application is used to gather screen time data and to allow stable Bluetooth communication with a smartwatch/peripheral. After gathering and processing the data, it is sent to the API.

# Conclusion

After designing the two levels of the C4 model diagram, the team can be sure that the context and its containers are properly designed and aligned with the product owner(s) and stakeholders. The first two levels of the C4 model help by creating a clearer picture of the software architecture and allows the Quantified Student team to plan accordingly. The next step is to design the components in a component diagram for each container in the container diagram.