Digitalization of a Pick and Place Module in the context of RAMI4.0

Author:

Carlos Josue Rene Avila Carrillo (7025691)

Examiners:

Prof. Dr. Armando Walter Colombo
M.Eng. Jeffrey Werman

M.Eng. Industrial Informatics - Semester Project - SS 2024

September 09, 2024

Hochschule Emden/Leer · Constantiaplatz 4 · 26723 Emden · http://www.hs-emden-leer.de

Contents

List of Figures

List of Tables

Abstract

asdfasdf

1 Introduction

The Plattform Industrie 4.0 is a German government initiative aimed at advancing the digital transformation of manufacturing and production industries. In a general sense this concept is known by other names such as: the fourth industrial revolution, smart manufacturing, industrial internet of things (IIoT), to name a few. The Plattform Industrie 4.0 name for this concept is **Industrie 4.0**.

1.1 Industrie 4.0

Industrie 4.0 refers to the intelligent networking of machines and processes for Industrie with the help of information and communication technology. Its fundamental purpose is to facilitate cooperation and collaboration between technical objects (assets), which means they have to be virtually represented and connected [2].

1.1.1 RAMI4.0

The Plattform Industrie 4.0, in partnership with many other stakeholders, has created the **DIN SPEC 91345**. This DIN SPEC describes the RAMI4.0 which is a reference architecture model in the form of a cubic layer model, which provides an architecture for technical objects (assets) in the form of layers, and allows them to be described, tracked over their entire lifetime and assigned to technical and/or organizational hierarchies. It also describes the structure and function of Industrie 4.0 components as essential parts of the virtual representation of assets. [2].

1.1.2 Asset Administration Shell

asdfa;sfaskljdfak;

1.2 Digital Factory

asfa

1.2.1 Pick and place module

Worked previously under the name of Delta robot

The goal of this project is to present a functional implementation of the pick and place module as an Industry 4.0 component by exposing its operational data as a digital twin using an AAS served through an OPC UA server.

This document is not only a report on the semester project but also aims to be a manual or guide into how to use the RAMI4.0 as a conceptual tool to digitalize an asset.

2 Methodology

The RAMI4.0 is used to create a model of an asset as an Industrie 4.0 component. The first step is to locate the asset in the RAMI4.0, this means that the dimensions of the RAMI4.0 are nav?????????????????

2.1 Asset position in the RAMI 4.0

The RAMI4.0 is a three-dimensional layered reference model. Figure 1 shows a visual representation of the RAMI4.0. Being a reference model simply means it is a model that is used as a reference or basis to create other models. In this case a model of the pick and place module as an Industrie 4.0 component.

Each dimension of the RAMI4.0 provides a context in which an asset is represented. In a broad sense navigating the RAMI4.0 involves three steps:

1. Determine the position of the asset in the Hierarchy Levels axis.
2. Determine the position of the asset in the Life cycle & Value stream axis.
3. Determine the contents of the architecture layers axis.

Steps 1 and 2 provide a notion of **what** data and information is relevant to digitalize given a context. Step 3 provides a notion of **how** to digitalize the relevant data, this is, how the data and information is going to be made available to the business.

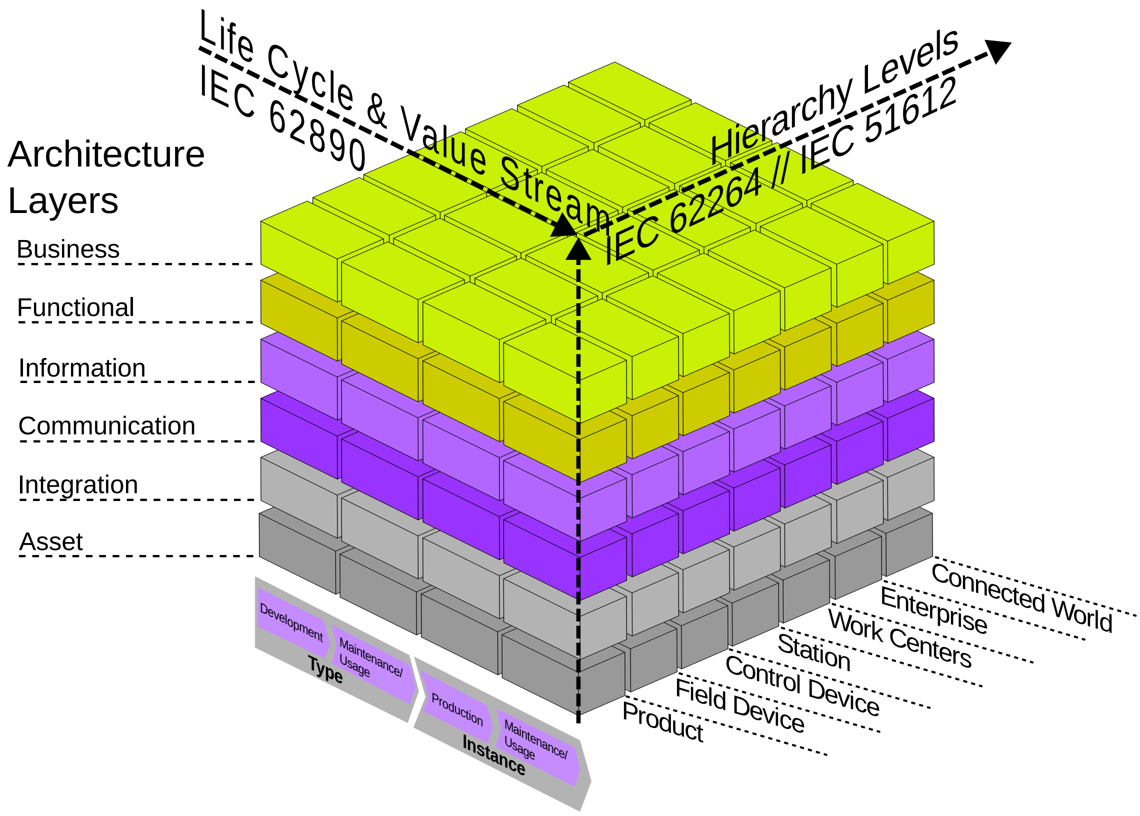

Figure 1: RAMI4.0

The following three sections explain these steps in more detail using the pick and place module as the asset to digitalize.

2.1.1 Position in the Hierarchy Levels Axis

The Hierarchy levels axis of the RAMI4.0 is based on the role-based hierarchy model of the IEC62264 [1]. This axis describes the assets of an enterprise that are involved in manufacturing.

The RAMI4.0 adds the “Connected World”, “Field Device”, and “Product” levels to reflect the needs of Industrie 4.0. Figure 2 shows an example of the levels in this axis referencing real components to give the reader a broader picture of where the pick and place module is located in this hierarchy and how it relates to other components in the enterprise.

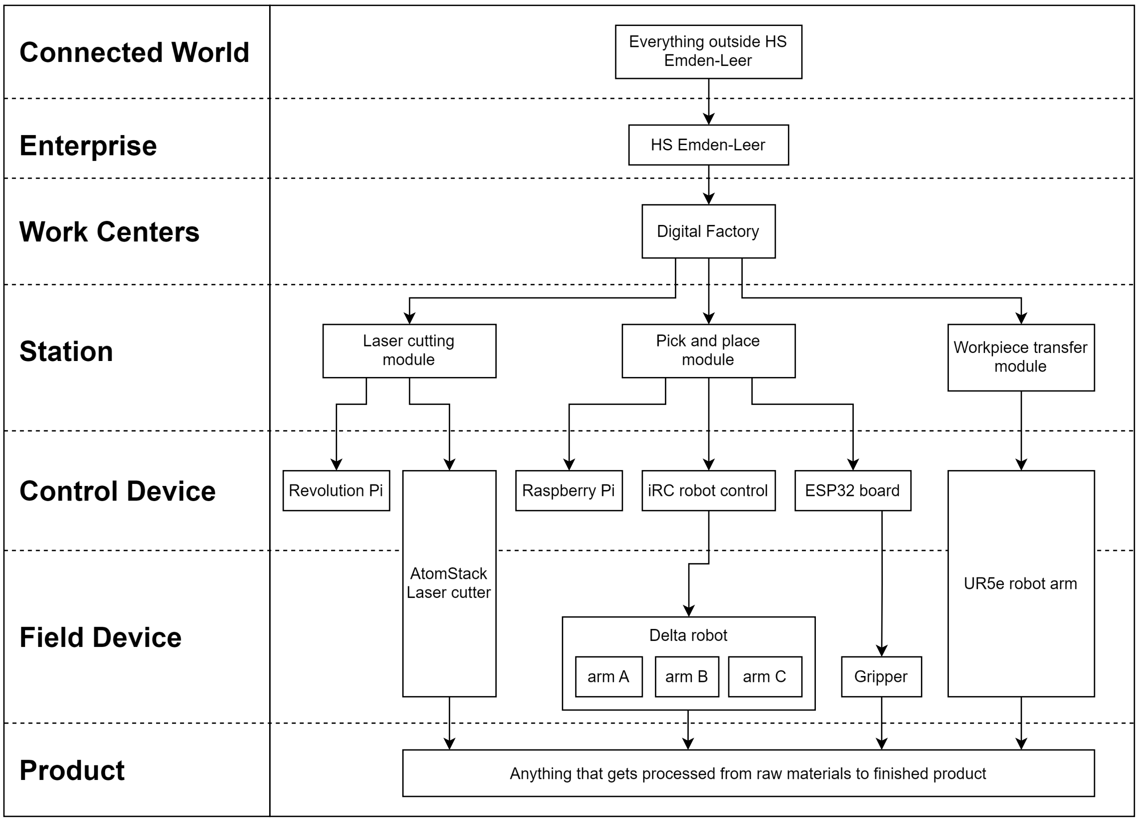

Figure 2: Role-based Hierarchy Levels

The role-based hierarchy defined in IEC62264 and the levels in the hierarchy axis of the RAMI4.0 do have a one-to-one mapping. This might lead to confusions when classifying assets in the hierarchy. This is for the modeler to determine so as long as it makes sense to the organization. For example, an enterprise, a site and an area may be all classified in the enterprise level.

In this case the pick and place module is classified as a **Station**

> **Note**
>
> A common, yet incomplete, definition of a station is an asset that is composed of sensors and actuators.
> A better definition of a station is an asset that has the equipment to manipulate a product (has sensors and actuators), has well-defined manufacturing capabilities and throughput capacities, it performs a segment of the manufacturing process, and is used for Level 3 functions (in a functional hierarchy)

2.1.2 Position in the Life Cycle & Value Stream Axis

The Life cycle & value stream axis is used to describe an asset at a particular point in time during its lifetime, from its conception and design, to its production and value-added use right up to its disposal. [2]

In this case the pick and place module is classified in the **instance** fase because it is already an object in the real world. Further more it is also in the **Maintenance/Usage** sub-phase because it is already operational (in use) and constantly improved (in maintenance).

2.1.3 Architecture Layers

The Architecture Layers axis describes the architecture in terms of properties and system structures with their functions and function-specific data in the form of layers [2].

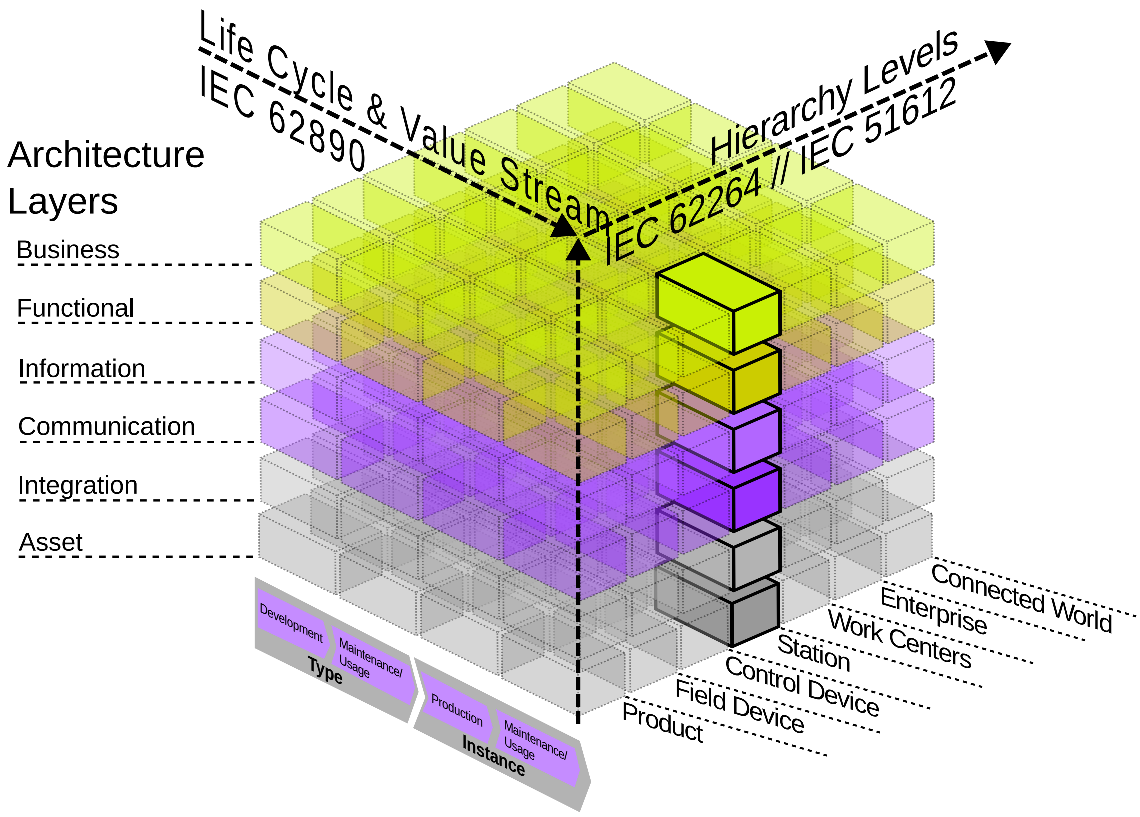

Figure 3: Architecture layers for station instance in maintenance/usage

By navigating the RAMI4.0 the scope of the model is reduced to the maintenance and usage data of a station instance. A visual representation of this process is shown in Figure 3 where each possible vertical column represents a set of data and information that describes the asset in a more specific context. Each of these columns can be considered a set of one or more digital twins of the asset. Here the interest is then in one single column.

The focus of this project is to translate the operational data of the pick and place station into digital data and information that enables and leverages the production of NFC business cards, as shown in Figure 4.

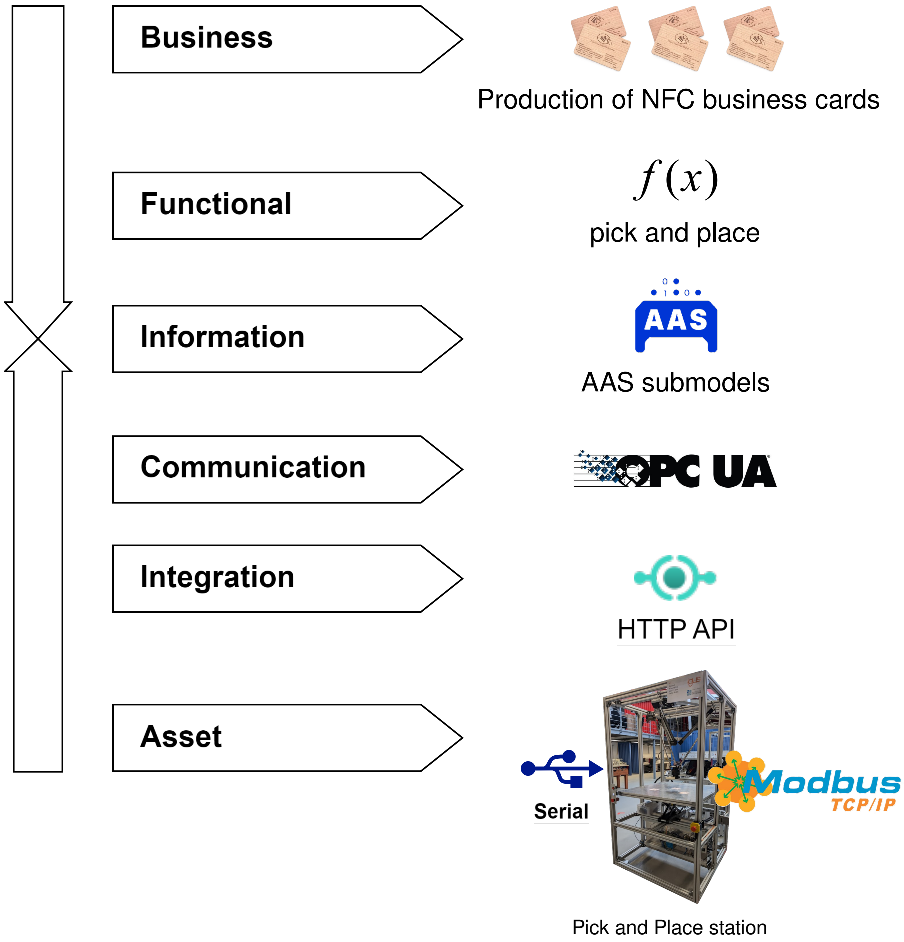

Figure 4: Architecture Layers Model

Notice that in Figure 4 there are two vertical arrows, one going from the business to the information layer, and another one going from the asset to the information layer. This is a top-down/bottom-up approach that allows a better understanding of the requirements of the actual implementation. The following is a description of this approach in the order mentioned.

Top-down:

- **Business Layer**: The business is given by the organization. In this case the organization wants to produce NFC business cards.
- **Functional Layer**: Once the business is known it is required to know what business or manufacturing functions enable this business. Examples here are: cutting

the cards, engraving the information on the cards, writing data to NFC chip of the card, transporting the cards. In this case a function is to pick and place cards for transportation, sorting, etc.

- **Information Layer**: Once the functions are determined it is required to know what information is consumed by these functions.

Bottom-up:

- **Asset Layer**: This determines the asset to digitalize. In this case it is the pick and place station.
- **Integration Layer**: Once the asset is known it is required to determine how data can be read from the asset as well as written to the asset. In other words, how to interact with the asset digitally.
- **Communication Layer**: Once the interaction with the asset is established it is required to transport the digital information in a Industrie 4.0-compliant way. This means that the information can be consumed by other Indutrie 4.0 components.

> **Note**
>
> Notice that the bottom-up approach helps to determine **how** the information is transported and consumed. The top-down approach determines **what** information is needed and how it should be presented.

3 Implementation

A model for the pick and place station as an Industrie 4.0 component has been created as shown in Figure 4. This model is the basis for the implementation of the component using software and hardware tools.

3.1 Existing hardware and software

As was seen in Figure 2 the pick and place station includes a Raspberry Pi as a control device. This adds an implementation detail which is that the AAS is to be embedded in the station as opposed to being hosted in a cloud.

The Igus Delta robot has its own robot control hardware and software which implements a modbusTCP service.

There is an ESP32 board controlling the gripper.

As a result from the previous work on the module there is a python library that interfaces with the robot control's modbusTCP service which greatly simplifies the interaction with the delta robot. There is also an Arduino based firmware for the ESP32 to control the gripper and a python library that wraps the communication to the board using a serial interface.

3.2 New implementations

3.2.1 AAS Package Explorer

We want to use the AASX file which is created by the package explorer.

Specifically, the information layer is to be implemented using an AAS and the communication layer is to be implemented using OPC UA. This implies the use of a tool that can expose the AAS information model as an OPC UA service. Moreover it is required to feed and consume real-time operational data through the OPC UA interface.

3.2.2 FA³ST Service

A common way to implement an AAS is to export the AAS information model to the OPC UA NodeSet schema and then use this schema to create an OPC UA server.

This projects present a different way implement an AAS using the FA³ST service tool which takes the AASX package file and directly exposes the information model as an OPC UA service. The FA³ST service also provides an interface to feed and consume real-time data from the asset called AssetConnection.

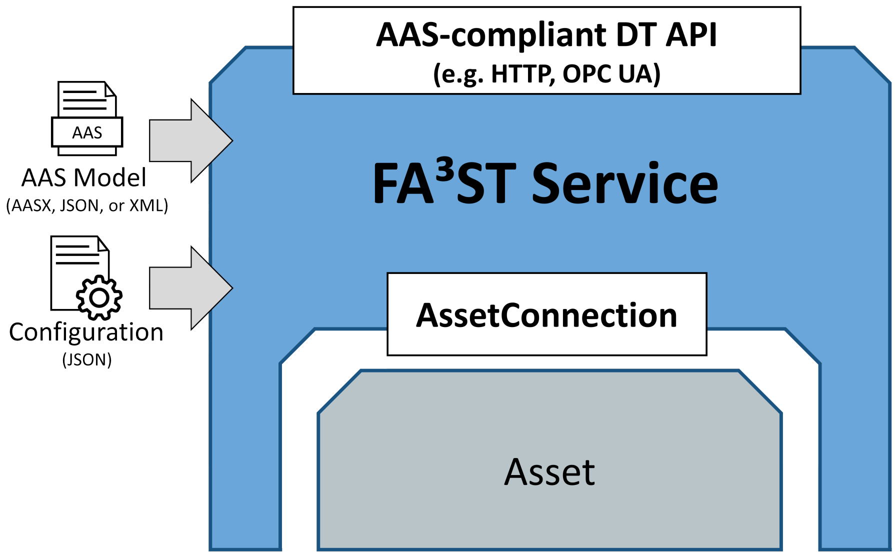

Figure 5: FA³ST service overview[1]

3.2.3 HTTP API, Delta robot library, and Gripper firmware

A new firmware was created to control the gripper. This firmware improves the communication by using JSON formatted messages which allow the exchange of status data from the gripper. more????

The python library to communicate with the delta robot's modbusTCP service was reused and refactored for easier usage and maintenance.

An HTTP API was created to implement the integration architecture layer. This API comes as a necessity because the FA³ST service asset connection only supports OPC UA, HTTP, and MQTT connections. From these three options an HTTP API is the easiest to implement.

3.2.4 Docker containers

Finally, in order to keep the implementation free from software compatibility and integration issues, it was decided to use Docker containers as run environment. This way ???????

Figure 6 shows a diagram of the actual implementation.

[1] https://faaast-service.readthedocs.io/en/latest/_images/overview.png

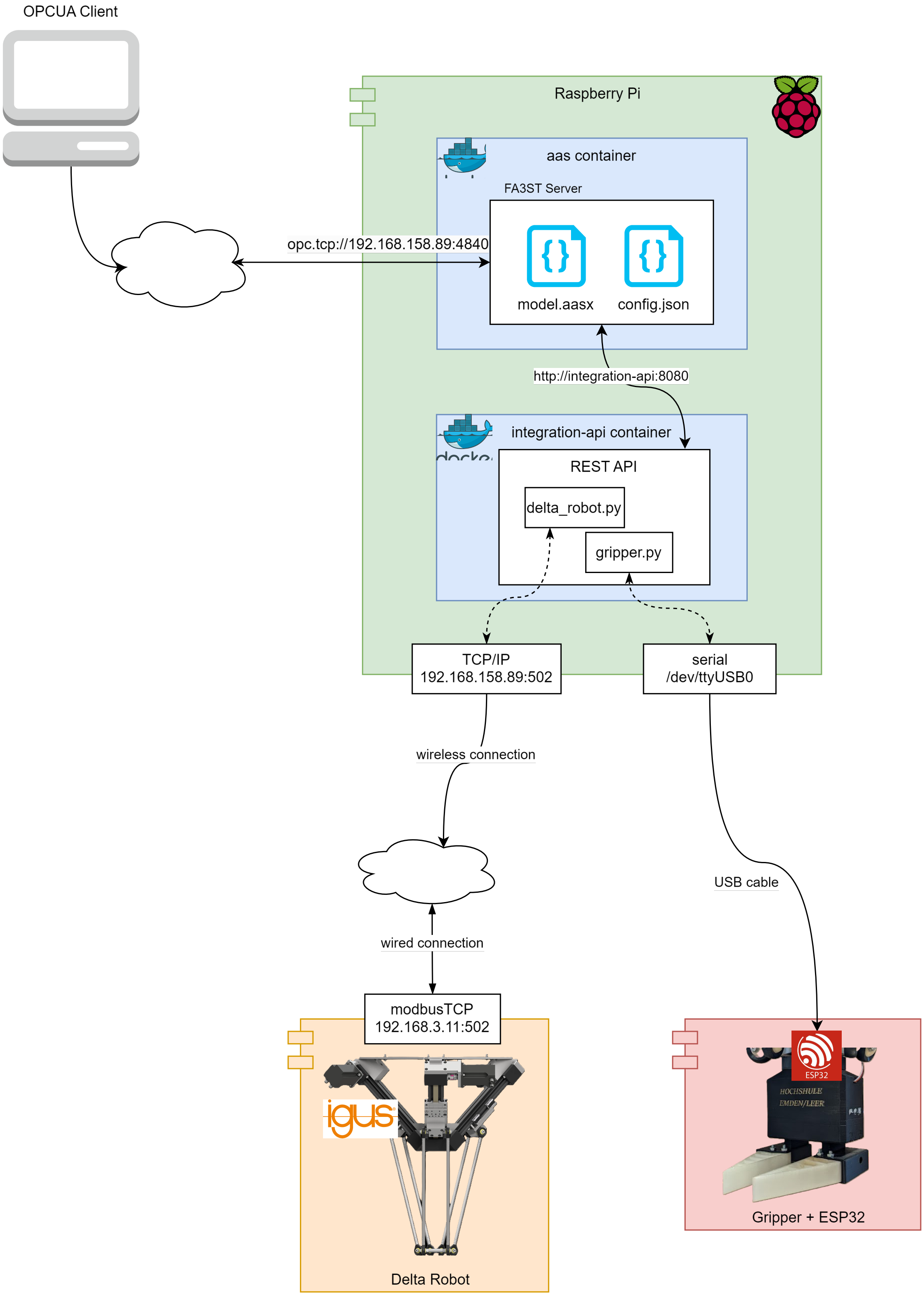

Figure 6: Pick and place station implementation diagram

4 References

[1] *IEC 62264-1: Enterprise-control system integration - Part 1: Models and terminology*. Second edition. International Electrotechnical Commission, 2013. URL: https://webstore.iec.ch/publication/6675.

[2] *Reference Architecture Model Industrie 4.0 (RAMI 4.0)*. DIN SPEC 91345. Accessed: 2024-06-23. Berlin, Germany: Deutsches Institut für Normung, Apr. 2016. URL: https://www.din.de/resource/blob/229091/38ec5291c94e5d7e5e7bce7b3fc4c06e/din-spec-91345-pdf-data.pdf.