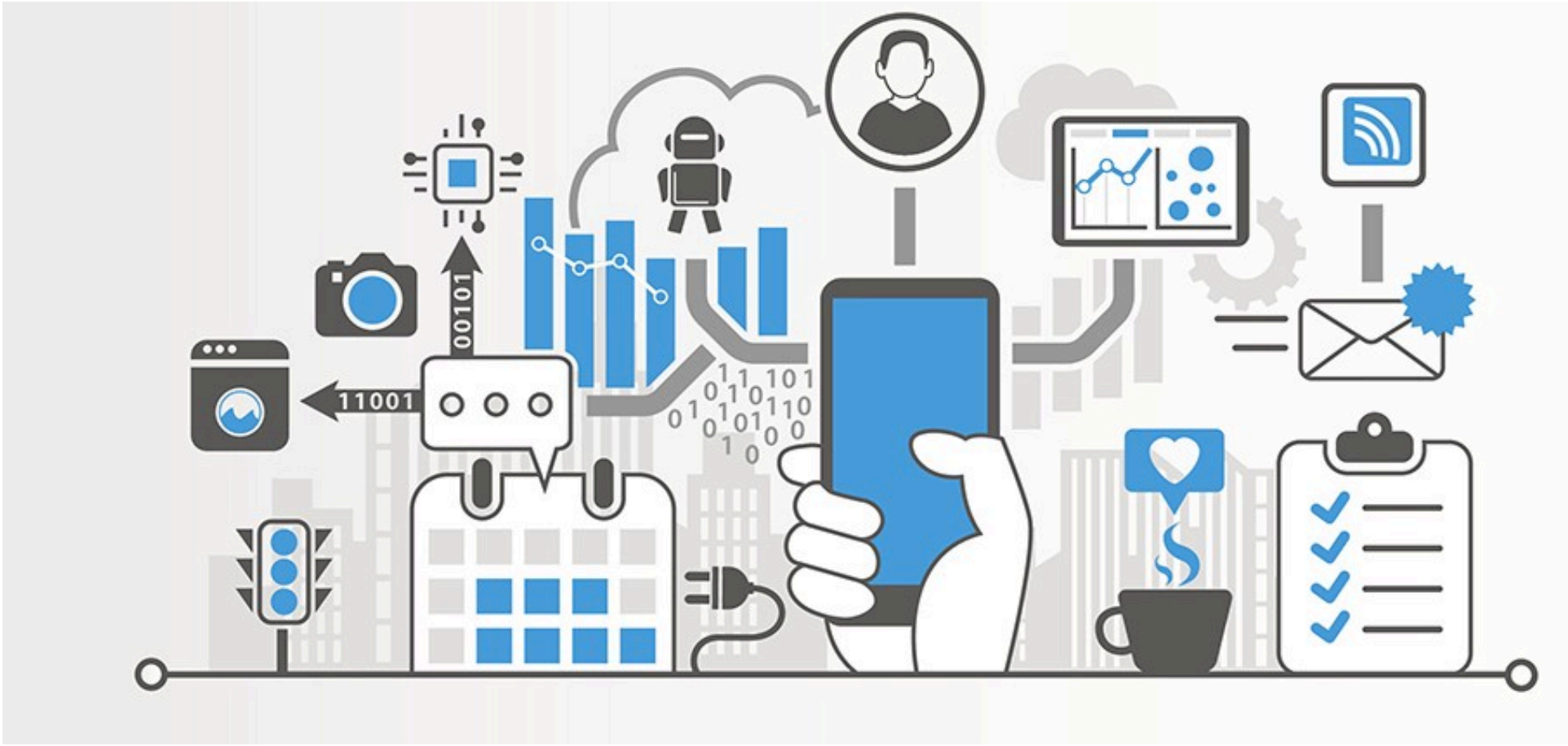
Programming for IoT Applications

Edoardo Patti

Project Proposal



Project Proposal - Introduction

Name of the use case: IoT platform for Smart Home

Scope: The proposed IoT platform aims at providing services for a smart home management.

Objective: The expected results consist on providing a smart control of appliances to minimize the energy waste and promoting green behaviours via user-awareness applications.

Domain(s): Smart home, Smart Building, Smart Grid.

Stakeholder(s): Home inhabitants, Energy aggregators.

Project Proposal - short description

The proposed IoT platform aims at making smart our homes. It integrates different IoT devices for managing appliances in home environments. It provides control strategies for lighting and heating systems to minimize the energy waste. The overall platform provides unified interfaces (through both REST and MQTT) to integrate the home into Smart Building and Smart Grid environments. Hence, Demand/Response policies can be applied. Finally, the platform provides end-users with detailed knowledge of the household consumption for each appliance.

Project Proposal - short description

Summarizing, the main features it offers are:

- remote control of appliances;
- control strategies for lighting and heating systems;
- unified interfaces (i.e. REST Web Services and MQTT queues) available to enable Demand/Response;
- end-user applications for energy-awareness.

Project Proposal - complete description and diagram

The proposed IoT platform for Smart Home follows the micorservices designing pattern. It also exploits two communication paradigms: i) publish/subscribe based on MQTT protocol and ii) request/response based on REST Web Services.

In this context, ten actors have been identified and introduced in the following:

Project Proposal - complete description and diagram

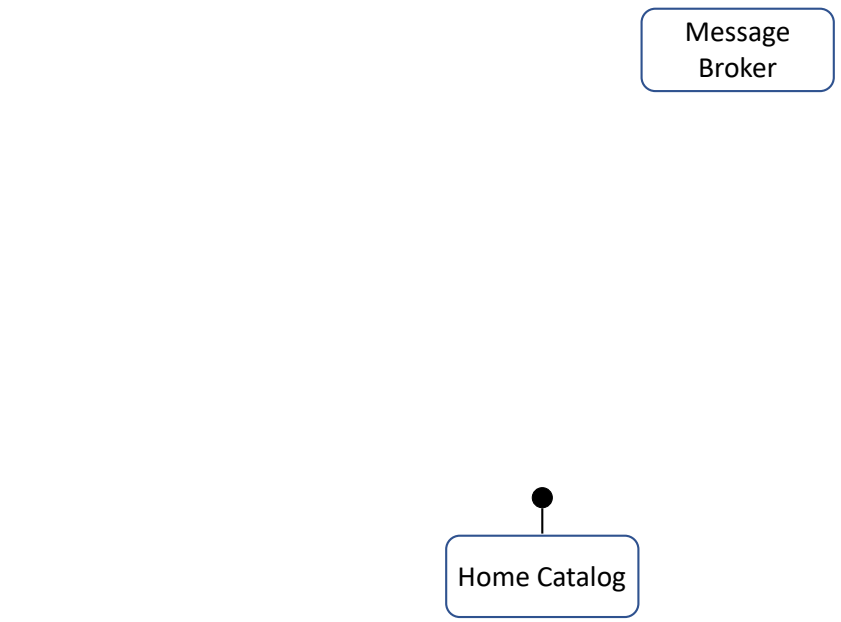
The **Message Broker** provides an asynchronous communication based on the publish/subscribe approach. It exploits the MQTT protocol.

Legend:

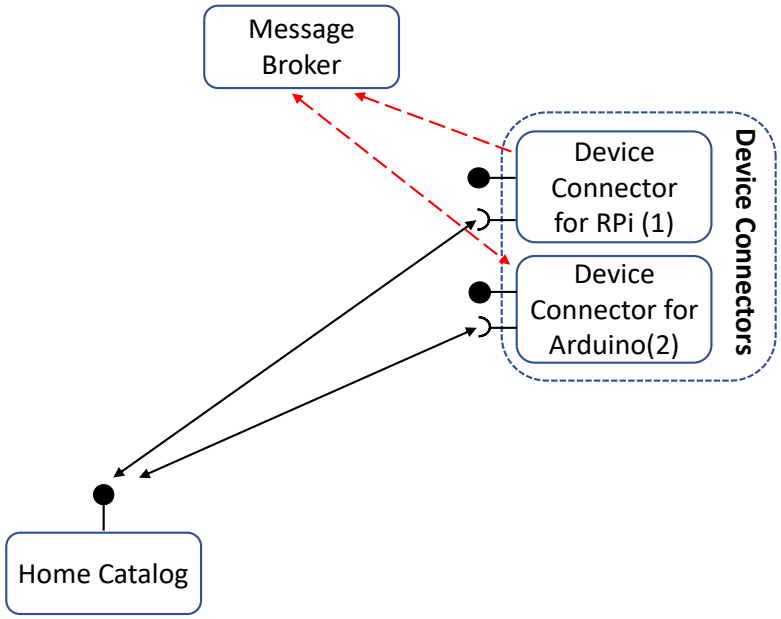
- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Project Proposal - complete description and diagram

The **Home Catalog** works as both **service and device registry system** for all the actors in the system. It provides information about end-points (i.e. REST Web Services and MQTT topics) of all the devices, resources and services in the platform. It also provides configuration settings for applications and control strategies (e.g. timers, list of sensors and actuators). Each actor, during its start-up, must retrieve such information from the Home Catalog exploiting its REST Web Services.



Project Proposal - complete description and diagram

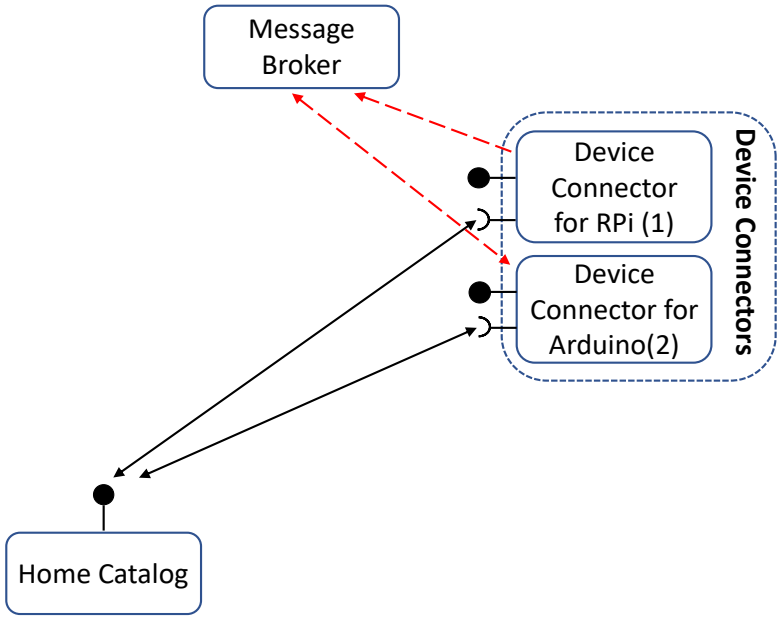


Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

The **Raspberry Pi Connector** is a *Device Connector* that integrates into the platform raspberry pi boards. Each raspberry is equipped with motion, temperature and humidity sensors to provide environmental information about the status of a room. It provides Rest Web Services to retrieve environmental information (i.e. temperature and humidity). It also works as an MQTT publisher sending information on user presence (when detected) and environmental data (every 5 minutes).

Project Proposal - complete description and diagram

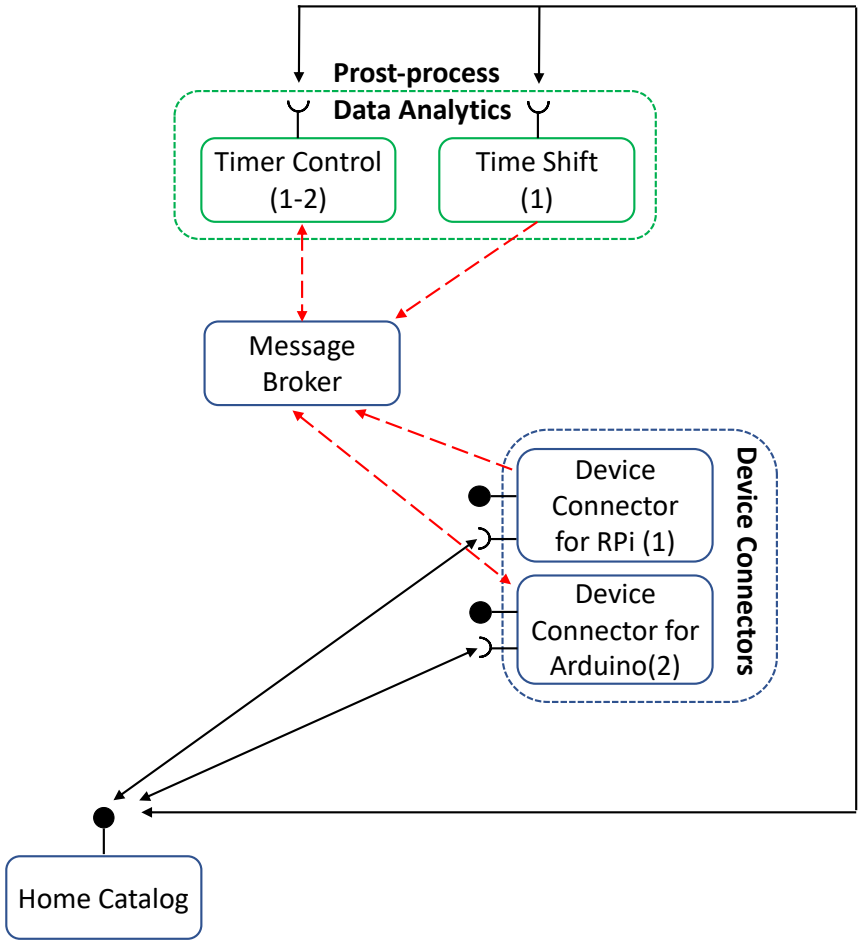


Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

The **Arduino Pi Connector** is *Device Connector* that integrates into the platform Arduino boards. Each Arduino is equipped with two relays to switch on and switch off the connected appliances. It provides Rest Web Services to retrieve and change the status of the connected appliances (on/off). It also works as an MQTT subscriber to receive actuation commands from other actors that exploit the MQTT protocol (e.g. Control Strategies).

Project Proposal - complete description and diagram

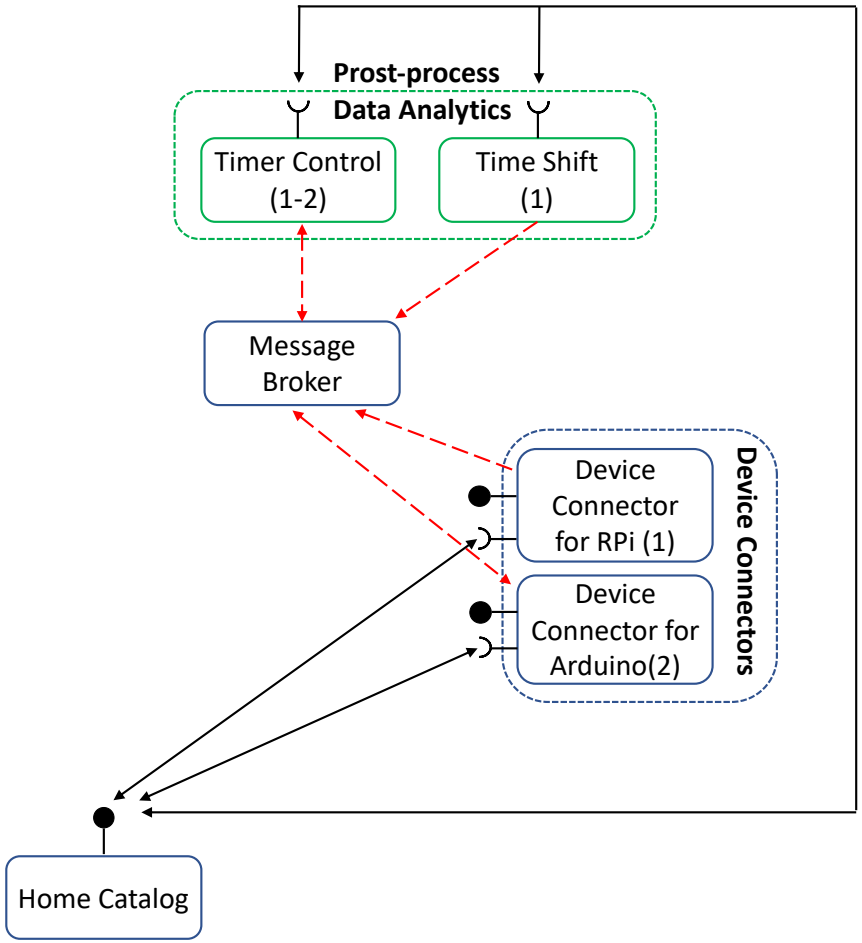


The **Timer Control** is a control strategy that manages the lighting system in rooms depending on user presence. Each room is managed by an instance of this strategy. It works i) as an MQTT subscriber to receive information on user presence; ii) as an MQTT publisher to send actuation commands to IoT Devices.

Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Project Proposal - complete description and diagram

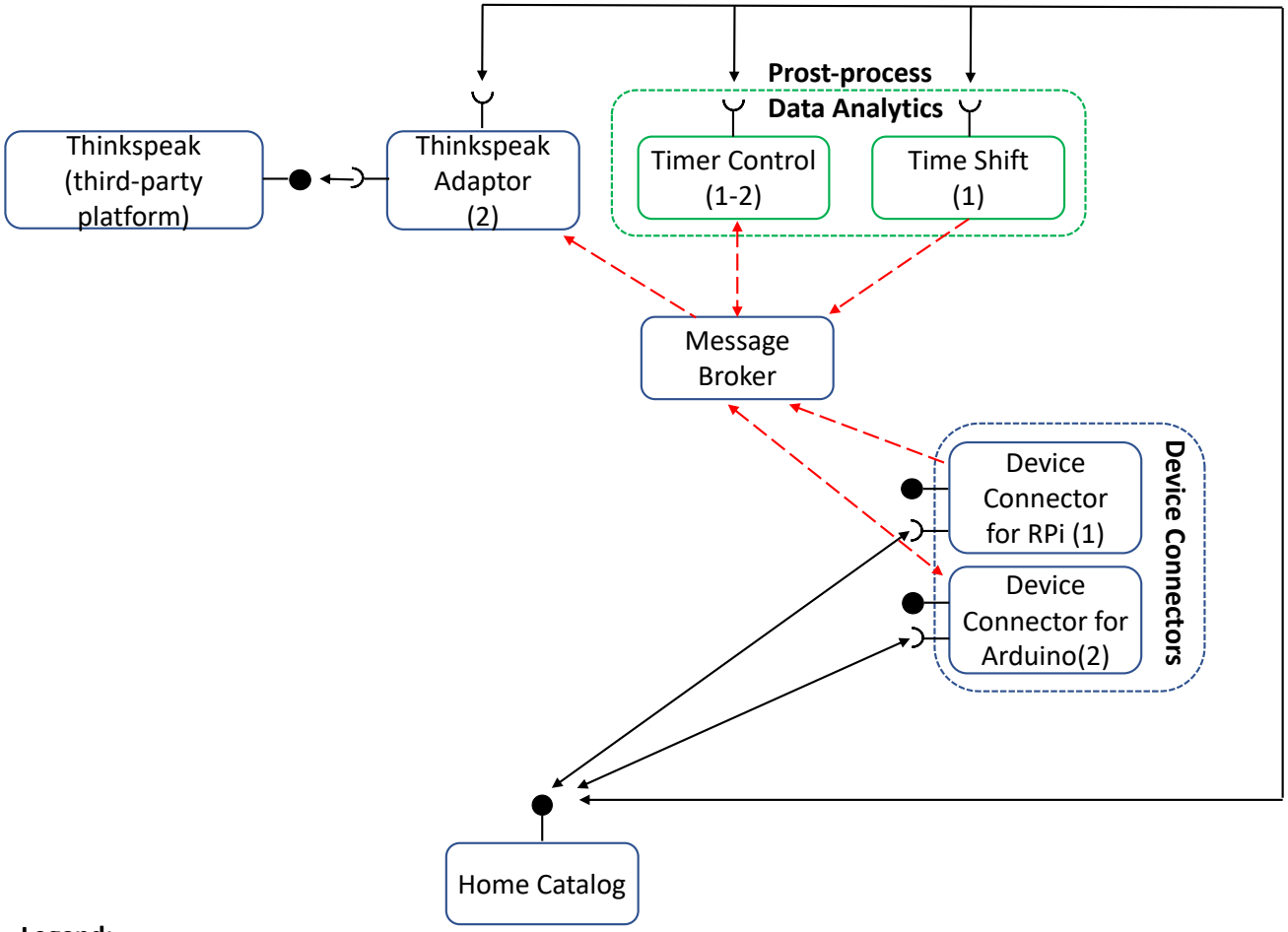


The **Time Shift** is a control strategy to manage appliances depending on time-schedules provided by the **Home Catalog**. For example, it allows users to switch on the washing machine from 19:00 to 7:00. It works as an MQTT publisher to send actuation commands to IoT Devices.

Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Project Proposal - complete description and diagram

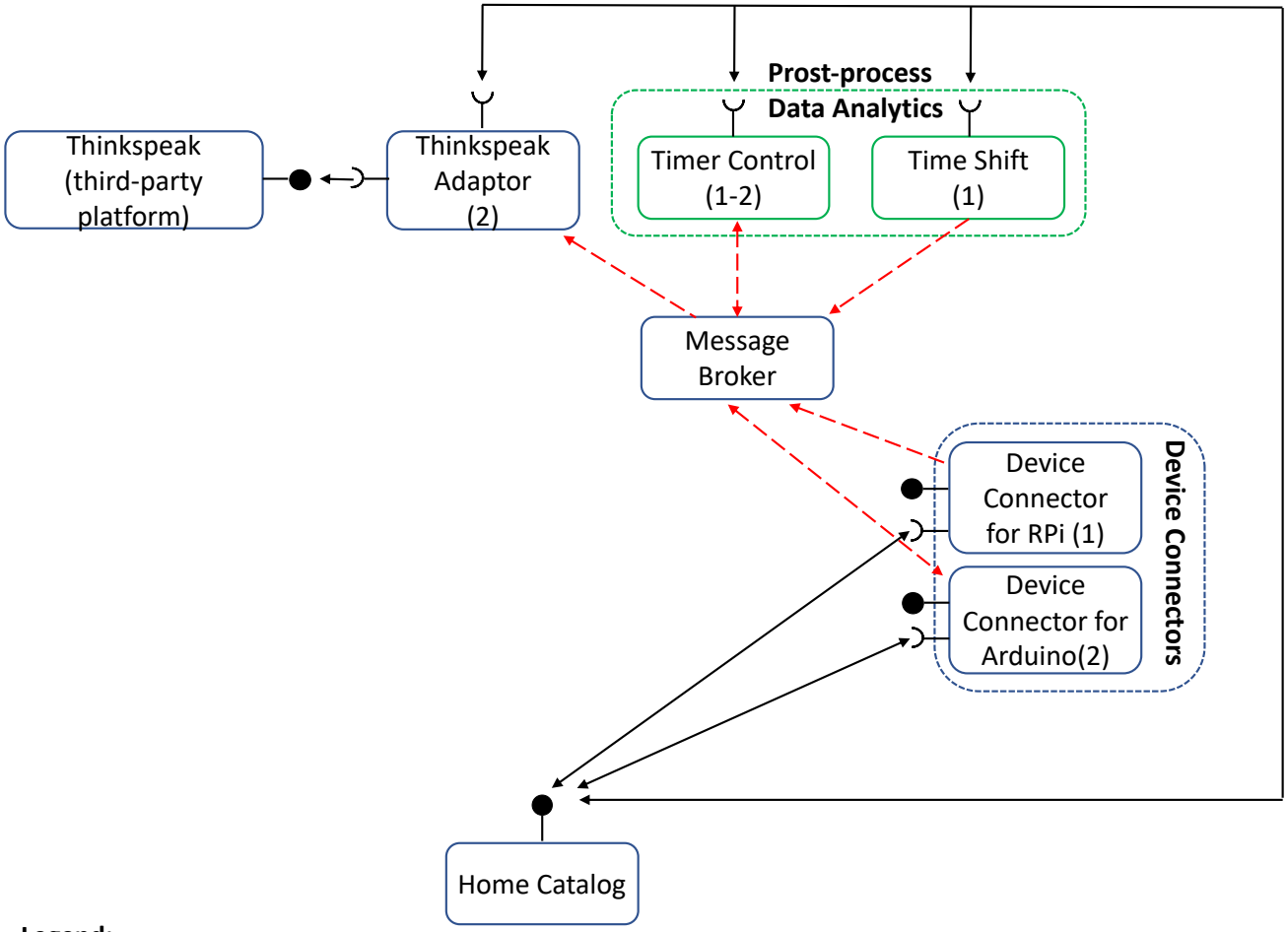


The **Thinkspeak Adaptor** is an MQTT subscriber that receives measurements on environmental measurements and upload them on **Thinkspeak** through REST Web Services.

Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Project Proposal - complete description and diagram

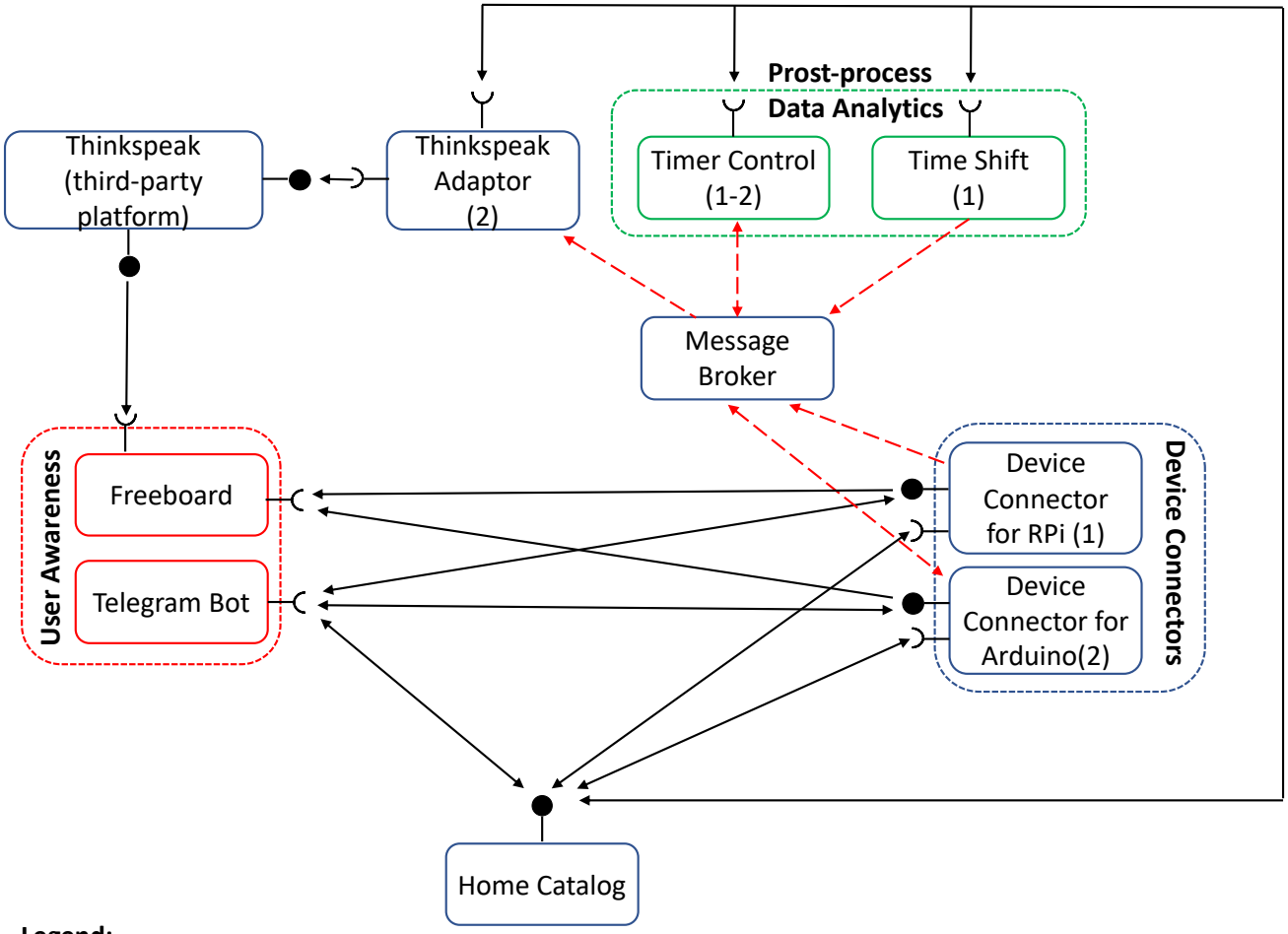


Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Thinkspeak is a third-party software (https://thingspeak.com/) that provides REST Web Services. It is an open-data platform for the Internet of Things to store, post-process and visualize data (through plots).

Project Proposal - complete description and diagram

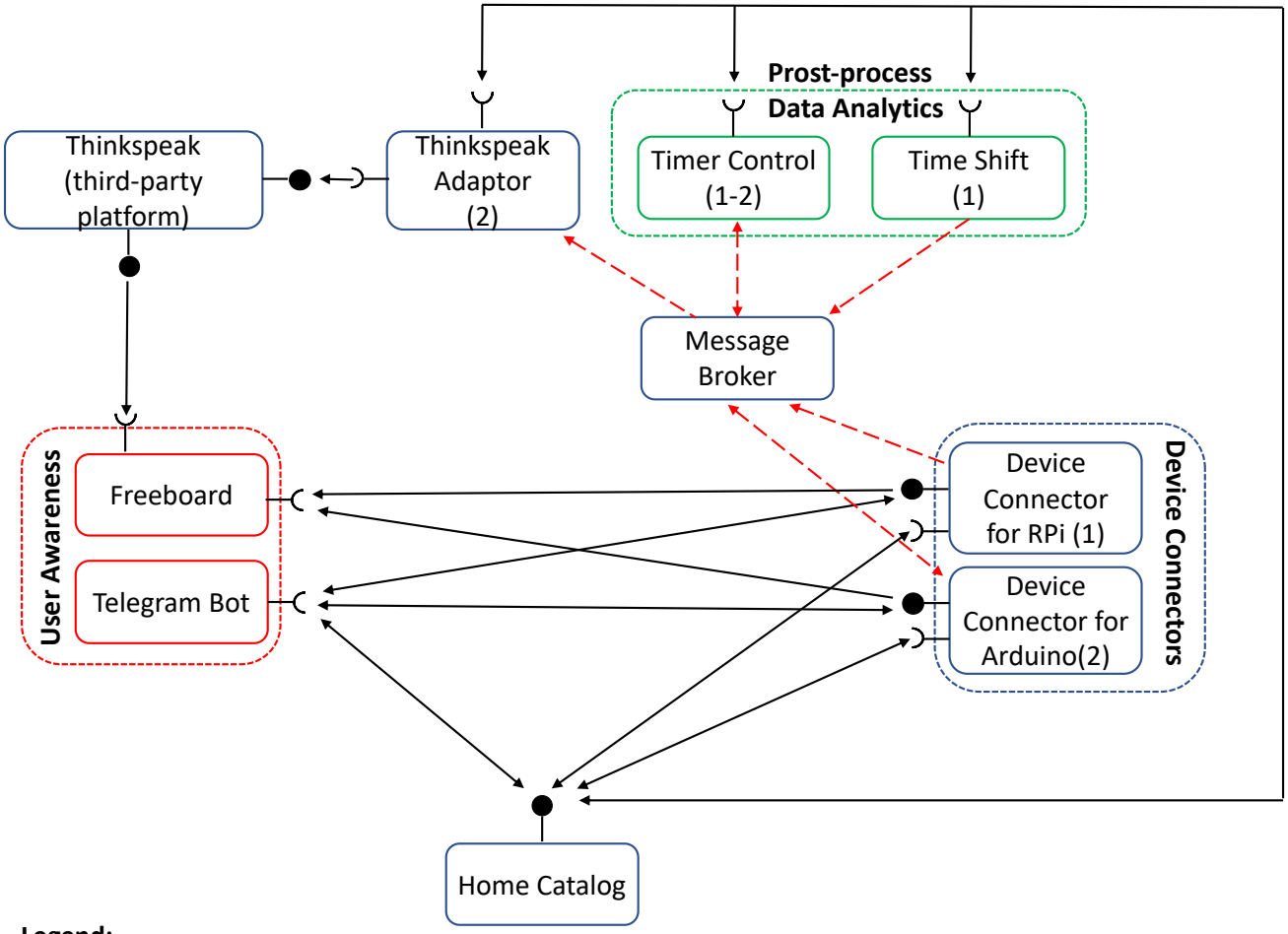


Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Freeboad is a dashboard to retrieve data from IoT devices and visualize them exploiting the REST Web Services provided by **Raspberry Pi** and **Arduino Connectors**. It also exploits the **Thinkspeak** Web Services to import plots about environmental measurements.

Project Proposal - complete description and diagram



Legend:

- REST Web Services (provider)
- REST Web Services (consumer)
- MQTT Communication
- (1) Publisher
- (2) Subscriber

Telegram Bot is a service to integrate the proposed infrastructure into Telegram platform, which is cloud-based instant messaging infrastructure. It retrieves measurements from IoT devices exploiting the REST Web Services provided by **Raspberry Pi** and **Arduino Connectors**. It also allows users on sending actuation commands to IoT devices again exploiting REST.