Lancaster University Leipzig School of Computing and Communications

An Open Source Smart home Platform

Niklas Harnish

12/05/2023

Supervisor: Dr Amna Asif

B.Sc. (Hons) Computer Science

Number of words = 0
This includes the body of the report only

Declaration of Originality

Put some text similar to the following in here:

I certify that the material contained in this dissertation is my own work and does not contain unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work.
I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.
Name:
Date:

Abstract

Put your abstract here. You should create a short abstract (200 words at maximum) which is on a page by itself. The abstract should be a very high-level overview: for example 1–2 sentences on the aims of the project, 1–2 sentences on the kind of design, implementation, or empirical work undertaken, and 2–3 sentences summarising the primary contribution or findings from your work. The abstract appears in the front matter of the report: after your title page but before the table of contents.

what should go in here:

aims

design

implementation

findings and primary contribution

If you want to dedicate to someone in particular

Acknowledgements

General acknowledgements ...
your supervisor, your family, your friends, ...

Contents

List of Figures

List of Tables

1 Introduction

actually write this section

Will write a couple of words about the project here, similar projects, inspirations etc. Citation here to remember how to do it :)

1.1 Aims & Objectives

When researching available smart home technology, one major gap I came across was the availability of open source software. While options exist for someone interested in connecting their proprietary device to an open source platform (view), there was no solution for anyone looking to build their own device and then connect it to an open source hub. In fulfilling this goal, to build an open source platform for both devices and the hub they will connect to, there are multiple objectives that will need to be met along the way:

find the link to this

1. Create a Library and API (Application Programming Interface) for building smart home devices.
2. Build a Server with an API for the smart home devices to communicate with. This will act as a hub and will control clients connected to it.
   a) This API should be well documented, so a user can interact with the hub, without using the Library.
3. Create a frontend, which will be populated with devices currently connected to the smart home. It will also be used to control clients connected to the server.
   a) The API provided by the server for this frontend should also be easy to use, so the user can create their own frontend environment.
4. The code of all of the above should be hosted in a public repository, with instructions for how to build and use every component of the system.
   a) An appropriate license should also be selected for this repository, so the code within it can be copied or modified by third parties.
   b) This repository should provide important links and provide information on the inner workings of the system, to support interested parties.

1.2 Project Overview

ask about this section

Each bullet point below would give a small summary of the section

1. **Background Research**
2. **Design of the System & Technology Decisions**
3. **Implementation**
4. **Results**
5. **Conclusion and Reflection**

2 Background

2.1 IoT System Architectures

Kamienski et al. describe a simple three layer architecture of an IoT system in [7]. Within this architecture, the top layer is the "Input System", from which any data that will influence the decisions of the IoT system will come from. Included in this are sensors, but also user facing interfaces. The second layer, known as the "Process System", is where any algorithms are run and system behavioral decisions are made. The goal of this layer is to gain an "improved understanding of the system where the data comes from" [7]. The bottom layer is the "Output System", which are where decisions made by the Process System will be enacted. This is often represented as the devices connected to the IoT system.

This three layer architecture is expanded upon by Bansal and Kumar within [2], where three more architectures are described which expand upon the ideas within the three layer architecture. They are however more specialized than the three layer architecture. The first of these is a "Middleware Based" architecture, which can take many forms, but is usually combined with another type of architecture, with a middleware layer. The different types are described in detail by Zhang et al. in [9]. The second is known as a "Fog Based" architecture, where certain tasks, usually those with less processing requirements, are calculated on device to reduce latency. More computationally expensive tasks are however calculated on a server in the cloud [8].

The most relevant architecture to this report is known as a "Service Based" architecture (SBA). The SBA is defined around the concept of the Service Oriented Architectural (SOA) style [5] of software design. SOA is defined by the Open Group Foundation as an "architectural style that supports service-orientation", where a service is a "logical representation of a repeatable business activity that has a specified outcome" [1]. Each service is a "black box" any device interacting with it. Other devices use interfaces and API endpoints to make requests to the service and receive a result. A SOA is composed of many different services. In SBA, services are used to offer device functionality using interfaces, often using web based concepts such as SOAP or REST APIs [3]. This allows devices with different capabilities and purposes to interact with the same system, allowing for an IoT system that is more flexible.

Add section about which one i chose

2.2 The Smart Home System

Sethi and Sarangi [8] define six components that need to be present within a social IoT setting. A social IoT system is defined as a IoT system where devices form relationships with other devices. While our smart home system will not be a social IoT system, some of these concepts are still of interest. These are:

ask if this is ok with citation as its quite similar

1. ID: the device within the system needs to have a way of identifying it.
2. Meta-data: the device should have information regarding its form and purpose

3. Security Controls: the system should have some way of distinguishing between different users. It should also be able to distinguish what types of devices it can connect to or can connect to it.
4. Service Discovery: each device should be able to discover other devices connected to the system and what services they offer.

There are some specific constraints specific to Smart Homes. Reliability is a key concern, due to the lack of a trained professional being available to fix any issues that arise. This is contrast to more industrial IoT settings, where there might be someone to fix any issues that arise. Another concern is the security and privacy of the system. Due to smart homes inherently having access to sensitive data (due to their position in someone's home), one must ensure that the system is both ethically sound and secure. The issue of security is further discussed in Subsection 2.4.

further add to this section, just not sure what yet

2.3 APIs and Web Interfaces

The book "Designing Web APIs" makes an important distinction about APIs that can often be forgotten by developers. "Although APIs are designed to work with other programs, they're mostly intended to be understood and used by humans writing those other programs" [6] (Chapter 1). Do to this reality, one must remember to design APIs appropriately.

probably rewrite this paragraph, or add another line here

"Request Response" APIs (RRA) expose their interface through a web server, to which clients can make requests. A client will request data and will receive a response from the server. Common formats for responses include JSON and XML. [6]. One popular type of RRA is known as Representational State Transfer (REST). Two important properties of REST is that it is used in Client-Server scenarios and that every request is stateless [4]. This means that every API request from the client to the server, must contain all information required to complete that request, without the server storing any of that information. Instead, all state is stored on the client. While this constraint might seem strange, it makes any API implementing REST easily scalable, and potentially easier/-faster to build. The downside being inherently increased network traffic, with less control application behavior. [4].

Another popular implementation of a RRA is known as the Remote Procedure Call Architecture (RPC). The key difference between RPC and REST is that RPC is about making an action on the server. In REST the client supplies the server with the information required to take an action, whereas using RPC the client tells the server what action to take. RPC APIs can usually express more nuance in their requests and are generally stateful. While RPC generally also uses JSON or XML for requests and responses, there are multiple other frameworks, such as gRPC (Google RPC) and Apache Thrift. These are usually serialized are therefore consume less network traffic than non-serialized formats such as JSON [6].

2.4 Security

2.5 Networking

2.6 Open Source and Licensing

3 Implementation

3.1 Server & Hub

3.1.1 Security

3.1.2 Threads & Concurrency

3.1.3 Device API

3.2 Device Library

3.3 Example Device

Perhaps add a section about networking here, talk about different things tried

3.4 Web & CLI Frontend

3.5 Open Source

Bibliography

[1] Soa source book.

[2] S. Bansal and D. Kumar. Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication. *International journal of wireless information networks*, 27(3):340–364, 2020.

[3] I.-R. Chen, J. Guo, and F. Bao. Trust management for soa-based iot and its application to service composition. *IEEE transactions on services computing*, 9(3):482–495, 2016.

[4] R. T. Fielding. Architectural styles and the design of network-based software architectures. *UCI Donald Bren School of Information & Computer Sciences*, 2000.

[5] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE transactions on services computing*, 3(3):223–235, 2010.

[6] B. Jin. *Designing Web APIs : building APIs that developers love*. O'Reilly, Beijing, 1st edition. edition, 2018.

[7] C. Kamienski, R. Prati, J. Kleinschmidt, and J. Soininen. Designing an open iot ecosystem. Belem, Brazil, 2019.

[8] P. Sethi and S. R. Sarangi. Internet of things: Architectures, protocols, and applications. *Journal of electrical and computer engineering*, 2017:1–25, 2017.

[9] J. Zhang, M. Ma, P. Wang, and X. dong Sun. Middleware for the internet of things: A survey on requirements, enabling technologies, and solutions. *Journal of Systems Architecture*, 117:102098, 2021.

ask about SOA source book citation

A Original Project Proposal

Put a copy of your proposal here

B Another Appendix Chapter

This could be about your experiments