Project Molecule

Planning Document v2 Revised 14-Nov-2016

May1739
project.molecule@outlook.com

Dr. Arun Somani – Advisor
arun@iastate.edu

Ryan Wade – Team Leader
ryanwade@iastate.edu

Nathan Volkert – Communications Lead
nvolkert@iastate.edu

Daniel Griffen – Key Concept Holder
dgriffen@iastate.edu

Alex Berns – Webmaster & Scribe
tagger94@iastate.edu

1 CONTENTS

2 Introduction

2.1 Project Statement & Purpose

Project Molecule is a system for smart home living. Unlike other solutions, we sought to create a high level ecosystem which can encompass all smart home devices. The framework is designed from the ground up to be fault tolerant. To this end, control is distributed amongst an array of low cost nodes located throughout the home. These nodes securely communicate with each other to make decisions, disseminate information, and collectively control smart devices connected to the network. This allows the end user to conveniently operate their devices from anywhere in the home. Project Molecule demonstrates how technology can enable a smart revolution in the home while maintaining security and reliability.

2.2 Goals

We have a good number of goals to achieve throughout our work on this senior design project. First, we have general learning to accomplish. This includes learning the Rust Programming Language and getting a better understanding of fault tolerance. Next, we hope to achieve network communication between the Raspberry Pi's in a safe and reliable manner. We want to maintain integrity of the system while still retaining functionality. We want to create nodes in such a way that they can have functionality, such as turning on a light, something the users could benefit from. We want to create a UI that is convenient to use to operate the nodes. Another goal is to introduce some automation to the system using something like IFTTT (If This Then That) or web hooks. Stretch goals include installing a system in an actual home or apartment, more complicated application procedures, and getting the rights to potentially expand the project beyond the time period of senior design. For our prototype system, each node will run on a Raspberry Pi.

3 Deliverables

Primary Goals

- Network of Raspberry Pis
- Web application to control network
- Scale model demo
- Source code on GitLab
- Documentation explaining setup and use

Extra Goals

- Demo video of real world use

4 DESIGN

4.1 PREVIOUS WORK/LITERATURE

As it stands today, many companies are working on the individual components which make up a Smart Home System. For example, there are several key competing standards for embedded communication. Some devices use generic WiFI while others use Zigbee or BLE Mesh which address problems, such as power consumption. Beyond communication, other groups are working on toolkits to empower device creators. This makes it easier for small companies to bring new products to market. Amazon, Google, and Microsoft are working on Speech Recognition and AI “skills”. This enables a user to conversationally interact with their technology. All of these technologies provide the building blocks necessary to create a smart home, but a system still needs to be developed to connect them all together.

Zigbee: One of the most widely used standards for IoT devices, being used with large companies such as Comcast and Time Warner. Zigbee is also available for do-it-yourself-ers. This platform is much more controlled, the standards are developed with a democratic process involving active developers. A new standard in the Zigbee system is only created when it is found that the current system is lacking something that could be used in market relevant products [1].

BLE Mesh: A cloud connect IoT platform using Bluetooth Low Energy as the communication method. Each node is part of a mesh network that perform local actions and can communicate with other Bluetooth devices. The network connects to any cloud services via a mesh gateway or controller interface (like a phone) [2].

Bluetooth SIG: The group that develops the Bluetooth communication system. They have started to add tools to the developer toolkit to make it easier for developers to create connected IoT devices. The ability to connect to modern smartphones and an ever increasing set of sensors allows smaller developers to create products for the market [3].

Amazon’ Alexa Skill API: The back-end to Amazon’s Echo product, Alexa is a voice service that can be integrated into a home. Alexa and Echo can understand speech and allow developers to easily react to spoken words. Skill API allows developers to “teach” Alexa new skills and expand her functionality. This system can be used to control a small set of IoT devices, which Amazon hopes to expand [4].

4.2 Proposed System Block diagram

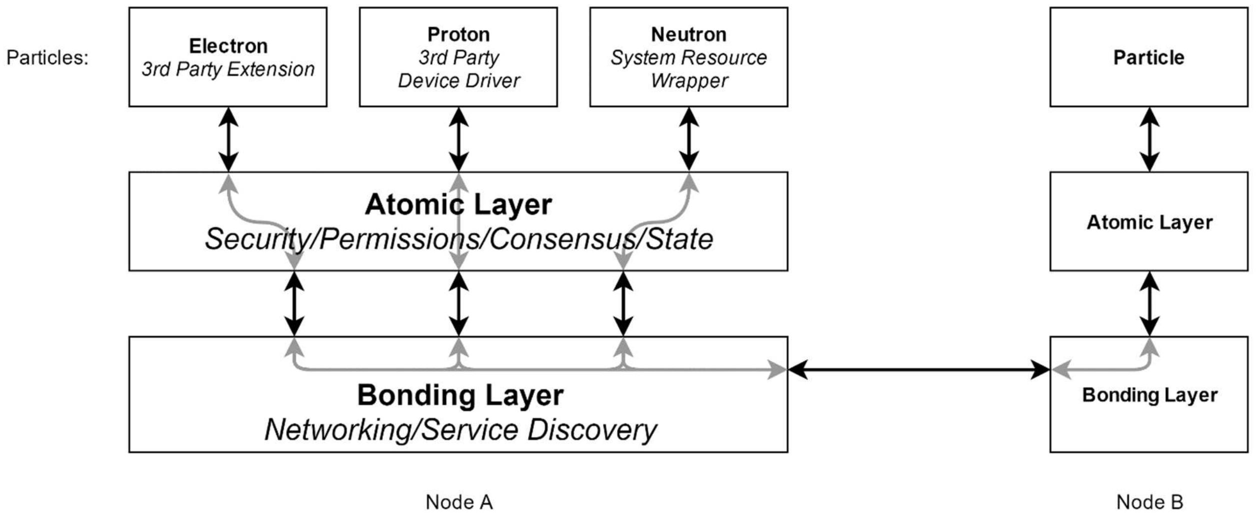

Figure 1: Block Diagram

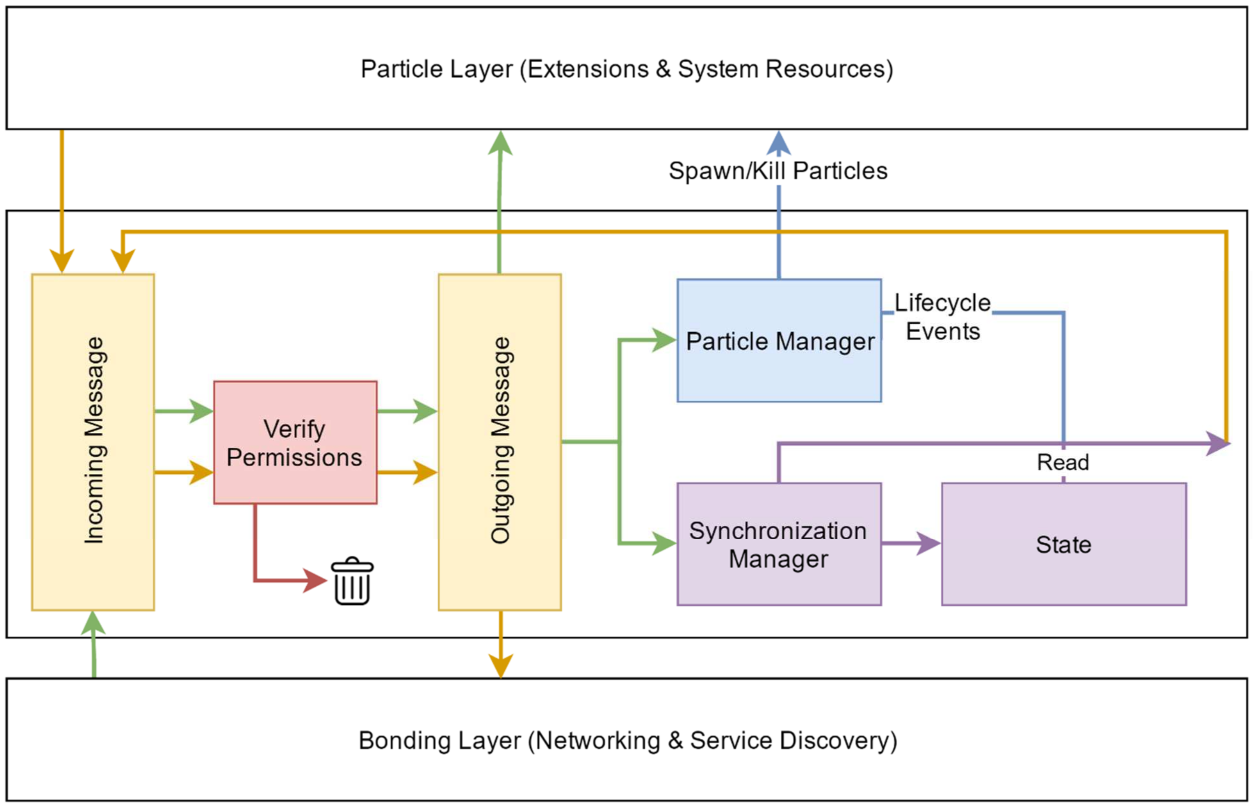

Figure 2: Atomic Layer Diagram

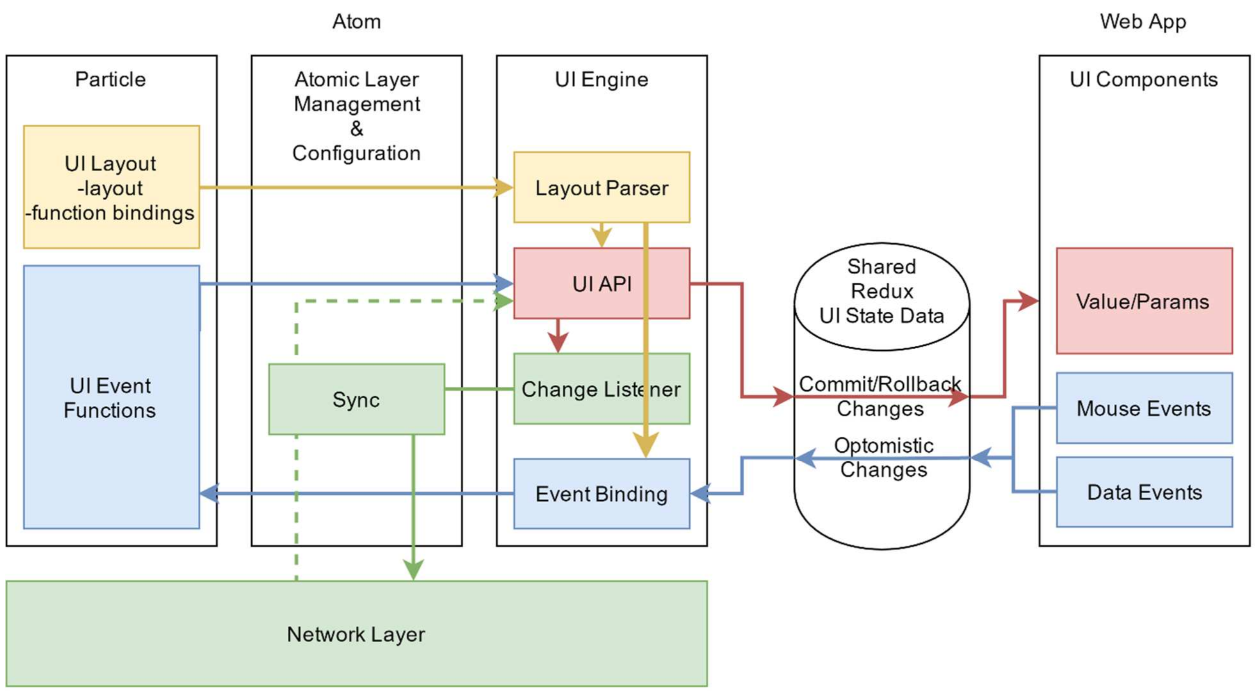

Figure 3: UI Engine Particle & Interactions

4.3 Assessment of Proposed methods

The easiest version of the smart home we could think of was to create a master cloud-based controller for each node. All commands that were given had to come from this single cloud server, each node knew only of the cloud server. But this is not the best solution as a slow internet or a lack of one altogether could cripple the system.

Next we looked at making a single master controller that was hosted locally. As long as the method of communication was still working, the system would not be down. But we foresaw issues with security. A smart home will control a large part of the user's life and may even be hosting files for them, a single point of access (the master controller) and any attacker has complete control over the system. There is no good way to remove the attacker for the common user without a hard reset of the system. This would lead to frustration and bad reviews.

With the idea of a local system with a high degree of security that would also be easy to set up, we decided on a mesh network of computers. Thus, Project Molecule was born. Our implementation has the benefit of a being completely locally hosted, so an internet outage does not leave the house unusable. And the mesh network with communication and security as its key focus, built in a memory safe language, lowers the chance of an attacker gaining access to an entire system via a single attack.

4.4 Validation

The best way to validate Project Molecule is to do a real world test. Our first idea is to retrofit a doll house, or some sort of simulated house. We will add a small speaker and LEDs in certain rooms for testing music playback and lighting control. A monitor will be attached on the side of our demo to emulate the living room media player. Behind the scenes, an external hard-drive with movies will also be connected for the media server. Then we will have a laptop or tablet set up as our UI controller. We can issue commands to the nodes.

5 Project Requirements/Specifications

5.1 Definitions

Project Molecule is composed of many components and arranged similarly to a molecule. As such, naming conventions are drawn from that domain.

Particle Layer: Smallest component of the system are divided into the following types

Electrons: 3rd Party Extensions

Protons: System Resource Wrappers

Neutrons: Device Level Drivers

Atomic Layer: Manages **particle** lifecycle, provides data synchronization, and enforces permissions

Bonding Layer: Networking handles **particle** to **particle** communication and discovery

5.2 Functional

Particles

- shall be isolated from other layers
- shall provide a common package format
- shall be unique (identifiable instance)
- may share configuration (common service type)
- shall implement common **interfaces** for service discovery
- shall advertise implemented **interfaces**
- shall respond to lifecycle events
- may consist of:
    - **Electrons** which
        - Shall not have direct System Access
    - **Neutrons** which
        - Shall encapsulate peripheral devices
    - **Protons** which
        - Shall wrap specific System Resources

Interfaces

- Shall describe features of each **particle** (Switch, Media Player, Audio Player, Outlet Control)

Atoms

- Shall manage **particle** Lifecycle. They:
    - shall ensure messages between **particles** is permitted
    - shall ensure inter-**particle** communication is secured
    - shall ensure only the destination **particle** may read the message
- Shall manage **particle** Configuration:
    - **Atoms** shall synchronize data between other **atoms**
    - **Atoms** shall ensure **particles** have permission to access the configuration
    - Configuration may be partitioned:
        - Each partition shall have independent permissions

Shall store **particle** permissions:

Permissions must be synchronized between all **atoms**

Bonds

Shall facilitates inter-**particle** communication which includes inter-**atom** and **atom** – **atom** communication

Shall route multiple instances of the same **particle**.

Shall have a routing layer which

- Must determine the best **particle** instance to use for a given message

- Must ensure successive requests for a **particle** by the same service will resolve to the same **particle** instance

Must be aware of all **particle** types and instances on the network

Shall provide an API for **particle** registration and discovery

5.3 Non-Functional

The system must have sufficient bandwidth to stream HD (1080p) video

A single node failure must not bring down the entire system

The system must not be down for more than 10 minutes in a given year (0.001% downtime)

A single compromised node should not be able to exploit the entire system

A stable API for extensions to use should exist

Cost of an individual node (Raspberry Pi, storage + power supply) should be no more than $100

Extension API should be fully documented

6 Challenges

We do not expect to face any challenges with costs or materials. Raspberry Pi's are relatively inexpensive and have excellent software support. Any additional equipment will be in the form of small LED lights for indicators and wires to make the connection to GPIO pins. Our team already has these materials in lab kits from previous classes.

The area that may cause us some concern is the knowledge of the area we are working in. The system we are creating has two complicated components to it, the networking layer and the security layer. The networking layer requires that all the nodes in the system can talk to one another without any data race issues. This could be quite difficult to make bug free. The security layer is also a source of complexity since each extension needs to be sandboxed and prevented from doing anything malicious to the system. No one on the team has prior experience with system security so learning enough to correctly implement the security layer could take us some time.

7 Timeline

7.1 First Semester

Milestone #1

- API development (dummy code)
- Bonding/Atomic/Sub-Atomic is one executable
- Hard coded UI

Milestone #2

- Object/class based particles
- Simple service networking router (only one instance of each particle type)
- Compile time extensibility (not runtime)

Milestone #3

- Demonstration

7.2 Second Semester

Milestone #4

Routing Algorithm

- Work towards multi-instanced particles and service discovery

UI

- Work towards markup based UI generation

Resources

- Create basic Neutrons (System Resources)

Milestone #5

Atomic level integration

System Resource/Networking

- Request file resource from System Resource particle

UI/Networking

- pass a markup document to a UI particle

Milestone #6

Particle Package (process based)

Security

Synchronized configuration

Milestone #7

Demo

8 Conclusions

Project Molecule aims to make connected living simple, cost effective, secure, and reliable. The platform is built to be extensible and integrate with new and existing sensors and devices. By sharing common configuration and computing resources, the network can remain operational even after a node failure. These fault tolerant features provide unique challenges to the project but enable us to provide a robust system for home automation.

9 REFERENCES

[1] "Smart Homes," *ZigBee Alliance.* [Online]. Available: http://www.zigbee.org/what-is-zigbee/494-2/. [Accessed: Oct. 18, 2016].

[2] "What is BLE-MESH," *BLE-MESH.* [Online]. Available: http://ble-mesh.com/. [Accessed: Oct. 18, 2016].

[3] "Developing for the IoT? Bluetooth SIG makes it easy with updated line of developer toolkits," *Bluetooth.* [Online]. Available: https://www.bluetooth.com/. [Accessed: Oct. 18, 2016].

[4] "What is the Alexa Skills Kit?," *Amazon.* [Online]. Available: https://developer.amazon.com/alexa-skills-kit. [Accessed: Oct. 18, 2016].