

CPRE 491 WEEKLY REPORT 04

Project Molecule

27 September – 03 October 2016

May1739

may1739@iastate.edu

Dr. Arun Somani

Ryan Wade – Team Leader

Nathan Volkert – Communications Lead

Daniel Griffen – Key Concept Holder

Alex Berns – Webmaster & Scribe

1 CONTENTS

2	Weekly Summary	2
3	Past week accomplishments	2
4	Individual contributions	2
5	Comments and extended discussion	3
5.1	Technical description of layered architecture	3
6	Plan for coming week:.....	5
7	Summary of weekly advisor meeting.....	5

2 WEEKLY SUMMARY

This week we met with our advisor and worked through several use cases for our project. We started brainstorming the architecture of our API, system architecture, and fault tolerance methods

3 PAST WEEK ACCOMPLISHMENTS

All Members:

- Meet to discuss system arch and fault tolerance. Diagrams Below
- Advisor Meeting
- Register for room access

Ryan Wade:

- Investigated Layered Architecture and Created Diagram

Nathan Volkert:

- Meeting minutes, Communicated and researched getting parts (pi 3s)

Daniel Griffen

- Created description of program architecture

Alex Berns

- Created report
- Meeting minutes

4 INDIVIDUAL CONTRIBUTIONS

NAME	Hours this week	Hours cumulative
Ryan Wade	5	15
Nathan Volkert	4	11
DanielGriffen	4	11
Alex Berns	4	11

5 COMMENTS AND EXTENDED DISCUSSION

5.1 TECHNICAL DESCRIPTION OF LAYERED ARCHITECTURE

The Program is structured into three logical “layers” (see figure 1). The extension layer, the permissions/security layer, and the routing/discovery layer. Each layer has clearly defined responsibilities and limitations. The extension layer is where smart-home features will be implemented, for example: a lightbulb controller or a media server. Extensions have very limited access to system resources, their only option for accessing resources is to request access through the permissions layer.

The permissions layer sits between the extensions and the routing layer. The permissions layer is responsible for ensuring that components of the system are only given access to what they have permission for.

The routing layer is what handles communication between nodes. It is the only layer that has knowledge of or can talk to other nodes in the system. Similar to the permission layer, all requests must go through this layer at some point during the execution of a task.

A standard request would follow a pattern similar to the following. A media player extension needs access to a music file to play. It makes a request to the permissions layer for the file. The permissions layer then verifies that the extension has the correct privileges to ask for that file. The permissions layer then asks the network layer to find the requested file. The network layer determines what node the file is on and sends a request to that node for the file. The network layer on the other node receives the request and sends it through its permission layer. The permission layer on this node checks to make sure that the other node has permission to access this file. The permission layer then asks a system resource extension for a handle to the file. This file handle is then sent back to the original node, where it passes down the layers to the media player extension that originally requested the file.

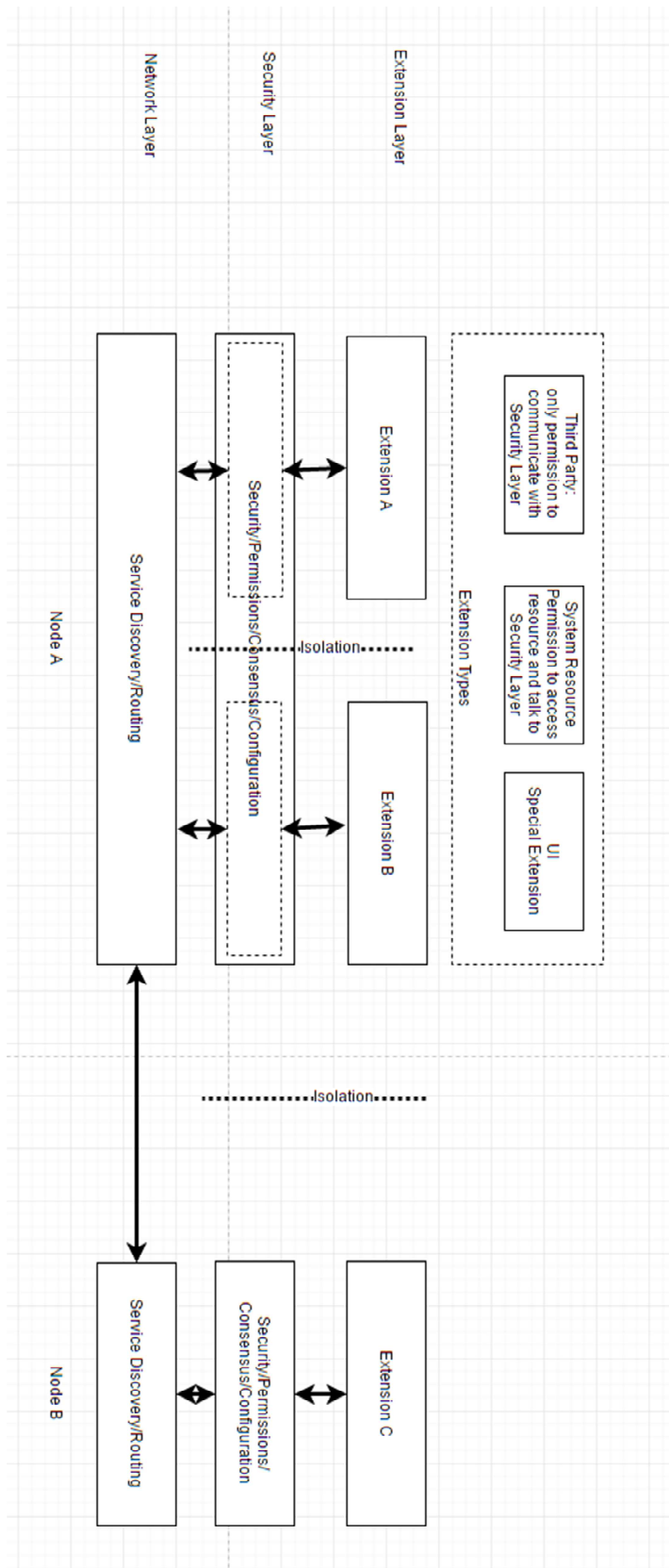


Figure 1: Layered Architecture

6 PLAN FOR COMING WEEK:

All Members:

- Continue refining API support (layered approach?)
- Continue refining System Architecture
- Consider possible alternatives for fault tolerance

7 SUMMARY OF WEEKLY ADVISOR MEETING

Met with advisor to discuss system architecture plans and fault tolerance.