ANDROID • iOS • WINDOWS PHONE • BLACKBERRY

SMART HOME

making your smart home smarter!

JASON POLLMAN • MICHAEL BATTALGIA • LAWRENCE ISMAIL • PATRICK SINNOTT

THE SMARTHOME PROJECT

JASON POLLMAN • MICHAEL BATTALGIA • LAWRENCE ISMAIL • PATRICK SINNOTT

TEAM 9

WHY SMARTHOME?

Creating a smart home means using a lot of device specific apps...
Philips Hue app, Belkin WeMo app, etc. etc.

or...

Pay Time Warner to setup your home with their equipment.
Monthly service charge? No Thanks.

The SmartHome Project allows users to use a single mobile application to control all of their smart devices from anywhere.

PROJECT OVERVIEW

Our project is motivated by the rapidly growing need for home automation in a new, emerging smart device market.

As consumers become intrigued by the concept of a *smart home*, the need for a *generic* mobile application to control different and multiple smart devices is becoming apparent.

Most manufactures create their own standalone applications to control their devices rather than conforming to a standard—and currently there isn't one. Consumers are forced to choose between using a single manufacturer's product line from which to purchase their smart devices, or use an unnecessary number of applications to manage devices from different manufactures.

The SmartHome project aims to ease the pain of operating a smart home by providing a single, central application to control and manage various "smart" home devices.

SmartHome interfaces with and controls various environmental agents such as thermostats, light sensors, sound sensors, power switches, and more—and it does this based on the users currently logged in to a home's local network and the users' preferences via communication with our "SmartHome API."

SMART API, SMART DESIGN

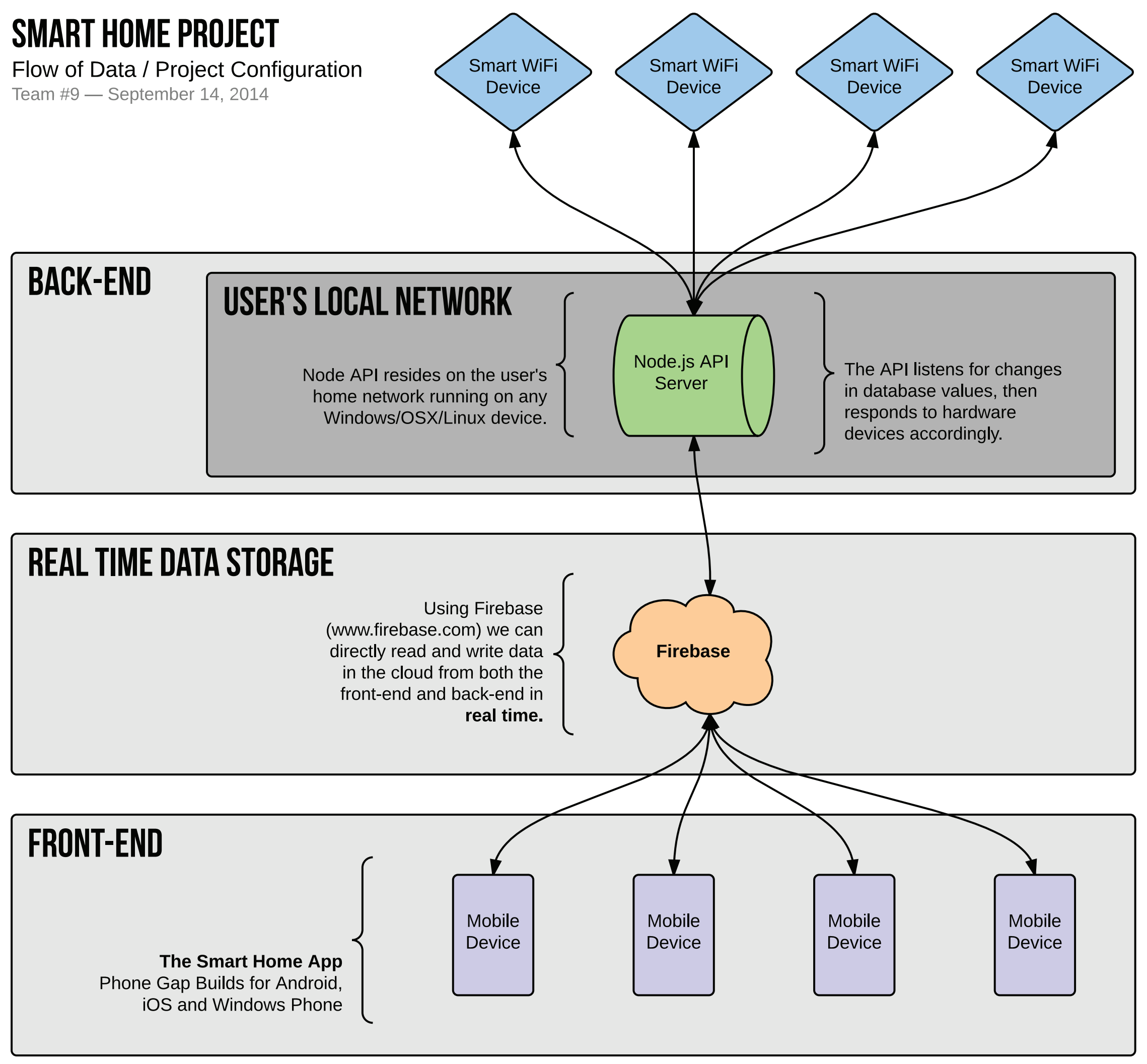
INTUITIVE & COMPATIBLE

The SmartHome API receives requests from users in the SmartHome App, interprets and forwards them to the particular smart device to modify, and then responds to the user with the device's response.

It's that simple. By acting as the "middle man," SmartHome allows manufacturers to focus on what they do best: hardware. Their typically crude and 'thrown' together apps can now replaced by SmartHome!

Adding or updating devices for SmartHome compatibility is as easy as writing a single device driver—about 500 lines of code on average.

Our API generalizes each smart device's *manufacturer specific* API into a generic one; this means that any Wi-Fi smart device with an open API can be controlled via our app by writing a simple manufacturer and model specific "device controller"—JavaScript Objects that implement code for specific methods that define the operations a device can perform.

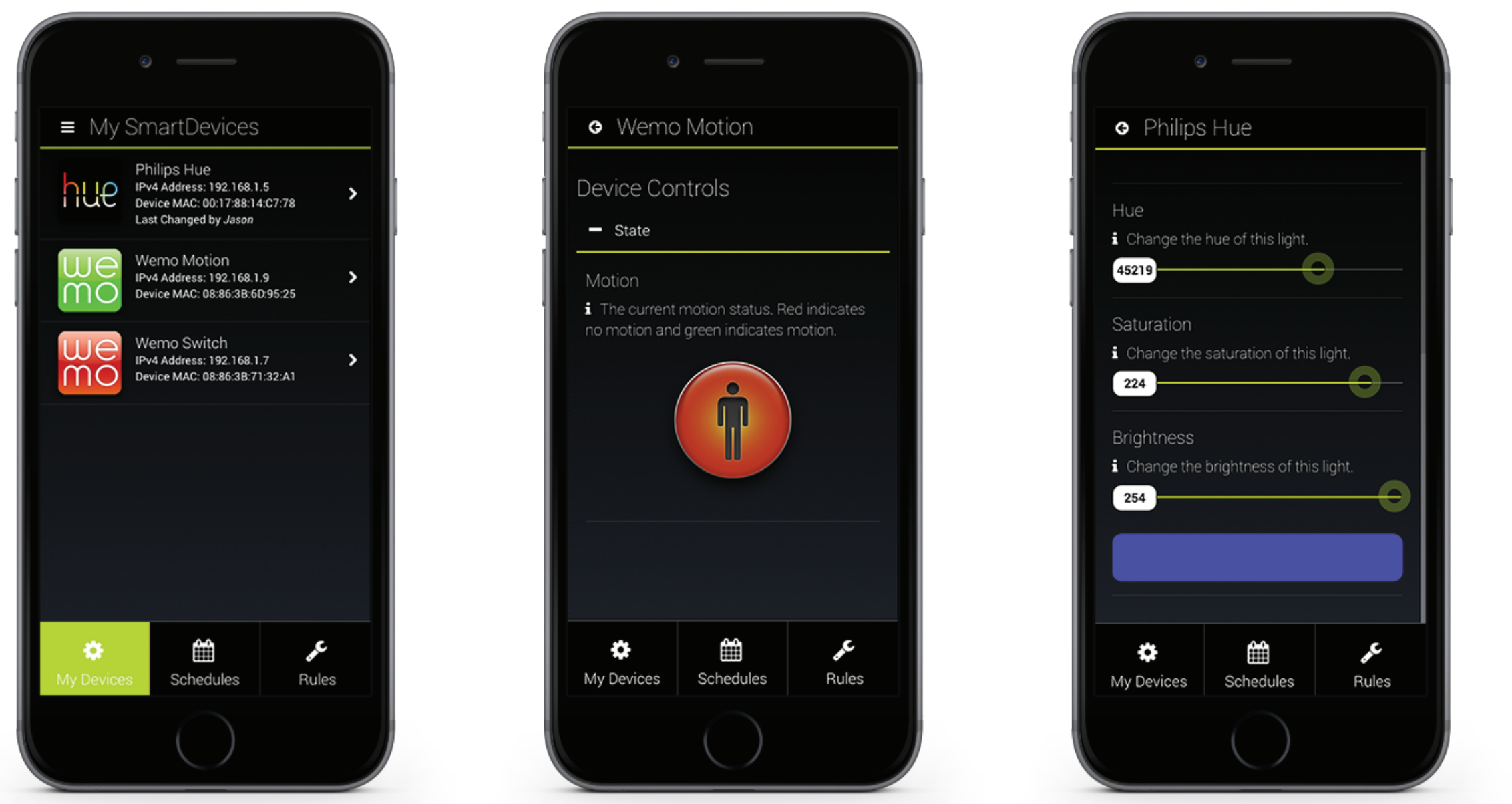

SMART HOME PROJECT
Flow of Data / Project Configuration
Team #9 — September 14, 2014
Smart WiFi Device
Smart WiFi Device
Smart WiFi Device
Smart WiFi Device
BACK-END
USER'S LOCAL NETWORK
Node API resides on the user's home network running on any Windows/OSX/Linux device.
Node.js API Server
The API listens for changes in database values, then responds to hardware devices accordingly.
REAL TIME DATA STORAGE
Using Firebase (www.firebase.com) we can directly read and write data in the cloud from both the front-end and back-end in real time.
Firebase
FRONT-END
The Smart Home App
Phone Gap Builds for Android, iOS and Windows Phone
Mobile Device
Mobile Device
Mobile Device
Mobile Device

THE FRONT-END

The SmartHome App utilizes the power of jQuery Mobile and PhoneGap together, intuitively.

Why PhoneGap and not a native app? It's Business. PhoneGap allows us to develop for a wider audience, across multiple mobile platforms. If the product takes off, then we start thinking native! Until then, exposure is the key to success.

However, by combining jQuery Mobile and PhoneGap together, we've achieved a beautiful and fluid AJAX powered application with the look and feel of a native app without any of the vendor specific hassle.

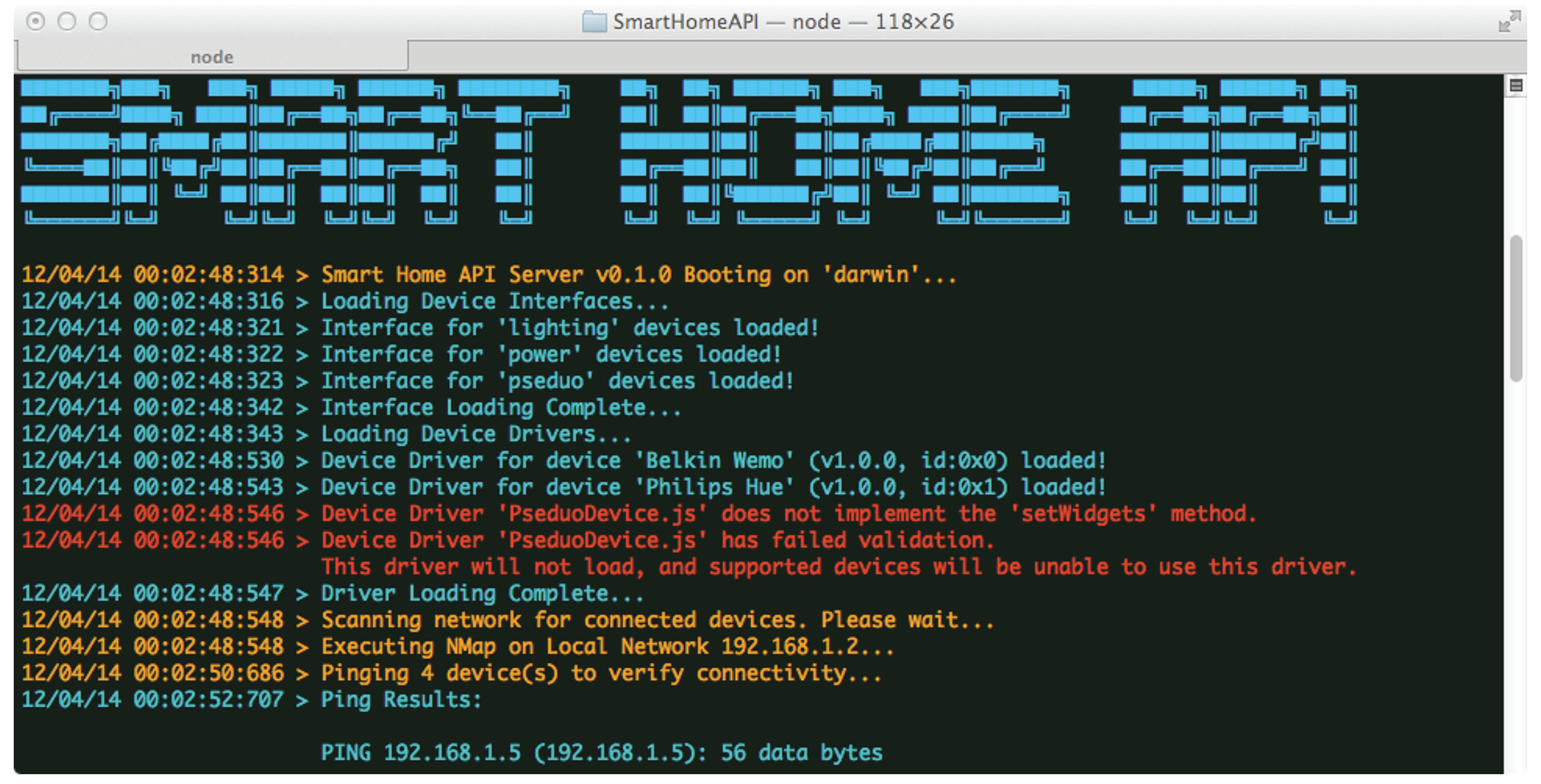

My SmartDevices
Philips Hue
IPv4 Address: 192.168.1.5
Device MAC: 00:17:88:14:C7:78
Last Changed by Jason
Wemo Motion
IPv4 Address: 192.168.1.9
Device MAC: 08:86:3B:6D:95:25
Wemo Switch
IPv4 Address: 192.168.1.7
Device MAC: 08:86:3B:71:32:A1
My Devices
Schedules
Rules
Wemo Motion
Device Controls
State
Motion
The current motion status. Red indicates no motion and green indicates motion.
My Devices
Schedules
Rules
Philips Hue
Hue
Change the hue of this light.
45219
Saturation
Change the saturation of this light.
224
Brightness
Change the brightness of this light.
254
My Devices
Schedules
Rules

THE BACK-END

Dynamite comes in small packages! Just like Node.js.

Our SmartHome API is written in JavaScript, utilizing the power of Node. By using Node.js, we've developed a scalable and modular back-end with small, loosely coupled components.

Below: The SmartHome API booting. The API has scanned the user's local network and added all supported smart devices to the user's profile for user interaction.



```
12/04/14 00:02:48:314 > Smart Home API Server v0.1.0 Booting on 'darwin'...
12/04/14 00:02:48:316 > Loading Device Interfaces...
12/04/14 00:02:48:321 > Interface for 'lighting' devices loaded!
12/04/14 00:02:48:322 > Interface for 'power' devices loaded!
12/04/14 00:02:48:323 > Interface for 'pseduo' devices loaded!
12/04/14 00:02:48:342 > Interface Loading Complete...
12/04/14 00:02:48:343 > Loading Device Drivers...
12/04/14 00:02:48:530 > Device Driver for device 'Belkin Wemo' (v1.0.0, id:0x0) loaded!
12/04/14 00:02:48:543 > Device Driver for device 'Philips Hue' (v1.0.0, id:0x1) loaded!
12/04/14 00:02:48:546 > Device Driver 'PseduoDevice.js' does not implement the 'setWidgets' method.
12/04/14 00:02:48:546 > Device Driver 'PseduoDevice.js' has failed validation.
                        This driver will not load, and supported devices will be unable to use this driver.
12/04/14 00:02:48:547 > Driver Loading Complete...
12/04/14 00:02:48:548 > Scanning network for connected devices. Please wait...
12/04/14 00:02:48:548 > Executing NMap on Local Network 192.168.1.2...
12/04/14 00:02:50:686 > Pinging 4 device(s) to verify connectivity...
12/04/14 00:02:52:707 > Ping Results:

                        PING 192.168.1.5 (192.168.1.5): 56 data bytes
```

ASYNC ARCHITECTURE

Node.js' event-driven design model has enabled us to write a lightning fast, 'non-blocking', and asynchronous back-end.

This means, controlling multiple smart devices at once can be done without performance lag.

REAL-TIME, CLOUD BASED DATA

Users don't have the patience to wait even a few seconds for their lights to click on or off. So we've used a dynamic, real-time object database system: Firebase.

Firebase is at the heart of the SmartHome Project—and it's real time, cloud based data processing is the key.

We've utilized a data-centric design; user requests are sent to our Firebase app, our API receives them, talks to smart devices and pushes the results back to Firebase, and finally the app picks up the results. It's a fast and simple solution to a complex problem.

Also, storing user configuration in the cloud means that users can control their smart devices from anywhere they have an internet connection!

ANDROID • iOS • WINDOWS PHONE • BLACKBERRY

SMART HOME

making your smart home smarter!

JASON POLLMAN • MICHAEL BATTALGIA • LAWRENCE ISMAIL • PATRICK SINNOTT

node JS

making your smart home smarter!

jQuery mobile