Brillo OS: Implementing Miniature Smart Home

Constantin Musca – Intel Corporation
Outline

I. Context
II. Problem statement
III. Solution
IV. Implementation details
V. Conclusions
VI. Future work
Context
Problem statement

- IoT Standardization and compatibility
- Device integration
- Open devices and products
What is Brillo?

- Embedded OS
- Core services
- Developer kit

Source: Google
The Brillo Operating System

- Open Source
- Maintained
- Supports devices with a small footprint
- Customizable

Source: Google
The Brillo Core Services

- Weave
- Metrics
- Crash reporting
- OTAs

Source: Google
The Brillo Developer Kit

- Based on the Android.mk architecture
- Standard testing
- Android debug bridge (adb)

Source: Google
Weave

- Protocol for device discovery, provisioning, authentication, and interaction
- Schema Driven (JSON)
- OAuth 2.0 Authentication

Source: Google
Edison Arduino board

- High performance dual core Intel CPU
- Wi-Fi antenna within the Edison chip
- 4GB EMMC storage
- Support for USB, I2C, SPI & GPIOs

Source: intel.com
Philips Hue RGB Bulb

- High quality light that you can dim smoothly
- Large range of colors
- Away-from-home control
- Smart control through an open REST interface

Source: www.newscenter.philips.com
Nest thermostat

• Automatically adapts to life and season changes
• Helps saving energy
• Smart control through an open REST interface

Source: nest.com/press
Grove Starter Kit

- Relays
- Motors
- Buzzers
- Temperature Sensor
- Light Sensor
- LEDs
- Buttons
Product Design

Bedroom 2

Bedroom 1

Garage

Living Room

Alarm system based on sensors

Hardware System Connections

Software System Architecture

- Weave Enabled Applications
  - Android Application
  - Weave Developer Console

- SHWeaveService

- SHBinderService
  - Philips Hue Support
  - Nest Cloud Support
  - Grove Starter Kit Support

- Weave

- Binder

- Brillo OS

- SHBinderClient
Implementation: SHWeaveService

- Connects to the SHBinderService via Binder
- Acts as a middleman for all Weave – SHBinderService communication
- Maintains and updates the Weave device state

Source: developers.google.com/weave
Implementation: Weave Commands Schema

```
"commands": {
    "set_bedroom_1_door_state": { ... } 
    ...
    "set_garage_door_state": { ... },
    "set_target_temperature": {
        "parameters": {
            "temperature": {
                "type": "number"
            }
        }
    },
    "set_light": {
        "parameters": {
            "onState": {
                "type": "string"
            },
            "saturation": { 
                "type": "number"
            },
            "brightness": { 
                "type": "number"
            },
            "hue": {
                "type": "number"
            }
        }
    }
}
```

Source: developers.google.com/weave
Implementation: Weave State Schema

```
"state": {
    "bedroom_1_door": {
        "type": "string",
        "enum": ["locked", "unlocked"]
    },
    "garage_door": {
        "type": "string",
        "enum": ["opened", "closed"]
    },
    "nest_thermostat": {
        "target_temperature": {
            "type": "number"
        },
        "current_temperature": {
            "type": "number"
        }
    },
    "philips_hue_light": { ... }
}
```

Source: developers.google.com/weave
Implementation: SHBinderService

- Exposes all the connected devices through a Binder interface
- Interacts with the Philips Hue & Nest thermostat via a HTTP client
- Exposes the I2C connected sensors via the Sensors HAL
- Uses libmraa to control the GPIOs and libupm for the motors
Implementation: SHBinderService details

SHBinderService
- getLight();
- setLight(bri, sat, hue);
- getTemperature();
- setTemperature(target);
- getDoorState(id);
- setDoorState(id);
- ...

PhilipsHue
- getLight();
- setLight(bri, sat, hue);

NestThermostat
- getTemperature();
- setTemperature(target);
- getHumidity();

HomePeripherals
- getDoorState(id);
- setDoorState(id);
- ...
Implementation: SHBinderService AIDL interface

```java
interface ISHBinderService {
    boolean set_bedroom1_door_state(String state);
    ...
    boolean set_garage_door_state(String state);
    double get_temperature();
    boolean set_target_temperature(double temp);
    void set_light(boolean on, double saturation,
        double brightness, double hue);
    ...
}
```

Source: pixabay.com
### Example for changing the light bulb state

<table>
<thead>
<tr>
<th>Address</th>
<th>http://&lt;bridge ip address&gt;/api/&lt;user&gt;/lights/1/state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>{&quot;on&quot;:true,&quot;sat&quot;:230,&quot;bri&quot;:180,&quot;hue&quot;:10000}</td>
</tr>
<tr>
<td>Method</td>
<td>PUT</td>
</tr>
</tbody>
</table>

Source: www.newscenter.philips.com
Example for changing the thermostat target temperature

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>{&quot;target_temperature_c&quot;: 22.5}</td>
</tr>
<tr>
<td>Method</td>
<td>PUT</td>
</tr>
</tbody>
</table>

Source: nest.com/press
Grove Starter Kit Support

- SHBinderService
- Sensors HAL
- libmraa/upm
- Kernel
- Sensors
- Motors, Relays, LEDs & Buzzers
Conclusions

Brillo OS
Smart Home

- Easy
- Secure
- Flexible
- Extensible
Future work

- Real life project implementation
- Add intelligence to the house
- Integrate more devices

Source: www.flickr.com/photos/thecourtyard/4213586581
Legal Notices and Disclaimers

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

© 2016 Intel Corporation.