



# Brillo/Weave Part 1: High Level Introduction

Open IoT Summit

Bruce Beare

April 2016



# Introduction to Brillo\* and Weave\*

- Introduction to Brillo
- Introduction to Weave
- Brillo Security Features
- Source Code Structure
- Intel Starter Boards
- Next Steps



# Brillo/Weave

## Brillo: Google's OS for IoT MPU devices

- Targeted at smart homes
- Expanding to buildings and industry
- Supports MPU devices w/ min 35MB of RAM.

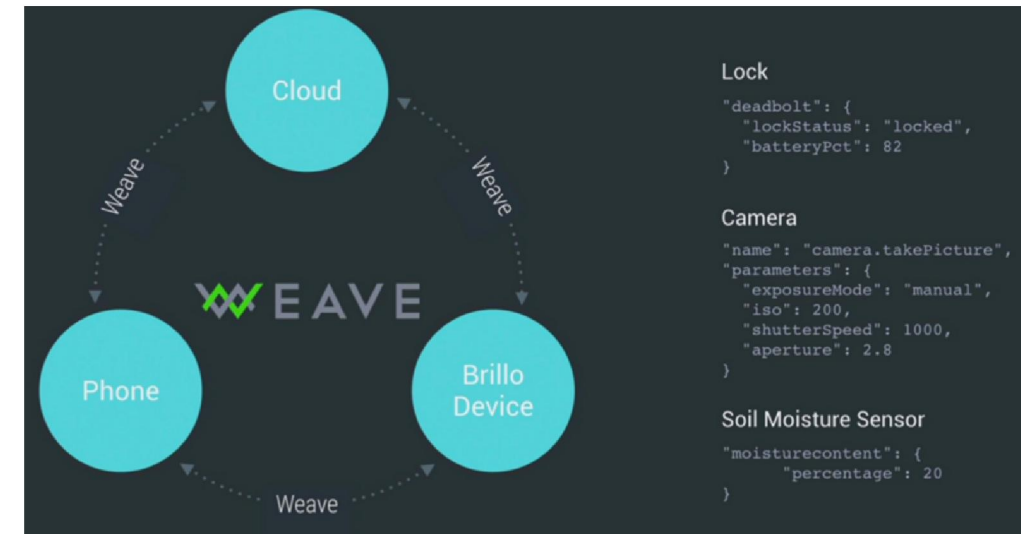
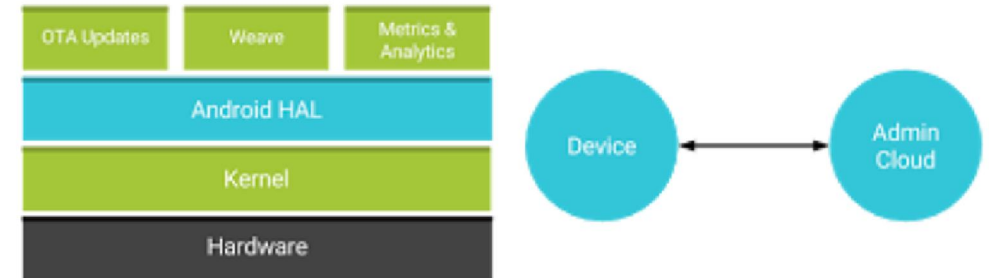
## Weave: An open communications protocol

- Devices can connect and interact with one another, the internet, and your phone
- Ensures that access to your device and user data is secure and private.

## Libuweave: A tiny Weave implementation

- Communications Stack intended to be portable to MCU based devices

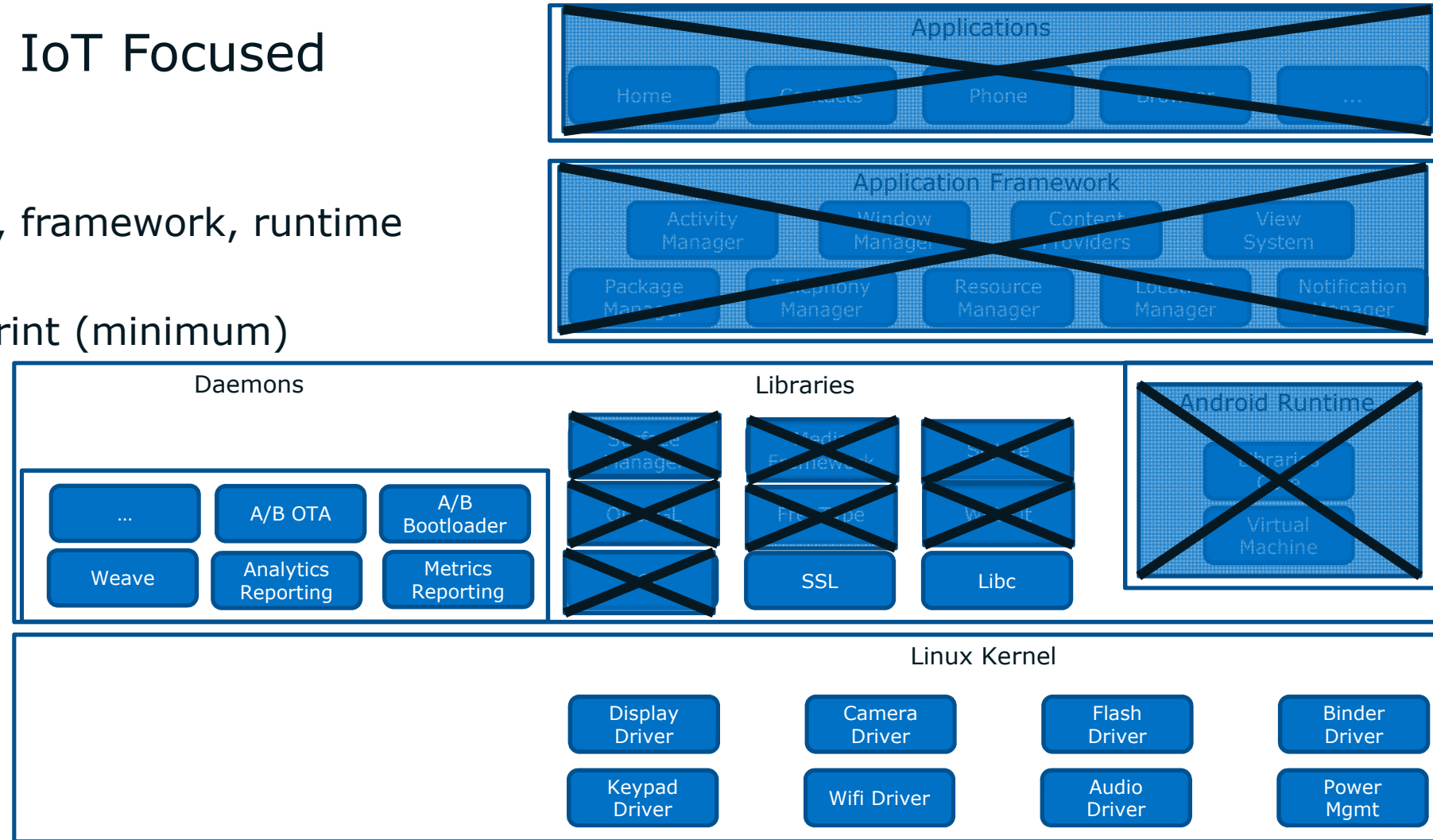
## Brillo OS



# Brillo: an Operating System based on Android

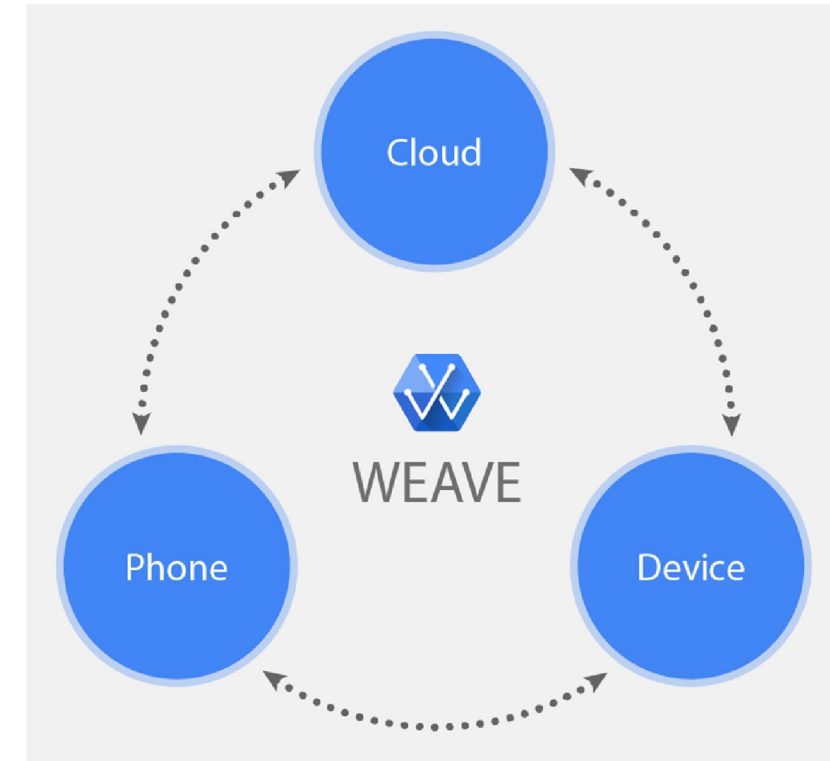
Simpler... Smaller... IoT Focused

- C/C++ environment
- Binder IPC
- No Java Applications, framework, runtime
- No Graphics
- 35MB Memory Footprint (minimum)



# Weave\* is..

- A communications platform for IoT devices
- Device setup, phone-to-device-to-cloud communication
- User interaction from mobile devices and the web
- Transports: 802.15.4 (zigbee, threads), BLE, WiFi, Ethernet, Others possible
- Schema Driven (json)  
Associates Weave XMPP requests with application function invocations
- Web apps may be written with Google\* API support
- OAuth 2.0 Authentication, Google as AS



# Weave\* Device Schema

- Device manufacturers are encouraged to support a standardized schema for their device (toaster, fridge, thermostat, ...)
- IHV may extend the schema
- The user will be able to share control of their devices. Friends, 3rd party Weave (web) applications
- Cloud based apps will better integrate all of a user's devices over time

*Google Weave*

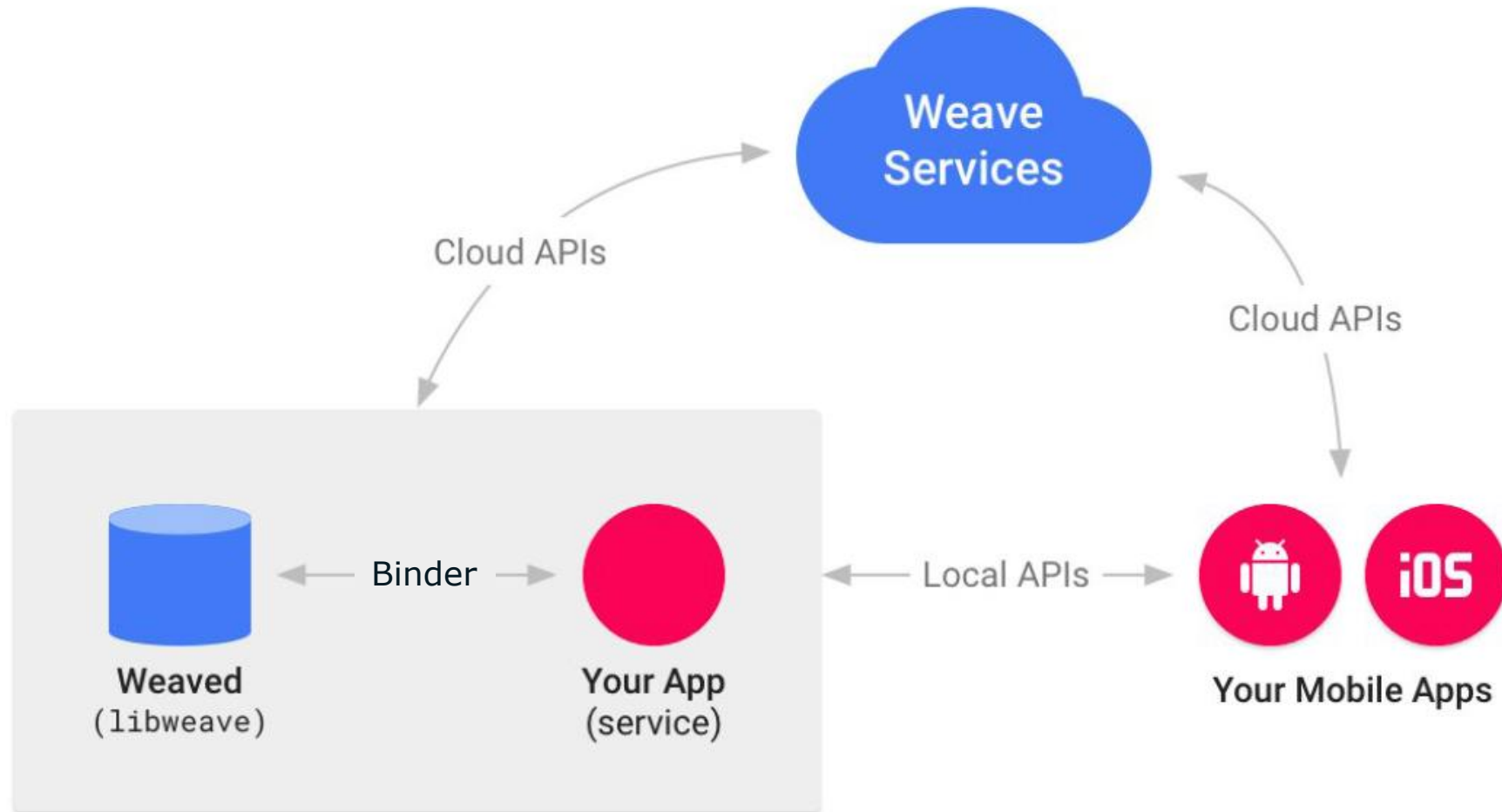
*JSO Alliance*

*DeviceFIVE*

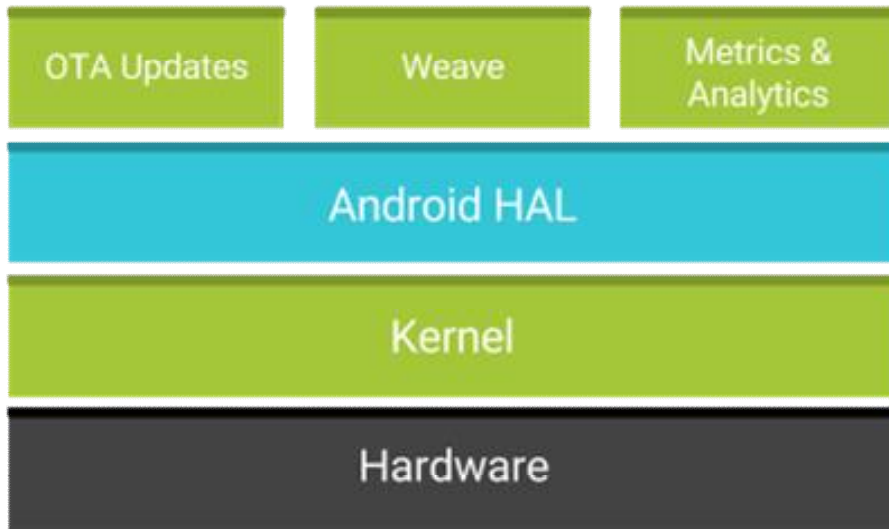
*Allentia*

*LoJivity*

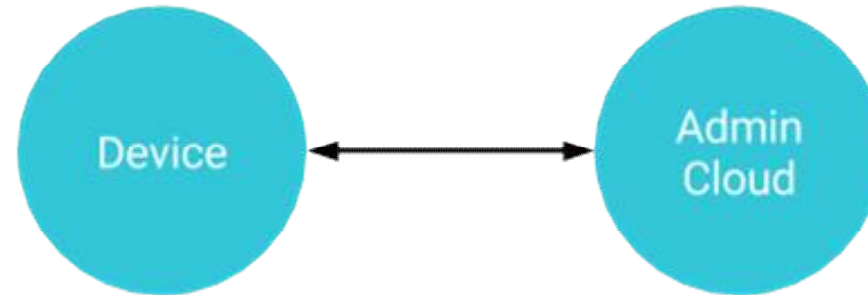
# Weave\* command/data flow



# Brillo Devices are Administered via the Google Cloud



- View Device Status
- Rename Devices
- Share Devices with apps, friends
- Send commands to Devices



FAQ: Do I have to use Google's Cloud?

There's nothing to prevent you from modifying Weave to use a different cloud service; but there's nothing to make this easy either. It is unlikely that Google will release the cloud-side sources.

You could certainly add a completely different cloud back-end as well and develop a product with the Brillo OS but without Weave.



# Registering (provisioning) a Brillo\* Device

Connect to the Brillo's WiFi AP via a Mobile Device

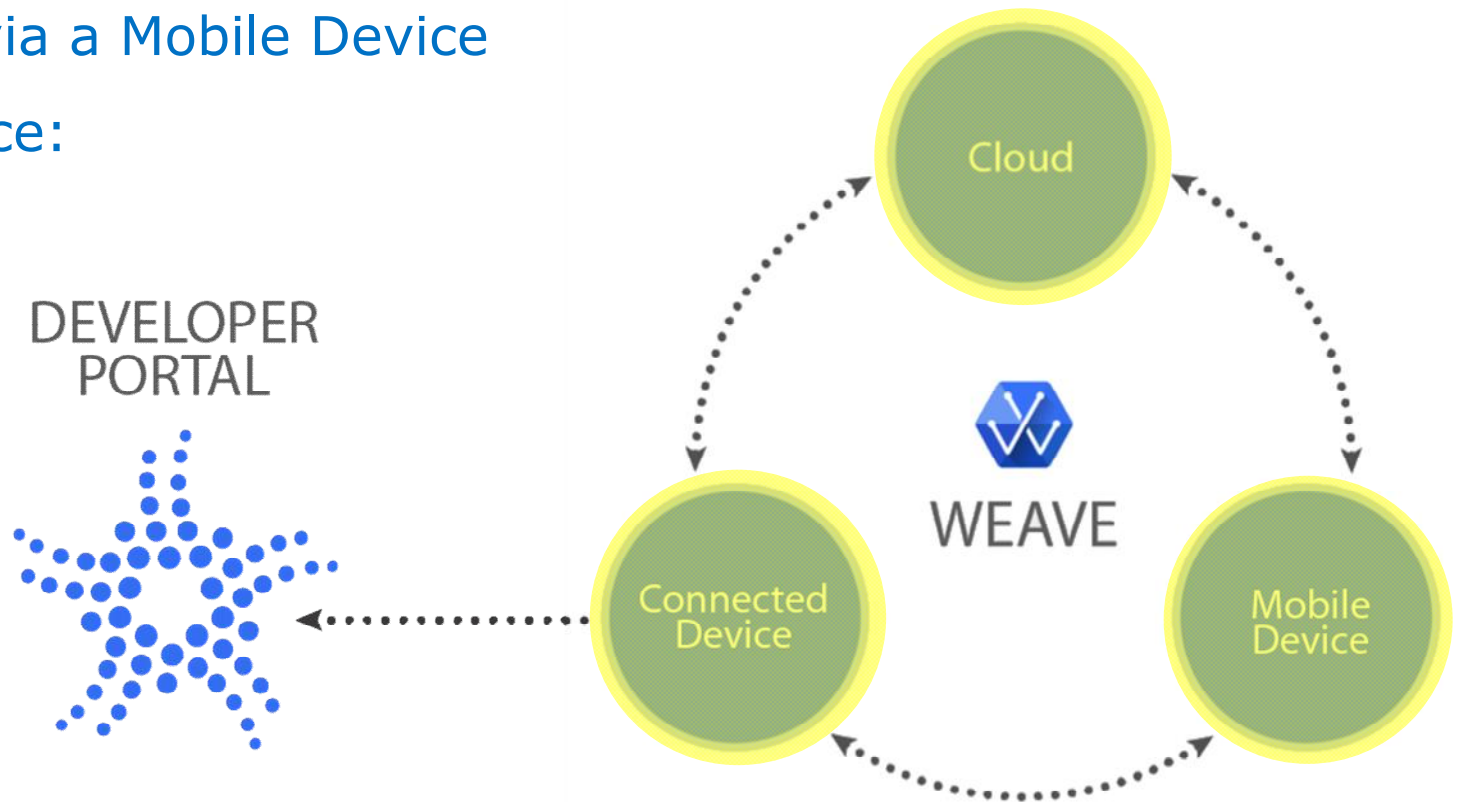
Provide information to the Device:

- OAuth\* 2.0 identity
- WiFi Access Point
- PIN Code

Device contacts the Cloud

- Downloads updates

The Device is Ready



# Brillo Security Features



# Brillo\* Security Features & Requirements

Requirements are largely the same as Android:

- Google\* Verified Boot
- HW Keymaster 1.0 (but limited to 2k keys)
- Cryptographic Chain of trust
- SE-Linux\* Enforcement
- Limit root processes

# Weave\* Security Features

Weave provides link-level security and data privacy:

- SSL/TLS for Link Level Data privacy
- Device and Application access may be shared with friends or apps
- OAuth\* 2.0 used for Authentication, crypto Certificates
- Google\* provides the Authentication Server

# Brillo vs. Ostro

	Brillo	Ostro
Based on (Kernel)	Linux	Linux
Based on (Userland)	Android	Yocto
Over the Air Updates	Yes	Yes, but needs a cloud service to support
Compatibility with Ubuntu, Fedora Source Packages	Moderate (bionic library, build system)	High
Compatibility with Android Source Packages	Yes	No
IPC Mechanism	Binder	D-BUS, Linux IPC
Open Source	Yes (AOSP)	Yes (github)

# Weave vs. OCF/IoTivity

	Weave	IoTivity
Existing/Available Cloud Infrastructure	Google	No
Open Source Cloud Infrastructure	No	Yes
User Data Privacy	Yes for transport layer. Yes for Local Link. No End-to-End through Cloud.	Yes for transport layer. Yes for Local Link. End-to-End in design.
Certificate based Authentication	Yes	No
Able to self-organize a network	No	Yes
Request/Response Protocol	HTTPS (tcp)	CoAP (tcp, udp)
Multi-OS, Multi-Platform	Limited	Yes
Gateway Support to non-native devices	Planned	Yes
Certification Available	Yes	Yes

1011101011101010101101  
01010111101000101010101  
01010101010110101111001  
1010101110101010101010

Brillo\* Code Structure





# Brillo\* in the AOSP (Android\*) Tree

## Golden Rule

- Google\* projects are upstream first

## Board Support Package

- vendor/bsp/intel  
*Pre-builts*  
*Externally sourced, not part of AOSP*
- hardware/bsp/intel  
*HALs, SOC Definitions*
- device/intel/<board>  
*Board Definitions*
- hardware/bsp/kernel/intel/\*  
*Transition to a google common kernel*

Android AOSP Projects: 506

Brillo AOSP Projects: 190

Projects common to both: 146

Projects Unique to Brillo: 42

- device/generic/brillo
- platform/external/gentoo/integration
- platform/external/gentoo/overlays/gentoo
- platform/external/gentoo/portage
- platform/external/shflags
- platform/system/peripheralmanager
- platform/tools/bdk
- product/google/common
- product/google/example-ledflasher
- platform/hardware/bsp/kernel/common/v4.4
- kernel/common
- device/intel/edison
- platform/hardware/bsp/intel
- platform/hardware/bsp/kernel/intel
- device/qcom/dragonboard
- platform/hardware/bsp/qcom
- platform/hardware/bsp/kernel/qcom



# Brillo\* BDK (board development kit)

## Procedure:

1. Setup your Workstation (Ubuntu\*)
2. Download the BDK
3. *brunch bsp download edison*
4. Structure your project/product
5. *brunch product build*
6. *brunch provision*
7. *brunch product tool fastboot reboot*
8. *brunch fastboot tool adb shell*
9. Register your device with the cloud

The *brunch* command is a python front-end for many of the product development tasks.

```
brunch product {build,gdb,provision,envsetup,tool,create} ...
    build      Build a product project from the current directory
    gdb        Run gdbclient.py for a given product
    provision   Run provision-device for a given product
    envsetup   Emit the environment for the product in the CWD
    tool       Run a host tool for a given product
    create     Create a product project in the current directory

brunch bsp {status,update,install,download,refresh,list} ...
    status     Get detailed information on the current status of a
               device BSP.
    update     Update an existing BSP, or Download a new BSP.
    install    Alias, see brunch bsp update.
    download   Alias, see brunch bsp update.
    refresh    Refresh an existing BSP. Removes and re-downloads all
               packages for a specified device.
    list       List available BSPs, and their current installation
               status.
```

# Brillo\* BSP Structure and Contents

hardware/bsp/<vendor>

- SoC Definition .mk files
- init \*.rc files for the SoC, HALs
- SELinux Policies
- HAL Source Code
- Tool Source Code
- Library Source Code
- Example Source Code

```
/home/bjbeare/BOSP/hardware/bsp/intel
soc
├── baytrail
├── chv
│   └── sepolicy
├── common
│   ├── bootctrl
│   └── tools
├── edison
│   └── sepolicy
```

```
/home/bjbeare/BOSP/hardware/bsp/intel
peripheral
├── audio
│   └── generic
├── bluetooth
│   └── bcm43340
├── displays
│   ├── Grove_LCD_RGB
│   └── SparkFun_OLED
├── examples
│   └── mraa
├── libmraa
│   ├── api
│   ├── cmake
│   ├── docs
│   ├── examples
│   ├── include
│   ├── src
│   └── tests
├── libupm
│   ├── cmake
│   ├── docs
│   ├── doxy
│   ├── examples
│   └── src
├── light
│   └── mraa
├── sensors
│   └── mraa
├── wifi
│   ├── bcm43340
│   ├── iwlwifi
│   └── rtl8192cu
```

# Brillo\* BSP Structure and Contents

device/<vendor>/<board>

Board Specific Files - defines the lunch target

- Product Structuring \*.mk files
- init \*.rc scripts for the board
- Flashing scripts
- File System Mount Table
- Partitioning Table
- SELinux Policy files

```
/home/bjbeare/BOSP/device/intel
edison
├── AndroidBoard.mk
├── base_product
│   ├── AndroidProducts.mk
│   ├── devicesetup.sh
│   ├── edison.mk
│   └── weaved.conf
├── BoardConfig.mk
├── build
│   └── tasks
│       └── flashfiles.mk
├── flash_tools
│   ├── brillo-flashall-edison.bat
│   ├── brillo-flashall-edison.sh
│   ├── brillo-provision-edison.sh
│   ├── FlashEdison.json
│   ├── LICENSE.txt
│   └── README
├── fstab.device
├── gpt.ini
├── sepolicy
│   ├── file_contexts
│   └── sensorservice.te
```

# Brillo\* BSP Structure and Contents

vendor/bsp/<vendor>

## Prebuilt Binaries

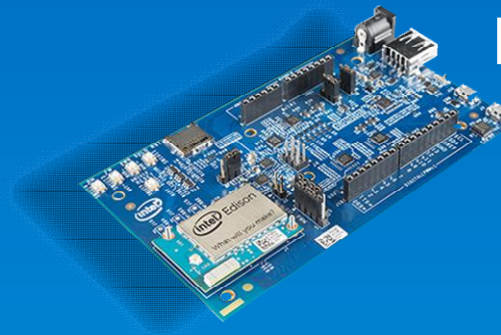
- Downloaded via the Brunch command
- Not in an AOSP Repository
- Bootloader binaries
- Microcode/firmware
- BIOS/firmware

Include information on how to rebuild the firmware where applicable.

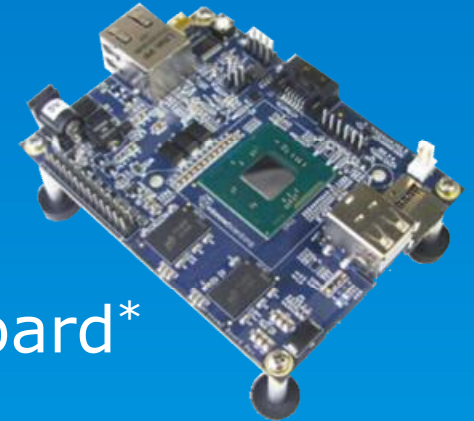
```
/home/bjbeare/BOSP/vendor/bsp
intel
├── edison
├── efibootmgr
├── efibootmgr
├── efivar
├── gnu-efi_prebuilt
├── i2c-tools
├── iniparser
├── kernelflinger
├── pciutils
├── peripheral
├── sbsigntool
├── uefi_shim
└── userfastboot
```

```
/home/bjbeare/BOSP/vendor/bsp/intel
edison
├── ifwi_firmware
│   ├── edison_dnx_fwr.bin
│   ├── edison_dnx_osr.bin
│   ├── edison_ifwi-dbg-00.bin
│   ├── edison_ifwi-dbg-01.bin
│   ├── edison_ifwi-dbg-02.bin
│   ├── edison_ifwi-dbg-03.bin
│   ├── edison_ifwi-dbg-04.bin
│   ├── edison_ifwi-dbg-05.bin
│   ├── edison_ifwi-dbg-06.bin
│   └── LICENSE.txt
├── uboot_firmware
│   ├── LICENSE.txt
│   ├── README
│   ├── u-boot-edison.bin
│   └── u-boot-edison.img
└── peripheral
    ├── bluetooth
    │   └── bcm43340_firmware
    └── wifi
        ├── bcm43340_firmware
        ├── iwlwifi_firmware
        └── rtlwifi_firmware
```

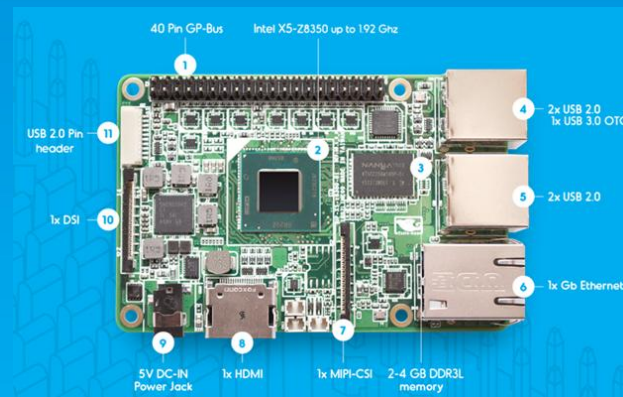
# Brillo\* Starter Boards from Intel



Edison + Arduino\*



MinnowBoard\*



UP Board (CherryTrail)



Edison + SparkFun\*

# The Intel® Edison Kit for Brillo\*

- The Intel® Edison kit with the Arduino\* Expansion Board was the first Brillo Starter Board
- Intel is providing the board support package
- MANY sensors and devices are available for prototyping with this board
- Brillo Starter Board
- Supported in AOSP since Nov '15.

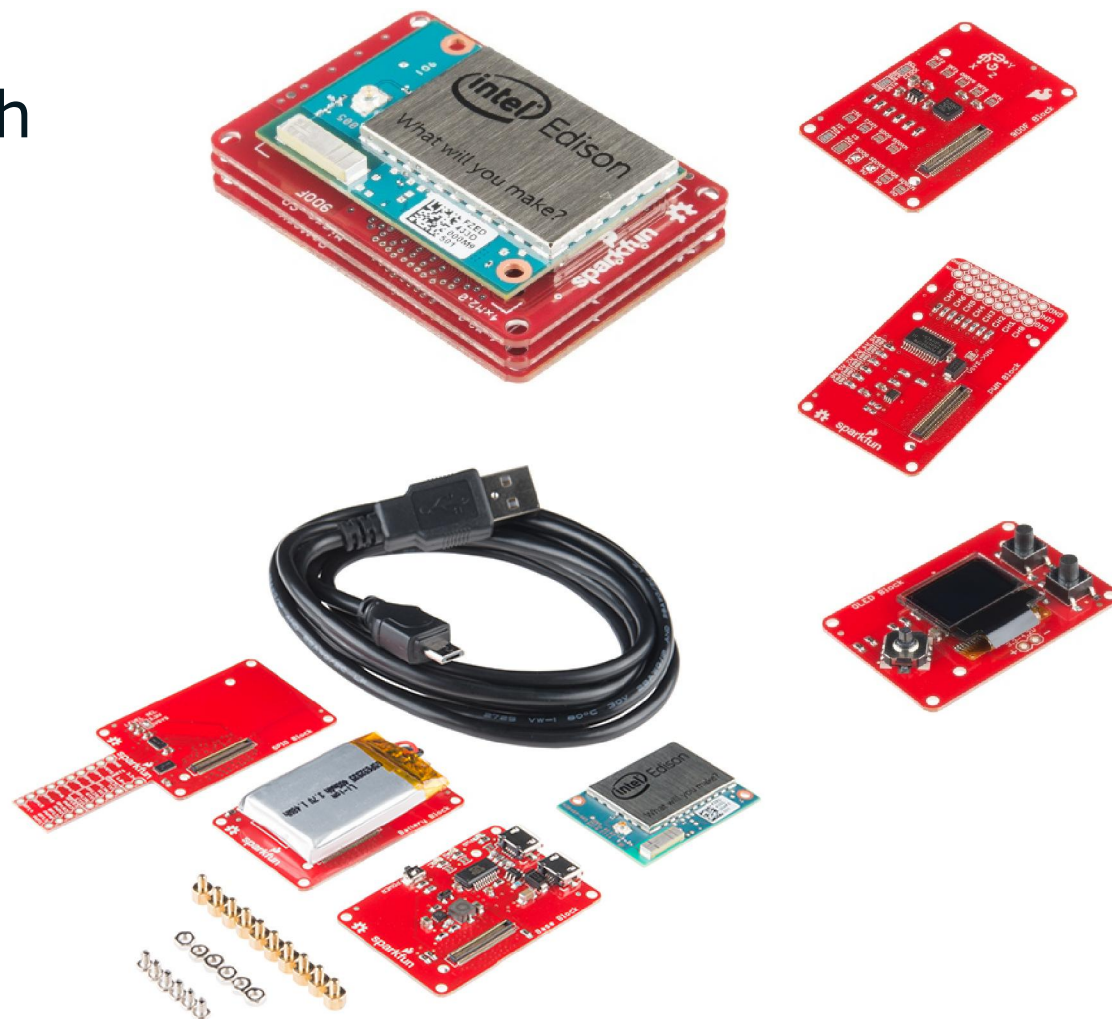




# Intel® Edison Board for SparkFun\*

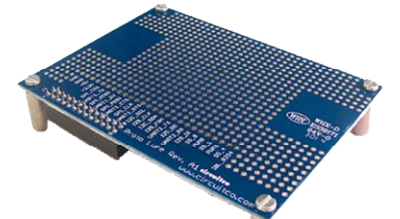
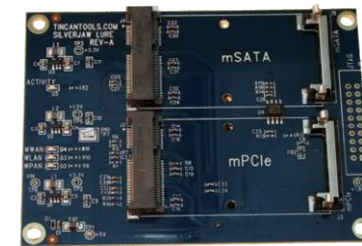
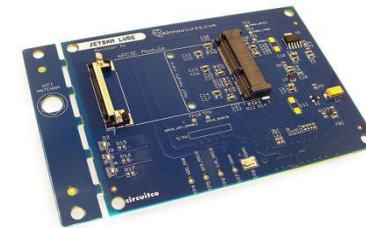
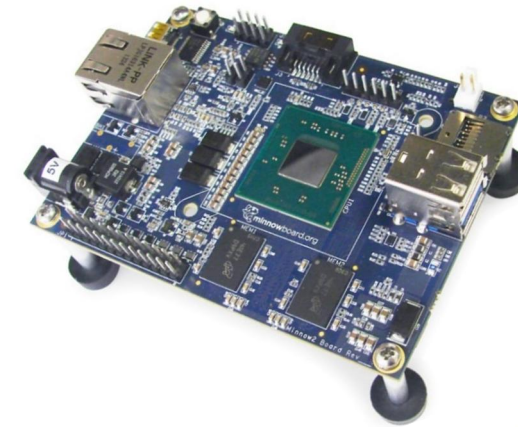
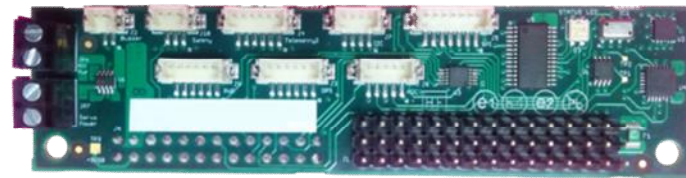
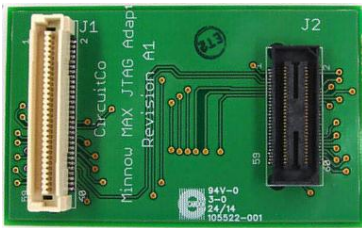
Compare to the Intel® Edison kit with Arduino:

- Same Edison CPU Module
- Simpler GPIO/Bus structure
- Stack on boards
- Smaller form factor
- Battery power available
- Many sensors available
- Brillo Starter Board
- Supported in AOSP with the February Brillo General Availability



# Intel® Expansion Kit with MinnowBoard\* (Baytrail-M)

- Similar Bus availability to Intel® Edison kits
- More powerful processor, Graphics and GPU enabled
- Larger RAM
- Additional storage options
- Add “lures” to expand to additional busses, sensors
- Brillo\* Starter Board
- Supported in AOSP TIP
- Will be part in the BDK for General Availability

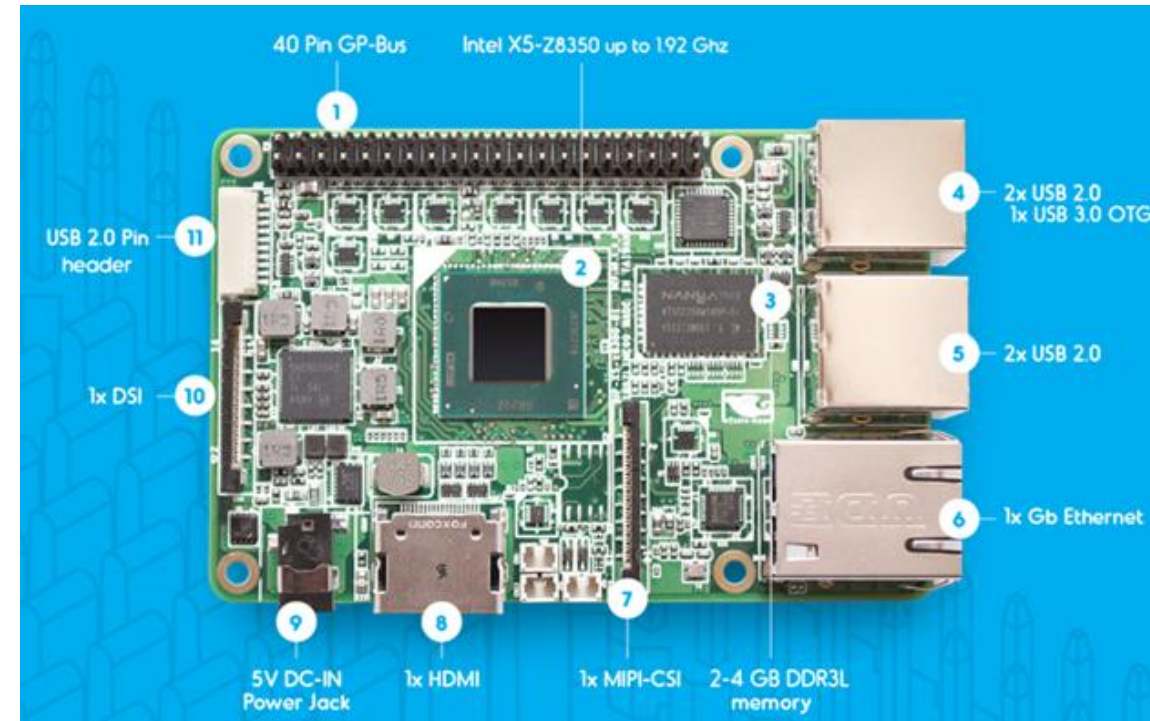




# Aaeon UP Board

- Kickstarter UP-board is a likely AOSP Starter Board
- Adds MIPI/DSI, MIPI/CSI ports
- Adds USB-Peripheral (for fastboot, adb)
- CherryTrail Platform

<http://up-board.org>



# Monitoring and Analytics



- Performance Analysis
- Crash Reporting

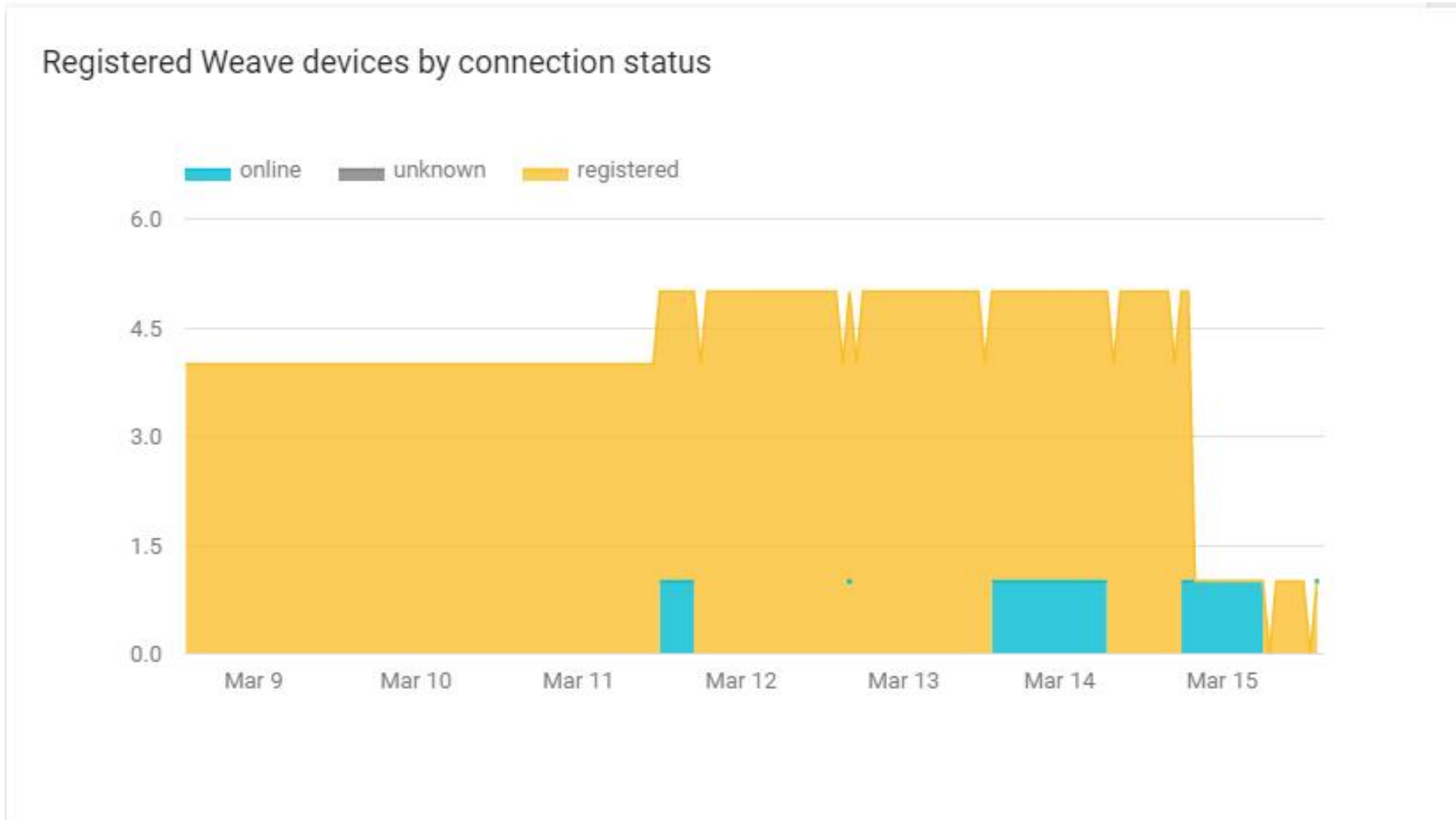


# Monitoring: Registrations

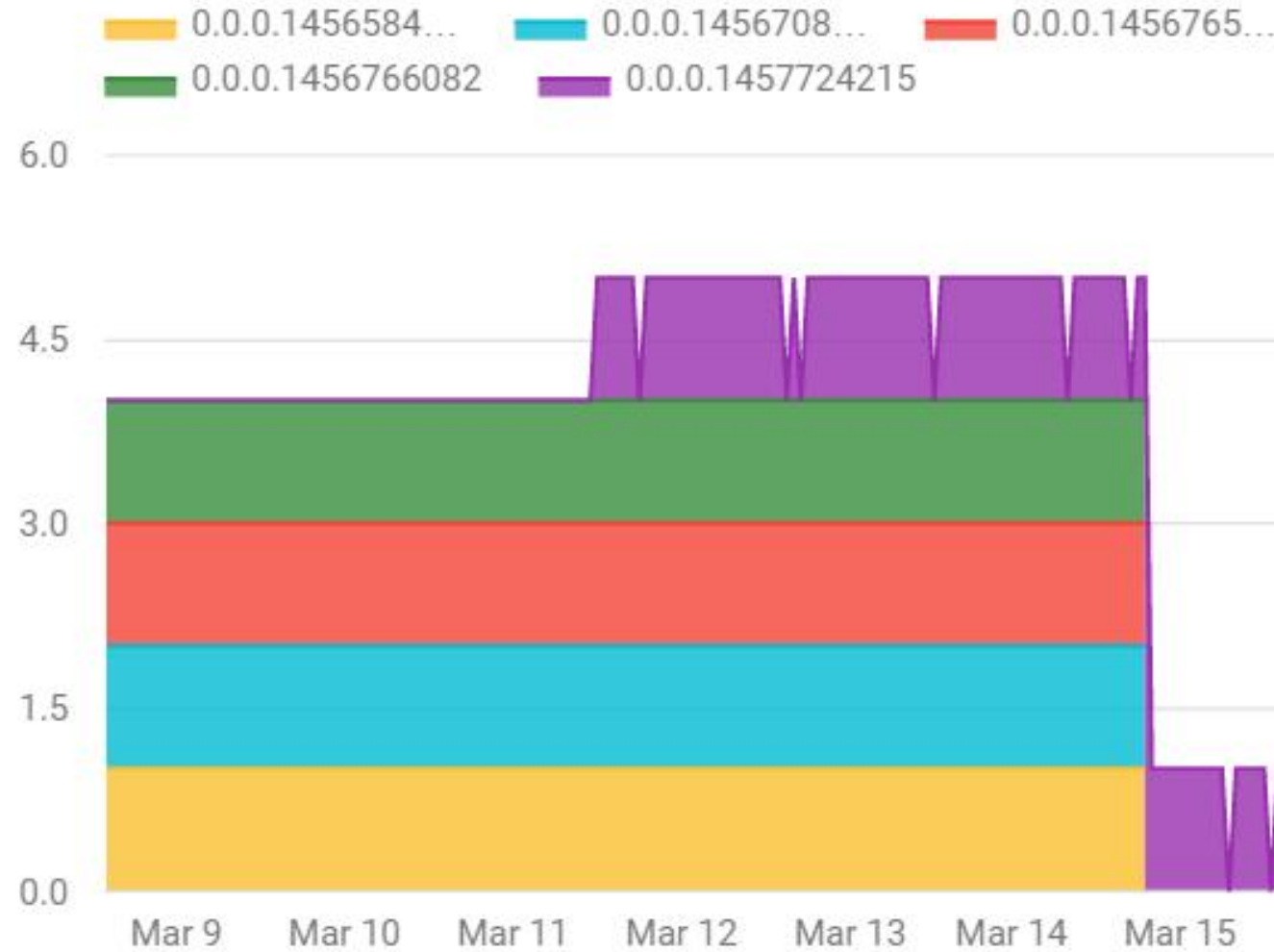
New Weave device registrations per hour



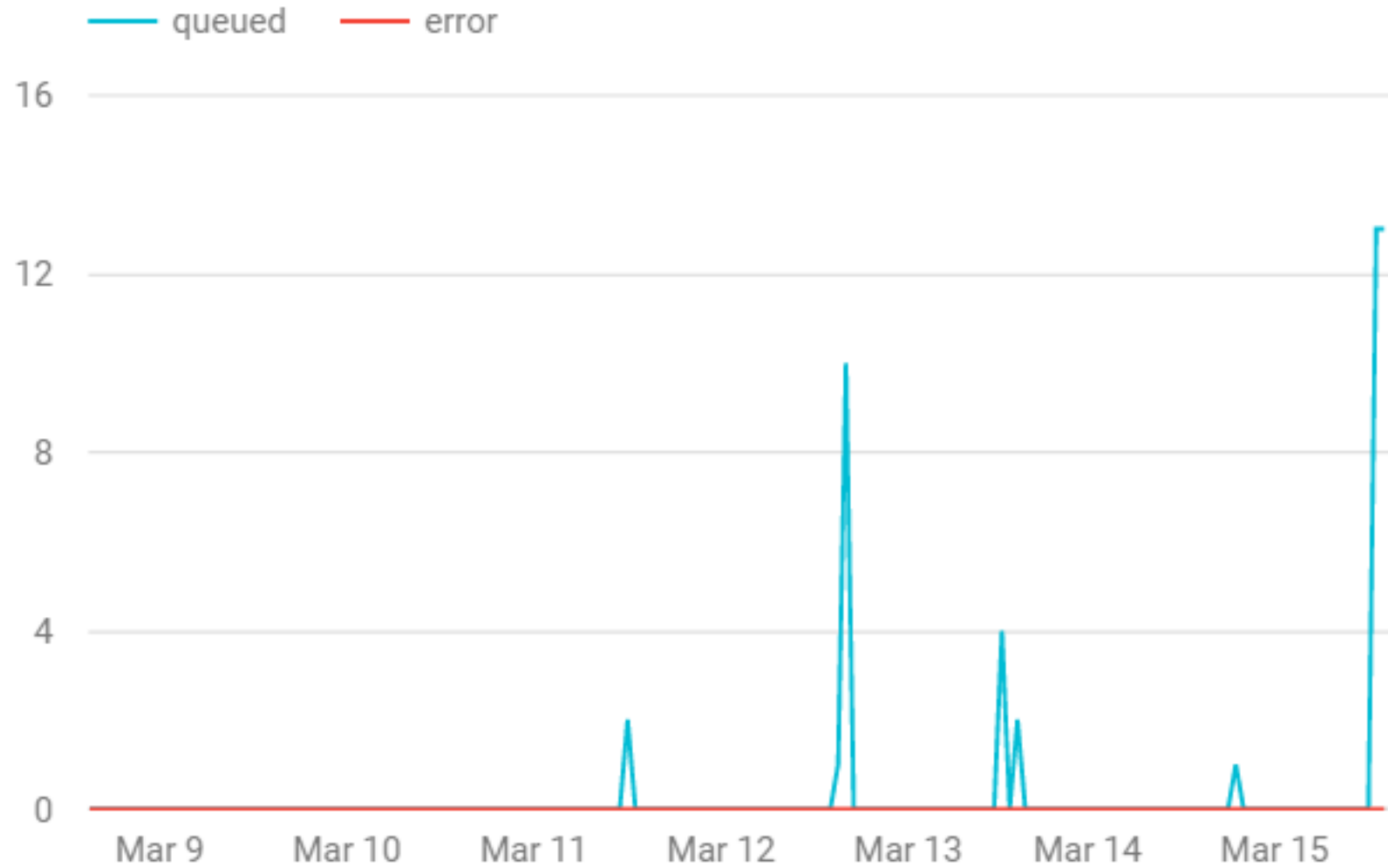
# Monitoring: Registered Devices



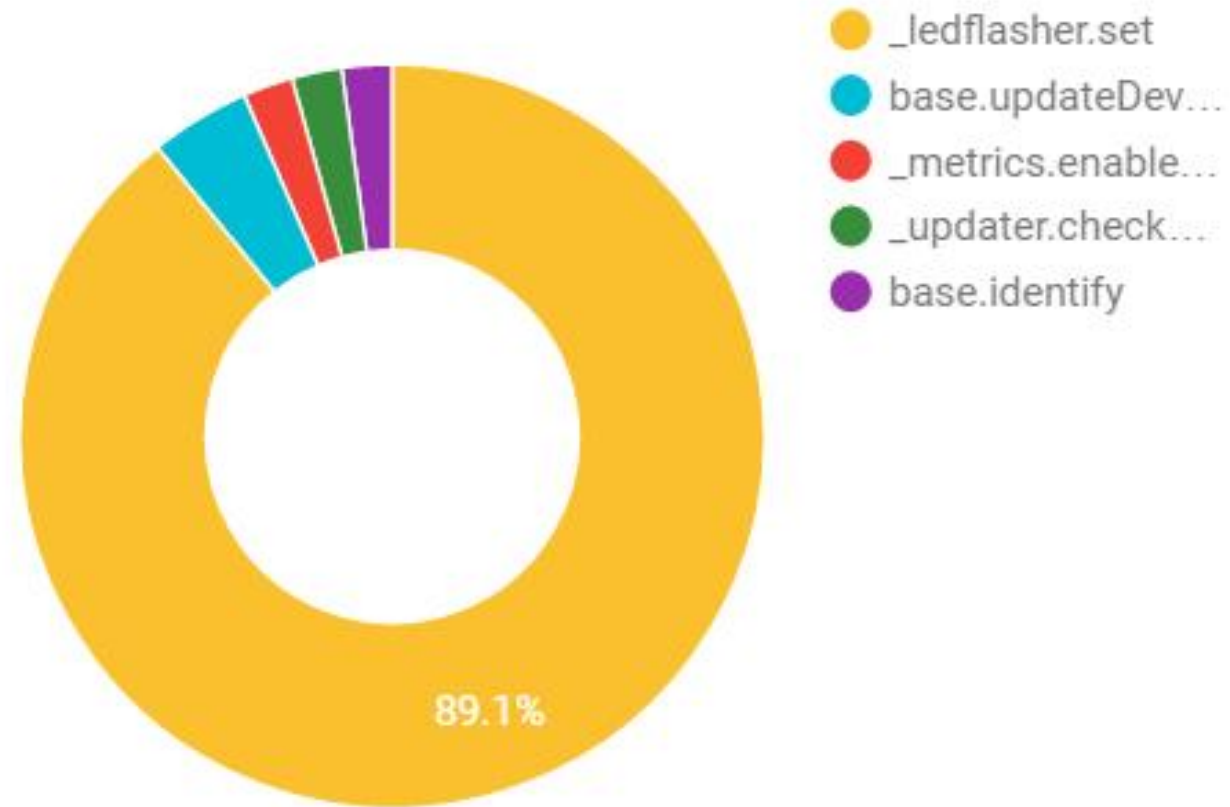
# Monitoring: Product Versions



# Monitoring: Weave Commands Per Hour

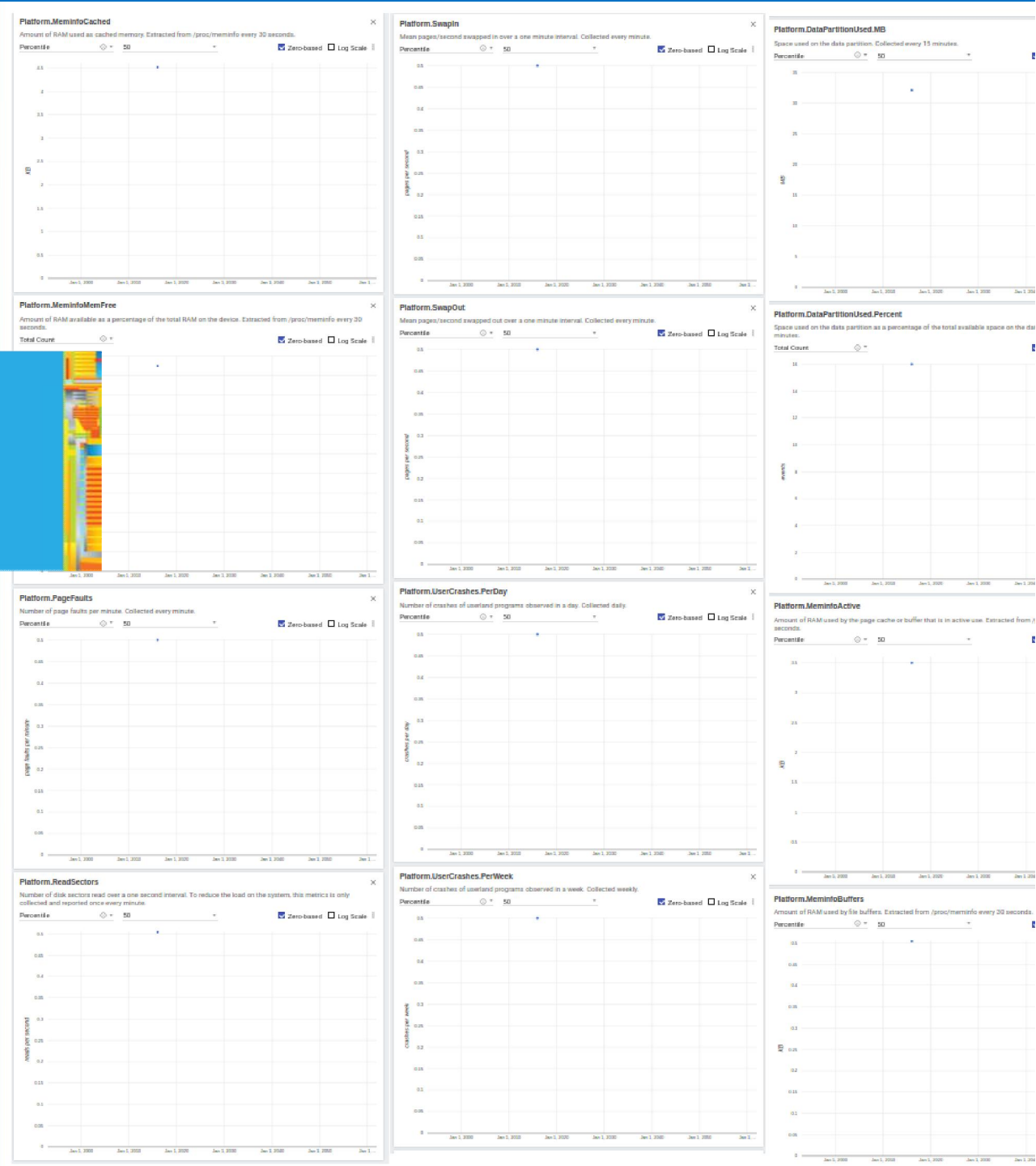


# Monitoring: Weave Commands by Name



# Logging

- Crash Reporting
- CPU Usage
- Use Time
- Data Partition Use
- Memory Used
- Page Faults
- Storage Activity





## Next Steps



### Likely futures for Brillo:

- Java\*
- Android\* Framework
- Camera Support
- Display via surface flinger
- Bluetooth\*
- Voice Command and Control
- Other connectivity options

### Intel Exploring:

- JavaScript
- Node.js
- NW.js (for display)

# What's in Part 2 of this Talk?



- Hardware Access Layer
- Code Labs
  - Hello World Brillo IO Programming
  - Hello World Weave Integration
  - Enable Google Services and APIs
  - Web Service Development

Questions?



# 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](http://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.

