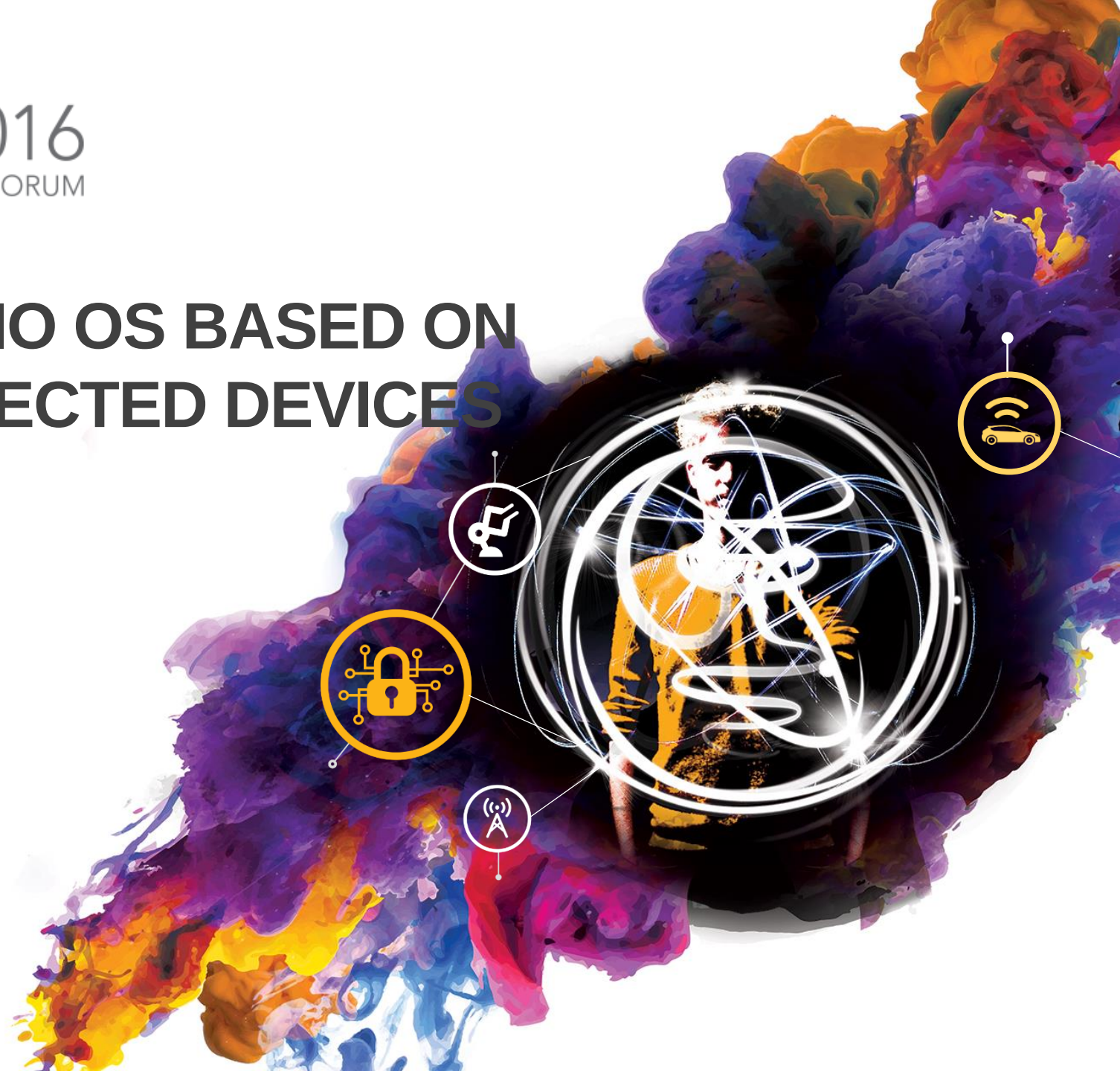
FTF 2016
TECHNOLOGY FORUM

INTRODUCING BRILLIO OS BASED ON ANDROID FOR CONNECTED DEVICES

TF-HMB-N1937

PHANTHAVONG DOUANGPHACHANH
I.MX APPS ENGINEER
ROBERT THOMPSON
I.MX ECOSYSTEM
FTF-HMB-N1937
MAY 17, 2016



AGENDA

- Overview of Brillo and Weave
- Brillo Target Markets
- Brillo, NXP & You
- Brillo Technical Overview

What is Brillo: Overview

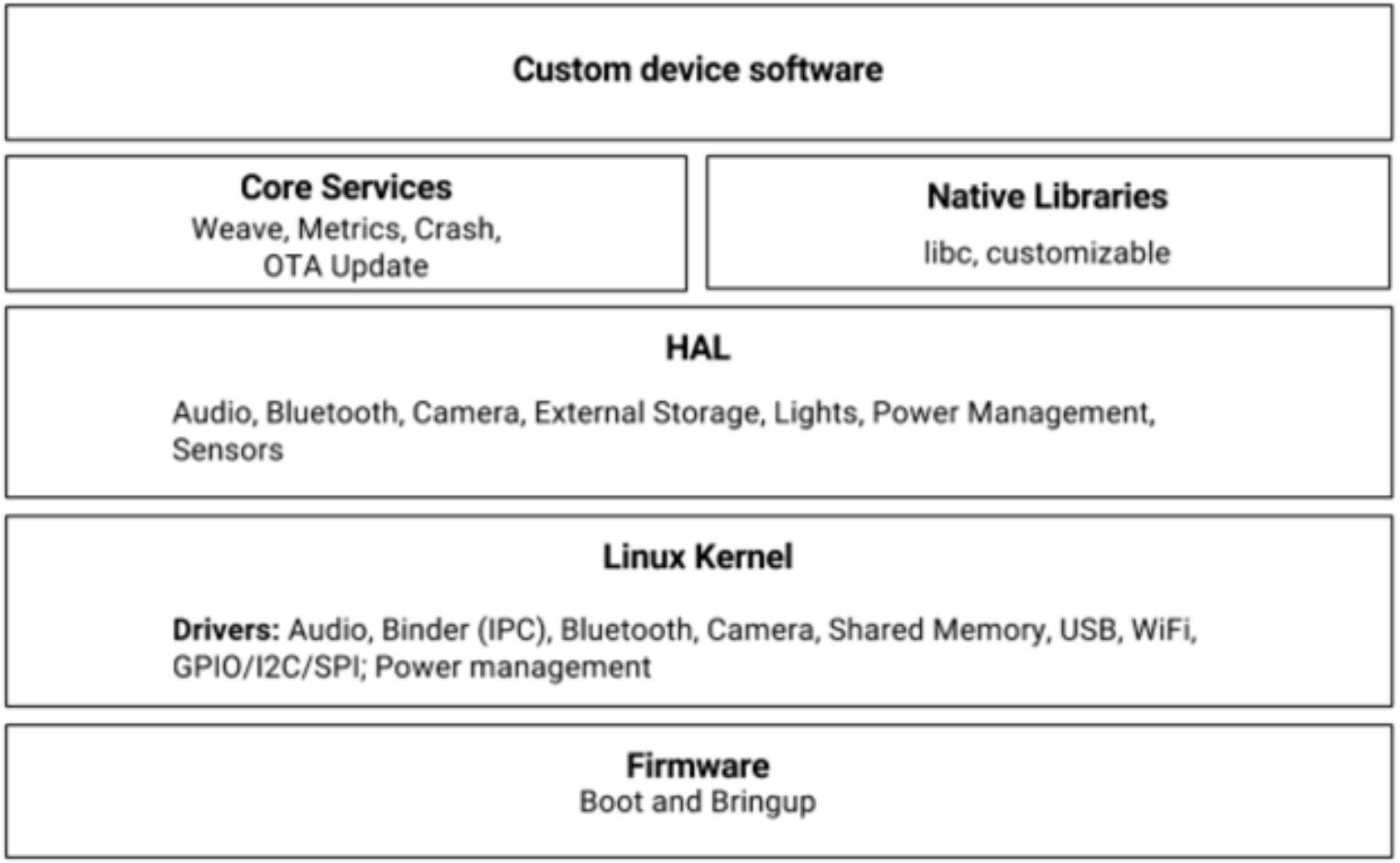
Brillo:

- Is an open source embedded OS based on Android
- Has a small memory footprint that makes it ideal for smart connected devices
  - Target RAM: 32MB+, Flash: 64MB+ (size depends on components loaded)
- Targets home and office devices such as locks, door controls, fire alarms thermostats, appliances, lights, plugs, moisture & temp sensors
- Core Services provide the ability to manage and monitor devices in the field
- Program is a device certification process that guarantees connectivity between certified mobile devices and the cloud

https://developers.google.com/brillo/?hl=en

What is Brillo: Main Components

Brillo has four components:
1) OS
2) core services
3) developer kit
4) developer console.



https://developers.google.com/brillo/?hl=en

Brillo OS

- Brillo provides a reliable, maintained, secure-by-default, and customizable operating system that lets you focus on your code
  - **Reliable:** Brillo is based on the Android Open Source Project (AOSP)
  - **Maintained:** Google provides minor (non-breaking) updates on a 6-week schedule and LTS releases on a 6-months schedule. Tools for updating your software and devices in the field are provided.
  - **Secure-by-default:** Brillo provides a verified computing base and software fault isolation architecture for all code, tools, and documentation.
  - **Customizable:** Brillo builds your product image from source, allowing you to customize nearly any behavior to your needs. The build architecture separates the chip-vendor "board support package" (BSP) from the Brillo core system, and allows both of these things to be configured from your product code and configuration.

https://android.googlesource.com/brillo/manifest/

Brillo: Developer Kit

- The Brillo Developer Kit (BDK) provides software & tools to build, test, and debug your target hardware:
  - **Build**: the BDK can build your entire product from source or perform incremental builds when possible. It is based on the Android.mk build architecture and is designed to integrate with your existing editor and source control/review work flows.
  - **Test:** the BDK provides standard ways to write and run unit tests locally on your workstation and integration tests that are controlled by your workstation but actually run on your target device. These can be used locally by each developer or integrated into a continuous build and integration testing infrastructure.
  - **Debug:** Use the "adb" tool over Ethernet or USB to test, inspect, and update your target device without special equipment.
  - The BDK runs on Ubuntu 14.04. The full set of requirements can be found at AOSP build requirements:

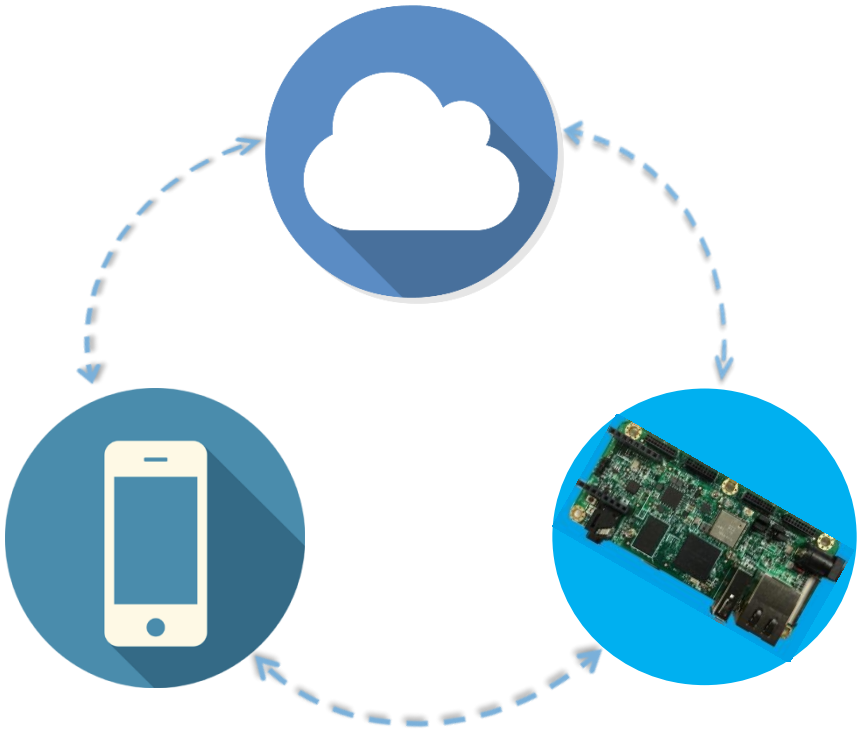
https://source.android.com/source/requirements.html

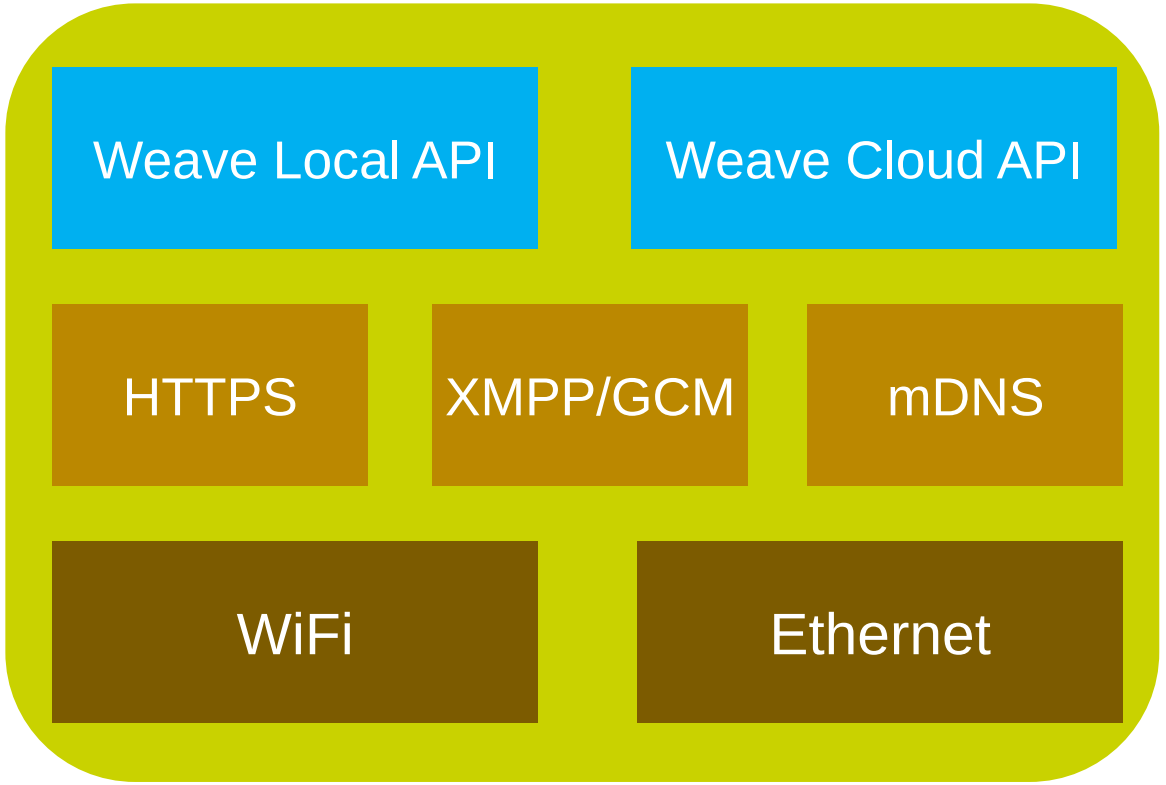
Brillo OS: Core Services & Developer Console

- Brillo provides four core services for connected devices out-of-the-box. These services are part of your device software and interact with Google servers to help your devices get better over time. You use the Weave Developer Console to administer and analyze these services.
  - **Weave:** Weave enables phone-based device bring up (get on Wi-Fi network, etc.), device-to-device and device-to-cloud communication, and user interaction from mobile devices and the web. Weave and Brillo are designed and tested to work well together.
  - **Metrics:** Brillo provides aggregate analytics data collection from devices in the field. You can view and analyze this data to understand end user behavior and validate product requirements and assumptions.
  - **Crash reporting:** Brillo provides crash statistics and debug data from devices in the field. You can view and analyze this data to improve the reliability of your products and manage the OTA update process.
  - **Over-the-air auto-updates:** Brillo enables you to upload new builds and within minutes, apply them to all your devices in the field (or a subset thereof). Monitor the health of new updates via the metrics and crash reporting services.

What is Brillo & Weave:

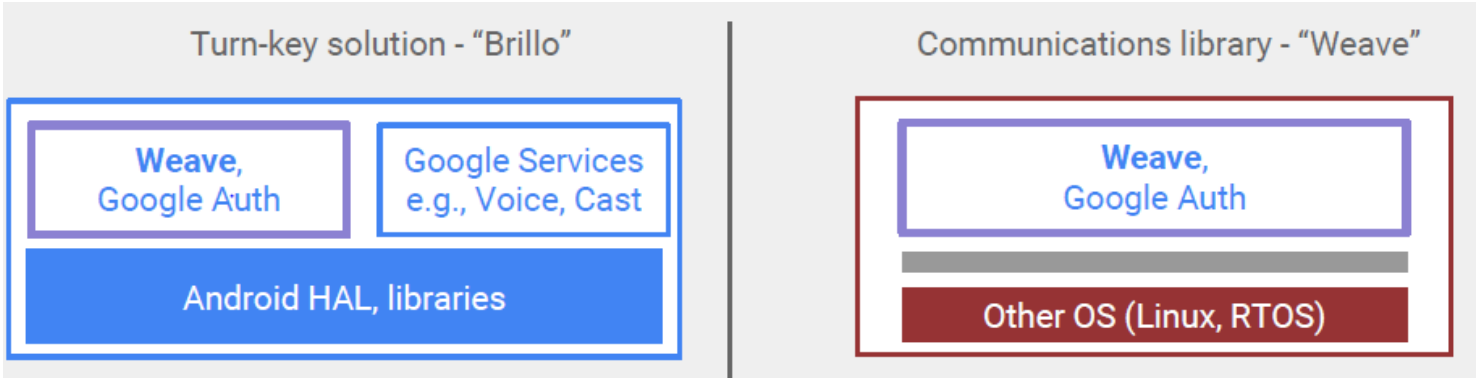
- Brillo is part of Weave communications platform for connected devices.
- To understand Brillo, let's take look at Weave ecosystem





What is Weave

- **Weave provides a framework for local and remote connectivity**
- Weave is a communication protocol that enables Brillo devices to connect locally to other devices or remotely through the cloud that include:
  - A set of services to enable a device to join a Wifi network (BT & Thread in 2H 2016) and provide secure access to the device
  - Mobile SDK for app developers on iOS and Android
  - A cloud service that manages state, history sharing and Updates



https://developers.google.com/weave/

Weave Ecosystem

- Weave ecosystem includes devices, client applications and the Weave cloud service
  - **Devices** expose Weave-compatible commands and device state. Examples of devices include autonomous vacuum cleaners, refrigerators, or parking meters.
  - **Client apps and services** are mobile or web applications which a user has authorized to monitor or access their devices. These include companion applications provided by the device developer, or third party applications and services the user authorizes to interact with his or her devices.
  - **The Weave cloud service** provides remote access to a user's devices if they choose to enable it. All Weave-compatible devices support local access, whereas only internet-connected devices support remote access.
  - **Transports** ensure a consistent experience across several classes of devices. It includes a REST API for remote access using the Weave cloud service, a local network API for direct network access, and a Bluetooth Low Energy (BLE) API for low-power devices. Weave libraries make it easy to always take the shortest route in communicating with a device - if it's on the same network, it's accessed there. If not, the Weave cloud service is used.
  - Remote access to devices is always authenticated, and requires the user to explicitly grant access to the relevant client app or service. Users can share varying levels of access to their devices with other users, groups, and client apps and services.

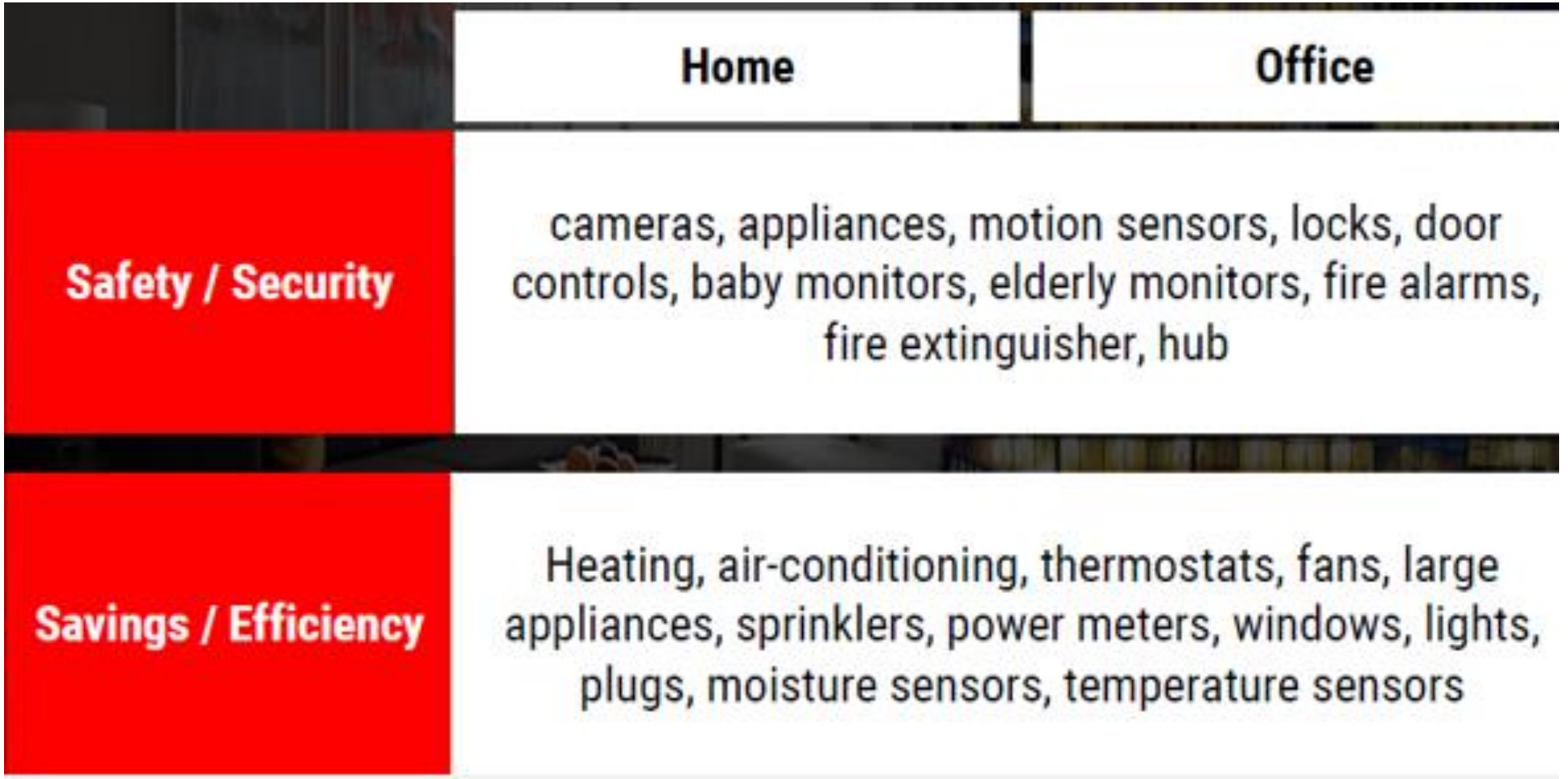
TARGET MARKETS

Brillo Target Platforms

Google has identified the following platforms for Brillo…

In summary, any device with some intelligence behind them (not just on/off), but not a device that requires a high degree of complex graphical user interface – those would use Android.

Brillo Target Platforms



BRILLO & NXP

Brillo & NXP: A Brief History

Starter Board Agreement

- 6/15 Signed the starter board agreement with Google

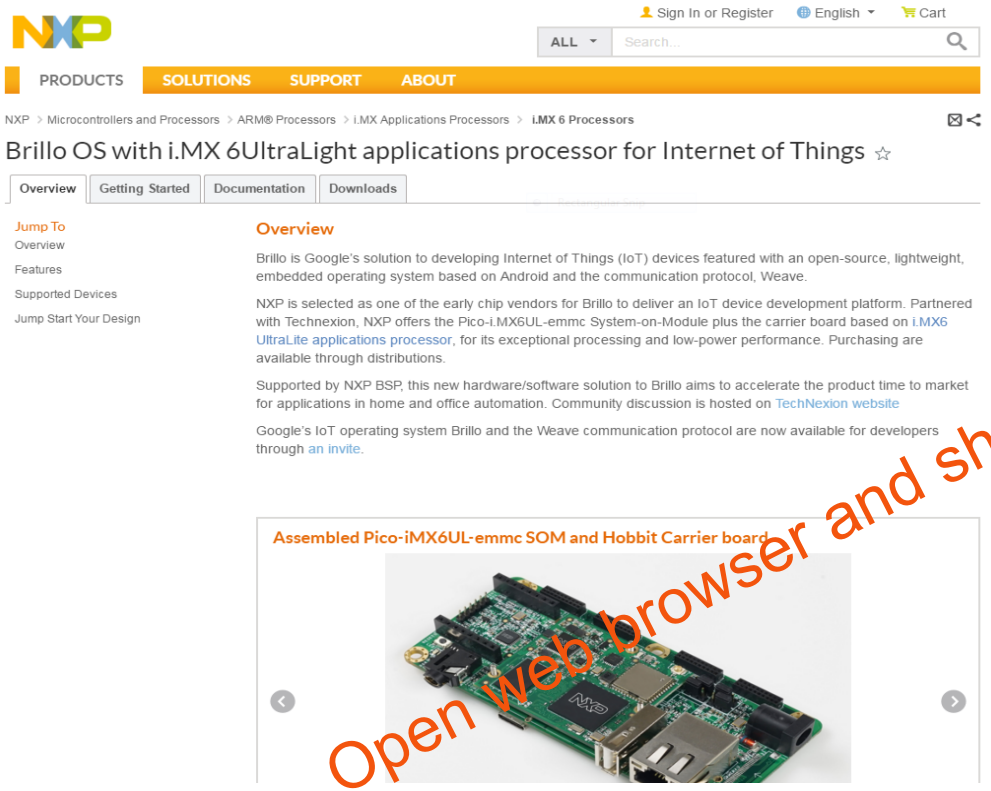
i.MX6UL Pico i.MX6UL

- Google wanted to offer low /cost platform equals –i.MX6UL
- We choose the PicoIMX6UL from Technexion as this met all the requirements of the Brillo platform spec & offered community support

Brillo Platform Test Suite

- Our Android team, worked with the Brillo team to port the code to the Pico6UL and pass the Brillo Platform Test Suite (PTS).

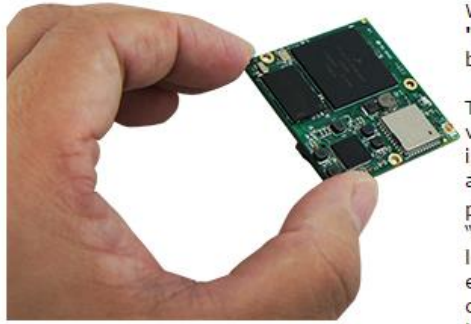
Brillo & NXP.com



Open web browser and show Brillo landing pages on NXP.com

NXP & TechNexion

Hobbitboard Brillo



Wandboard is proud to introduce the **"Hobbitboard Made for Brillo"**. Your building block for the Internet of Things (IoT).

The Hobbitboard is an ubiquitous computing, very compact, high performance platform that is highly optimized for mobile Internet of Things applications. Using a pin-compatible scalable platform that not only utilizes the Intel® "Edison" connector connectivity for sensors and low-speed I/O, but also adds additional expansion possibilities for multimedia and connectivity, giving you cutting edge technology that can easily be expanded and implemented into Industry 4.0 applications.

The **"Hobbitboard Made for Brillo"** puts any IoT project in fast track mode and help you build your next home automation project, drone, 3D printer, climate control, robot or remote sensor network in minutes.

What is Brillo?

Brillo brings the simplicity and speed of software development to hardware for IoT with an embedded OS, core services, developer kit, and developer console.

More information can be found at : **Google Brillo Project Page**

The hardware specifications

Hobbitboard exist of an System on Module (SoM) and an I/O carrierboard in a similar way as the original Wandboard i.MX6 making the Compute Module very easy to adapt to your specific projects.

The Hobbit Compute Module

Meet the Hobbitboard

PRE-ORDER your Hobbitboard Here

armkits · AVNET Embedded · dencomputer systems · Digi-Key Corporation · MOUSER ELECTRONICS · TEXIM EUROPE

Open web browser and show Brillo landing pages on Hobbitboard.com

BRILLO, NXP AND YOU

Brillo, NXP and You – FAQ

- **Q: How do I get access to Brillo source code?**
- A: Brillo (including Weave) is open source and the code is available at: https://android.googlesource.com/brillo/manifest/

- **Q: How do I gain access to the to the developer kit and console?**
- A: Access needs to be requested at https://developers.google.com/brillo/?hl=en – NXP cannot grant access to customers. Only one request is need for both Brillo and Weave.

- **Q: Where is this NXP specific code being posted?**
- A: A device tree Picoimx has been created on the AOSP Brillo manifest. On the Brillo landing page on NXP.com there is a link to the latest Brillo image and manufacturing tool for re-flashing of boards

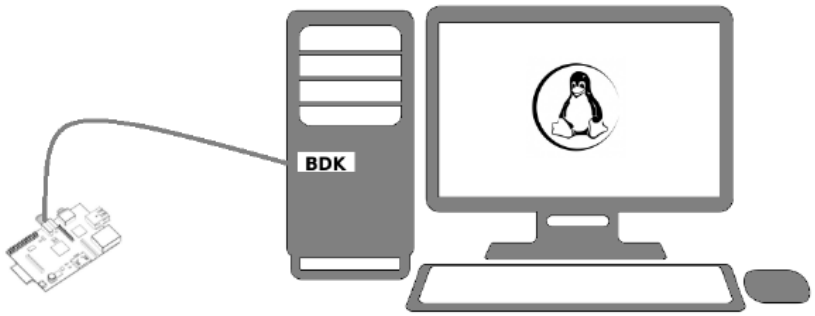
- **Q: Where do I get support?**
- A: Level 1: As an open-source project the first line of support is via the community @:
- Level 2: NXP will support customers using the Picoi.MX board via the i.MXCommunity
- Level 3: FAE's and apps engineers

BRILLO TECHNCIAL OVERVIEW

Technical Overview

- **What to expect:**
  - The Brillo platform is composed of certified hardware, APIs, runtime environment, libraries, and development tools.
  - Unlike Android developers, Brillo developers write in C or C++ directly in the system's userspace. To minimize the device footprint, there is no Java runtime, NDK, or Android Application Framework.

- **Development workflow:**
  - Brillo development, build, and unit testing takes place on the developer workstation.
  - The resulting image is pushed via USB or Ethernet to the target hardware, where further integration testing and debugging can take place.
  - The connection between the workstation and the target can be USB or Ethernet. On the workstation, you can use the adb and fastboot tools to update and interact with the target.



Brillo vs. AOSP Development

1. Brillo does not provide or support the Application Framework. All Brillo code is completely functional without using any Java code. Compatible boards are also required to be independent of the Application Framework to provide their functionality.

2. Brillo defines a strong separation of code from different providers: the board support package (BSP) code, the Brillo core system, and your product code are all stored in well defined locations. Interactions between components are clear and upgrading them is a straightforward exercise.

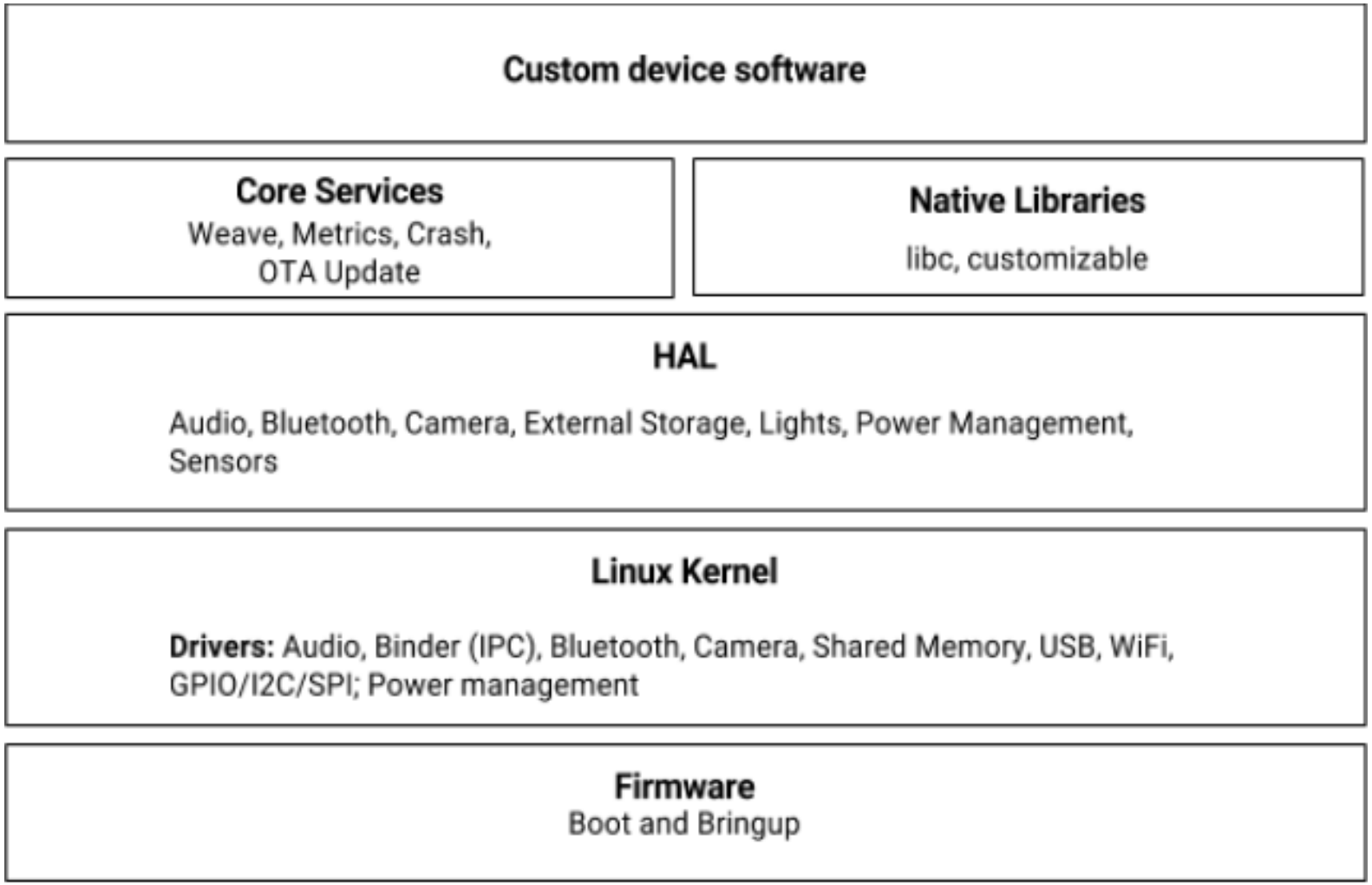
3. Brillo tools and workflow are designed to support C and C++ development directly in the native userspace.

Brillo vs. XYZ Embedded Linux

- Brillo has the following userspace differences from other Linux distributions, which means some code will need to be ported:
  - It uses the bionic C library used by Android
  - It follows the AOSP file system layout

- Instructions for porting your legacy code are provided. Included in the Brillo distribution is a package management system that allows you to automatically pull in 3rd party packages with minimal work.

Architecture



Operating System Overview

- The Brillo OS is based on the current version of Android. The components that make up Brillo's operating system are:

- Kernel and HAL:
  - Brillo compatible boards provide an Android kernel and HAL. Depending on the board's available features, the HAL implementation may be a subset of the complete Android HAL; for example, a board that is not intended to support graphics might skip an implementation of the graphics HAL.

  - While the majority of Brillo devices use Weave and undergo Weave certification for interoperability, Brillo does not impose this type of requirement. Devices built on Brillo require no additional certification process.

  - In some cases, Brillo provides C++ development interfaces that sit on top of the HAL.

Minimum Required Feature Subset

Component	Supported HAL	Development Interface
WiFi [required]	wpa_supplicant	Connection Manager
Bluetooth	coming soon	coming soon
Storage [required]	Standard kernel & libc interfaces	vold and config files
Input Devices	linux/input.h	
Audio (input & output)	audio.h	libmedia/stagefright
Camera/Video Input	Camera and Camera v3	Camera API v2
Video (output) + DRM	N/A	
Graphics	N/A	
Sensors	sensors.h	Sensors NDK API
LEDs	lights.h	
Power management	kernel wakelocks, cpufreq, and cpuidle	Native power management
SPI	sysfs	
I2C	sysfs	
GPIO	sysfs	

Brillo does not depend on the remaining parts of the Android HAL but board vendors may choose to provide them anyway. Brillo does not support graphics, video output nor associated DRM behaviors.

Run Time Environment

C Library	bionic
Init	Standard Android init system with improved support for native processes.
IPC	Binder (other IPC like dBus can be added)
Security/Sandboxing	Android native security support

- **Boot process**
  - Brillo compatible boards support a verified boot process that expects an A/B update strategy. This relies on the typical Android fastboot style bootloader and adds:
  - Support for A/B path
  - Verification of the kernel stage (which can then use dm-verity on partitions)
  - Automated rollback if the new path fails to boot
  - Recovery behavior if both paths fail
  - Rollback protection (with proper hardware support)

- For bootloader details, see Bootloader Reference
- For more about OTA updates see the OTA Update Overview

Developer Tools

- **BSP:**
  - In embedded operating systems, a board support package is the glue that allows an embedded developer to write code for a particular platform. It can include a kernel, drivers, toolchain, userspace code, and even a basic operating system.
  - In Brillo, the BSP concept has been formalized and modularized so that SoC vendors can define their architecture, platform, and board without modifying the rest of the tree. In Brillo, the BSP includes the bootloader, kernel, HAL implementations, any software needed for custom hardware, and optionally, a toolchain, but not the majority of the operating system.
  - The BSP is provided by the SoC vendor. Brillo compatible boards must meet minimum feature, implementation, and performance requirements. Brillo compatible BSP implementations must comply with the Brillo DPS.

Developer Tools

- **Product:**
  - The Brillo product structure illustrates where developers and SoC vendors should place their code and configuration files.

- **Testing:**
  - Provides default unit and integration testing framework to be used by the OS and the project. Unit tests run locally on the developer workstation and integration tests interact with the target device via adb.

- **Debugging tools:**
  - Brillo compatible boards provide full support for adb and fastboot over USB and/or TCP/IP (depending on board capabilities). For more information on using these commands, see Android Debug Bridge: http://developer.android.com/tools/help/adb.html

Getting Brillo Source and Building PicoiMX Image

- **Prepare Host PC (You will nee a 64-bit version of Ubuntu 14.04)**
  - Installing required packages:
    - sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++- \ multilib libc6-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa- dev \ libxml2-utils xsltproc unzip python-networkx bc android-tools-fastboot android-tools-adb
    - wget -S -O - http://source.android.com/source/51-android.rules | sed "s/<username>/$USER/" | sudo tee >/dev/null /etc/udev/rules.d/51-android.rules; sudo udevadm control --reload-rules
- **Getting Brillo source code:**
  - mkdir ~/brillo-aosp
  - cd ~/brillo-aosp
  - repo init -u https://android.googlesource.com/brillo/manifest -b master
  - repo sync –jn –c
    - where n is the number of simultaneous jobs. This option speeds up command execution on multi-core machines. Typical values start at -j2 and go up to -j24 or -j32
- **Building Picoimx target image:**
  - cd ~/brillo-aosp
  - source build/envsetup.sh
  - Lunch picoimx-eng
  - Make
    - It will take a while for the first build to complete. If there is no error, proceed to next slide to flash the image into your Picoimx board.

Flashing Picoimx Image

a) Make sure that your /etc/udev/rules.d/51-android.rules contains the 2 entries below:

```
# adb protocol for Brillo ADB
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="4ee7", MODE="0666", OWNER="lab“
# adb protocol for Brillo Fastboot
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1", ATTR{idProduct}=="0d02", MODE="0666", OWNER="lab“
```

Otherwise edit the file and add these 2 entries and issue “sudo udevadm control --reload-rules” command.

b) Connect you Picoimx board serial console and USB OTG port to your host PC.

c) Power up your Picoimx board and wait until it completed.

d) Open a terminal console your host PC and issue “lsusb” command and check for an entry:

e) “ID 18d1:4ee7 Google Inc”

f) adb reboot bootloader (wait for your Picoimx board get in fastboot mode)

g) cd ~/brillo-aosp/out/target/product/picoimx

h) export ANDROID_PRODUCT_OUT=~/brillo-aosp/out/target/product/picoimx

i) ./provision-device (and wait until the flash is completed)

j) fastboot reboot (wait for your Picoimx board to complete the boot-up)

Licenses and Certifications

- Brillo is released as open source available via the AOSP and follows the same licensing practices. Licensing information at:
  - https://source.android.com/source/licenses.html

- Official BDK releases are available at:
  - https://developers.google.com/brillo/eap/reference/downloads

- Devices that use the Brillo core services in production are required to obtain Weave Certification and conform to the Brillo Compatibility Definition Document (CDD)
  - https://developers.google.com/brillo/eap/reference/program/device-certification-program
  - https://developers.google.com/brillo/eap/reference/program/compatibility-definition-document

SECURE CONNECTIONS
FOR A SMARTER WORLD