



#SECONRU



МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Android Things & Google Weave

Звиад Карадава

Google Developer Expert: Internet of Things

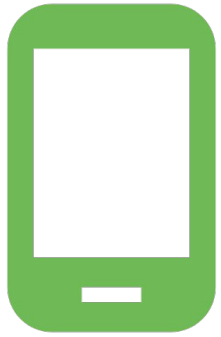
21-22 АПРЕЛЯ | ПЕНЗА



android things



Zviad Kardava - Google Developer Expert: IoT
+ZviadKardava
@ZviadKardava
fb.com/Zv1ad



Ideal for **powerful, intelligent devices on the edge**
that **need to be secure.**



Cameras
Gateways
HVAC Control
Smart Meters



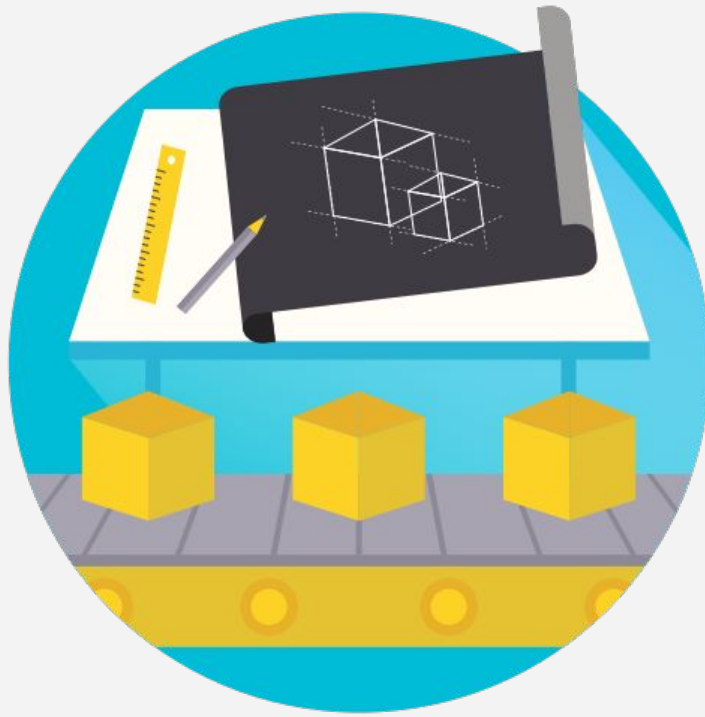
Point of Sale
Inventory Control
Interactive Ads
Vending Machines



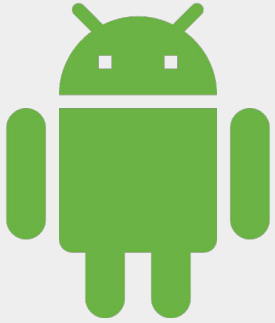
Security Systems
Smart Doorbells
Routers
Energy Monitors



Asset Tracking
Fleet Management
Driver Assist
Predictive Service



From Prototype to Production



Android SDK



Android Studio



Play Services



Firebase



Cloud Platform



IoT Developer
Console



Automatic
Security Updates



Signed Images



Verified Boot

Managed by Google

Android Framework

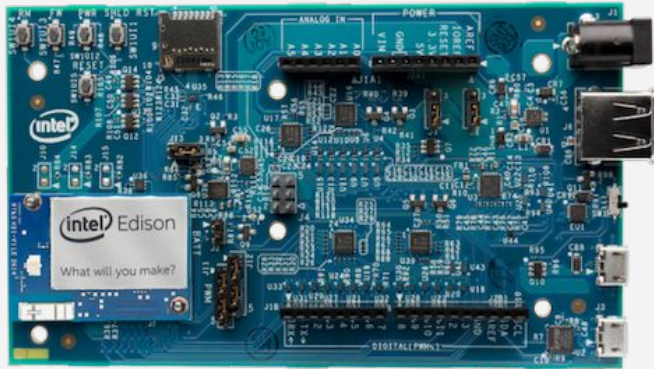
Hardware Libraries

Linux Kernel

Managed by Developers

Apps

User Drivers



SoM
Architecture



Google Managed
BSP



Android Things for Developers

Applications

Launcher

Phone

Messaging

Contacts

Calendar

Browser

Application Framework

Activity Manager

Window Manager

Power Manager

Resource Manager

XMPP Service

Content Providers

Wallpapers

SystemUI

Package Manager

Telephony Manager

Location Manager

Connectivity Manager

View System

Runtime Permissions

Soft Keyboards

Notifications

Libraries

Surface Manager

Media Framework

Chromium

SSL

HAL

Android Runtime

Core Libraries

Android Runtime (ART)

Audio Manager

SQLite

OpenGL

libc

Linux Kernel

Display Driver

Camera Driver

Bluetooth Driver

Binder (IPC) Driver

USB Driver

Audio Driver

WiFi Driver

Power Management

Applications

Launcher

Phone

Messaging

Contacts

Calendar

Browser

Application Framework

Activity Manager

Window Manager

Power Manager

Resource Manager

XMPP Service

Content Providers

Wallpapers

SystemUI

Package Manager

Telephony Manager

Location Manager

Connectivity Manager

View System

Runtime Permissions

Soft Keyboards

Notifications

Libraries

Surface Manager

Media Framework

Chromium

SSL

HAL

Android Runtime

Core Libraries

Android Runtime (ART)

Audio Manager

SQLite

OpenGL

libc

Linux Kernel

Display Driver

Camera Driver

Bluetooth Driver

Binder (IPC) Driver

USB Driver

Audio Driver

WiFi Driver

Power Management



Displays are Optional



Consider Alternate UI



```
dependencies {  
    provided 'com.google.android.things:androidthings:...'  
}
```

```
<application ...>  
    <uses-library android:name="com.google.android.things" />  
  
    <activity ...>  
        ...  
  
        <!-- Launch activity automatically on boot -->  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
            <category android:name="android.intent.category.IOT_LAUNCHER" />  
            <category android:name="android.intent.category.DEFAULT" />  
        </intent-filter>  
    </activity>  
</application>
```

Home Activity

Peripheral I/O

GPIO

PWM

I²C

SPI

UART

```
// Open a peripheral connection
PeripheralManagerService service = new PeripheralManagerService();
Gpio button = service.openGpio(GPIO_PIN_NAME);

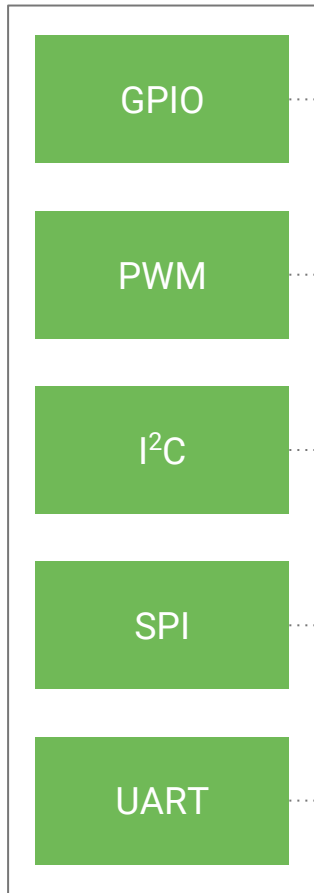
// Configure the peripheral
button.setDirection(Gpio.DIRECTION_IN);
button.setEdgeTriggerType(Gpio.EDGE_FALLING);

// Attach callback for input events
button.registerGpioCallback(new GpioCallback() {
    @Override
    public boolean onGpioEdge(Gpio gpio) {
        Log.i(TAG, "GPIO changed");

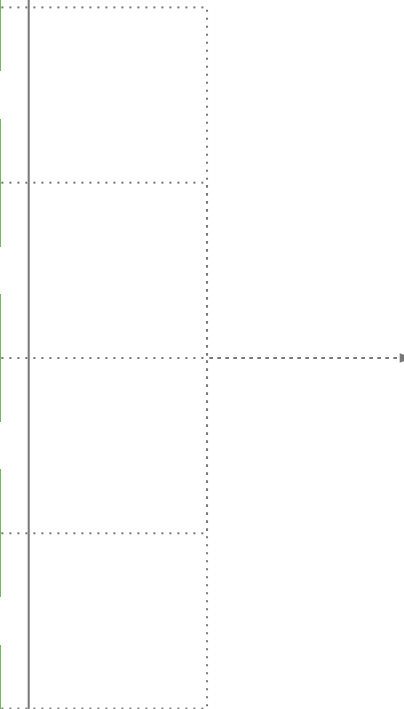
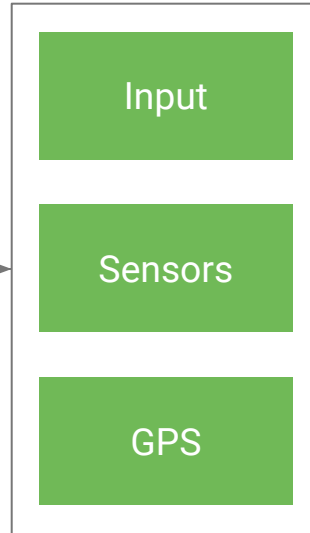
        // Return true to continue listening to events
        return true;
    }
});
```

Simple Peripheral I/O

Peripheral I/O



User Drivers




```
InputDriver inputDriver = InputDriver.builder(InputDevice.SOURCE_CLASS_BUTTON)
    .setName(DRIVER_NAME)
    .setVersion(DRIVER_VERSION)
    .setKeys(new int[]{KeyEvent.KEYCODE_SPACE})
    .build();
UserDriverManager.getManager().registerInputDriver(inputDriver);
```

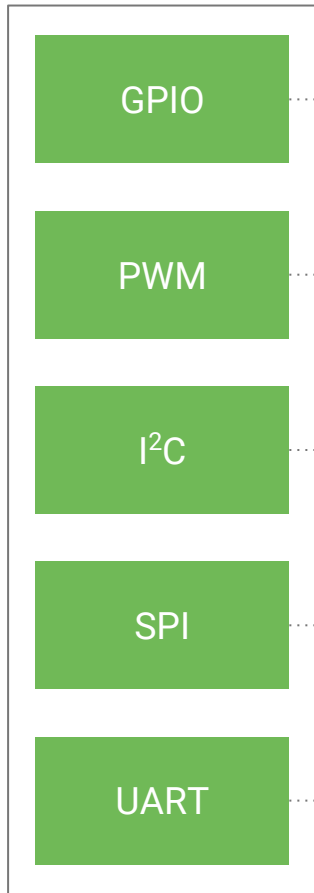
```
private GpioCallback callback = new GpioCallback() {
    @Override
    public boolean onGpioEdge(Gpio gpio) {
        int keyAction = gpio.getValue() ?
            KeyEvent.ACTION_DOWN : KeyEvent.ACTION_UP;

        inputDriver.emit(new KeyEvent[]{
            new KeyEvent(keyAction, KeyEvent.KEYCODE_SPACE)});

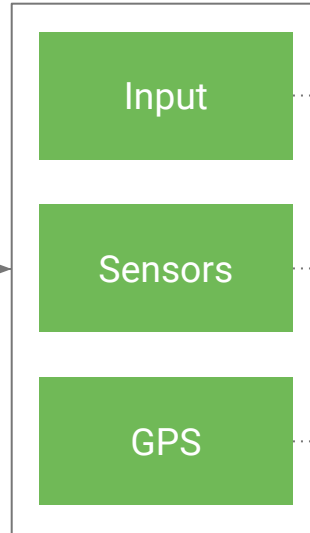
        return true;
    }
};
```

Button User Driver

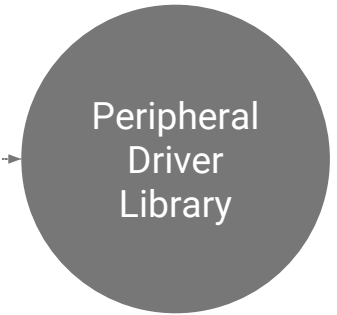
Peripheral I/O



User Drivers



Peripheral
Driver
Library



```
dependencies {  
    compile 'com.google.android.things.contrib:driver-button:...'  
}
```

```
ButtonInputDriver inputDriver = new ButtonInputDriver(  
    GPIO_PIN_NAME,  
    Button.LogicState.PRESSED_WHEN_LOW,  
    KeyEvent.KEYCODE_SPACE);  
  
inputDriver.register();
```

Button Library Driver



The Power of
Android



Managed by
Google



Automatic and
Secure

androidthings



Google's IoT Developers Community
<https://g.co/iotdev>



Google's IoT Solutions
<https://iot.google.com>



Android Things SDK
<https://developer.android.com/things>



Zviad Kardava - Google Developer Expert: IoT
+ZviadKardava
@ZviadKardava
fb.com/Zv1ad



Weave



Zviad Kardava - Google Developer Expert: IoT
+ZviadKardava
@ZviadKardava
fb.com/Zv1ad



Device SDK

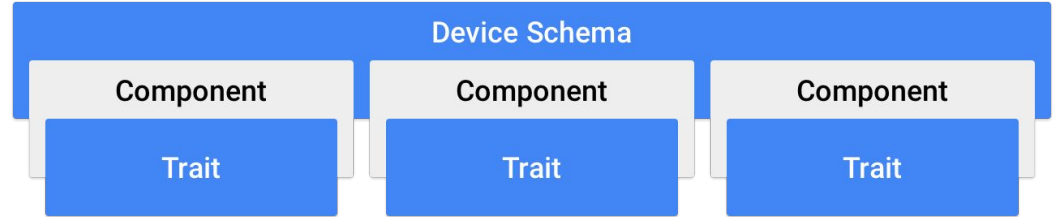


Cloud Service

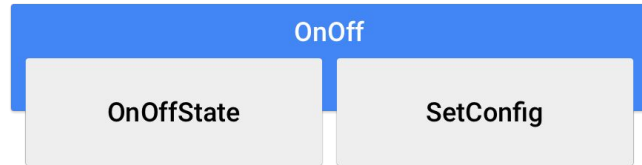
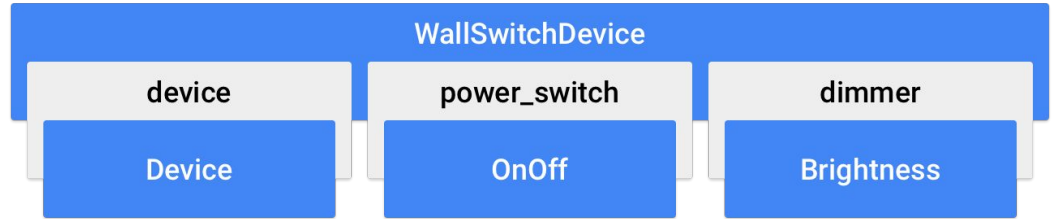


Developer
Tools

Device Schemas



Device Schemas



```
// Create the device interface.
GoogWallSwitchDevice* wall_switch = GoogWallSwitchDevice_create(
    GoogWallSwitchDevice_WITH_ALL_COMPONENTS);
IotaDevice* iota_device = iota_device_create_from_interface(
    (IotaInterface*)wall_switch, (IotaModelManifestId){"AIAAA"});

// Create the platform daemon.
IotaOauth2Keys oauth2_keys = (IotaOauth2Keys){
    .oauth2_api_key = IOTA_OAUTH2_API_KEY,
    .oauth2_client_id = IOTA_OAUTH2_CLIENT_ID,
    .oauth2_client_secret = IOTA_OAUTH2_CLIENT_SECRET,
};

IotaDaemon* iota_daemon = host_iota_daemon_create(
    iota_device, "switch",
    (HostIotaDaemonOptions){.oauth2_keys = &oauth2_keys});
```

Device Initialization

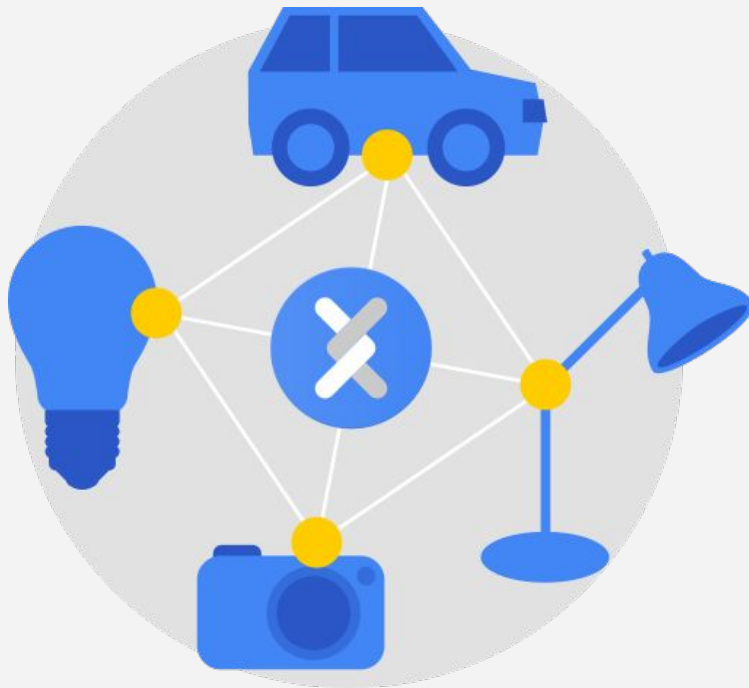
```
// Set default power switch configuration.
GoogOnOff* onoff_trait = GoogWallSwitchDevice_get_power_switch(wall_switch);
GoogOnOff_set_callbacks(onoff_trait, daemon,
    (GoogOnOff_Handlers){.set_config = &wall_switch_on_off_trait_setconfig});

IOTA_MAP_SET_DEFAULT(GoogOnOff_get_state(onoff_trait), state,
    GoogOnOff_ON_OFF_STATE_ON);
```

```
IotaTraitCallbackStatus wall_switch_on_off_trait_setconfig(
    GoogOnOff* self,
    GoogOnOff_SetConfig_Params* params,
    GoogOnOff_SetConfig_Results* result,
    GoogOnOff_Errors* errors,
    void* user_data) {
    GoogOnOff_OnOffState new_state = IOTA_MAP_GET(params, state);
    // ...handle state change command...

    return kIotaTraitCallbackStatusSuccess;
}
```

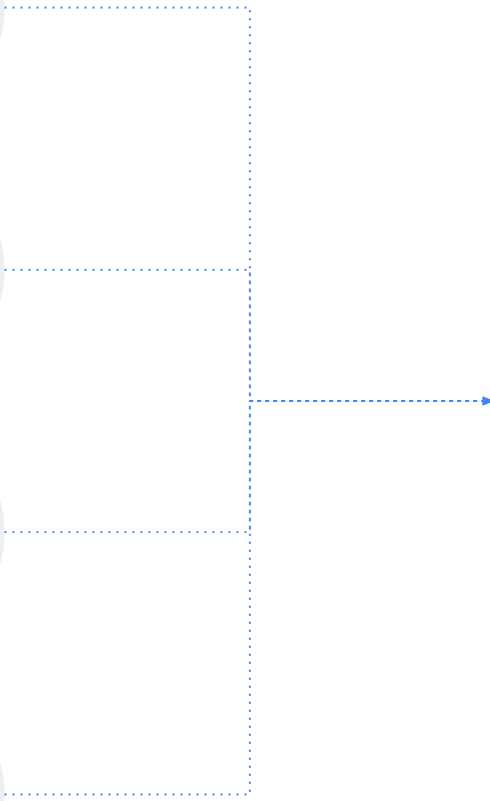
Trait Initialization

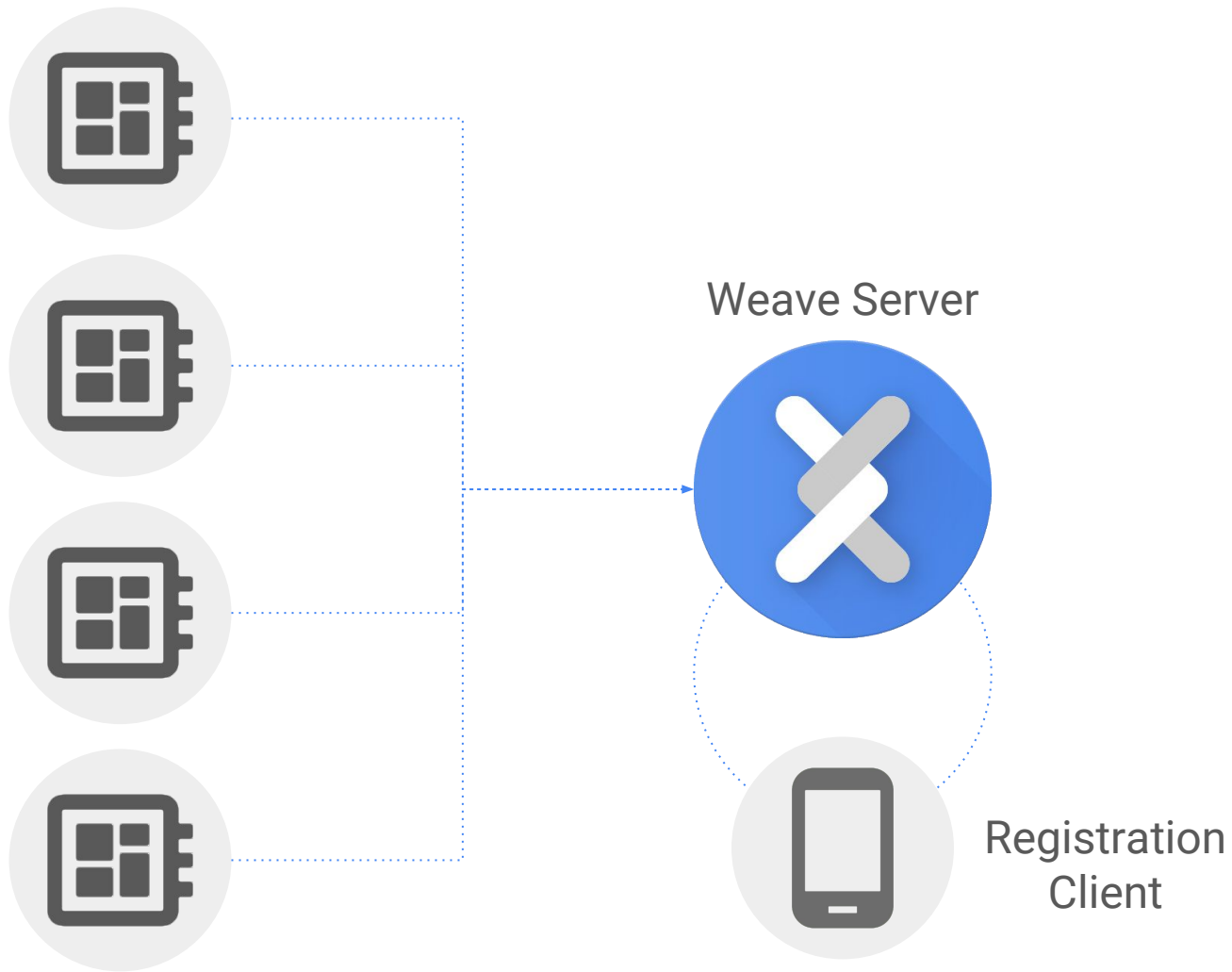


Weave Server



Weave Server



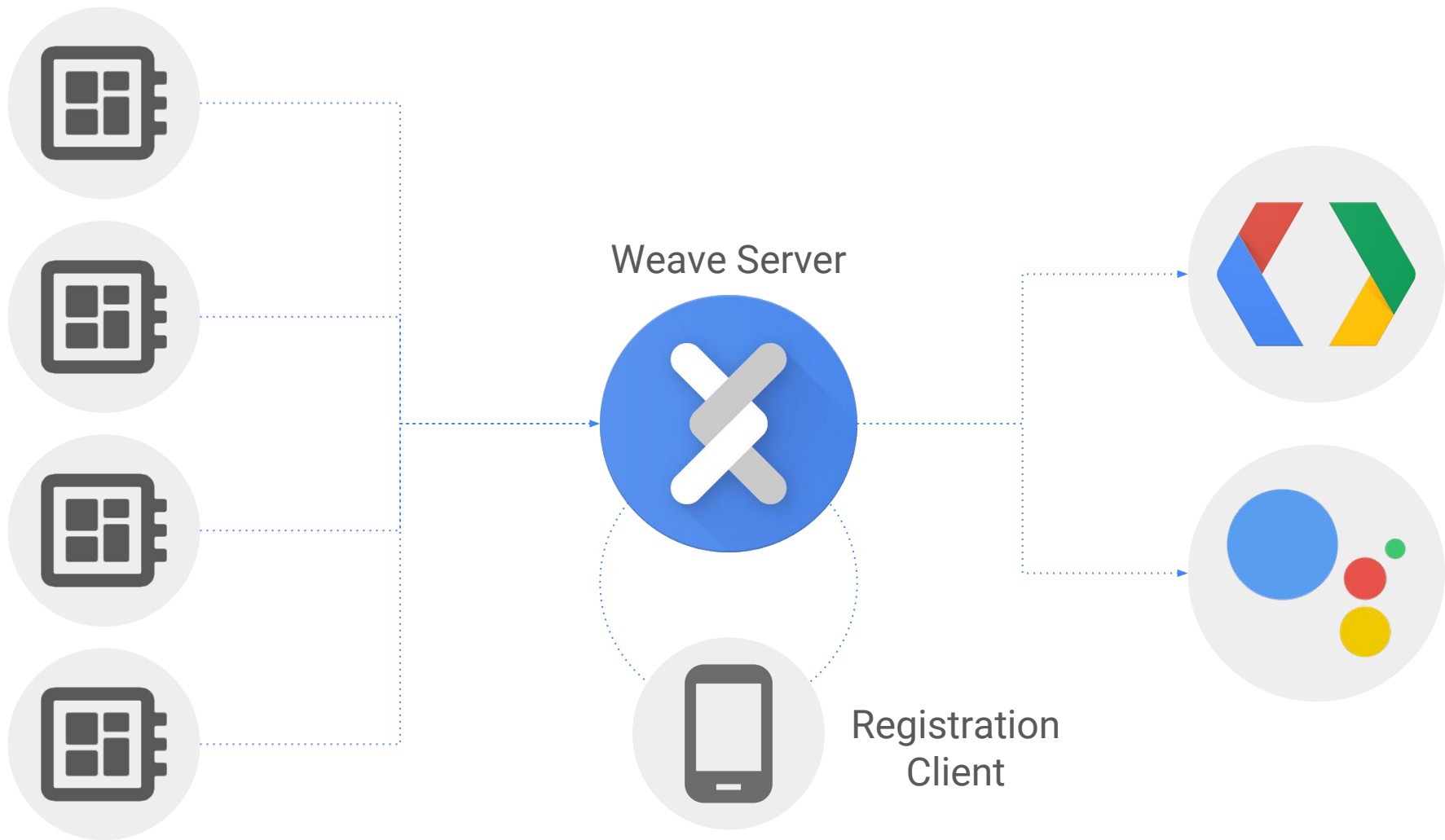


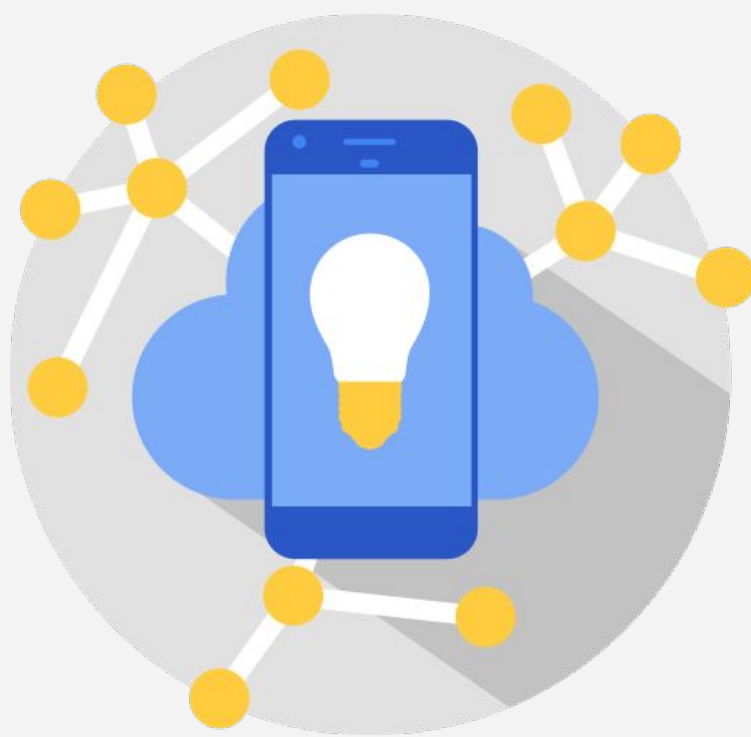
```
char* registration_ticket = ...;

// Notify libiota that the device is connected to the network
host_iota_daemon_set_connected(iota_daemon, true);
// Pass the registration ticket to libiota
host_iota_daemon_register(iota_daemon, registration_ticket,
                          registration_complete_, registration_ticket);
```

```
// Handle registration result
static void registration_complete_(IotaStatus status, void* context) {
    if (!is_iota_status_success(status)) {
        IOTA_LOG_INFO("Registration Failed, Status=%d.", status);
    } else {
        IOTA_LOG_INFO("Registration Succeeded.");
    }
}
```

Device Registration





Weave Developer Tools



Android Things



Weave Products



Weave Test Lab



Help



Send Feedback



CoolLightbulb



CONFIGURE

MONITOR

Product settings

Weave configuration

Weave models

Product settings

Cloud project

[LightbulbCloud](#)



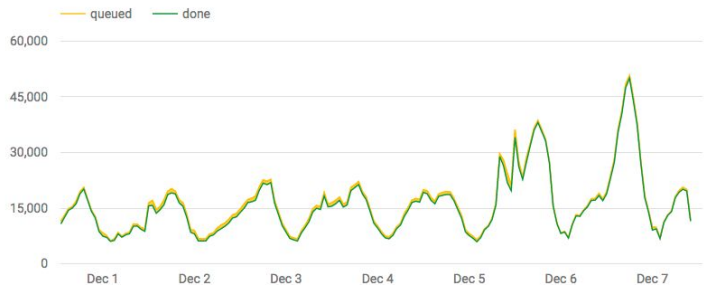
Product name

CoolLightbulb

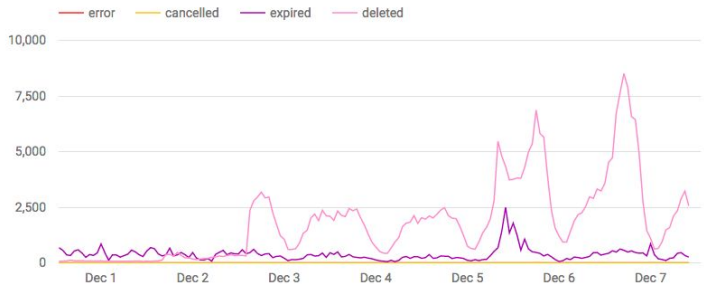


SAVE CHANGES

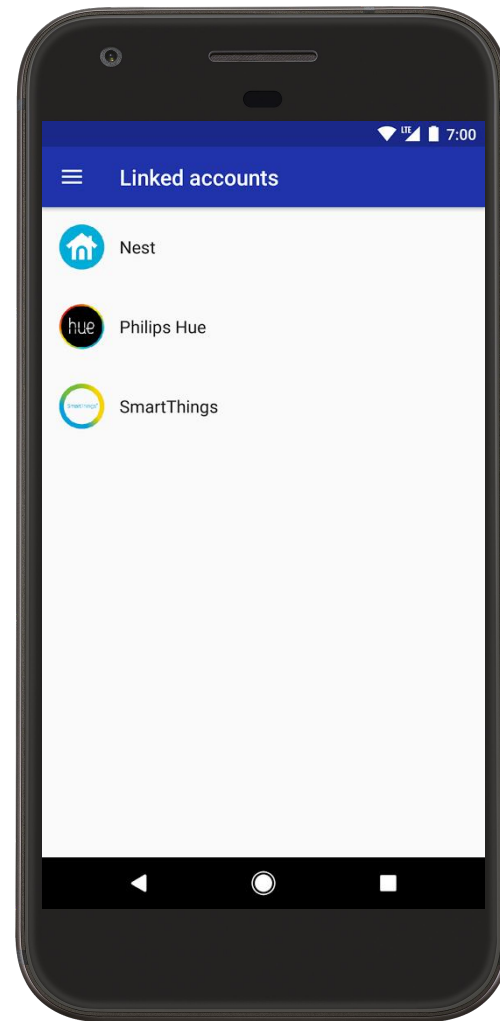
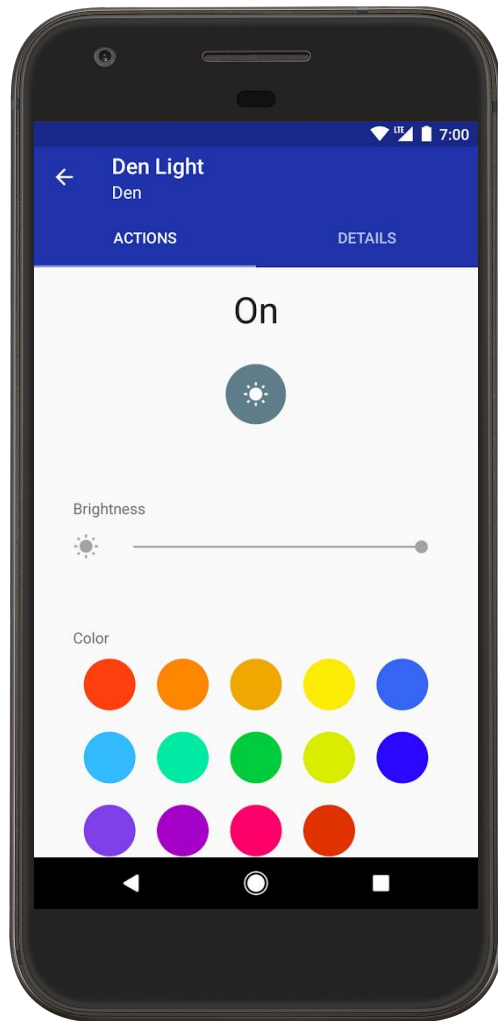
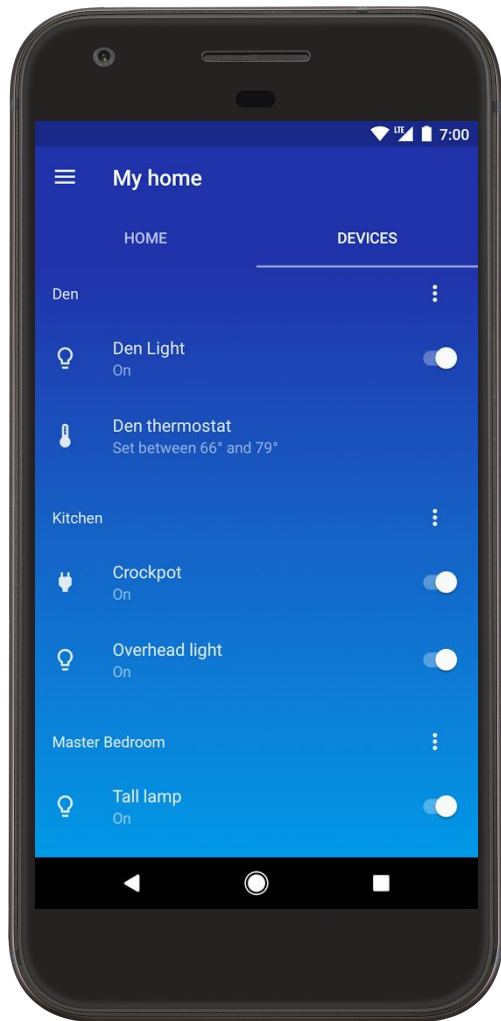
Weave commands queued by devices per hour and reaching done state



Weave commands with terminal states other than done per hour



Metrics and Reports





Weave



Google's IoT Developers Community
<https://g.co/iotdev>



Google's IoT Solutions
<https://iot.google.com>



Weave Documentation
<https://developers.google.com/weave>



Zviad Kardava - Google Developer Expert: IoT
+ZviadKardava
@ZviadKardava
fb.com/Zv1ad



IX МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Google Developers Experts: Internet of Things

zviadkardava@gmail.com

[@zviadkardava](https://twitter.com/zviadkardava)

