Mobile Device IoT using Blynk

Frank Walsh

Mobile Devices and IoT

- Critical component of many IoT solutions is the mobile phone/tablet.
  - IOS/Android dominate
- >2.6 billion users worldwide
  - Simple, mobile connection to the internet
- Provides nice features for IoT apps
  - Packed with sensors (Location, accelerometer, camera)
  - Can connect/interlink other smart devices using Bluetooth, BLE, etc.

Mobile Apps in IoT, Examples

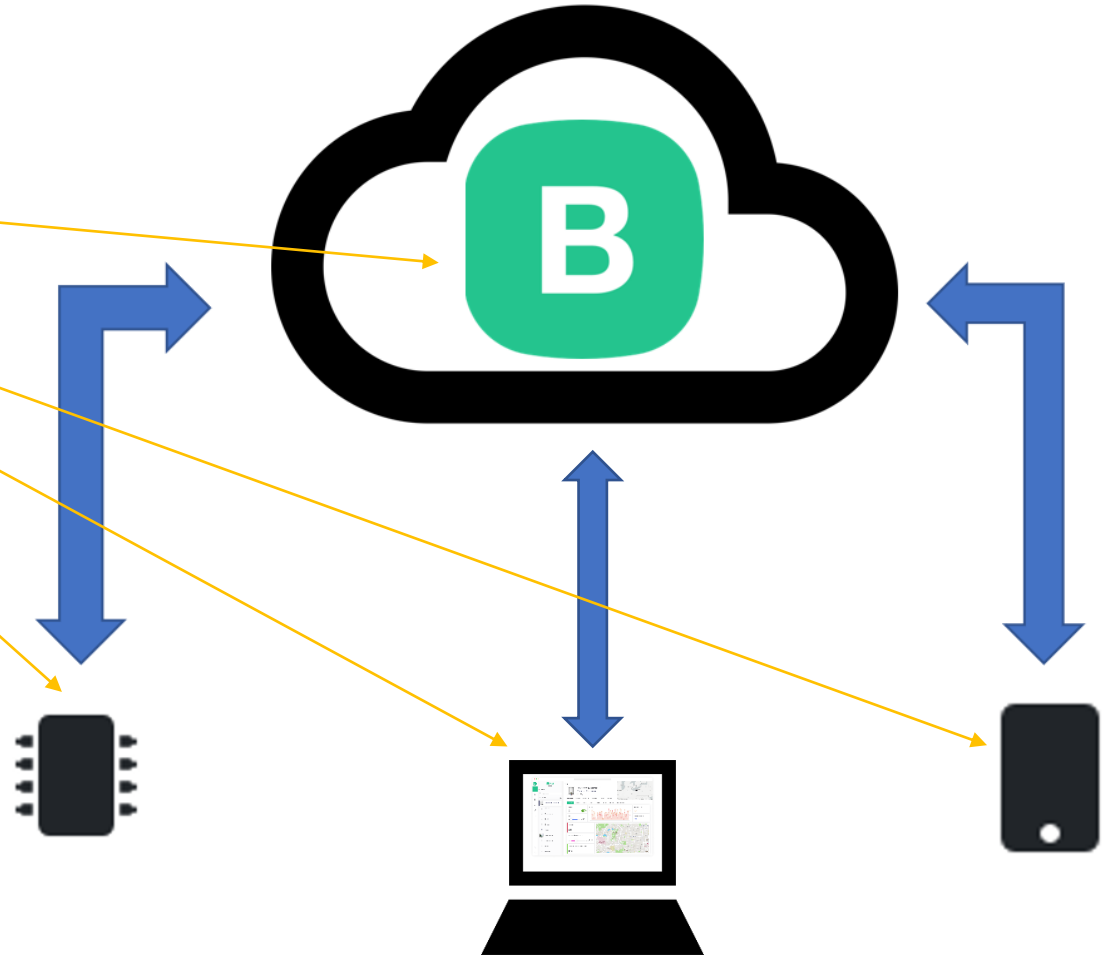
- Wearables
  - wristwatches, eyeglasses and rings
- Healthcare
  - Medical sensors obtain health data and transfer to a mobile app.
  - This data can be transferred remotely to doctor/ family members
- SmartHome
  - Nest – see who's at the door...
- AgriTech
  - MooCall

What's Blynk

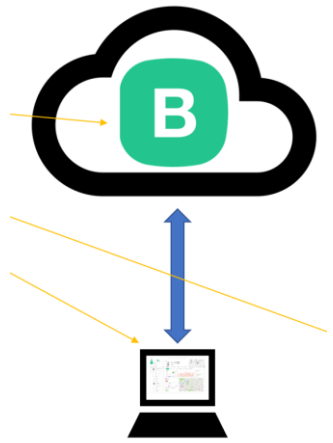
- Yet Another IoT Platform
  - Specialism: mobile application builder
- "Blynk is a full suite of software required to prototype, deploy, and remotely manage connected electronic devices at any scale: from personal IoT projects to millions of commercial connected products."

Platform Overview

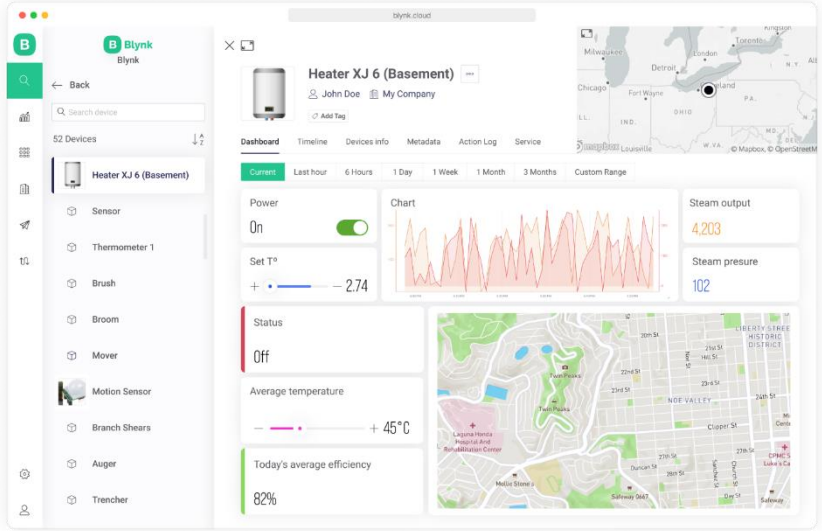
- 4 Compnents
  - Blink.Cloud
  - Blynk.App
  - Blynk.Console
  - Blynk.Edgent (Device Libraries)



Blynk.Console

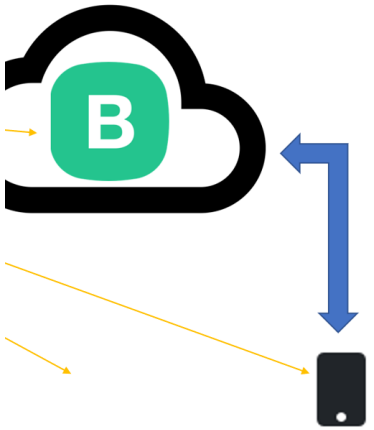


- **Blynk.Console** is a feature-rich web application
- Register with Blynk to gain access
- Use it to:
  - Configuration of how connected devices work on the platform + application settings.
  - Management of devices, their data, users, organizations and locations
  - Remote monitoring and control of devices



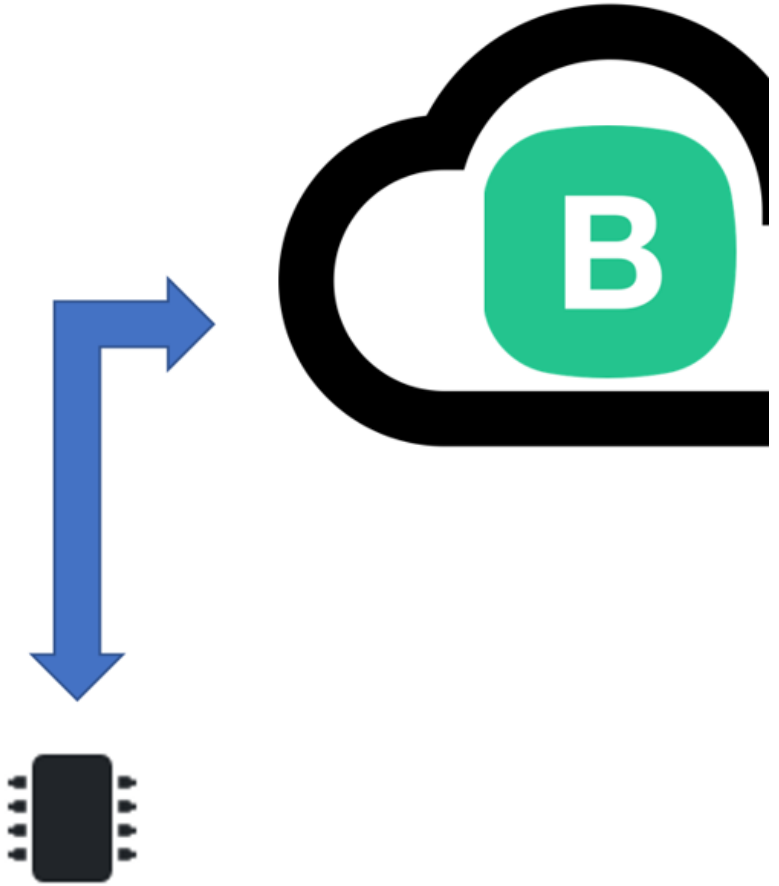
Blynk.App

- **Blynk.Apps** is a multi-functional native iOS and Android mobile application
- Use it to:
  - Remote monitoring and control of connected devices that work with Blynk platform
  - Configuration of mobile UI during prototyping and production stages
  - Automate work of connected devices
- Allows you to create mobile applications using drag and drop
  - Uses "widgets"
- No Code!!!
  - Don't worry – that comes later



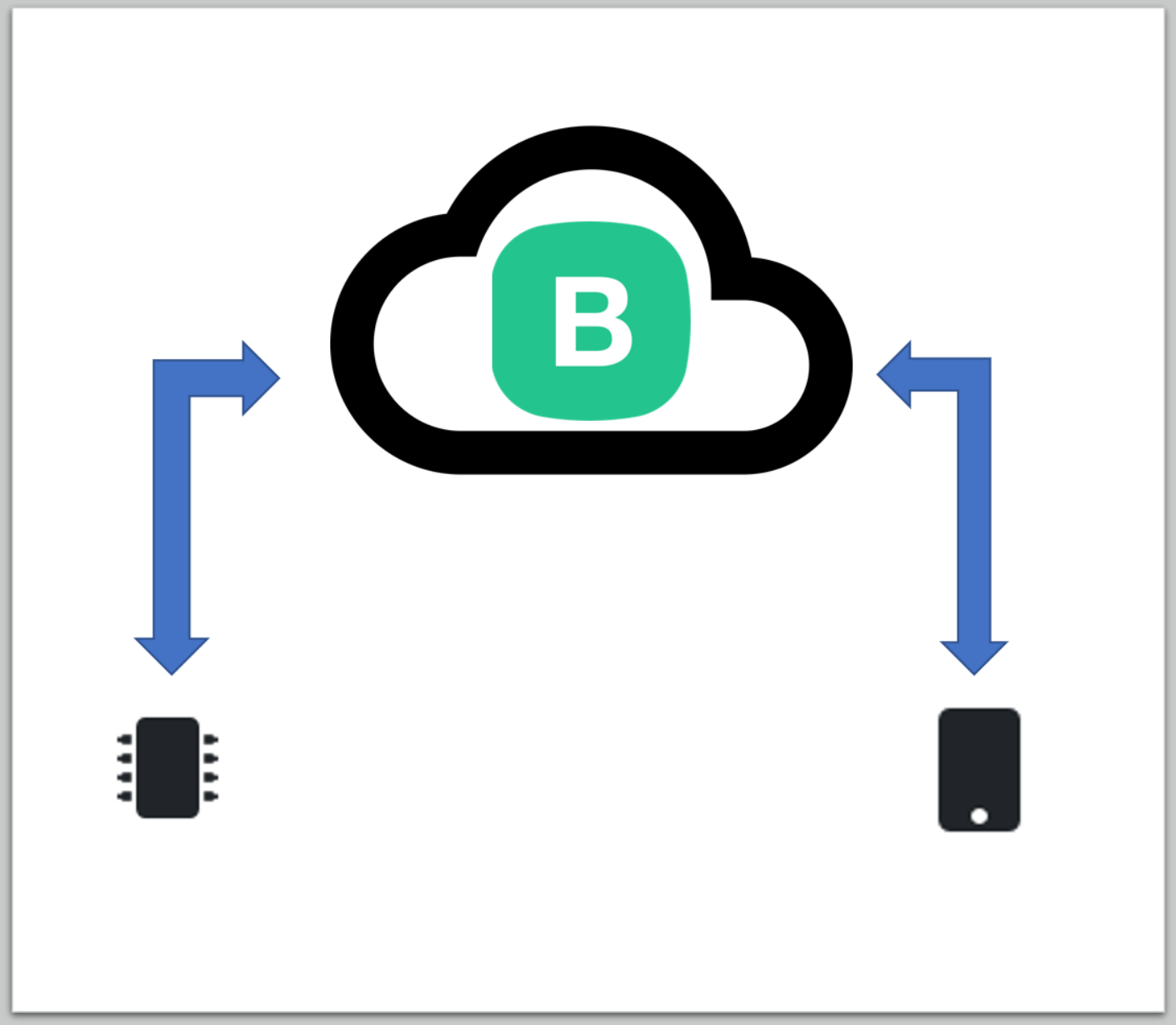
Blynk.Edgent(Device Library)

- Lightweight embedded library that runs on over 400 supported hardware models
- Use it to:
  - bring your device online and authenticating)
  - Connectivity management (WiFi, Cellular, Ethernet)
  - Data transfer Over-the-air firmware updates (for selected hardware models)
- Available for most popular hardware platforms
  - RPi, Arduino, ...
- Enable communication between the device and cloud API to work with specific Blynk.Apps and Blynk.cloud features
- Python, Javascript, C++ … on the RPi.



Blynk.cloud

- **Blynk.Cloud** is a server infrastructure – heart of Blynk IoT platform. Cloud is responsible for binding all the platform components together.
- Controls all the communications between the mobile device(e.g. your phone) and hardware (e.g. the RPi)
- Remember we talked about the benefits of indirect communication.
- It's Cloud-based but you can run your own private Blynk Server
- A bit like MQTT...(actually, after peeking at the source code, it uses the same MQTT libraries from the previous lab!)



Benefits

- Don’t need to be a mobile app developer
- Minimal code
- Reasonably mature (around since 2014)
- Very quick to create a prototype
- If you want, process to publish to App store/Google Play

Similar API & UI for all supported hardware & devices

Connections/ Protocols:

WiFi

Bluetooth and BLE

Ethernet

USB (Serial)

GSM

…

Set of easy-to-use Widgets

Connect device with no code writing

Easy to integrate and add new functionality using virtual pins

History data monitoring via SuperChart widget

Device-to-Device communication using Bridge Widget

Sending emails, tweets, push notifications,

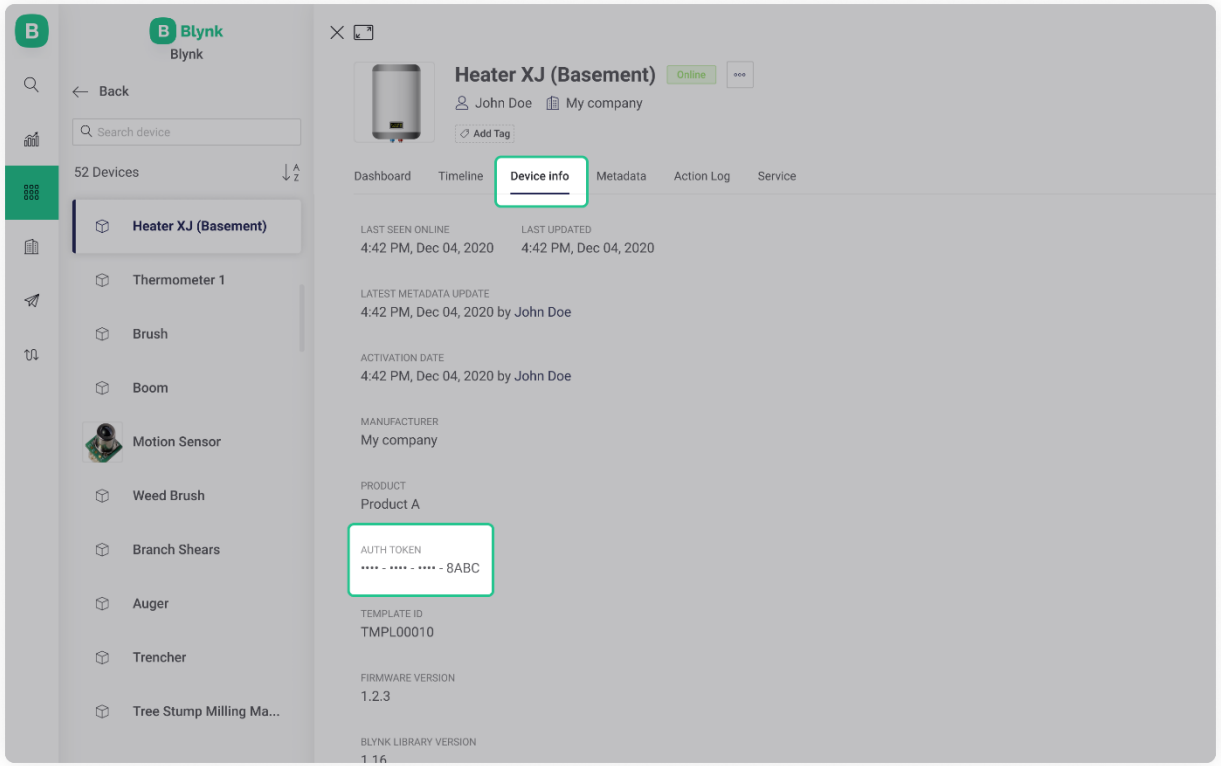
Features

Blynk Concepts

- Devices
- Templates
- Users

Blynk Concepts: Device

- A “device” can be:
  - A small MicroController based hardware (e.g. Arduino, Raspberry Pi, etc)
  - A finished physical product like a Smart Air Conditioner, or a virtual service that sends data to Blynk.Cloud using REST API.
- Each device has an “Authentication Token”. This is a unique identifier generated in Blynk.Cloud
- Every device uses a **Device Template.**

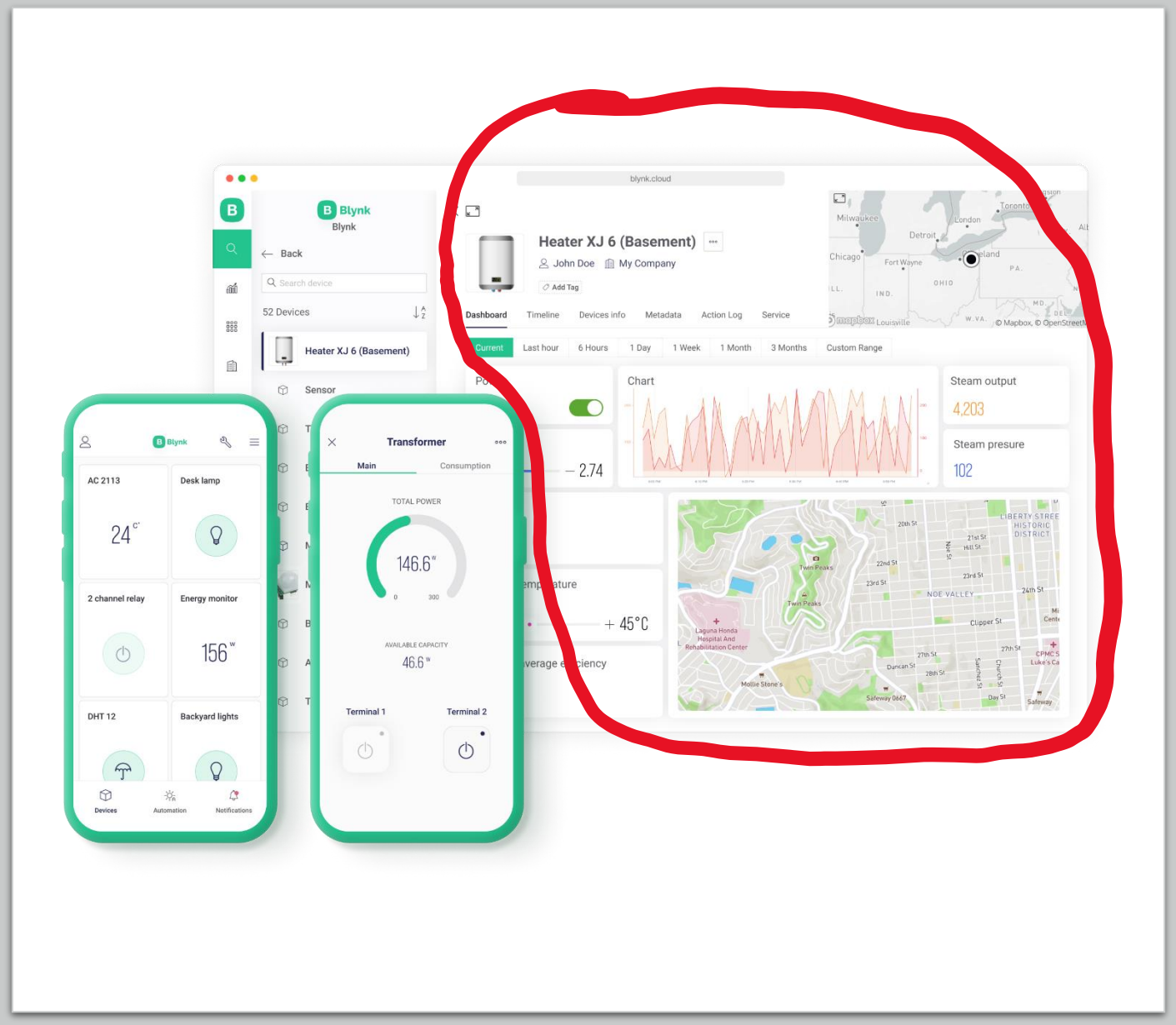


Blynk Concepts: Device Template

- **A Device Template is a set of configurations used by similar devices.**
- A device template specifies:
  - **General Settings:** general settings of the device(name, ID etc..)
  - **Metadata:** a table of **key:value** data attached to every device. For example Serial Number field, email field to use for email notifications
  - **Datastreams:** channels for any time-stamped data that flows in and out from the device to the cloud. For example sensor data should go through a Datastream. These are also known as “Virtual Pins”.
  - **Events: i**mportant events in the life of the device that should be logged and, if needed, used for notifications. Events can be triggered from the device itself or externally using HTTP API
  - **Dashboards:** Set of UI elements to visualise data and send commands/data to/from device
    - There are two dashboards: Web and Mobile

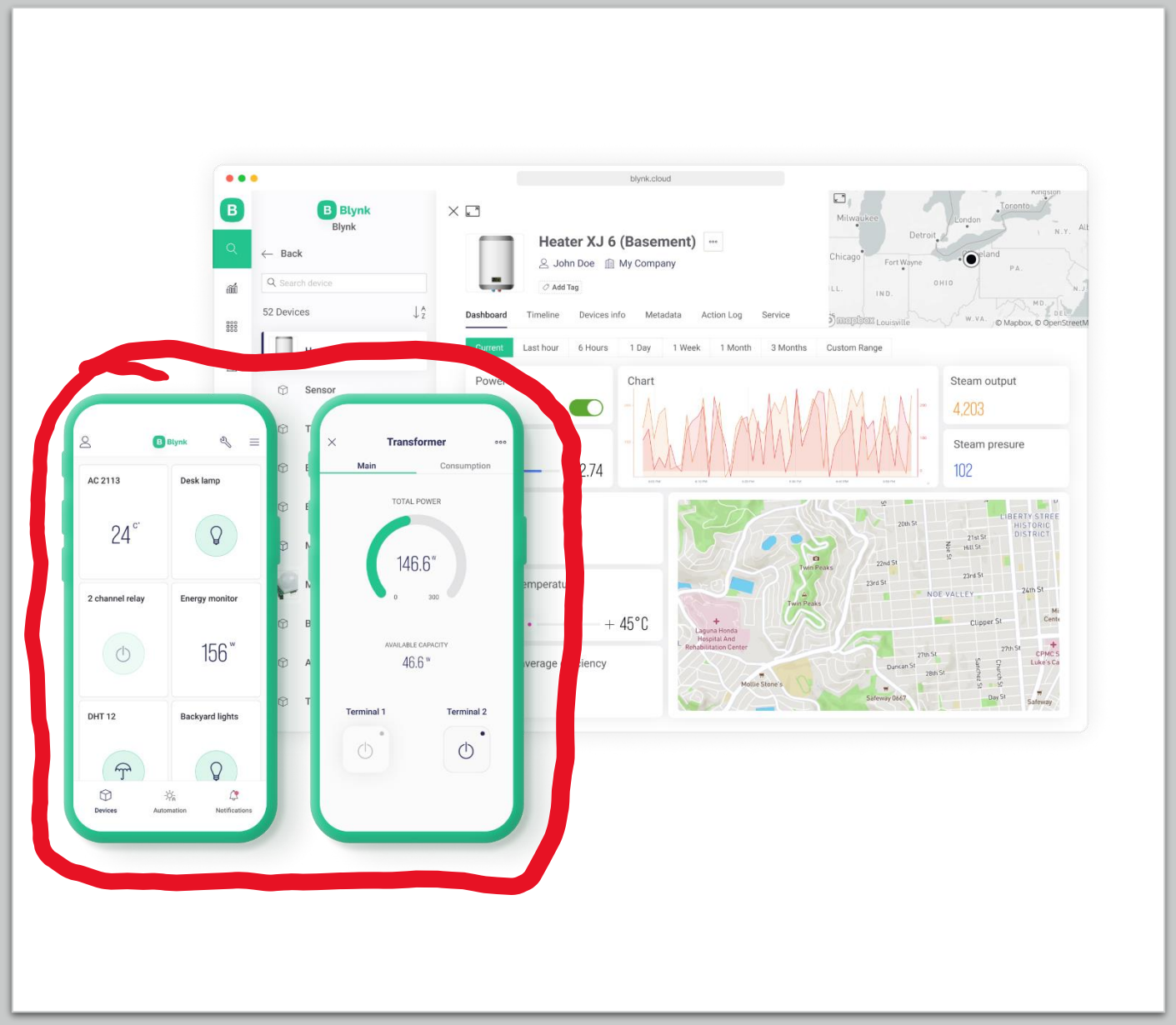
Blynk Concepts: Web Dashboard

- a set of UI elements (widgets) to visualize the data from the device accessible for the users in Blynk.Console – a web-based application.



Blynk Concepts: Mobile Dashboard

- a set of UI elements (widgets) to visualize the data in Blynk mobile apps for iOS and Android.
- Mobile apps also contain a template of how device is represented in the list of devices (tiles)



Development Steps

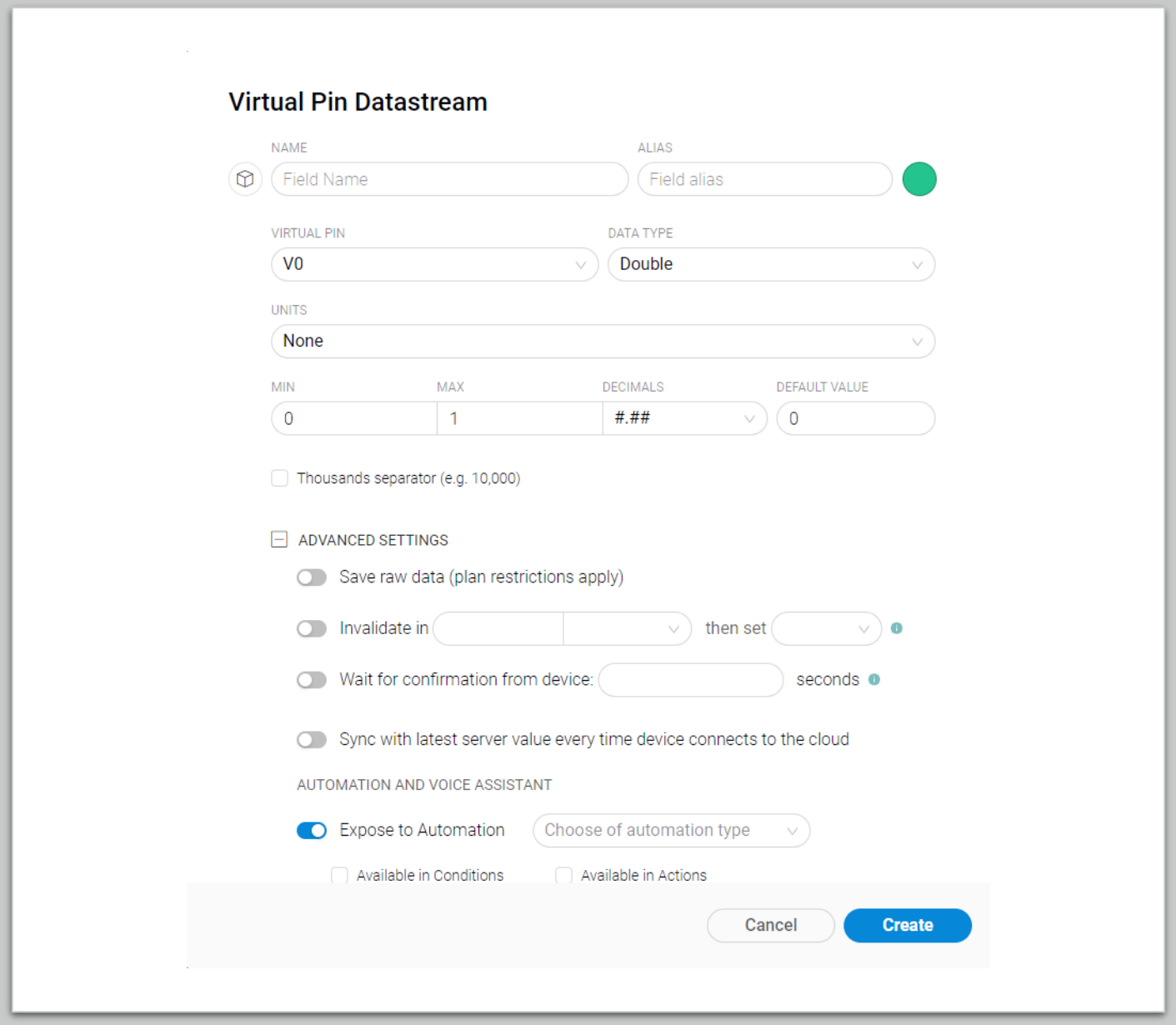
1. Create an Account with Blynk
2. Add a Device
    1. Choose your hardware
    2. Create “Template”
    3. This will result in an Authentication Credentials for that Device
3. Code the Hardware Device(e.g. Rpi)
    Use the relevant library(e.g. Python Library)
4. Install Blynk App on Phone
    1. Develop Blynk

Add a Device

- Create a new device
- Set Notifications:
  - Specify who gets notifications, which events will trigger notifications, and which channel should be used.
  - 3 options:
    - E-mail
    - Push notifications - a push notification is a message that pops up on a mobile device. Users don't have to be in the app to get them
    - SMS - notification will be delivered as a text message using mobile operator (only available in paid option though…)

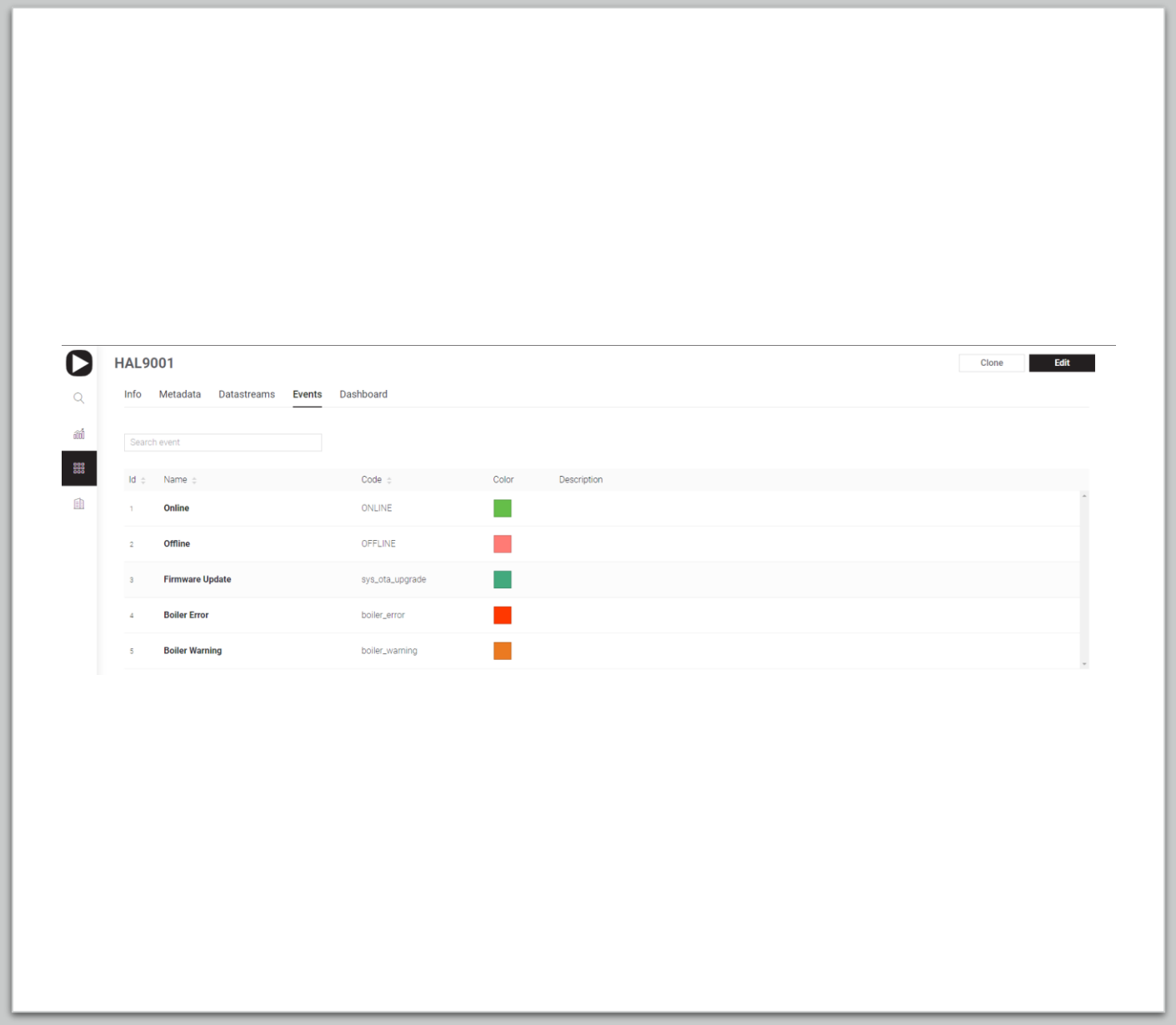
Working with Templates

- Datastreams is a way to structure data that regularly flows in an out from device.
- Use it for sensor data, any telemetry, or actuators.
- Virtual Pin is a concept invented to provide exchange of any data between hardware, web and mobile app.
- Virtual pins allow you to interface with any sensor, any library, any actuator.Think about Virtual Pins as a box where you can put any value, and everyone who has access to this box can see this value.
- You can set UNITS that will be viewed in the Widget by selecting them from the dropdown menu.
- It's a very powerful feature to display and send any data from your hardware to the application.**Please make sure you differentiate Virtual Pins from physical GPIO pins on your hardware.**



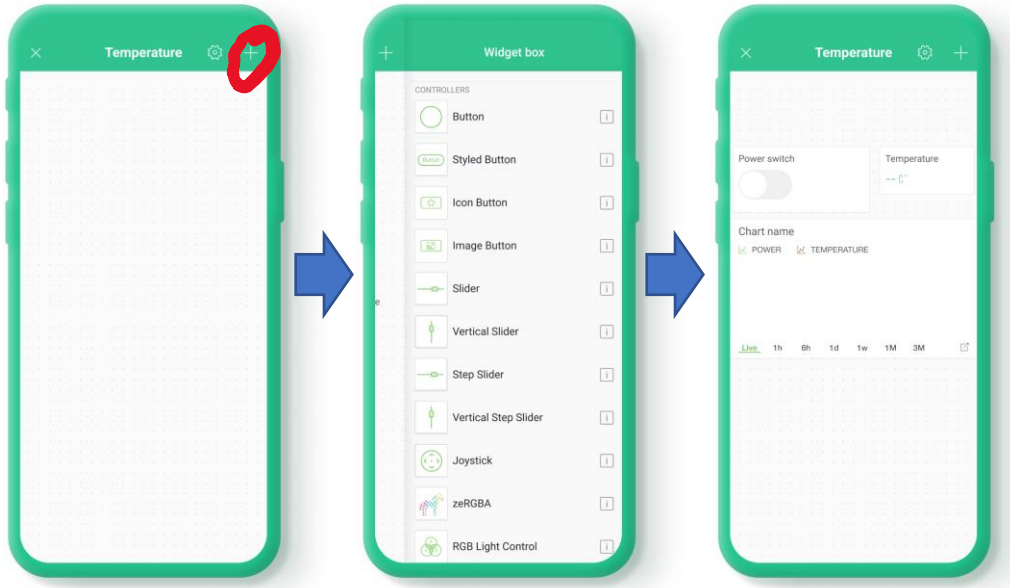
Working with Templates: Events

- Events are used to track important events happening on your devices.
- Events can trigger notifications which can be
  - sent over email
  - delivered as push notifications to user's phone
  - SMS
- Examples:
  - log a moment when a temperature reached a certain threshold and send a notification to selected users
  - You need to log a total working hours of the device. If it approaches or goes beyond a max value, you would need to notify technical support so that they can replace the device



Blynk.Apps

- Using the Blynk App, a Mobile Dashboard can be built from Widgets - modular UI elements which can be positioned on the canvas.
- To add Widgets to Canvas:
  - tap anywhere on empty canvas or press PLUS icon in the top right corner of the app.
  - A Widget Box will open
  - Find the widget you need and tap on it
  - Widget will placed in the first empty area required for this widget
  - Configure Widget.



Widgets

- Categorised as:
  - Controllers

  - Display

  - Notifications

- Interface
  - Tabs, Menu, Map..

Example

DEMO!!!!!