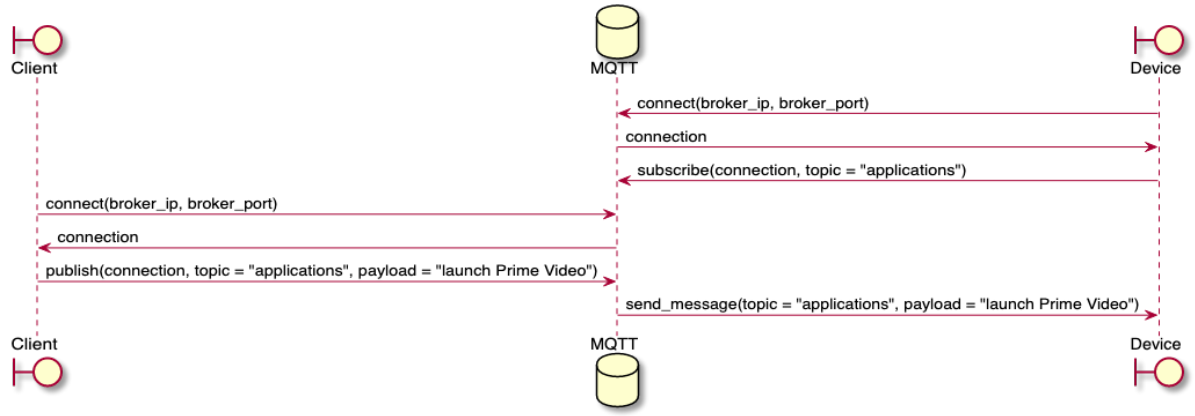
Security Overview

Device Automation Bus (DAB) is a protocol that enables the automation of a device. The protocol is built on top of the popular IoT protocol MQTT. To read more about MQTT, follow https://mqtt.org. MQTT is a communication protocol that requires a broker (e.g., mosquitto) to facilitate communication between the connected parties. It implements a publish/subscribe architecture built using topics.

A sample workflow using MQTT protocol:



Architectural note:

- A device is assigned to the dedicated MQTT broker.
- A device with the embedded broker would simply connect to localhost and the predefined port.
- DAB is operational once the device connects to its localhost broker and is ready to respond to the DAB commands/requests.

There are two basic states of DAB:

- **Disabled**. In this state, no remote control is possible. DAB will be considered disabled when any of the following is true::
    - The MQTT broker process is not running.
    - The MQTT broker is not accepting any incoming connections.
    - The MQTT broker is running but the device is not responding to any of the messages.
    - In a “bridged” implementation, the bridge process is not running or is not connected to the client device.
- **Enabled**. In this state, the clients can connect to the broker associated with the device, the DAB component on the device will be subscribed to the topics required by the specification, and the device will respond to the DAB messages published to these topics.

Why Does DAB Need to Be Secure?

DAB needs to be secure because it allows an actor to gain a similar level of control over a device to that of a user in physical proximity to that device.

A device with DAB in a **disabled state** is considered **secure.**

DAB provides the means to control a DAB-enabled device remotely over the network to aid the automation of testing/certification of devices. For example:

- Launching application(s)
- Interaction with the device (key presses)

The security layer is needed to mitigate the risk of any unauthorized party being able to remotely control the device.

Recommendation

We believe there are simple ways to enable/disable DAB on production FW through partner controlled or user-controlled enablement approaches.

Partner controlled approach

This approach will imply that the device partner would enable/disable DAB remotely on a production device upon request from one of the App partners supporting DAB. An example of this approach would be the remote provisioning or installation of DAB utilities on a device identified by partner-specific deviceId but located at one of the App partner's facilities supporting DAB. Once the protocol is enabled, clients can freely connect to the broker and publish messages to start controlling the device. This security recommendation makes DAB suitable only for partner-controlled debugging or development/automation scenarios.

User controlled approach

This approach will imply that the protocol is explicitly enabled/disabled by a person that has physical access to a DAB-capable device. As an example, the ADB model is a simple and effective way of providing security by using a keypress combination in a specific location of the device UI, which is unlikely to be generated by an end-user by accident.