MACHINEQ™

MachineQ OTA Updates – LoRaWAN Quick and Simple

09 . 12 . 19

- Introduction to MachineQ
- Demo – Delta FUOTA
- LoRa
- LoRaWAN
- Unicast Messages vs FUOTA
- FUOTA
- Questions

Introduction to MachineQ

Introduction to MachineQ

MachineQ is a Comcast Company focused on Enterprise IoT in the market

Founded in 2017, we help put the power of the Internet into everyday things, giving businesses access to vital information that streamlines their operations

MachineQ provides integrated, enterprise-grade hardware and software to serve as common infrastructure for low-bandwidth IoT solutions

- Premium LoraWAN network provider
- Super easy to setup and manage your network and devices
- Very cost-effective pricing model
- Great gateway and device health visualization
- Easy to use APIs make integration painless
- Device agnostic: Use our devices, BYOD, or design your own
- We can help you build your own custom device
  - We are an ST Partner and prefer STM32 MCUs
  - We can help ensure your device is complaint to the LoraWAN spec
  - Co-marketing and branding are possible as well

Hardware

Connect your devices from anywhere and collect, deliver, and process data into insights

Software

Take control of our highly-scalable, cloud-based hardware to maximize efficiency

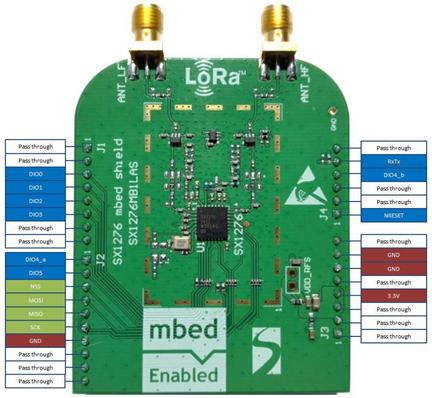
Demo – Delta FUOTA

Demo: Integration

- **Gateway:** MachineQ Area 8c
- **MCU:** ST STM32L476
- **LoRa Transceiver:** Semtech SX1276
- **Bootloader:** ST X-CUBE-SBSFU - Secure Boot and Secure Firmware Update
- **Stack:** Semtech LoRaMac-node
- **FUOTA:** MachineQ FUOTA library for STM32
- **Flash Management**: MachineQ flash abstraction
- **Image Compression:** Basicloader - Delta Update
- **IDE:** OpenSTM32 System Work Bench

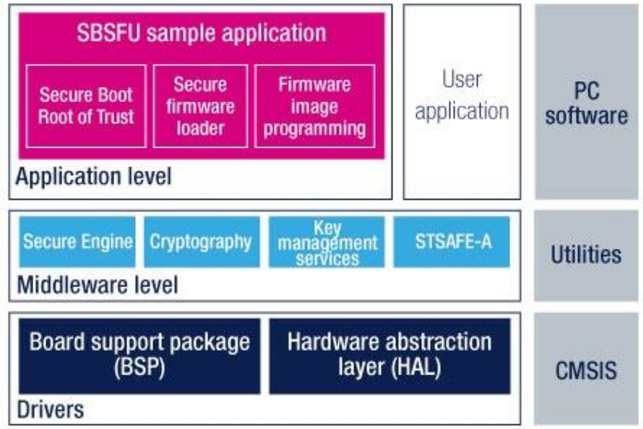
https://store.machineq.com/store/products/area-8c

https://github.com/lorabasics/basicloader



https://os.mbed.com/components/SX1276MB1xAS/

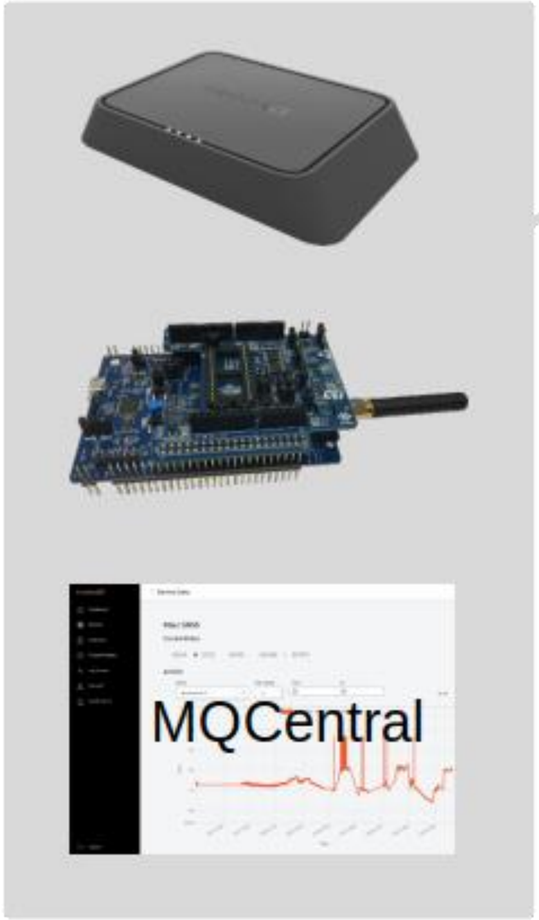
https://www.st.com/en/evaluation-tools/nucleo-f030r8.html



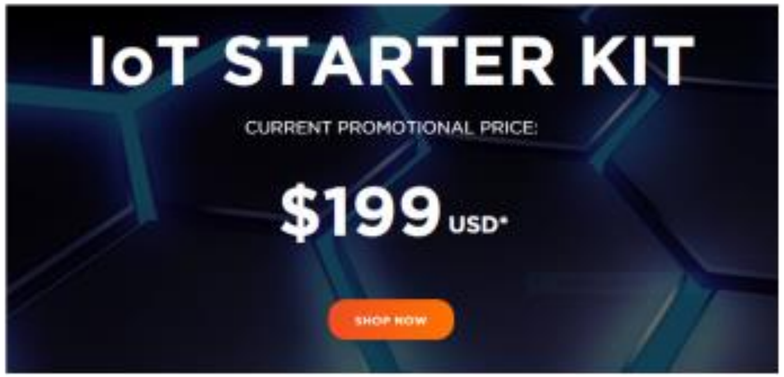
https://www.st.com/en/embedded-software/x-cube-sbsfu.html

Demo: Integration

MACHINEQ
A COMCAST COMPANY



ST LoRa® Dev-Kit Deal



ST LoRa Dev Kit

The ST LoRa development board and sensor is an advanced development kit optimized for engineers that are ready to prototype scalable solutions towards a complete IoT implementation.

https://store.machineq.com/store/products/machineq-starter-kit

Includes:

1 MachineQ Gateway (8 channel)
1 ST LoRa® Dev Kit (Select “Dev Kit: ST LoRa” in menu)
1 Year MQcentral IoT Management

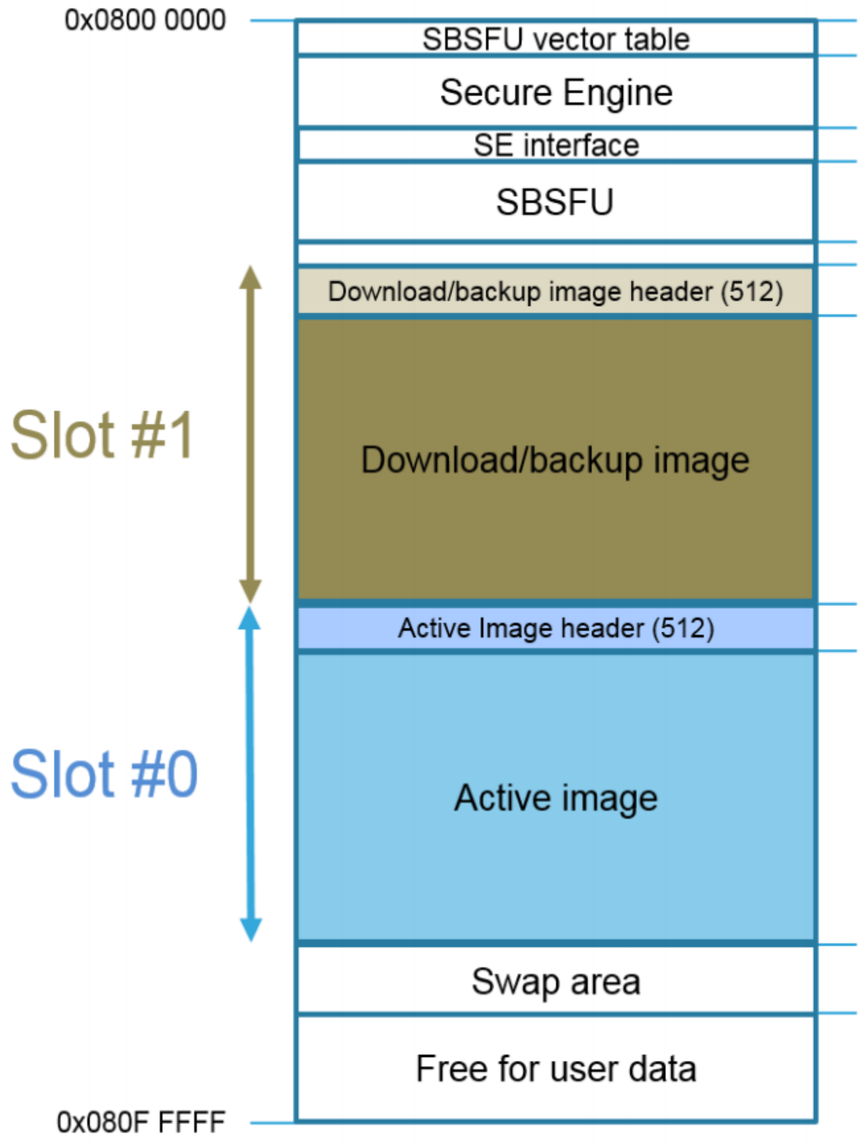
Demo: Process

1. **Image Generation: SBSFU + Basicloader Integration**
   1. LZ4 diff generation
   2. Encryption
   3. Dual header signing

2. **MQ FUOTA lib + MQ Flash Abstraction**
   1. Call back registration with MQ FUOTA lib
   2. Multicast Class C setup
   3. Fragmentation setup
   4. Image reconstruction using LDPC FEC in Slot #1

3. **Image Decryption + Decompression: SBSFU + Basicloader Integration**
   1. Decrypt delta image and swap slots
   2. Verify delta image
   3. Move to delta image to top of Slot #0
   4. LZ4 decompress to base Slot #0
   5. Verify decompressed image
   6. Boot new image

4. **MQ FUOTA lib**
   1. Report new firmware version



Demo: Availability For Qualified Customers

- **MQ FUOTA API**
  - **Status**: Beta
  - **Note**: Requires some setup before use

- **MQ FUOTA Lib**
  - **Status**: Released
  - **Note:** Slim customizable static library based on required features, MCU and FP implementation

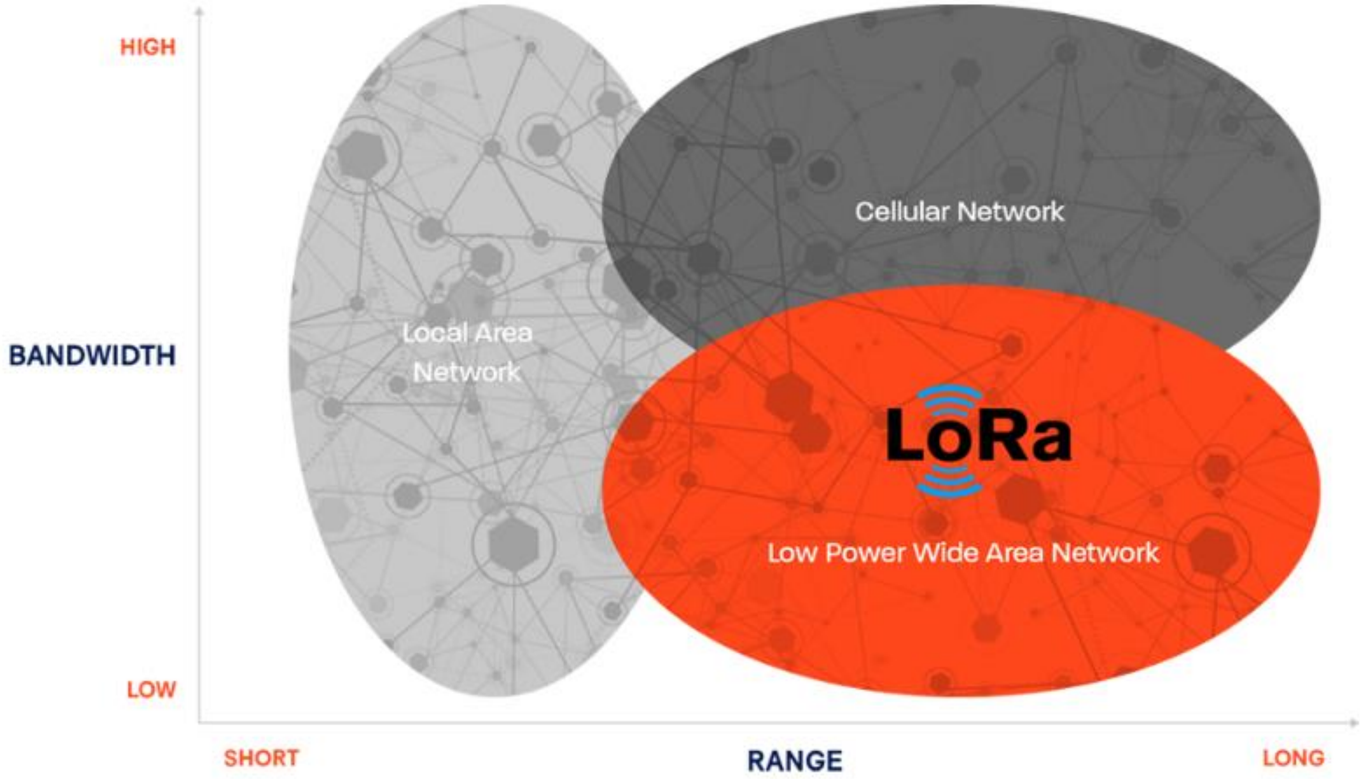
- **SBSFU + FUOTA + Compressed Update**
  - **Status**: Beta
  - **Note**: Currently tied to STM32L476

LoRa

LoRa: Introduction

Robust, long range wireless transmission

- **Long Range** - A few kilometers in urban areas to over 10 km in rural settings. Effective indoors and underground.
- **Low Power** - Can run on batteries for up to 10 years
- **Low Cost** - Lightweight protocol combined with low cost transceiver



LoRa: Modulation

- Chirp modulation tailored to long range or high noise environments
- A chirp is a frequency sweep
- The frequency direction encodes the symbol or bit being transmitted
- The sweep rate or Spreading Factor defines the data rate
- The faster the sweep the higher the data rate
- Higher data rates come at the expense of lower receive sensitivity
- Low data rates increase sensitivity, but increase the probability of interference / collision
- How data rate / spreading factor and TX power are managed is defined in the LoRaWAN spec

Data Rate	Configuration	bits/s	Max payload
DR0	SF10/125kHz	980	19
DR1	SF9/125kHz	1 760	61
DR2	SF8/125kHz	3 125	133
DR3	SF7/125kHz	5 470	250
DR4	SF8/500kHz	12 500	250
DR8	SF12/500kHz	980	41
DR9	SF11/500kHz	1 760	117
DR10	SF10/500kHz	3 900	230
DR11	SF9/500kHz	7 000	230
DR12	SF8/500kHz	12 500	230
DR13	SF7/500kHz	21 900	230

LoRaWAN

LoRaWAN

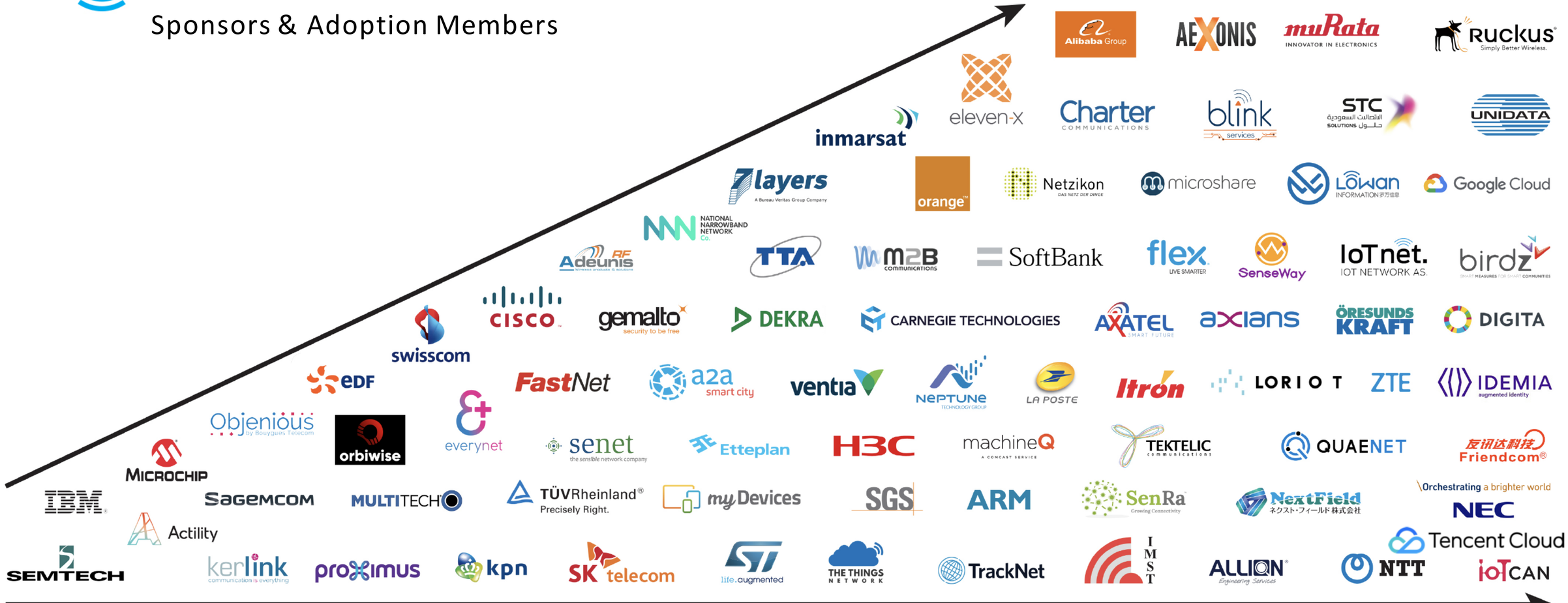
LoRaWAN™ is an open, global standard for secure enterprise-grade IoT connectivity. Deployed by 80+ major carriers worldwide, LoRaWAN has emerged as the de-facto global network protocol for low-power wireless IoT connectivity.

- LOW COST
- LONG RANGE
- DEEP COVERAGE
- LOW POWER
- SECURE
- SCALABLE
- STANDARDS-BASED
- GLOBAL ECOSYSTEM

LoRa Alliance™

Sponsors & Adoption Members

+ >300 Adopter Members



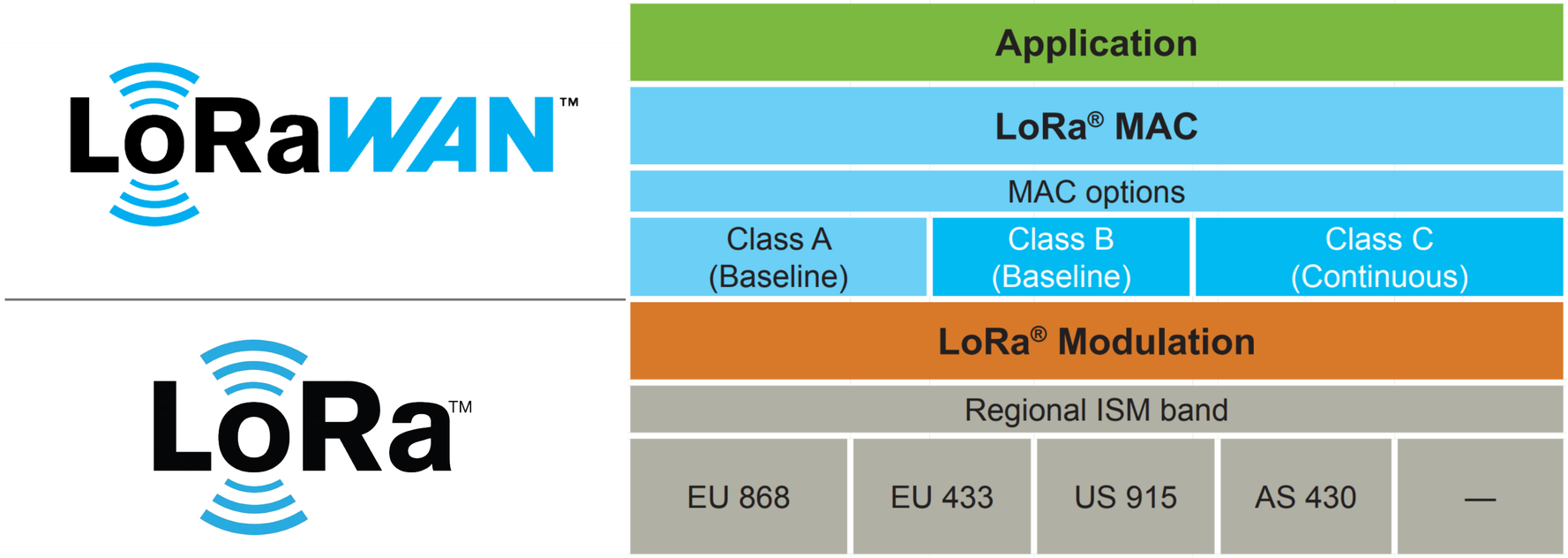
2015

2018

LoRa Alliance™

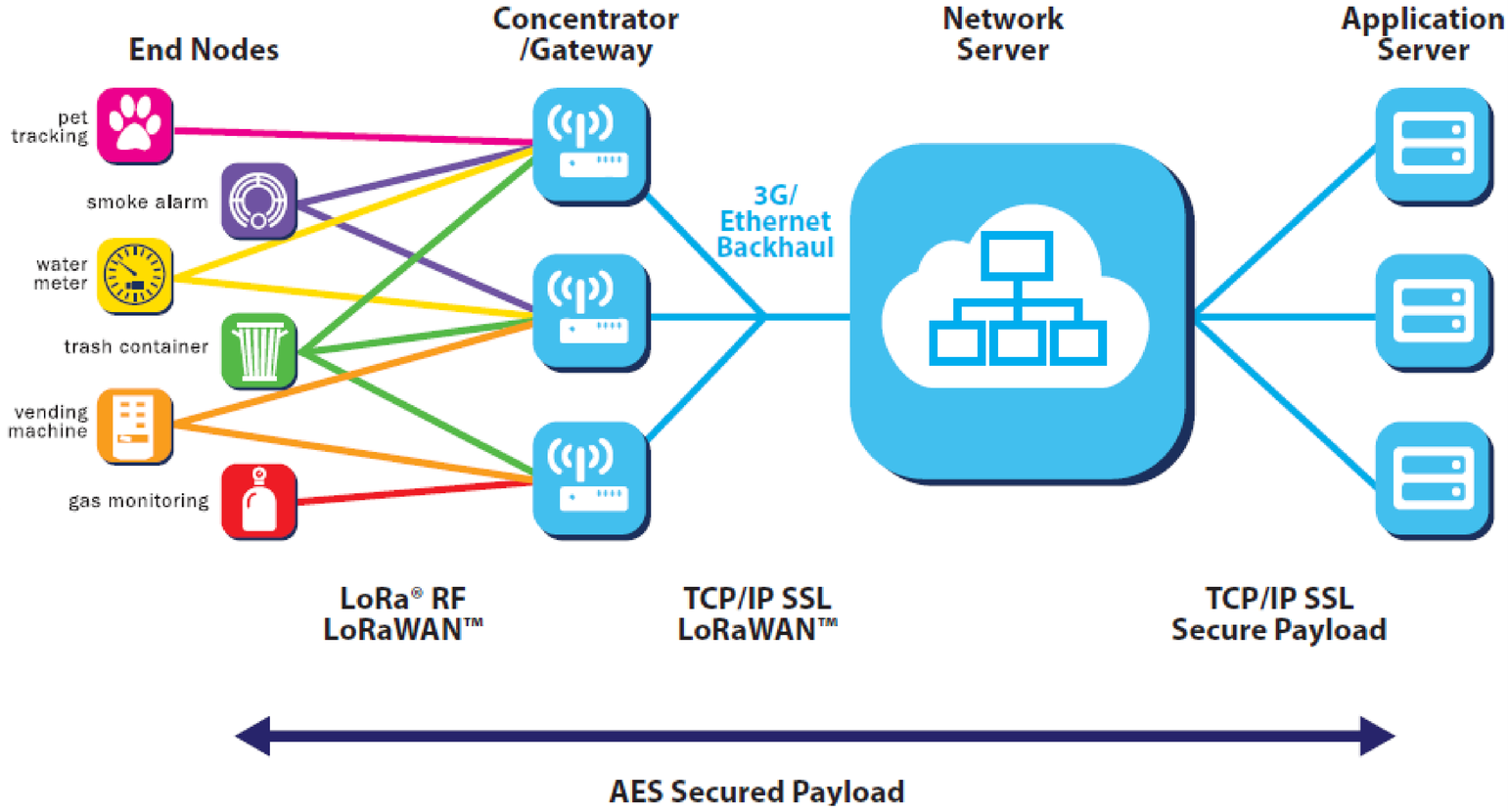
LoRaWAN: PHY vs MAC

- LoRa defines the PHY or how bits are physically transmitted over the air
- LoRaWAN defines MAC layer or how that physical layer is used
- Defines the “rules of the road” so that thousands of devices can effectively share the same spectrum and even the same access point



LoRaWAN: Network

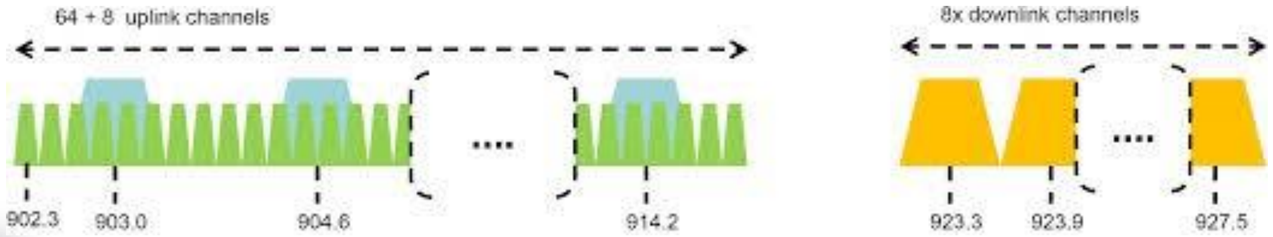
- Star-of-stars topology
- Devices use single-hop LoRa
- GWs are RF-to-IP bridges
- All messages are encrypted with an AES-128 network key
- Payloads are encrypted a second time with an application key
- Message may be received by multiple access points

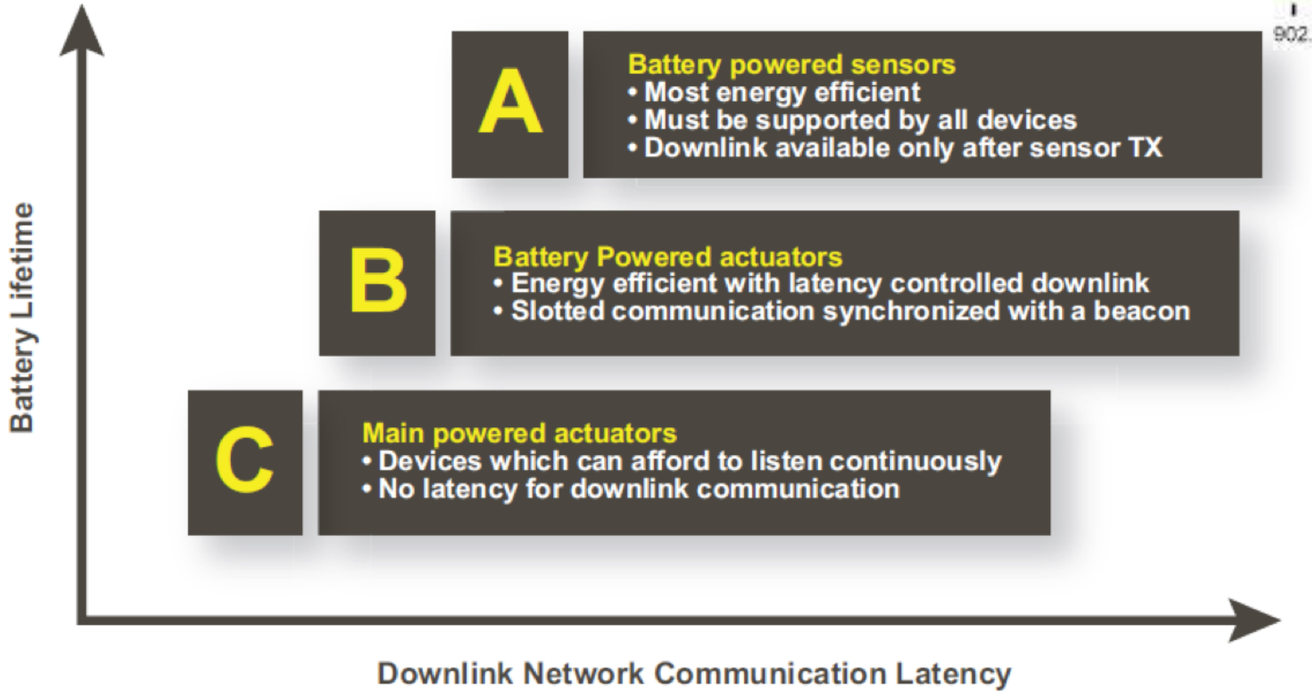


- The network server decides which AP to send downlinks based the AP with best signal quality (SNR)
- ARR (Adaptive Data Rate) is a feature that will ask your device to adjust its data rate and TX power to maximize battery life and network throughput

LoRaWAN: Classes + Channels

- In North America LoRaWAN operates in the unlicensed 900 MHz ISM band
- Uplinks or transmissions from a device use 125 KHz channels
- Downlinks use a 500 KHz channel to maximize the message rate
- Three classes of device. Each is largely a superset of the next.

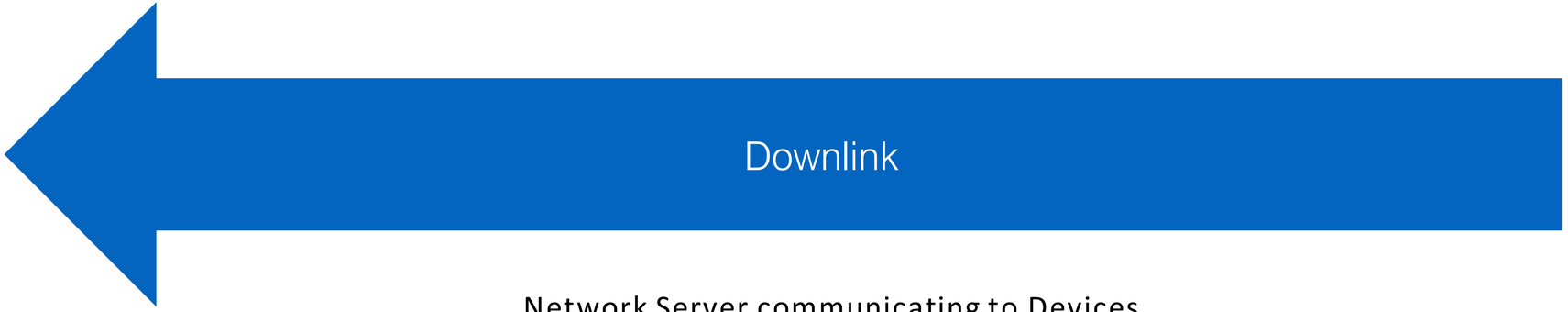




Unicast Messages vs FUOTA

Unicast Messages vs FUOTA: Unicast

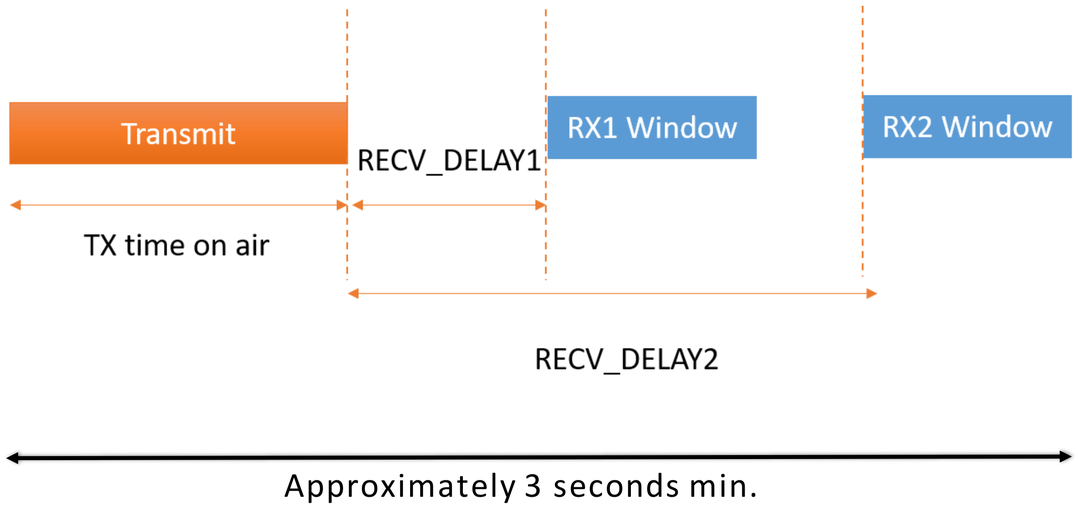
Devices communicating to Network Server



Network Server communicating to Devices

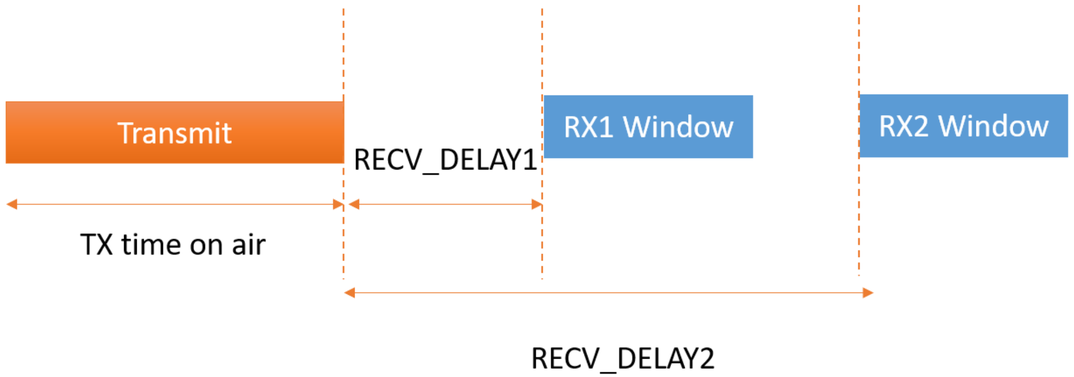
Uplink

- Can be sent anytime
- Sensor reading
- Device status



Downlink

- Only after an Uplink message
- Uses one of two possible receive windows
- Configuration
- Time sync



- Uplink and downlink messages are normal unicast application layer methods used to communicate with your device
- LoRaWAN is optimized to scale very well with thousands of devices that transmit small amounts of information infrequently
- Occasionally you may need to send information down to a device. This is typically to adjust configuration. Transmit interval, an alert threshold…
- The only time a message can be received by a class A device is after a transmission / uplink so it could be hours or even a day before that message is received
- Unicast messages are inherently inefficient if you are trying to send large amounts of the same data to many device (i.e. Firmware upgrade)

Configuration

Unicast Class A downlink messages

- Threshold parameters
- Reporting frequency

FUOTA

Binary image that is distributed to multiple devices simultaneously using a temporary multicast Class C session

FUOTA

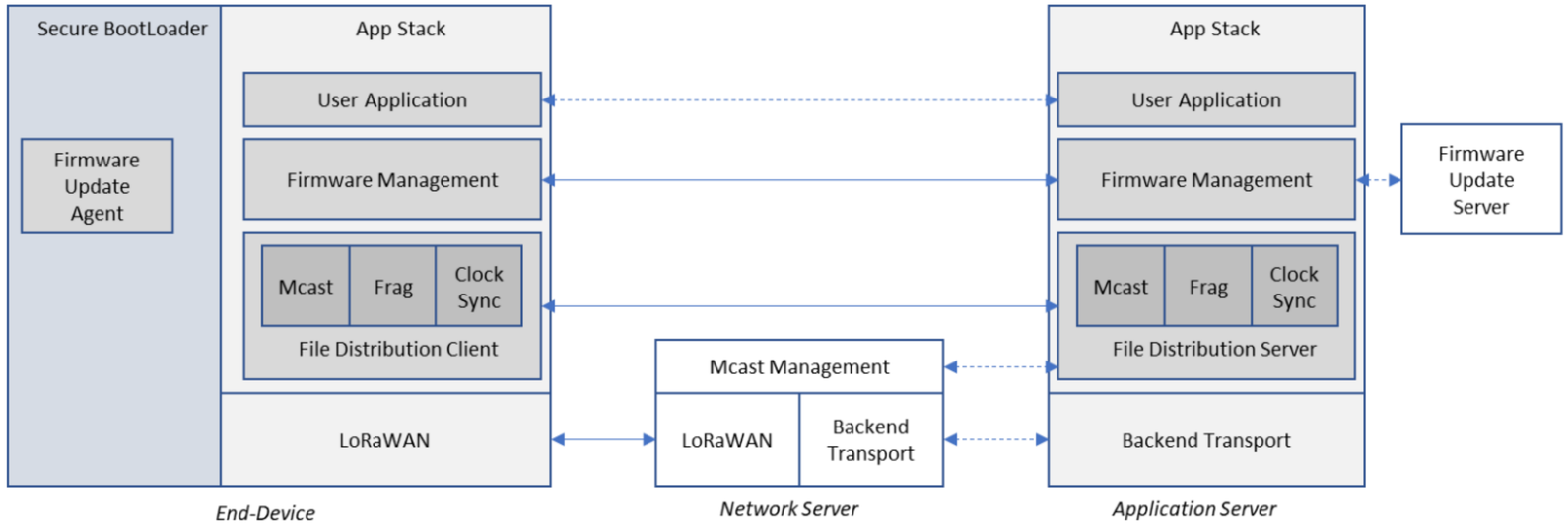
Firmware Update Over The Air

FUOTA

FUOTA is an application layer process that is designed to send a large chunk of information to many devices reliably and efficiently

Consists of roughly 5 steps

1. Clock Synchronization
2. Multicast group setup
3. Temporary transition to Class C
4. Fragmented Data Transmission
5. Firmware update and reboot



FUOTA: Server Side

MachineQ API

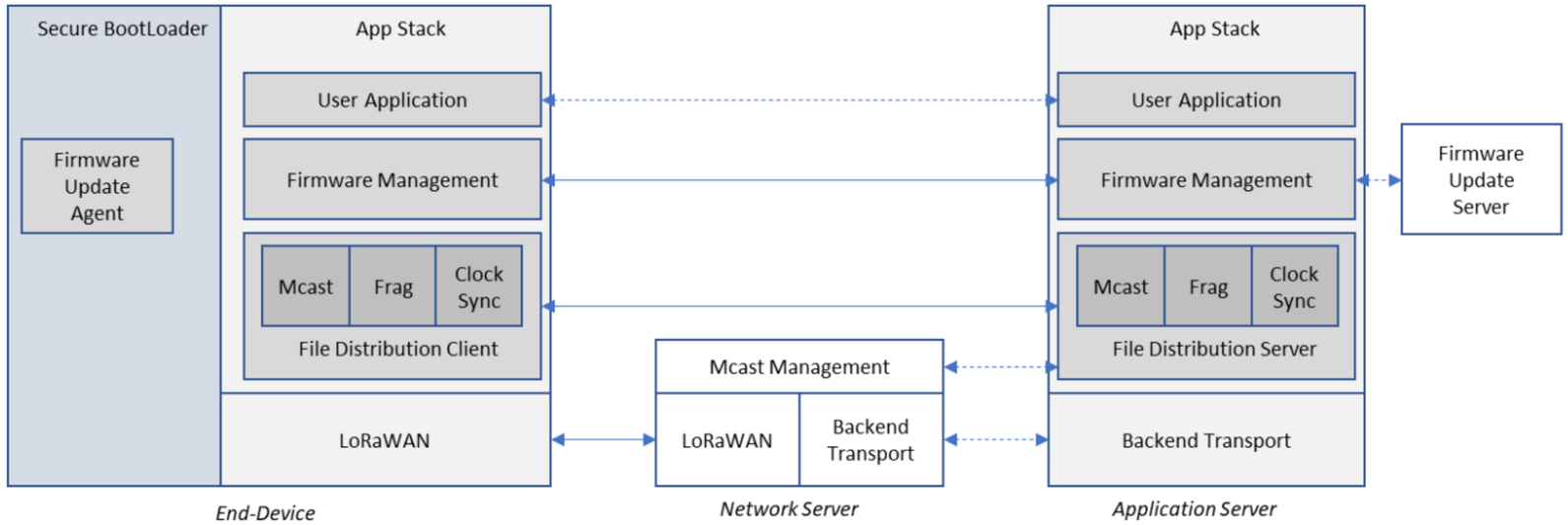
1. Firmware Image Upload
2. Multicast Group Definition
3. Deployment Creation
4. Deployment Start/Stop
5. Deployment Status

VS.

Direct NS + FUOTA Server

1. Device provisioning
2. Device Sets provisioning
3. Gateway tagging
4. Multicast Group provisioning
5. Region provisioning
6. File provisioning
7. Campaign Creation
8. Campaign Start/Stop
9. Campaign Status

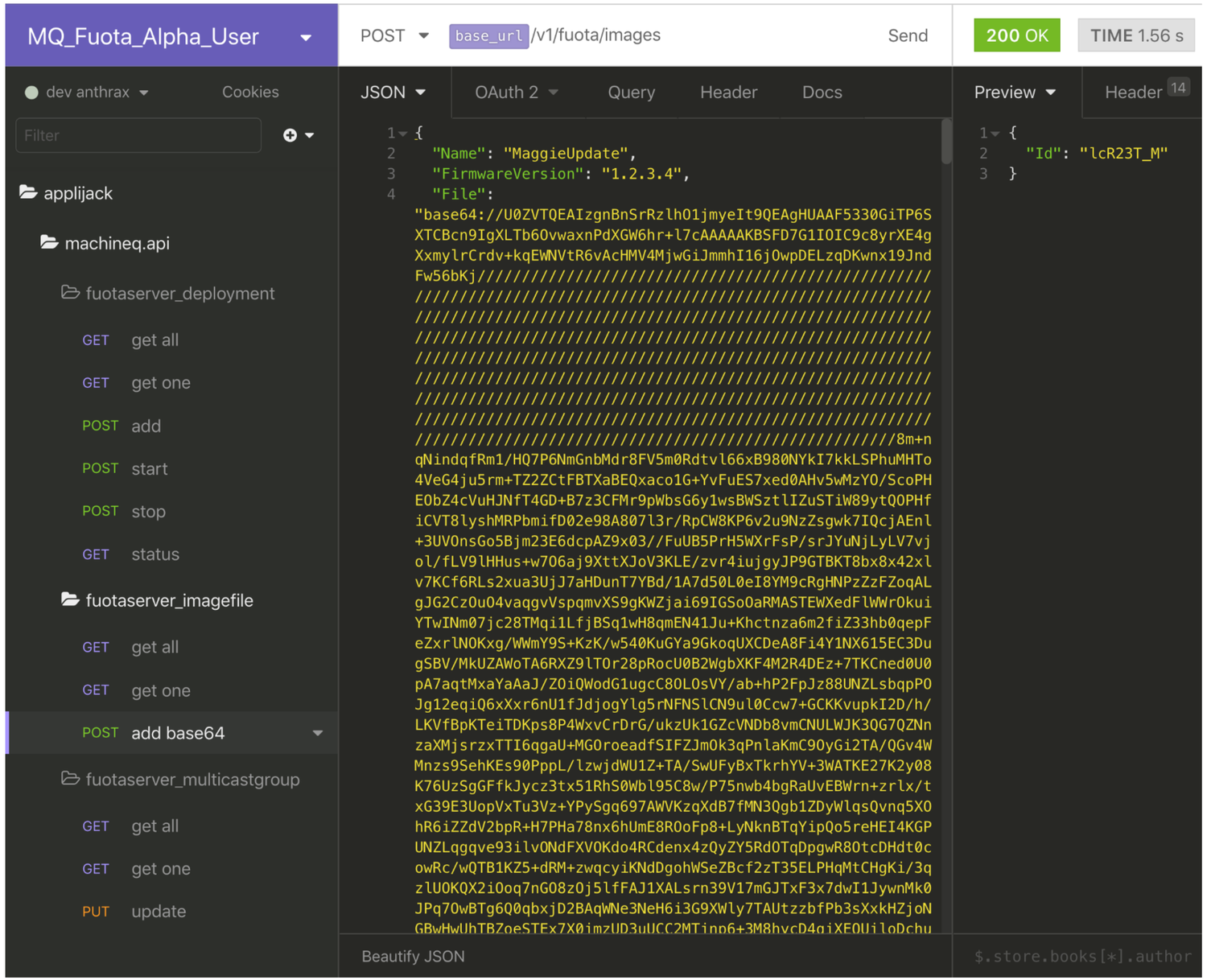
MachineQ FUOTA API and Embedded Library are 100% standards compliant



FUOTA: Server Side – Image Upload

Image Upload

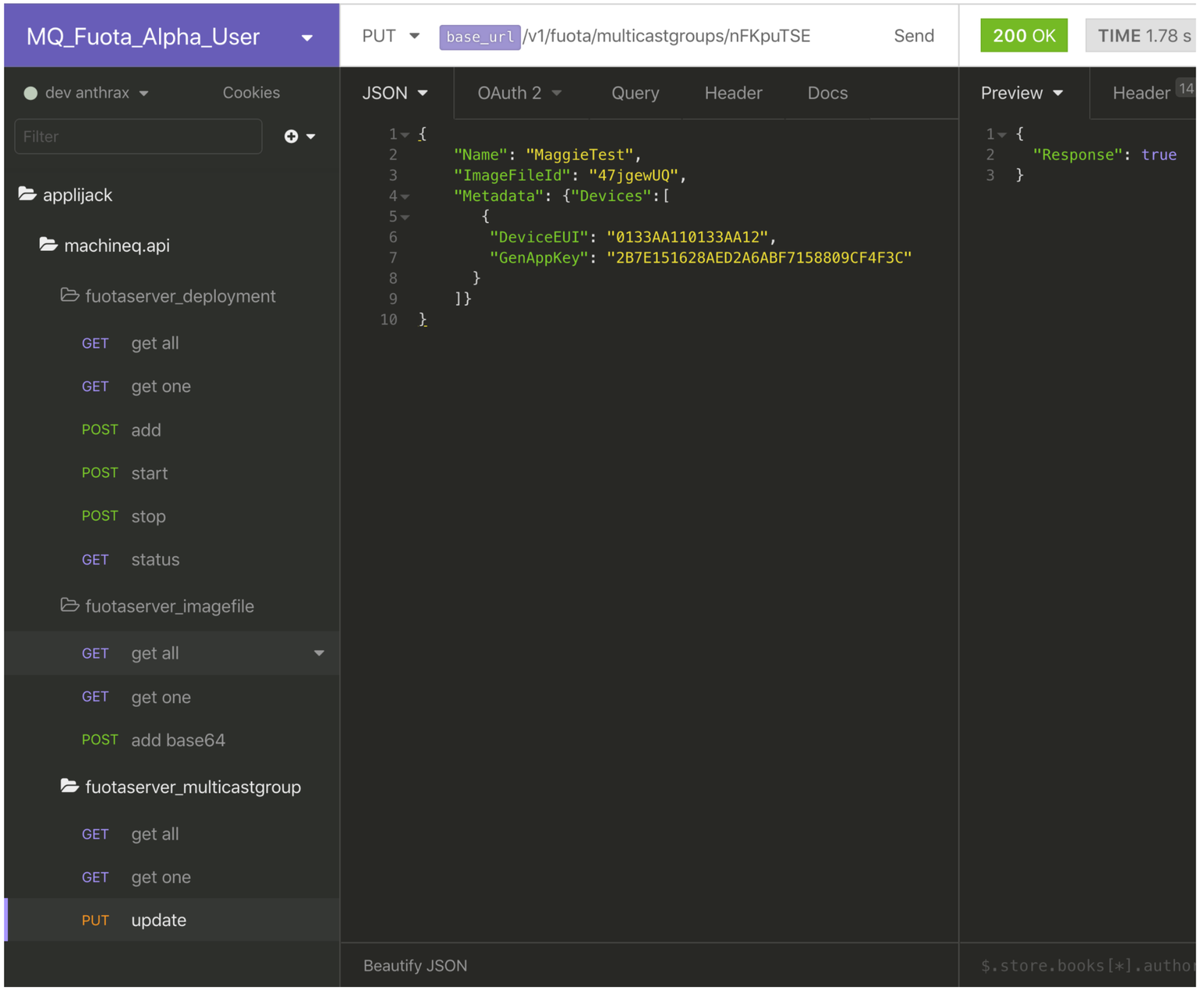
- Base64 encoded file
- Additional meta information
- Id returned used to create deployment



FUOTA: Server Side – Multicast Group

Multicast Group Population

- Selecting the devices to update
- Group Name
- Image ID
- Array of DevEUI and GenAppKey pairs



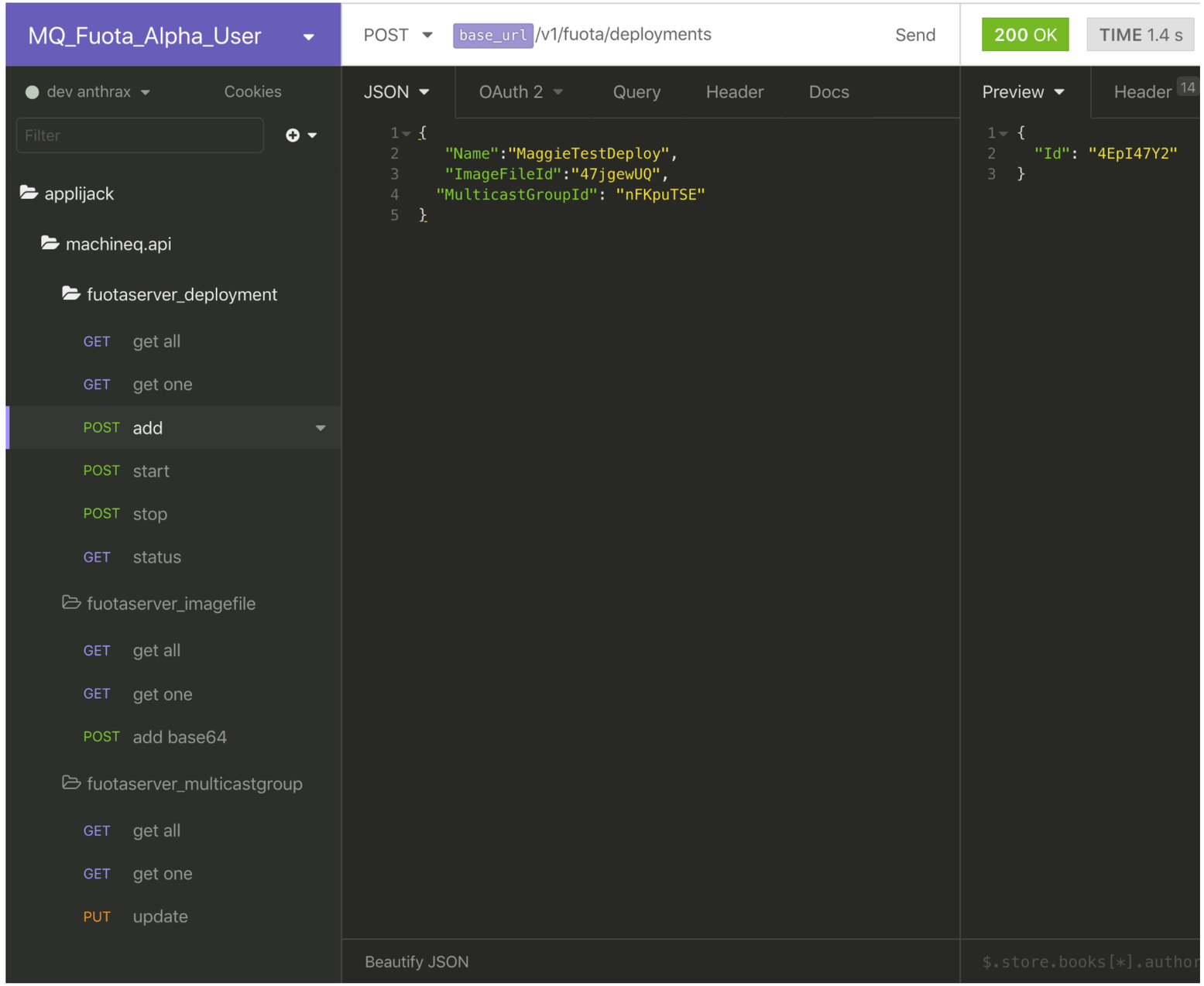
FUOTA: Server Side - Deployment

Creating a Deployment: Instance Of

- ImageFileID
- MulitcastGroupID

Deployment Operations:

- Start
- Stop
- Get Status



FUOTA: Device Side

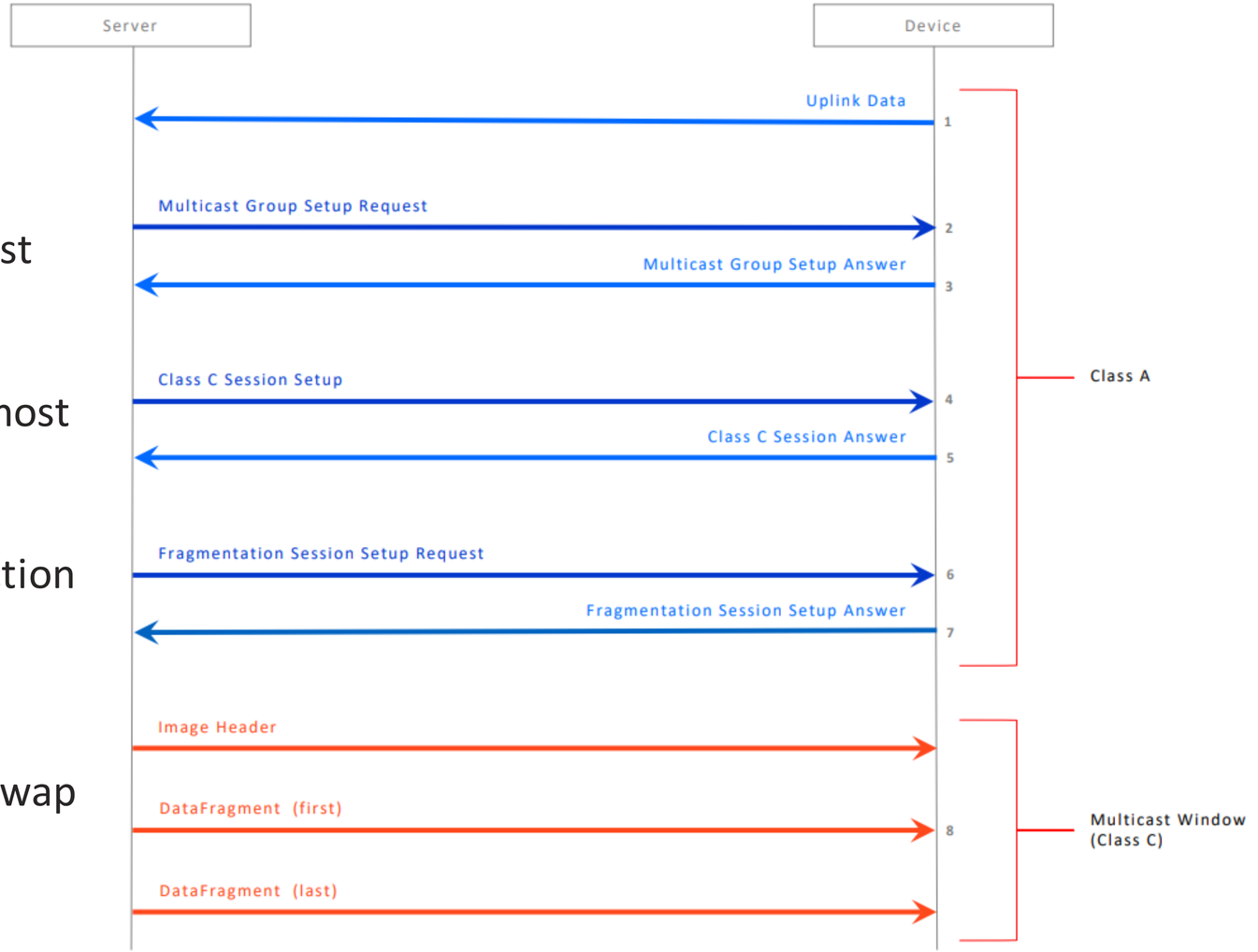
1. **Clock Synchronization:** Necessary to synchronize transition to class C (rendez-vous)

2. **Multicast group setup:** Sets up a common multicast address and encryption keys

3. **Class C Transition:** Allows downlink messages at most any time

4. **Fragmentation Setup:** Firmware image reconstruction with LDPC FEC. Fragment size, total number, redundancy...

5. **Firmware Update and Reboot:** Verify image and swap slots



FUOTA: MQfuotalib

Packages available in the library:

1. Remote Multicast Session Setup
2. Fragmented Data Block Transport
3. Application Layer Clock Synchronization
4. Firmware Management

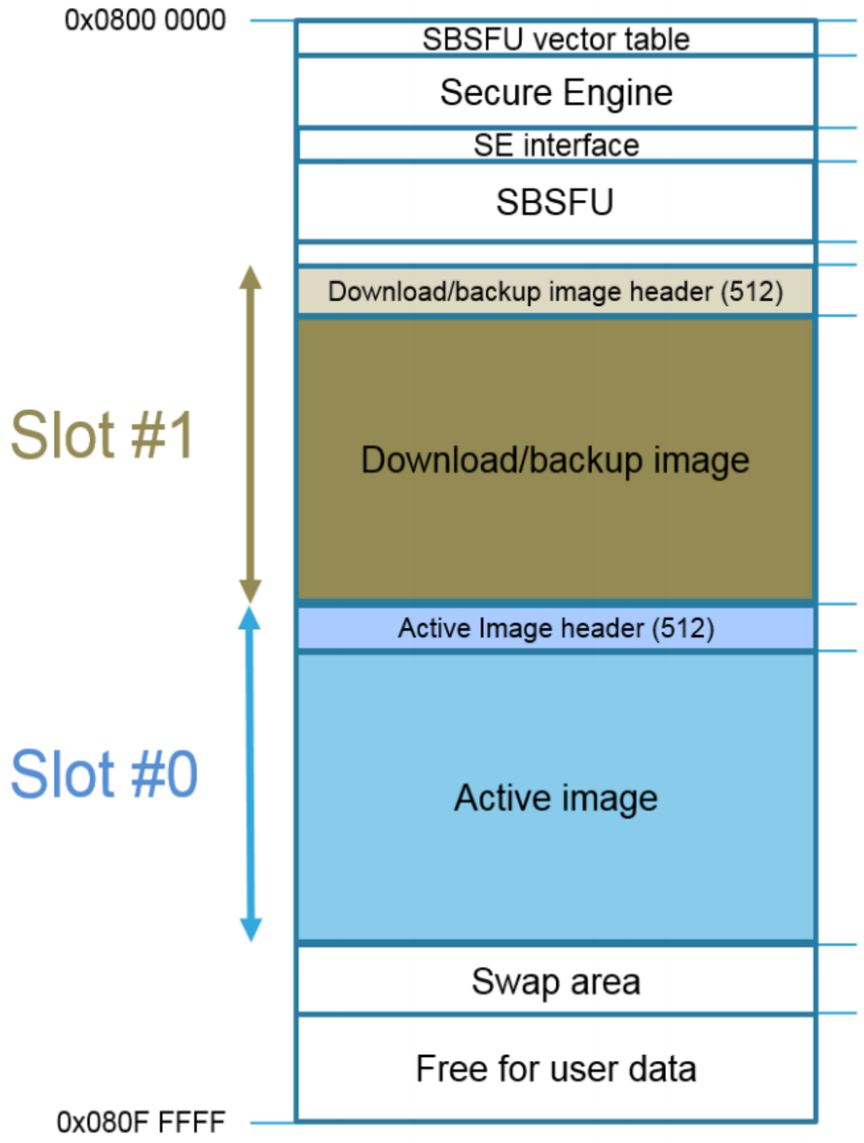
```
FuotaCallbacks_t fuotaCallbacks =
    {
        GetAvailableSpace,
        SendAppData,
        GetCurrClock,
        AesEncrypt,
        ClassCReqReceived,
        VerifyFreq,
        VerifyDR,
        GetDevVer,
        RebootDev,
        CancelReboot,
        GetUpgradeImageStatus,
        RemoveCurrFw,
        IsTimeSynced,
        FragDone,
        FragSetup,
        .decoderCallbacks =
            {
                FragDecoderRead,
                FragDecoderWrite,
            },
        CorrectTime,
        PeriodChange,
        ClockSyncCompleted,
        ForceADR,
        ForceNbTrans};
```

FUOTA: Design Recommendations

Hardware
- Simple Bootloader: 256K+ flash
- SBSFU: 384K+ flash

Dual Bank Bootloader

Stack Support
- Multicast
- Class C



FUOTA: Reference Documentation

- FUOTA Process Summary
- Clock Synchronization
- Multicast Setup Specification
- Fragmented Data Block Transport

Questions?

David Luksenberg

Senior Embedded Systems Engineer

David_Luksenberg@comcast.com

Thank You