EmPOWER™ Evaluation Kit

Getting Started

Date, October 20 2021 | Version 1.0

Contents

1. INTRODUCTION

EmPOWER™ is a SaaS solution that provides cloud services for device monitoring, management and update. On the cloud side is the Control Center featuring a web-based customer interface. On the device side is the Device-Client component. The EmPOWER™ Device-Client requires the EmSPARK™ Security Suite capabilities. **Figure 1** illustrates components.

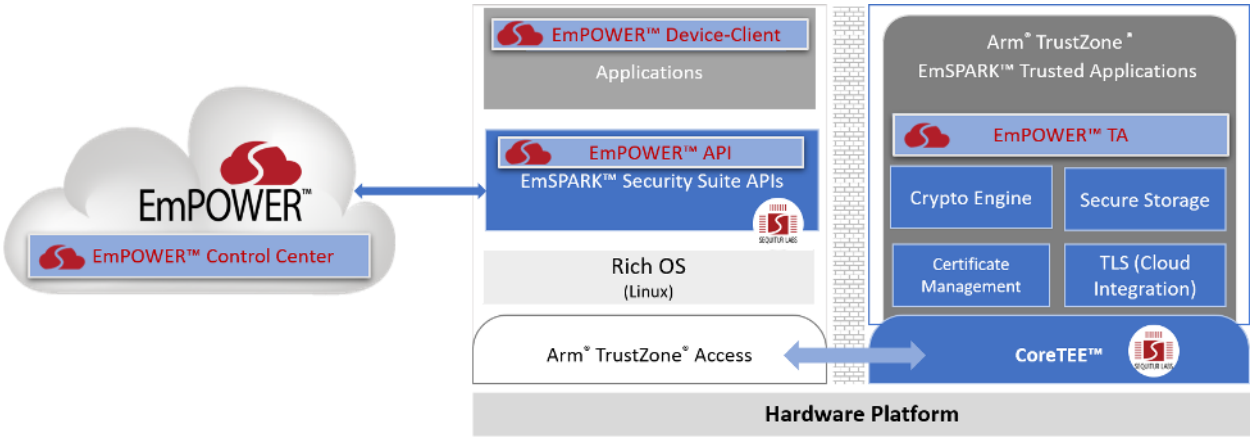

Figure 1 Cloud – Device Components

This document introduces the EmPOWER Evaluation Kit, using AWS as the cloud platform. The following section is an overview of the EmPOWER Suite Control Center and Device-Client for EmPOWER customers, and the difference between the EmPOWER Suite and the scope of the EmPOWER Evaluation Kit. The subsequent sections in the document explain the requirements, contents, installation and example execution of the EmPOWER Evaluation Kit.

1.1. Terminology and Acronyms

Certificate Store	Encrypted store of Certificate Authorities and Certificate Revocation List managed with the CoreLockr Certificate API.
Control Center	Web-based customer interface of the EmPOWER solution. It comprises the EmPOWER backend that enables communication with devices.
CoreTEE	Sequitur's Trusted Execution Environment (TEE), or secure OS, enabled by ARM's TrustZone™ architecture.
CoreLockr APIs	Equivalent to the EmSPARK Security Suite APIs
Customer	Product developer.
Device	Any device managed through EmPOWER.
Device Group	Customer defined device classification based on Customer's device management needs
Device Group UUID	Identification generated in the Control Center when Customer creates a Device Group. Customer uses the Device Group UUID when producing the device-client

Key Store	Encrypted store of keys managed with the CoreLockr Crypto API.
TA	Trusted Applications
TEE	Trusted Execution Environment

2. EmPOWER Suite Overview

This section highlights features of the Control Center and the Device-Client. Also, it describes the features that the Evaluation Kit illustrates. Note that the Evaluation Kit does not grant access to the web-based Control Center or provide Device-Client source code. Additional information about EmPOWER Suite is available upon request.

2.1. EmPOWER Control Center

EmPOWER Customers have access to the web-based Control Center that offers customers functionality to:

- Manage Devices: customers categorize devices in Device Groups.
- Manage OTA updates: Customer manages updates per Device Group.
- Monitor device status and health: the Control Center gathers data reported from the Device-Client installed on devices.
- Access Device Group data for analysis: health data for all devices in a Device Group.
- Manage response to device status: for example, Customer configures one or multiple update version sequences that devices must apply based on their reported status.
- Manage Control Center users: access control for customer users.

2.2. EmPOWER Device-Client

EmPOWER Customers have access to the Device-Client source code. The Device-Client is a software component that Customer configures, builds and installs on a device equipped with the EmSPARK Suite and its CoreLockr TAs and APIs. Customer configures the Device-Client for a specific Device Group enabling functionality to:

- Communicate with the Cloud: set up mutual authentication between device and Cloud using CoreLockr TLS IO and EmPOWER Cloud CA Cert and EmPOWER Device Cert.
- Connect to Cloud for Just-in-Time Registration (JITR): in the Cloud the device is associated with Customer and the corresponding Device Group.
- Send device status to Cloud: report to the Cloud device status data including system version, contact date and device health.
- Check for status response from Cloud: receive from the Cloud commands or responses corresponding to the reported device status.

2.3. EmPOWER Evaluation Kit Scope

Given that the Evaluation Kit does not grant access to the web-based Control Center or provides source code of the Device-Client, the artifacts included in the kit are already configured for installation and execution on a device provisioned with the EmSPARK Suite. The artifacts include a prebuild Device-Client. The following sections explain the Kit

contents, installation and example execution. Control Center screenshots illustrate how the sample Device-Client interacts with the Control Center and the Cloud.

3. EMPOWER EVALUATION KIT

The EmPOWER Evaluation Kit provides an example that illustrates a subset of capabilities in which the Device-Client establishes end-to-end trusted communication with the Cloud for a predefined sample Customer and Device Group. The following sections explain details of these capabilities:

- Communication with the EmPOWER Cloud: establish mutual authentication using CoreLockr TLS IO and EmPOWER Cloud CA Cert and EmPOWER Device Cert.
- Just-in-Time Registration (JITR): the Device-Client registers with the Cloud.
- Customer association: in the Cloud, the device is associated with the preconfigured sample Customer and Device Group.
- Communication with authorized servers: use of TLS IO API to mutually authenticate with only known trusted servers.
- Downloading a firmware update payload from a trusted source: the Device-Client sends to the Cloud a predefined device status data for which the Cloud responds with a command to download an update package.
- Secure update: installation of the downloaded update payload, verification of update payload during secure boot, and update system version on two boot stacks.

3.1. Prerequisites

This guide assumes that the following hardware and software are available:

- EmPOWER Evaluation Kit
- EmSPARK Suite – Evaluation Kit which includes the required firmware and filesystem
- Variscite VAR-SOM-MX8M-PLUS or DART-MX8M-PLUS device based on NXP i.MX 8M Plus provisioned with the contents of the EmSPARK Suite – Evaluation Kit
- Linux system to extract the Evaluation Kit packages, build the example applications and open a serial terminal to interact with the device

Please see **EmSPARK Suite – Evaluation Kit**, `GETTING_STARTED.pdf` for installation and board serial console access instructions.

3.2. EmPOWER Evaluation Kit Contents

Download and expand the Evaluation Kit tarball:

```
tar -zxvf empower_client_eval_variscite_[release].tar.gz
```

Expand the tarball containing the following files:

- `4401E9D6-998F-665E-947EEAAA2A5BA85E.stp`, EmPOWER Trusted Application
- `libseqr_empower.so`, the EmPOWER library companion of the TA
- `empower_client`, prebuilt sample Device-Client ready for device execution
- `DigiCertBaltimoreCA-2G2.crt.pem`, sample Certificate Authority required to ensure the device downloads the update payload from a trusted source
- `RELEASE_NOTES.txt`, information about the release

- `COPYRIGHT.txt`, the copyright notice
- `EMPOWER_GETTING_STARTED.pdf`, this guide

3.3. Installation Procedure

Install the EmPOWER TA and library in the root filesystem of a board provisioned with the EmSPARK Suite:

- Copy `4401E9D6-998F-665E-947EEAAA2A5BA85E.stp` to `/data/tee/` where additional TAs are installed on the board. Ensure the file permissions are the same as the other TAs.
- Copy `libseqr_empower.so` to `/usr/lib/`.

The installation of files needed to execute the example is described in the next section, `empower_client` (Device-Client) and `DigiCertBaltimoreCA-2G2.crt.pem` (Server CA).

4. EMPOWER DEVICE-CLIENT

Here is an overview of the Device-Client and specifics of the example.

4.1. Device-Client

The EmPOWER Device-Client is designed for Customer configuration, building and installation. The operations in the device TEE use TAs and their associated libraries, keys stored in the Key Store and certificates stored in the Certificate Store.
To associate the device with the Customer and Device Group in the Cloud, the Device Group UUID must be configured in Device-Client installed on the device. In addition, the device to Cloud communication must maintain a chain of trust that on the device starts at provisioning.

TAs and libraries:

- EmPOWER TA and library
- CoreLockr Crypto
- CoreLockr Secure Certificates
- CoreLockr TLS IO

Keys and certificates stored in the Certificate Store and Key Store:

- EmPOWER Root Certificate, used for EmPOWER cloud connectivity
- EmPOWER Public Key, public key contained in EmPOWER Root Certificate
- EmPOWER Device Key, generated during provisioning, immutable, used to uniquely identify the device
- EmPOWER Device Public Key, public component of the EmPOWER Device Key
- EmPOWER Device Certificate, used for TLS authentication with EmPOWER service

Information about all provisioned keys and certificates is available in the `CORELOCKR_LIBRARIES_GUIDE.pdf` included in the EmSPARK Suite – Evaluation Kit.

4.2. Configuration and Building

The first time the Device-Client executes and connects with the Cloud, in the Cloud the device is associated with the Customer and the Customer's designated Device Group. Subsequent communication with the Cloud maintains the same Device Group. In the Cloud the device is identified using the EmPOWER Device Certificate.
In a production workflow, the Device-Client configuration and building requires steps in the Control Center and the Customer's Development Environment:

- **Control Center**: Customer creates a Device Group based on its device management needs.
- **Control Center**: When the Device Group is added, the Control Center generates a `DEVICE_GROUP_UUID` that unique identifies the Customer and Device Group
- **Development Environment**: Customer configures the Device-Client with the `DEVICE_GROUP_UUID`
- **Development Environment**: Customer builds the Device-Client which needs to be installed on the device.

In this Evaluation Kit, the Customer and Device Group are already configured in the Control Center, as explained in **5**.

5. DEVICE-CLIENT CLOUD COMMUNICATION EXAMPLE

The example describes flow, scope and execution steps on the device and in the Cloud.

5.1.1. Configuration and Building Flow

The `empower_client` application is a binary configured for an example Customer. Unlike the Device-Client for a production environment, this example takes switches and parameters to illustrate the interactions with the Cloud. The following steps in the configuration and building flow are already executed and specific for the example:

- **Control Center**: in the Control Center, the Device Group is already created. Its `DEVICE_GROUP_UUID` is used in this example as a Device-Client input parameter.
- **Device-Client**: The sample `empower_client` is a special build that takes two switches, the first one to associate the device with the Device Group in the Cloud, and the second switch to set the device to a predefined system version configured in the Cloud for the sample `DEVICE_GROUP_UUID`.

Figure **2** shows the Control Center Device Group page and corresponding `DEVICE_GROUP_UUID`.

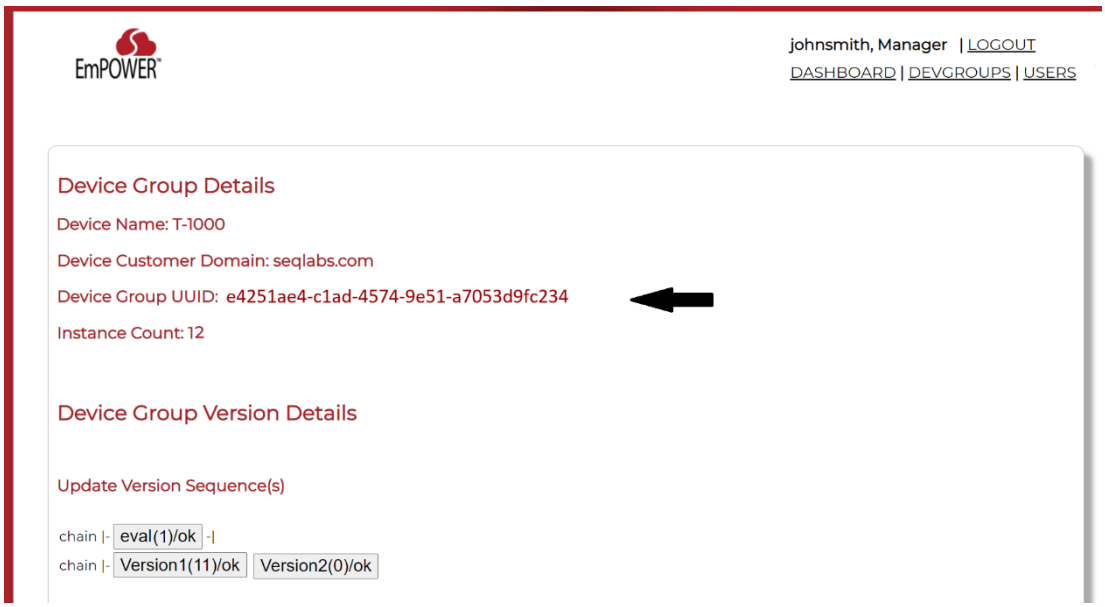

Figure 2 – Control Center, Device Group Details

5.1.2. Example Stages

The example execution illustrates the following sequences of events:

- JITR registration and Device Group association in the Cloud for a sample Customer and Device Group UUID
- Download update payload and update device system version
  - Add in the device Certificate Store the Certificate Authority of the server where the update payload will be downloaded from
  - Download an update payload from the Cloud
  - Apply update payload for one boot stack
  - Apply update payload for the other boot stack

5.1.3. Preparation

- From the **EmPOWER Evaluation Kit**, transfer the provided `empower_client` binary and `DigiCertBaltimoreCA-2G2.crt.pem` to the device to a directory of your choice
- From the **EmSPARK Suite – Evaluation Kit**, build the Secure Certificates example application. Please follow the instructions in `CORELOCKR_LIBRARIES_GUIDE.pdf`, *Certificate Authority Management* example section. Transfer to the board `clrsc_example` and `certs` directory.

5.1.4. Execution: JITR Registration and Customer Association

The example does JITR registration and Device Group association for the sample Customer and Device Group. The execution will register the device to the `e4251ae4-c1ad-4574-9e51-a7053d9fc234` Device Group. **Figure 2** shows the total of devices registered to the Device Group before the device registration. In the example, `Instance Count: 12`.

- Change to the directory where `empower_client` is on the device

- Execute the Device-Client to associate the device with the Device Group in the Cloud

```
./empower_client -a e4251ae4-c1ad-4574-9e51-a7053d9fc234
```

Where `-a` is a switch to associate the device with the Device Group in the Cloud

`e4251ae4-c1ad-4574-9e51-a7053d9fc234` is the sample `DEVICE_GROUP_UUID`

The application does the following:

- Device to Cloud Mutual Authentication using TLS
- Connection to Cloud for Just-In-Time Registration
- Use EmPOWER Cloud CA Cert and EmPOWER Device Cert
- Device Registration and Customer Association

In the Cloud, the device identity is associated with the Customer and Device Group UUID.

The application prints messages such as the following:

```
Running EmPOWER client for device id:
5dbca5e693e1d236f9f50cbf1070634ea279b9cb
Opening session to CoreLockr TLSIO TA:
Using EmPOWER Cloud CA Certificate
Using EmPOWER Device Credentials
Connecting to a3vjl794pxe1n3-ats.iot.us-west-1.amazonaws.com...
Initial connection failed.
This could be due to certificate Just-In-Time Registration.
Trying again...
Connected
Running device registration and association.
Associating:
Device: 5dbca5e693e1d236f9f50cbf1070634ea279b9cb
With Group: e4251ae4-c1ad-4574-9e51-a7053d9fc234
Subscribing to device association topics...

---------------------------------------------------------
Associating device with Customer Device Group in Cloud
---------------------------------------------------------
Association request sent... Calling loop

*****************Subscription callback
Parsing challenge...
Calling calculate challenge...
Posting challenge response...

*****************Subscription callback
Customer Challenge is accepted. Device can be associated.
```

In the Control Center, after the device registration the total of devices associated to the Device Group is incremented, `Instance Count: 13`, as shown in **Figure 3**.

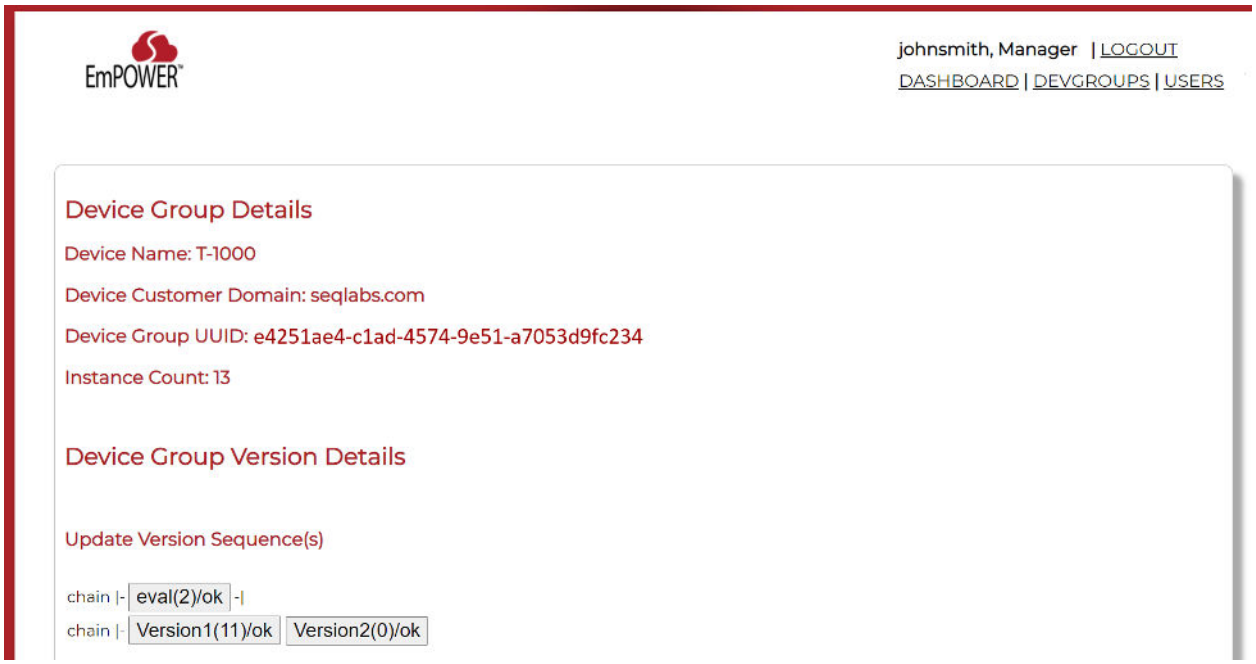

Figure 3 – Control Center, Device Group Updated after JITR

5.1.5. Execution: Download Update Payload and Update Device System Version

For a Device Group, the Control Center enables configuration of chains of system updates and their versions. In this example, the Device Group has configured two chains of Update Version Sequences, as shown in **Figure 3**:

- `eval`, which does not have configured subsequent updates
- `Version1` which has configured an update to `Version2`, i.e. devices reporting `Version1` status will receive response from the Cloud to download an update payload that updates the system to `Version2`

The device system version after provisioning with the Evaluation Kit is "`eval`". To facilitate the execution of the example, the following steps instruct how to set the device system version to `Version1` to download the expected payload. Then, the steps instruct how apply a system update to both boot stacks on the device to bring the device to system `Version2`.

The Control Center displays within parenthesis the total number of devices that have installed a given version. Before the device registration, the total number of devices in "`eval`" version was `1`, as shown in **Figure 2**. After the device registration, the Control Center shows that the total number of devices in "`eval`" version has increased to `2`, shown in **Figure 3.** Devices in `Version2` are `0`.

5.1.5.1. Add to the device Certificate Store the CA of the server where the update payload is available for downloading

To ensure that the payload is downloaded from a trusted source, using the CoreLockr TLS IO for mutual authentication requires that the CA of the server be known in the TEE. To add the server CA in the TEE:

- Change to the directory where the files of the Secure Certificate example `clrsc_example` and `certs` directory are located
- Ensure that the provided `DigiCertBaltimoreCA-2G2.crt.pem` is available on the board
- Execute `clrsc_example` with the following switch:

```
./clrsc_example a $certPATH/DigiCertBaltimoreCA-2G2.crt.pem
```

The application prints messages confirming the addition of the CA in the TEE, e.g.:

```
Running CLRSC OpenSSL example
Got CA cert
/*********Get DER cert from X509
/*********Getting command buffer
/*********Getting sha of buffer
/*********Signing hash with COMMAND PRIVATE KEY
/*********Calling TA to perform command: 256
Successfully saved certificate to TEE
```

5.1.5.2. Set system version to `Version1` to exercise the download of the update payload

Execute `./empower_client -v Version1`

The command does the following:

- Set the devise system version to `Version1`, the device sends status `Version1`.
- In the Control Center, `Version1` has configured a system update to `Version2` with an update payload uploaded on a server that uses the `DigiCertBaltimoreCA-2G2.crt.pem` CA added to the Certificate Store in step a).
- The Device-Client receives the commands to download the update payload from the server.
- Upon successful mutual authentication, the Device-Client downloads the update payload and completes commands such as writing the update payload in NVM

The application prints messages such as the following:

```
Running EmPOWER client for device id:
5dbca5e693e1d236f9f50cbf1070634ea279b9cb
Opening session to CoreLockr TLSIO TA:
Using EmPOWER Cloud CA Certificate
Using EmPOWER Device Credentials
```

```
Connecting to a3vjl794pxe1n3-ats.iot.us-west-
1.amazonaws.com...
Connected
Setting system version to: Version1
Loaded server response from manifest.
Parsing for additional commands to be performed...
Server Response:
Status: ok
Version: 1
Sending device status to EmPOWER...
Posted... Calling loop
Received status response from EmPOWER.
Got array. Parsing single command...
Server Response:
Status: ok
Version: 1
Retrieving and storing EmPOWER command values...
Starting download to... ./Firmware/updatepayloadeval_uboot.bin
Downloading [https://mdtestupdates02.s3-us-west-
2.amazonaws.com/updatepayloadeval_uboot.bin] to ->
[./Firmware/updatepayloadeval_uboot.bin]...

Verifying payload certificate...
clrscVerifyCertificate returned TEEC_SUCCESS! Certificate is
verified.
Result of download is: 0
Downloaded: 860760 bytes...
Starting download to... ./Firmware/
Running update on downloaded server payload...
Wrote: 860760 bytes to NVM
Command is complete.
Please reboot the device to apply the update...!
```

5.1.5.3. Reboot the device to apply the update on the non-primary boot stack

After the device provisioning, the primary boot stack is Plex A and the non-primary boot stack is Plex B. During boot, the update is applied to the non-primary boot stack, i.e. in this example, Plex B. Upon successfully update, Secure Boot activates the non-primary stack as the primary, i.e. when the device boots, Plex B is now the primary stack. The device serial console prints messages such as the following:

```
[check_boot_state] - Stateval: 0x1f
[SLI] - Update plex set. Updating non-primary plex
...
Copying update header from block offset...
Update header...
30830D2253020400000093304502202A
E0F48B7B75EB82B388D0CC8376A0645F
Total update size: 860760  860755
```

```
Calling verify_and_run_update...
Calling verify...
Update package signature for debugging!!!!
2AE0F48B7B75EB82B388D0CC8376A064
5F9A5D3FC78245ED935D5158B6DC6ED7
C614962BC55635F6C58AF07146FEAE95
E6445E7EAD519295948D21D61246B8B5
...
Updating components. Manifest: 0x00001000   plnode->content:
60000057
Update manifest found. Running update. SPL already: 0.
Setting MMC boot state to: 0x1f
Boot state from mmc is: 0x1f
Continuing with non SPL components.
New component[uboot] at update offset...
Copying: 0x60001057 to 0x70000000   size: d1200
Decapsulate Gold Update Blob
Encrypting component: SUCCESS (0)
Copying component to MMC from: ...
Saving manifest back to NVM: 0x00023000
SUCCESS (0)
Writing to the SF...6144 bytes written
Done update
...
[SLI] - Activate plex set. Activating non-primary plex
Setting MMC boot state to: 0x0c
Boot state from mmc is: 0x0c
Setting BLC to: 5
Setting BLC to: 4
[check_boot_state] - Stateval: 0x0c

Loading firmware version [eval_uboot] for Plex B
Calling component_setup
...
********************
Running UPDATED UBOOT!
**********************
```

5.1.5.4. Apply update payload for the other boot stack

The failover logic uses two boot stacks: Plex A and Plex B. In the previous step, Plex B was updated. The following steps will update Plex A, which is now the non-primary stack.

Execute `./empower_client`

The execution prints messages such as the following.

```
Running EmPOWER client for device id:
8a0b3ca4c8ba0215975fbee09a781a09cbcb41b5
```

```
Opening session to CoreLockr TLSIO TA:
Using EmPOWER Cloud CA Certificate
Using EmPOWER Device Credentials
Connecting to a3vjl794pxe1n3-ats.iot.us-west-1.amazonaws.com...
Connected
Setting system version to: Version1
Running update on 'other' plexSaving Boot state value: 0x0000000f
Resetting...
```

5.1.5.5. Reboot the device to apply the update on the non-primary boot stack, which now is Plex A

During boot, the console prints messages such as the following:

```
Loading firmware version [eval_uboot] for Plex A
Calling component_setup
Loading KERNEL from GOLD BLOB
Loading u-boot from GOLD BLOB
    DTB loaded from U-Boot
Loading CoreTEE from GOLD BLOB
Loading ATF from GOLD BLOB
Calling into ATF...
…
********************
Running UPDATED UBOOT!
************************
```

After the device system update, the Control Center displays the updated number of devices in "`Version2`" which in this example has increased to `1`, and the number of devices in "`init`" has decreased to "`1`", as shown in **Figure 4**.

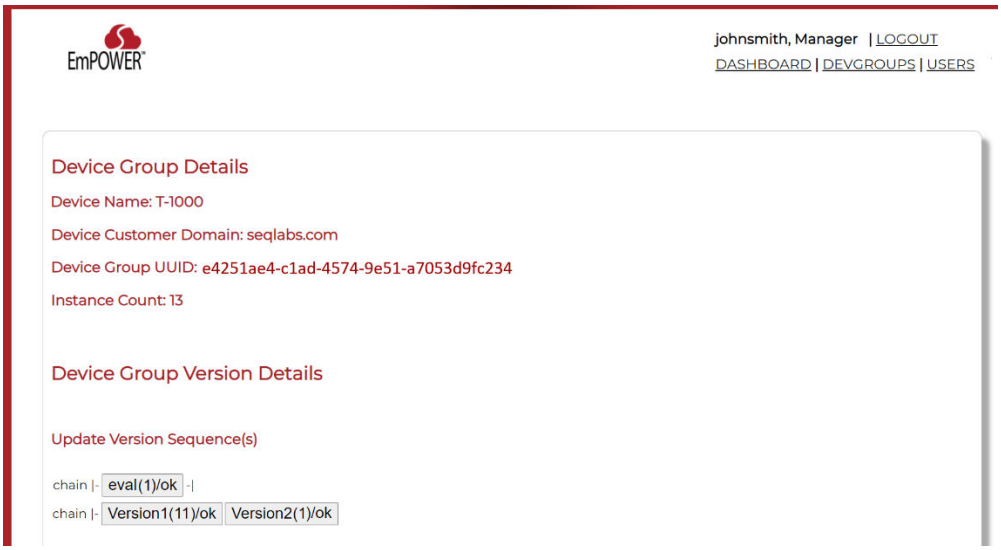

Figure 4 - Control Center, Device Group after Device System Version Update

This concludes the EmPOWER example execution.

CHANGE HISTORY

DATE	VERSION	RESPONSIBLE	DESCRIPTION
October 20	1.0	Julia Narvaez	Produced document for release.