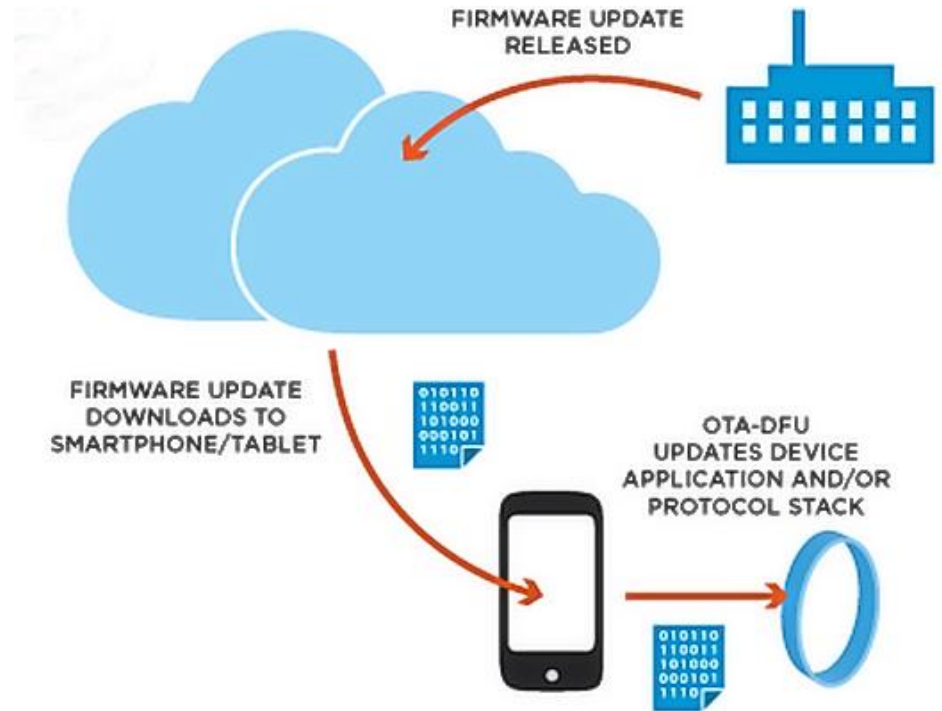


NORDIC tech TOUR

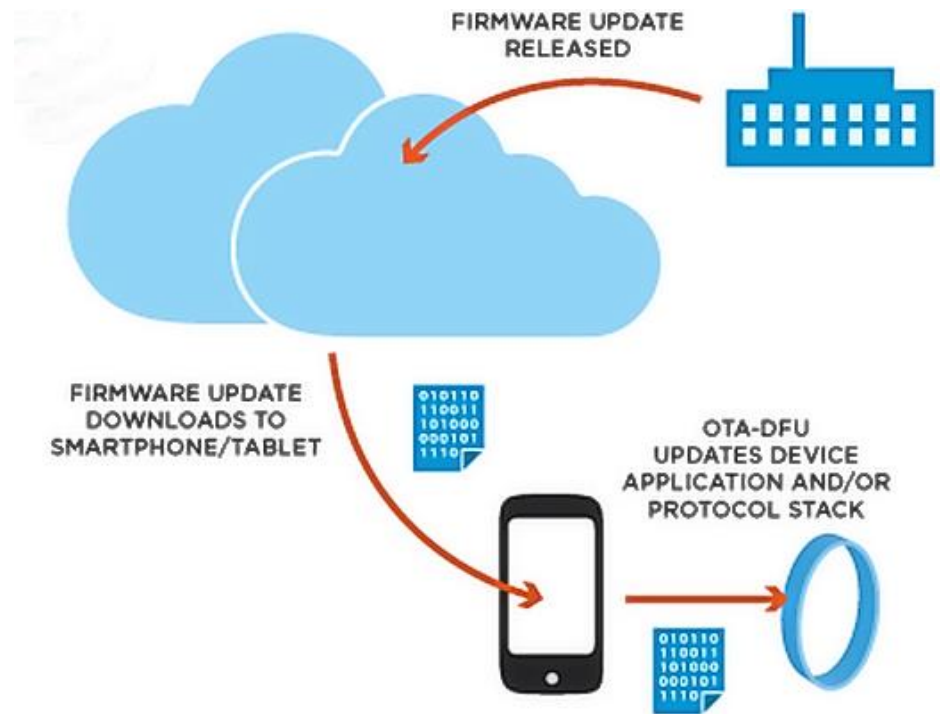
*nRF51 SoC Device Firmware Update
(DFU)*

Device Firmware Updates (DFU)



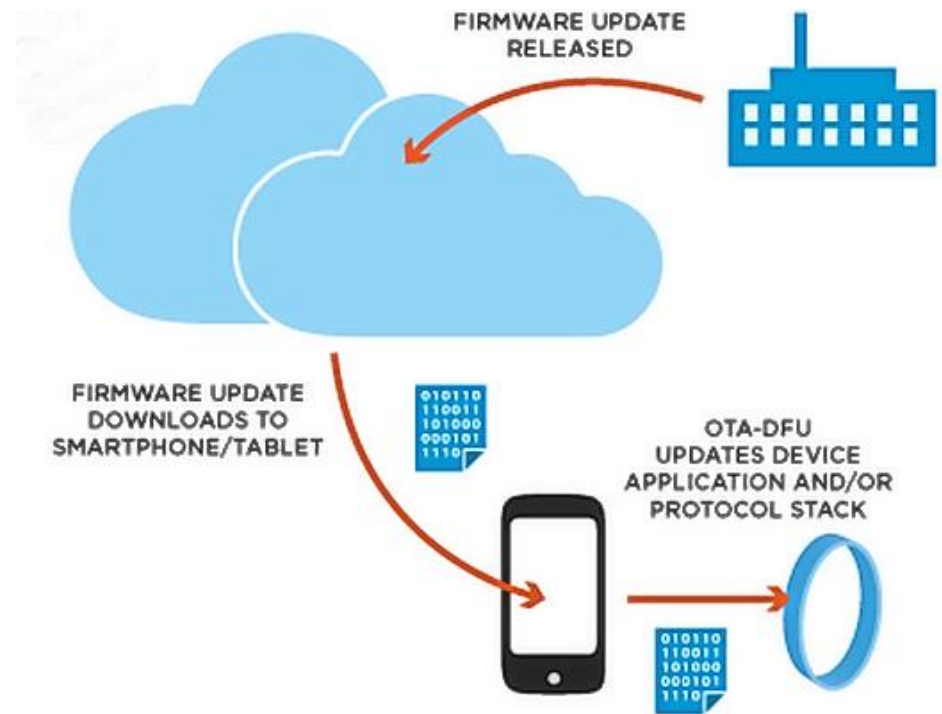
Device Firmware Updates (DFU)

- Firmware updates after the product is shipped
 - Bug fixes, new features and improvements
 - Via Over-the-Air (OTA), wired interfaces (UART, SPI, etc)



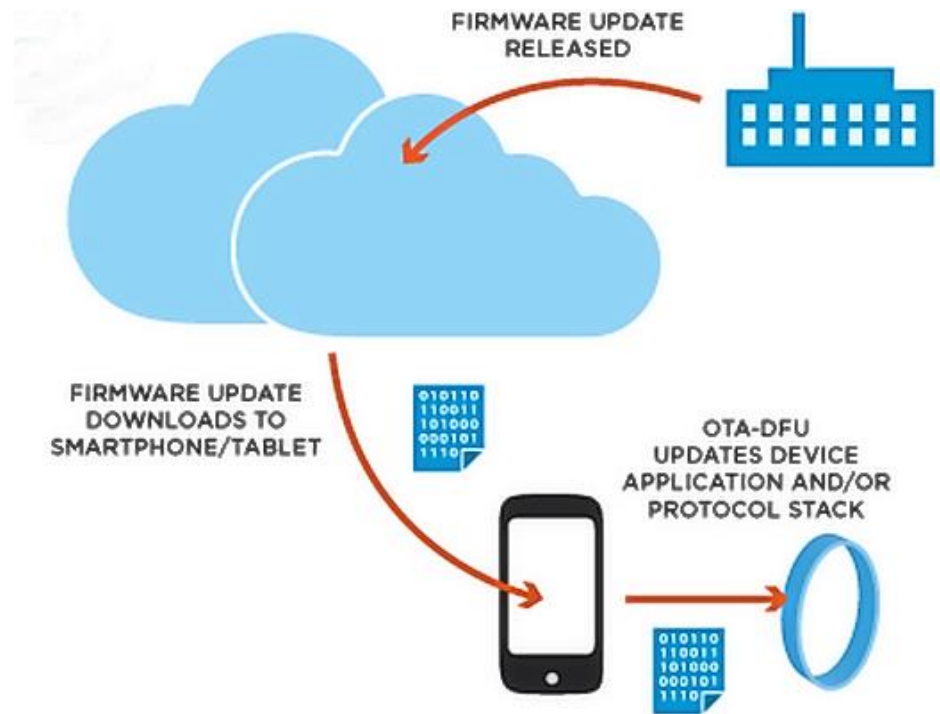
Device Firmware Updates (DFU)

- Firmware updates after the product is shipped
 - Bug fixes, new features and improvements
 - Via Over-the-Air (OTA), wired interfaces (UART, SPI, etc)
- Used to be optional, have become ubiquitous



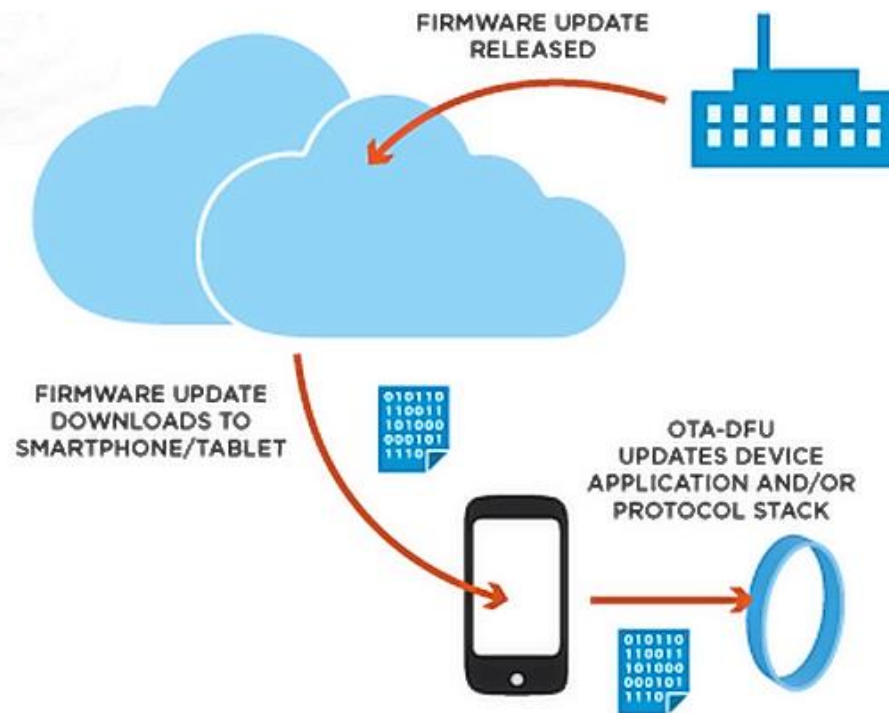
Device Firmware Updates (DFU)

- Firmware updates after the product is shipped
 - Bug fixes, new features and improvements
 - Via Over-the-Air (OTA), wired interfaces (UART, SPI, etc)
- Used to be optional, have become ubiquitous
- Consumers expect them, companies want to offer them



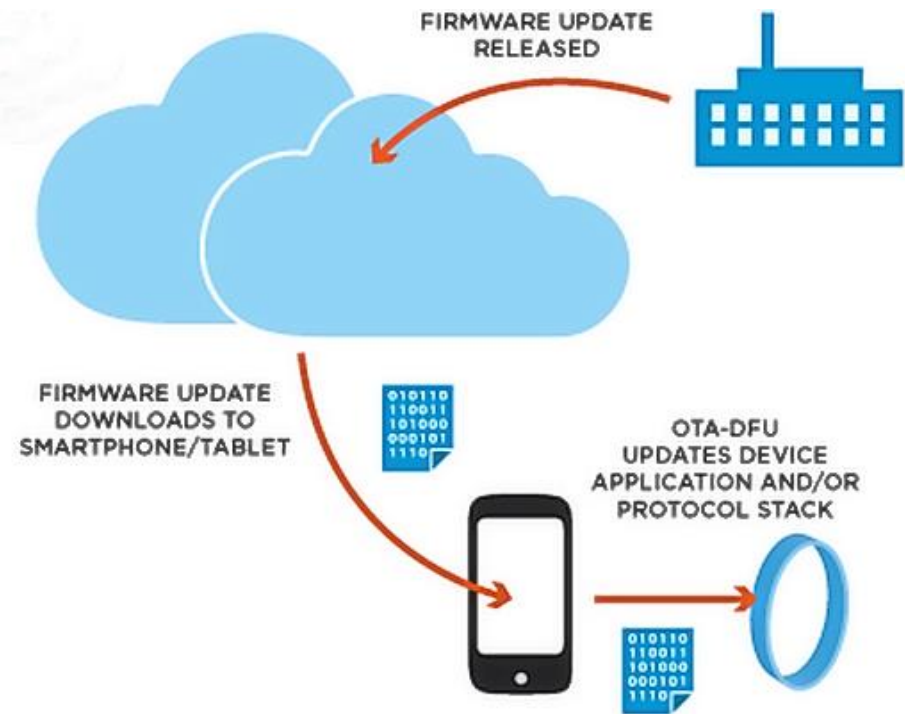
Device Firmware Updates (DFU)

- Nordic is developing an advanced suite of DFU solutions for nRF51 Series ICs and software
 - Flexible – freedom for developers to customize
 - Efficient and safe

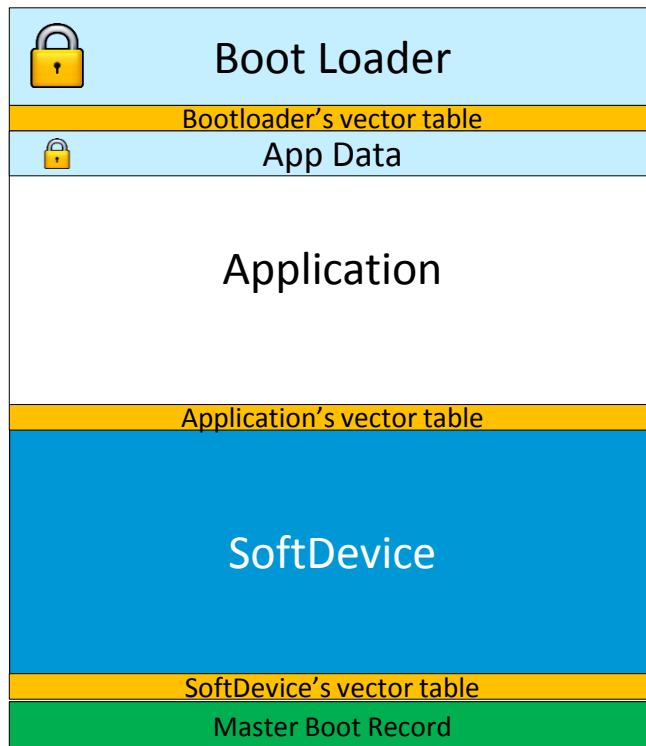


Device Firmware Updates (DFU)

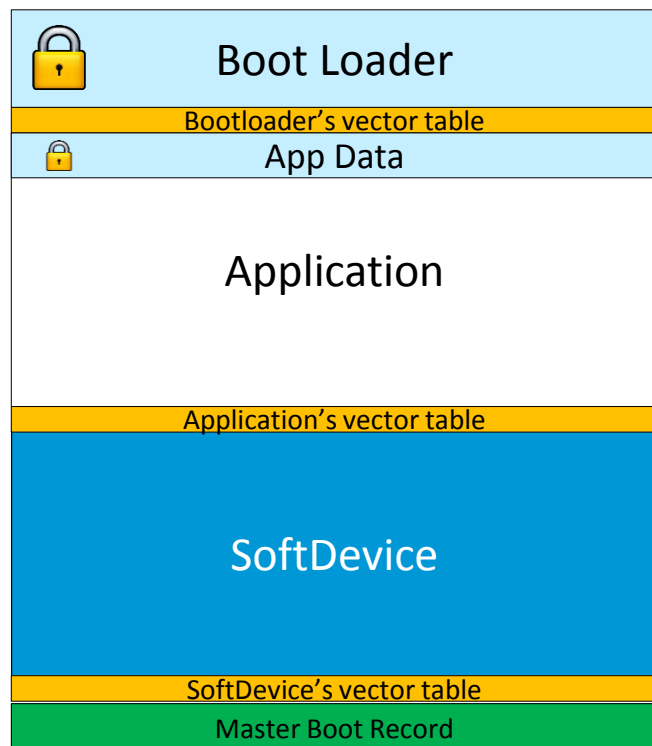
- Nordic is developing an advanced suite of DFU solutions for nRF51 Series ICs and software
 - Flexible – freedom for developers to customize
 - Efficient and safe
- Application space (user code) update, released Nov 2013, from SDK v5.0
- Bluetooth stack (SoftDevice) update, released Aug 2014, from SDK v6.1



Boot Loader

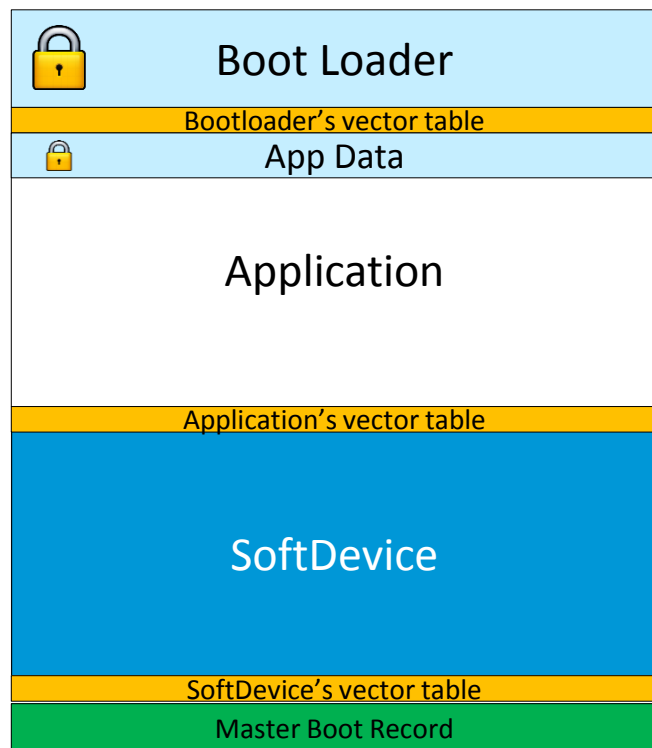


Boot Loader



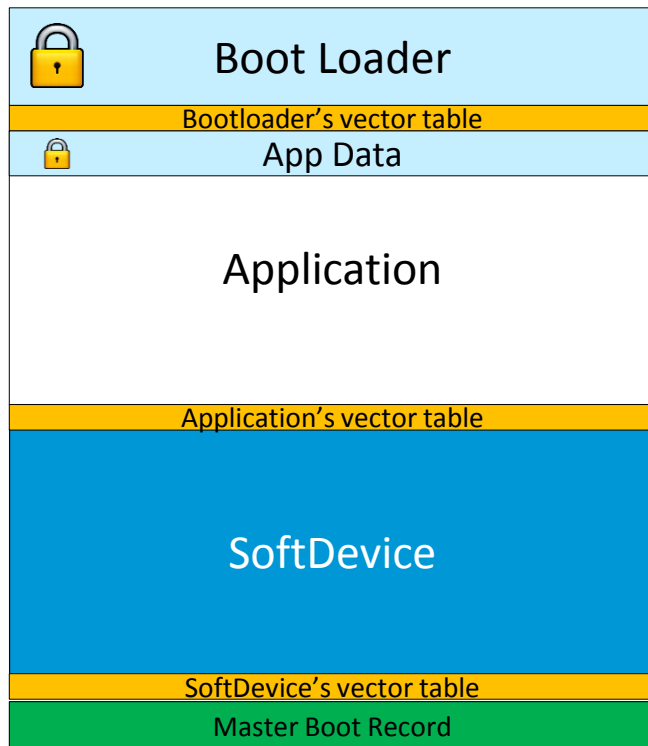
- A small program that's always executed first when device booting up.
 - Write and erase protectable (PROTENSET)

Boot Loader



- A small program that's always executed first when device booting up.
 - Write and erase protectable (PROTENSET)
- Functions:
 - Control device
 - Receive updates firmware images via OTA or UART
 - Swap the new images with the old ones

Boot Loader

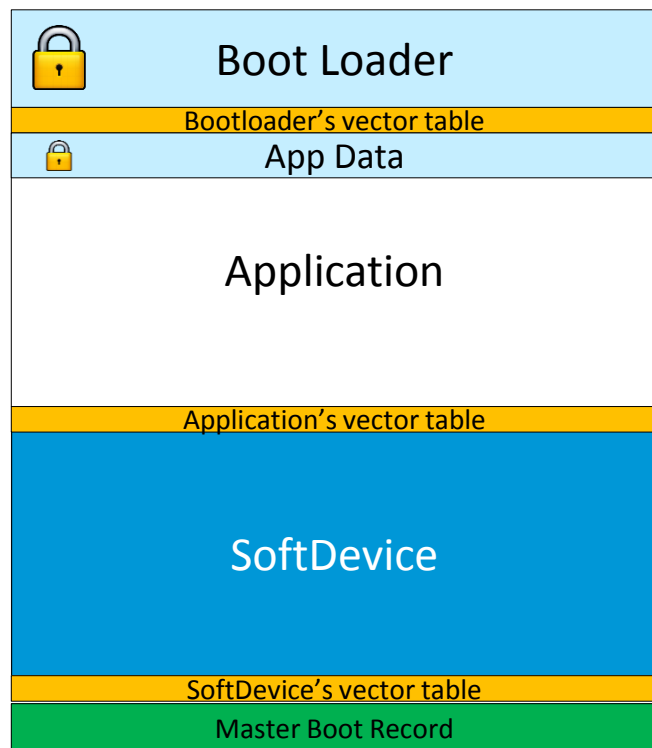


- A small program that's always executed first when device booting up.
 - Write and erase protectable (PROTENSET)

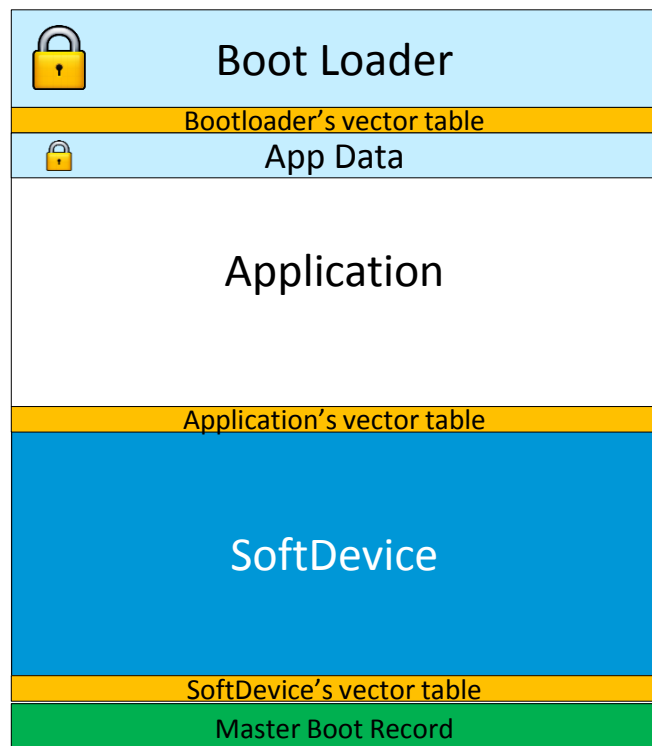
- Functions:
 - Control device
 - Receive updates firmware images via OTA or UART
 - Swap the new images with the old ones

- Available in nRF51 SDK
 - Source code
 - Can be used as is or customised

nRF51 SW architecture

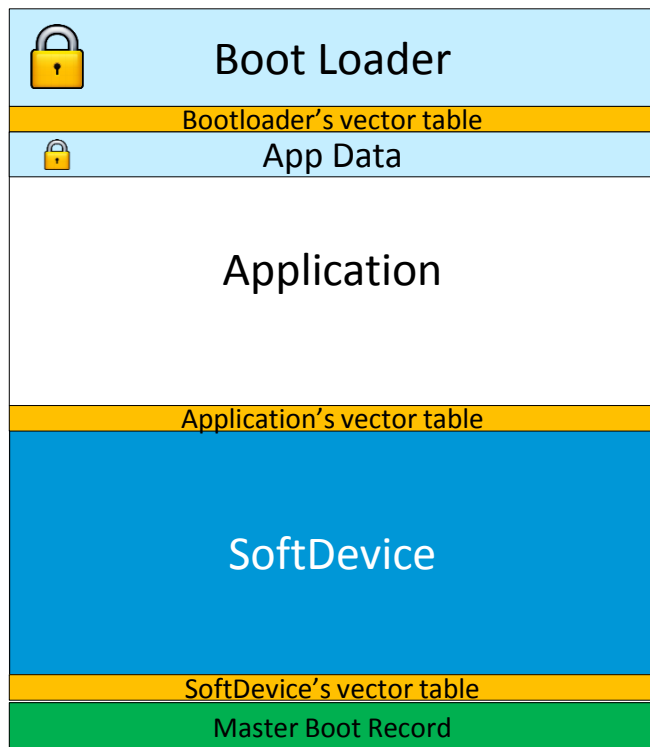


nRF51 SW architecture



- nRF51 Software architecture is unique
 - Clean split of protocol stack (BLE) and app
 - Separate hex files

nRF51 SW architecture



- nRF51 Software architecture is unique
 - Clean split of protocol stack (BLE) and app
 - Separate hex files
- Unique benefits for nRF51 DFU:
 - Use same BLE stack for DFU as normal communication
 - Split updates to only what is needed
 - Application only
 - DFU transfer size = application
 - SoftDevice only
 - DFU transfer = SoftDevice + application
 - Softdevice + bootloader
 - DFU transfer = Softdevice + bootloader + application

Application Update (Supported from S110 v5.0 and later)

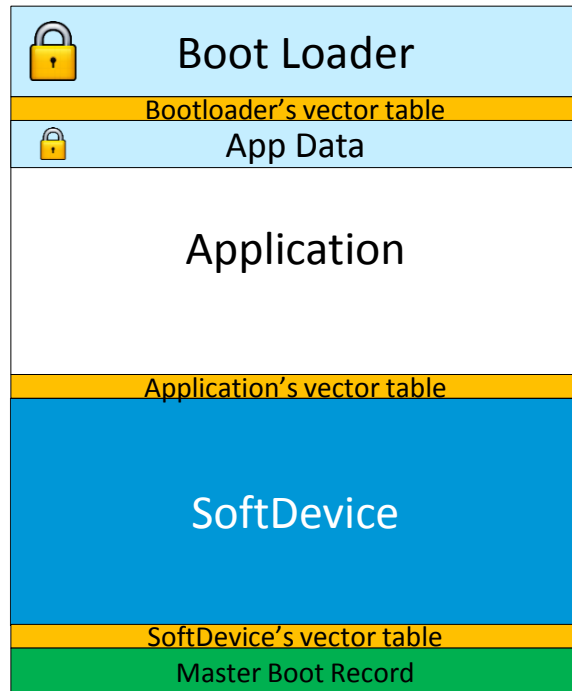
Two methods of updating:

Application Update (Supported from S110 v5.0 and later)

Two methods of updating:

Single Bank Update

Minimum code space overhead

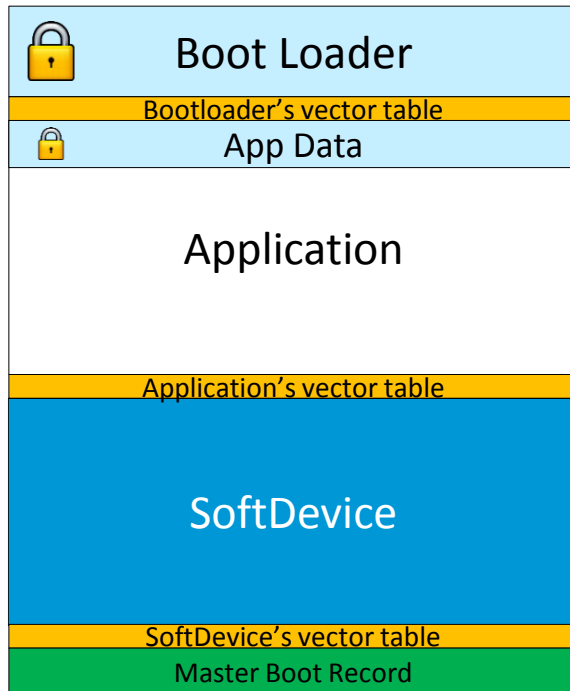


Application Update (Supported from S110 v5.0 and later)

Two methods of updating:

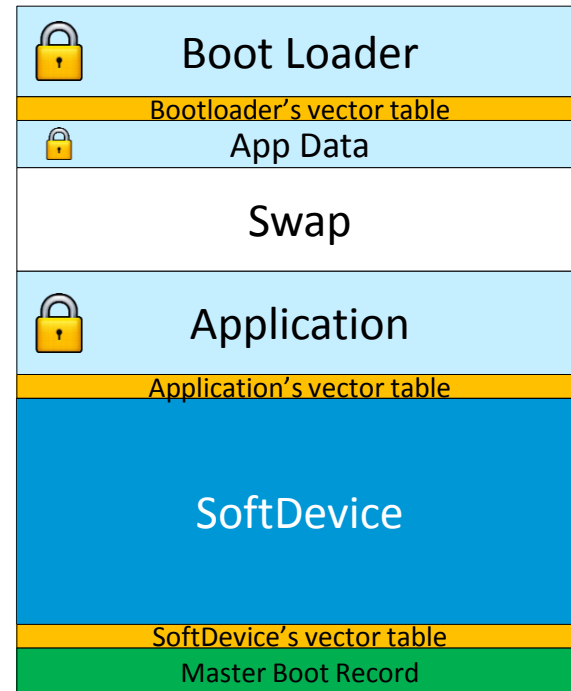
Single Bank Update

Minimum code space overhead



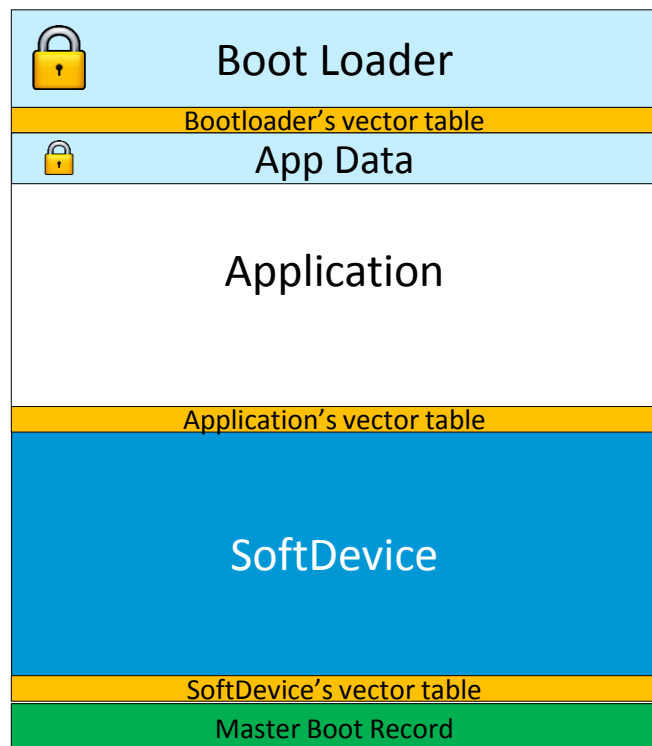
Dual Bank update

Maximum safety



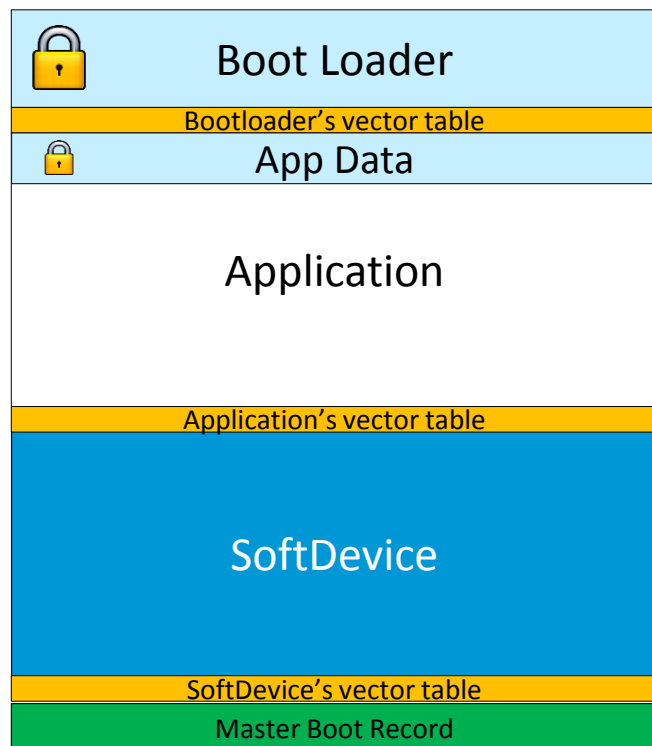
Single Bank update

Application



Single Bank update

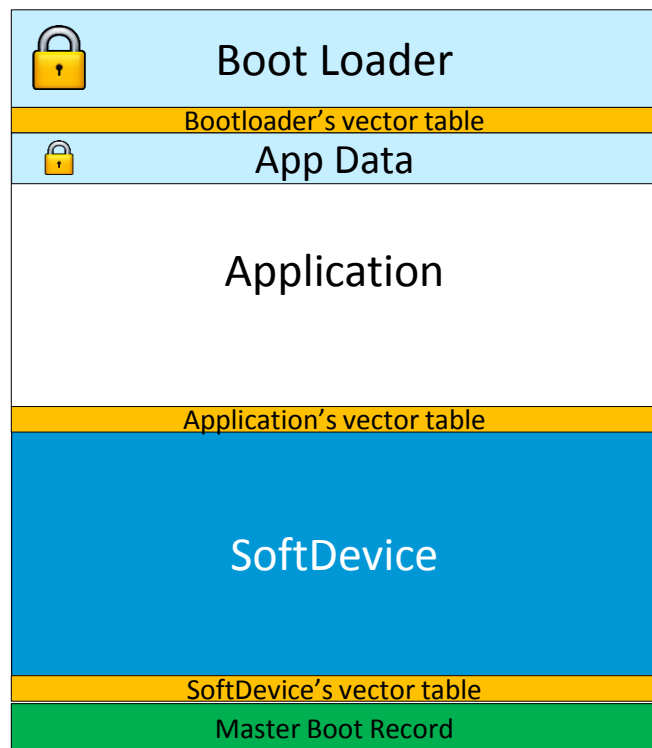
Application



- Single bank update:
 - Uses Serial or BLE for transport
 - Overwrites existing application
 - Resets and hand over control to the new application

Single Bank update

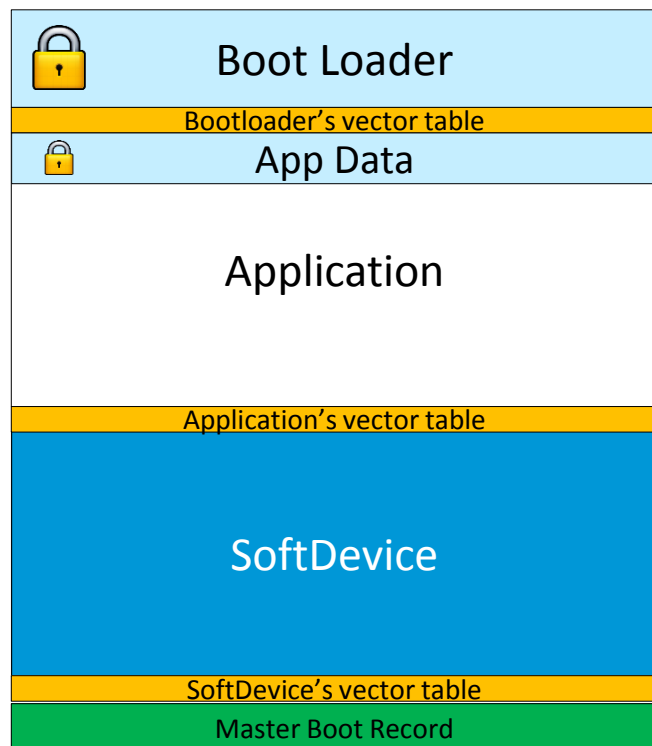
Application



- Single bank update:
 - Uses Serial or BLE for transport
 - Overwrites existing application
 - Resets and hand over control to the new application
- Characteristics:
 - Minimum code space overhead
 - If DFU fails Boot Loader **MUST** continue its attempts until update is successful...
 - Device is “bricked” until a successful update
 - Generally recommended for ‘stable’ transports

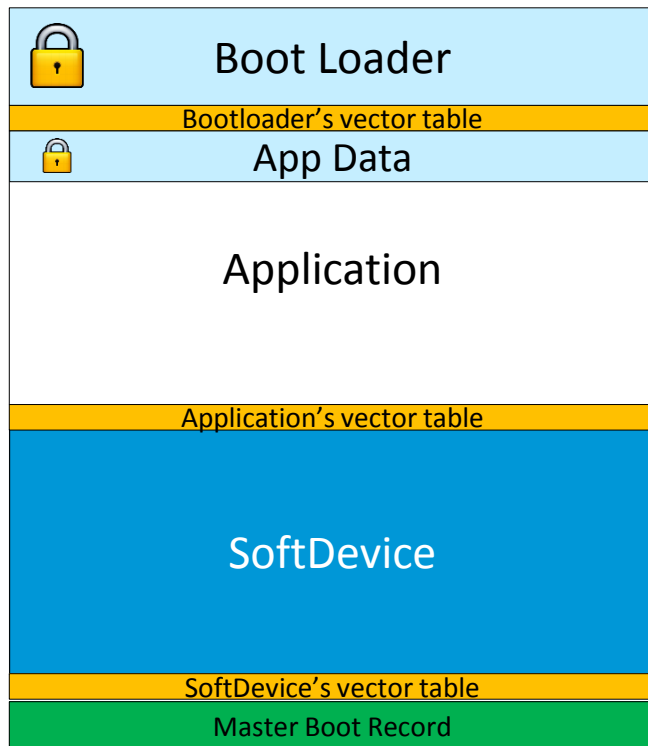
Single Bank update

Application



Single Bank update

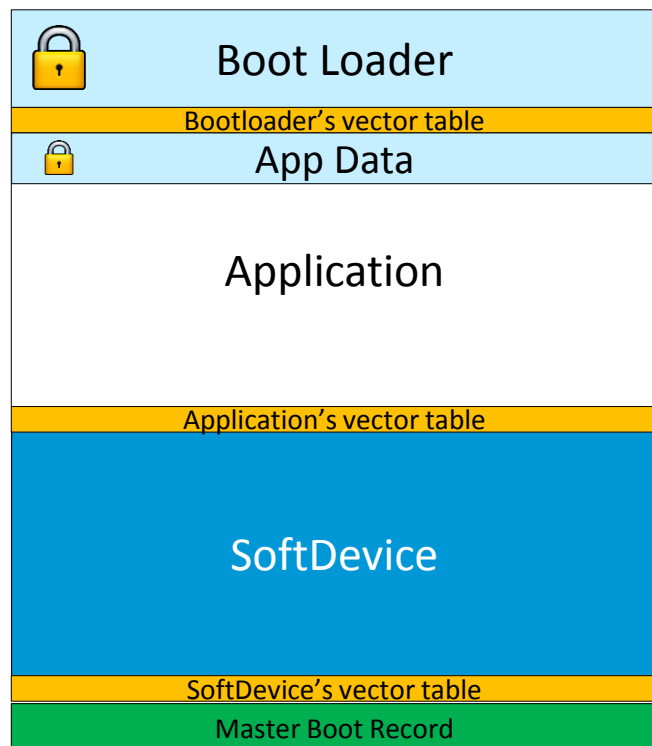
Application



- Current single bank bootloader:
 - Using UART transportation
 - Supported from nRF51 SDK v4.4.0
 - Code size of bootloader 14kB

Single Bank update

Application

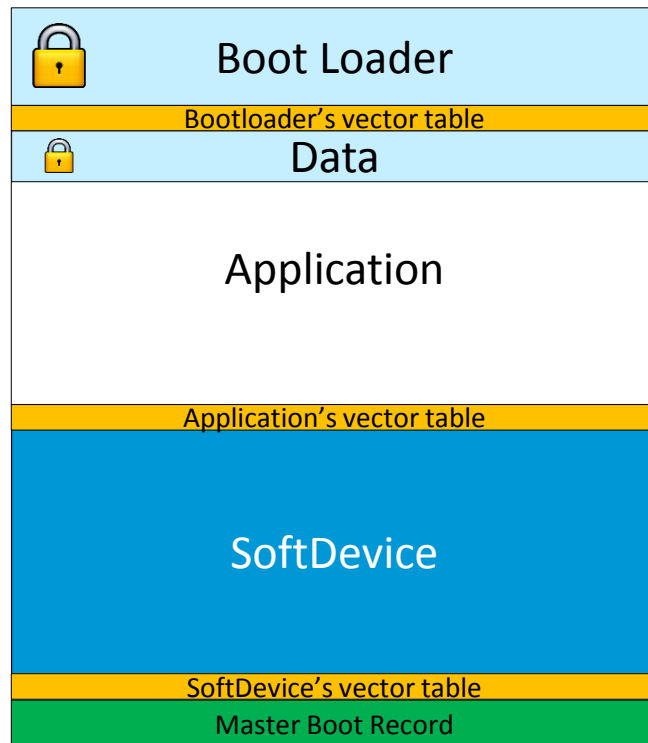


- Current single bank bootloader:
 - Using UART transportation
 - Supported from nRF51 SDK v4.4.0
 - Code size of bootloader 14kB
- Max application size:
 - $\text{App} = \text{Flash size} - \text{Softdevice} - \text{Bootloader}$
 - nRF51822 XXAA (256 kB): 146 kB*
 - nRF51822 QFAB (128 kB): 18 kB*

**Calculated base on S110 v8.0 size = 96kB*

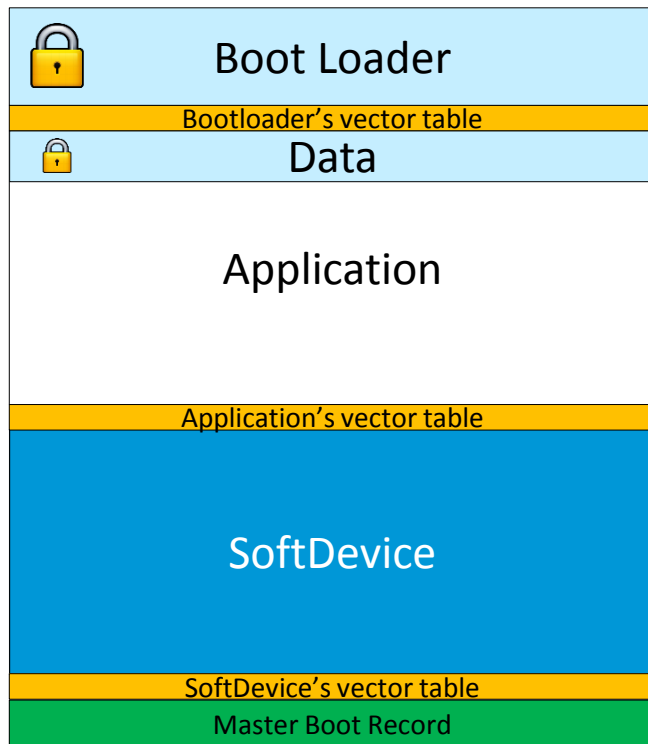
Single Bank update

Performance



Single Bank update

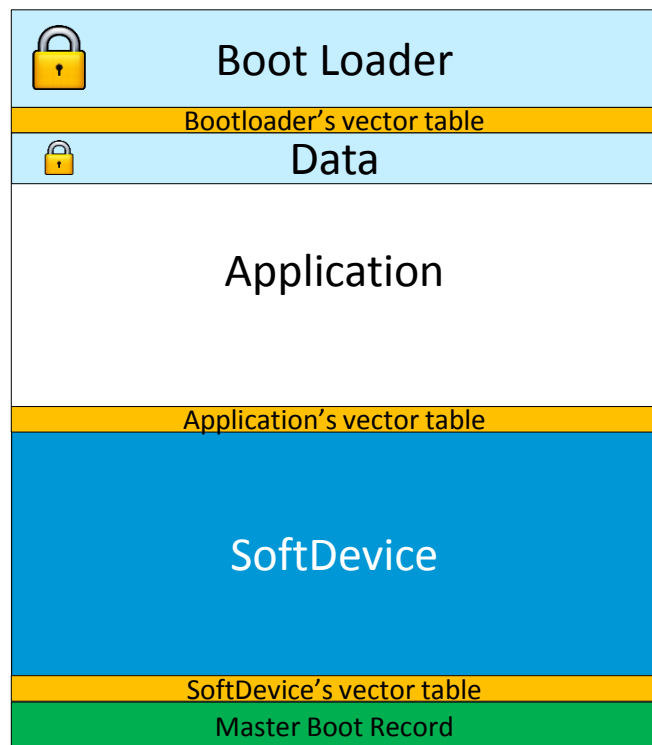
Performance



- UART baud rate supported:
 - nRF51 SDK example: 38400 baud
 - nRF51 SDK transport layer can do 1Mbaud
 - Production friendly, same interface as BT Direct Test Mode

Single Bank update

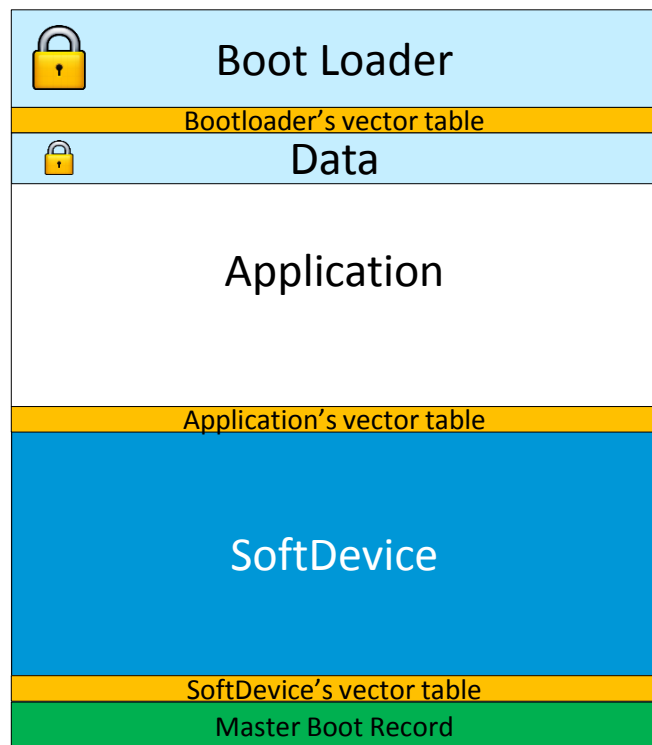
Performance



- UART baud rate supported:
 - nRF51 SDK example: 38400 baud
 - nRF51 SDK transport layer can do 1Mbaud
 - Production friendly, same interface as BT Direct Test Mode
- Transfer speed*
 - 20 kB application (e.g. HRM) @ 38.4 kbaud: **4.1 sec**
 - 155 kB (full app space) @ 38.4 kbaud: **33.6 sec**
 - 155 kB @ 1 Mbaud: 1.3 sec

Single Bank update

Performance



- UART baud rate supported:
 - nRF51 SDK example: 38400 baud
 - nRF51 SDK transport layer can do 1Mbaud
 - Production friendly, same interface as BT Direct Test Mode

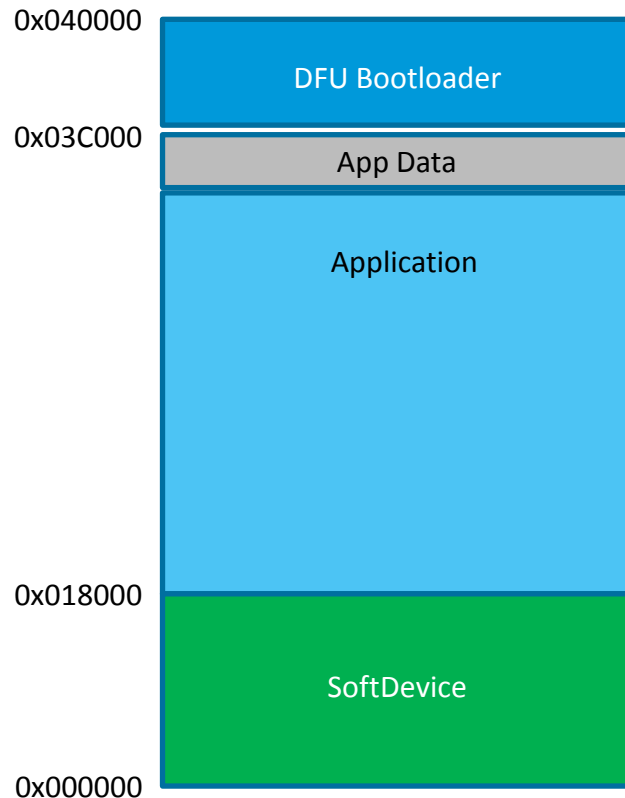
- Transfer speed*
 - 20 kB application (e.g. HRM) @ 38.4 kbaud: **4.1 sec**
 - 155 kB (full app space) @ 38.4 kbaud: **33.6 sec**
 - 155 kB @ 1 Mbaud: 1.3 sec

- Peer-side DFU master tool provided:
 - Windows command line utility
 - nRFgo studio

**for reference only*

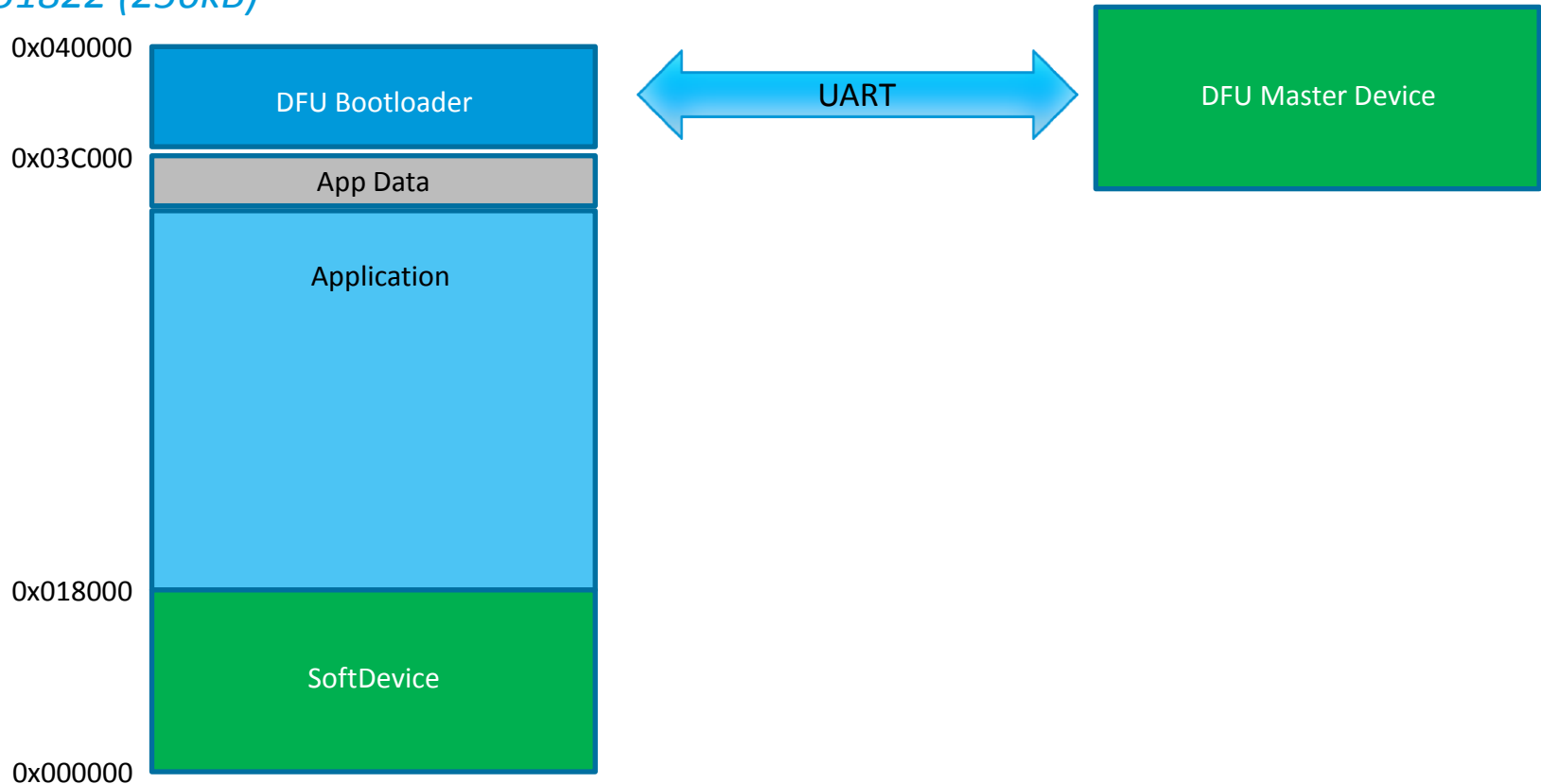
Application firmware single bank update

nRF51822 (256kB)



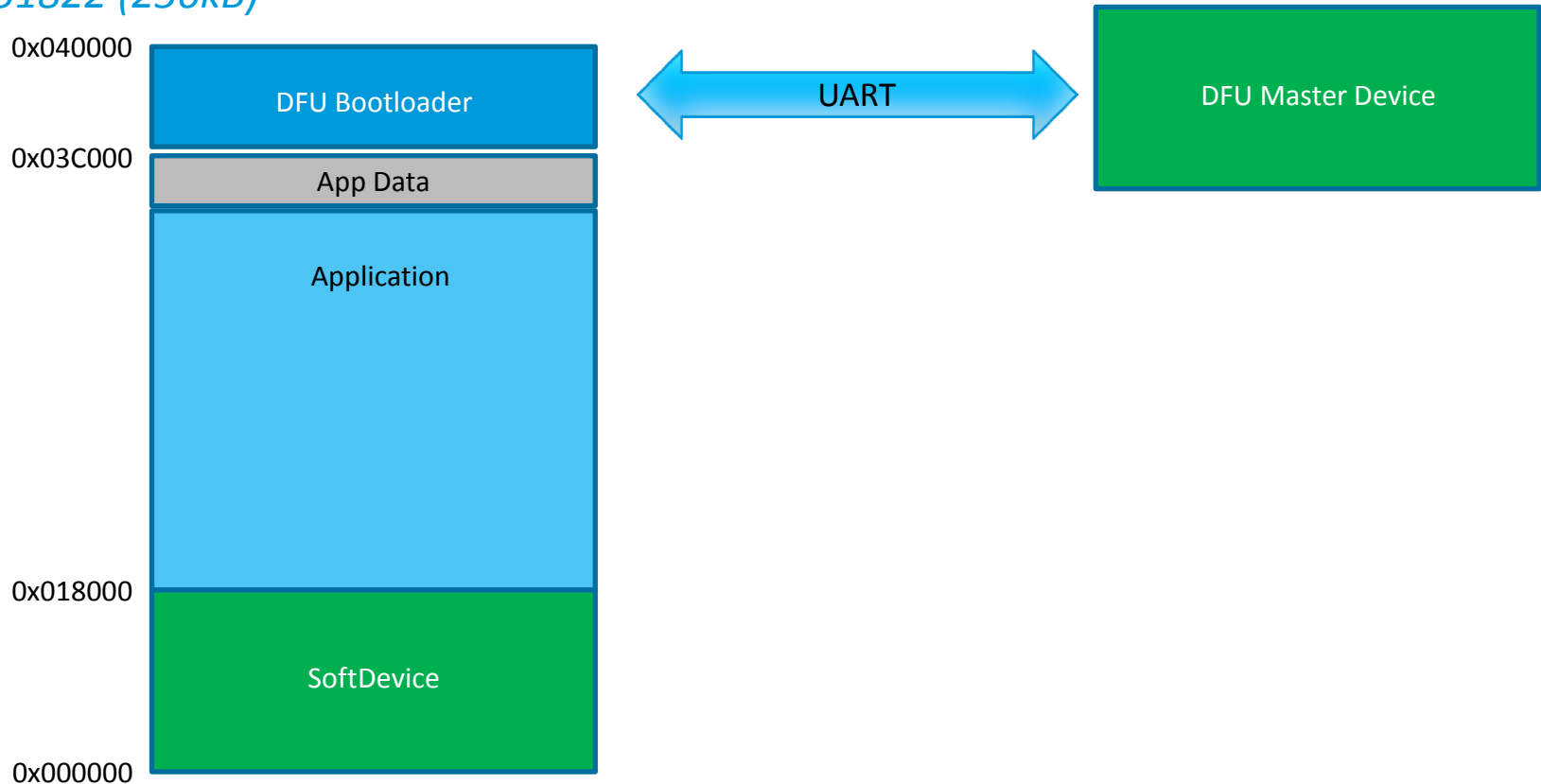
Application firmware single bank update

nRF51822 (256kB)



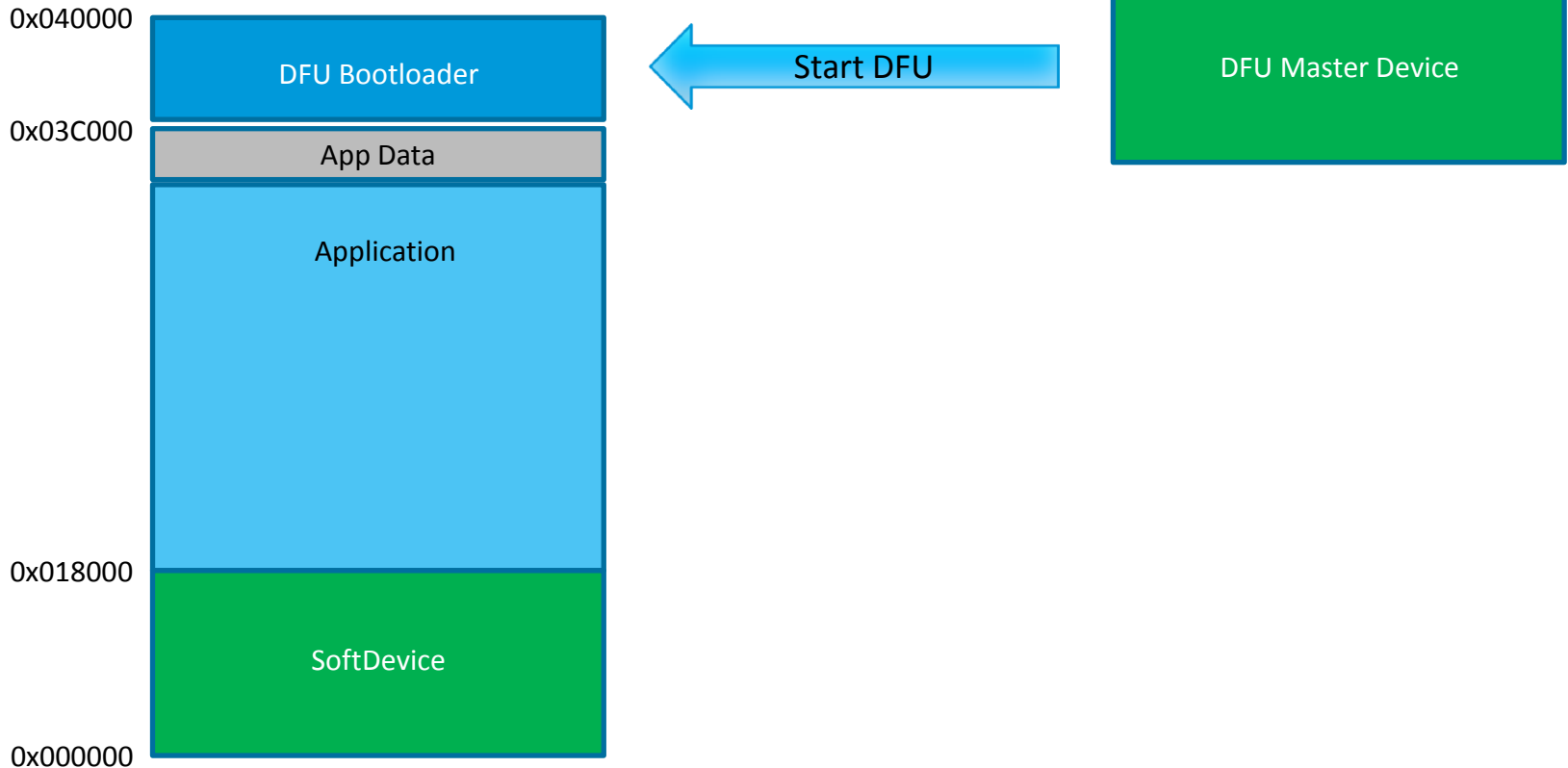
Application firmware single bank update

nRF51822 (256kB)



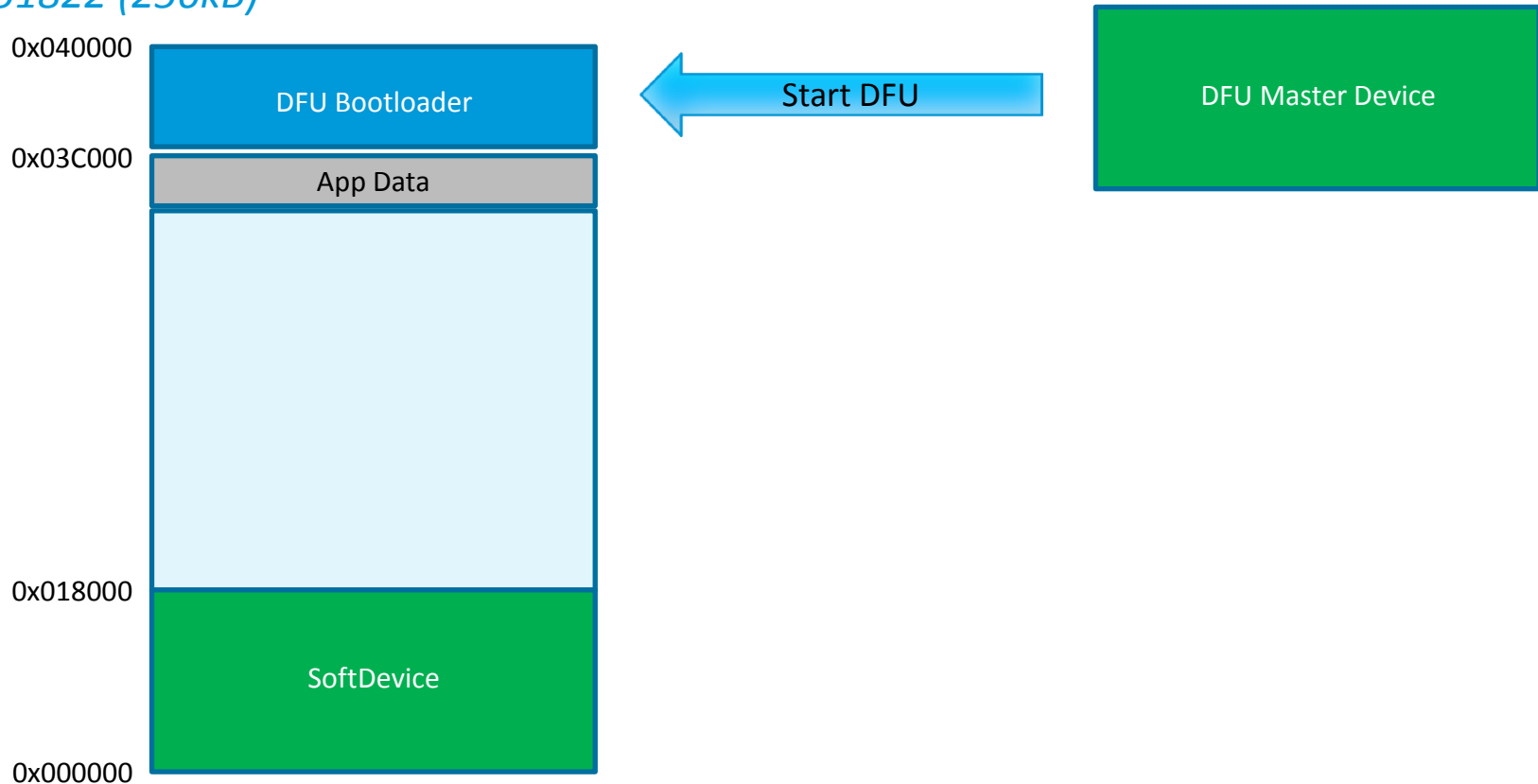
Application firmware single bank update

nRF51822 (256kB)



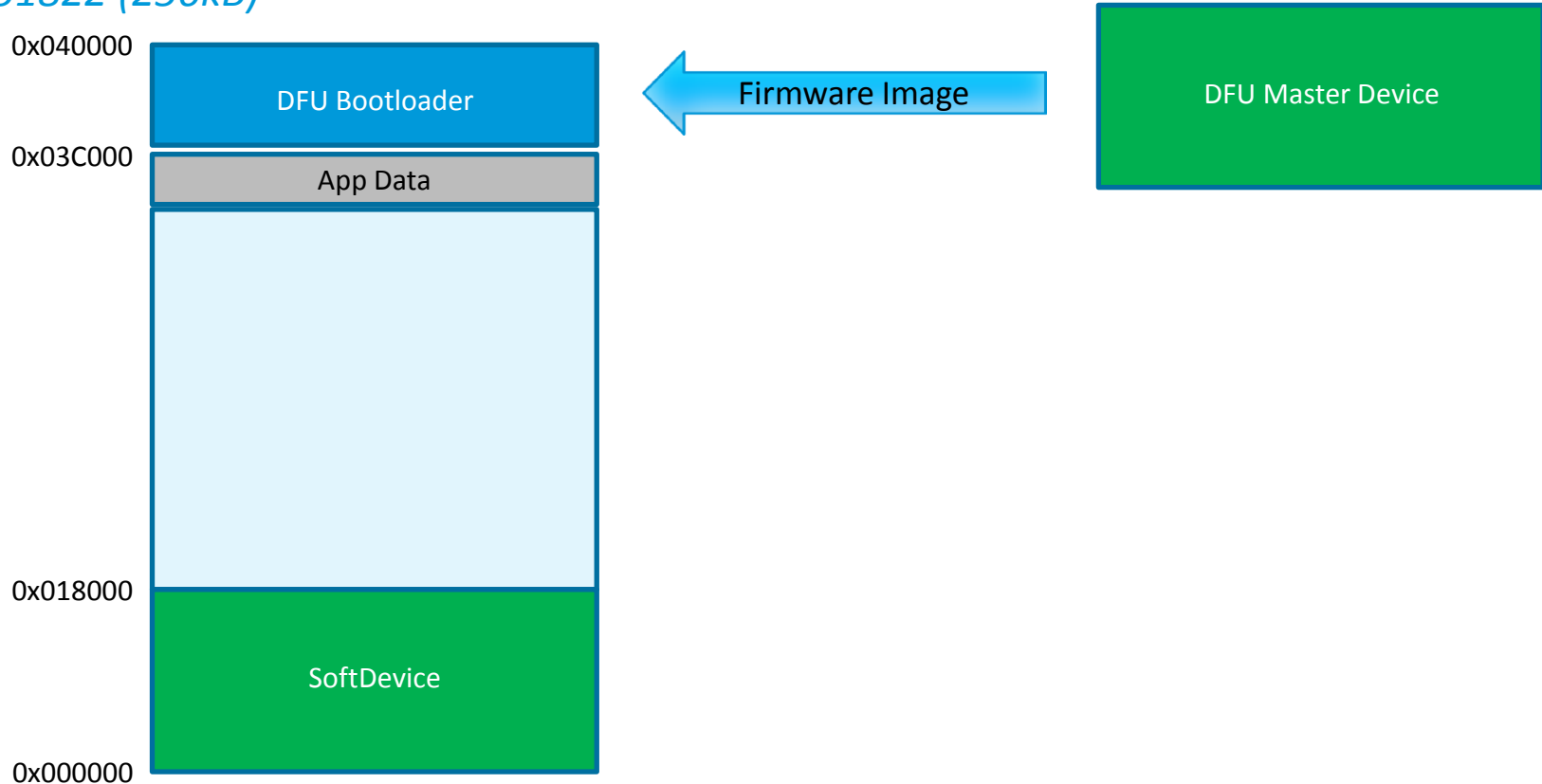
Application firmware single bank update

nRF51822 (256kB)



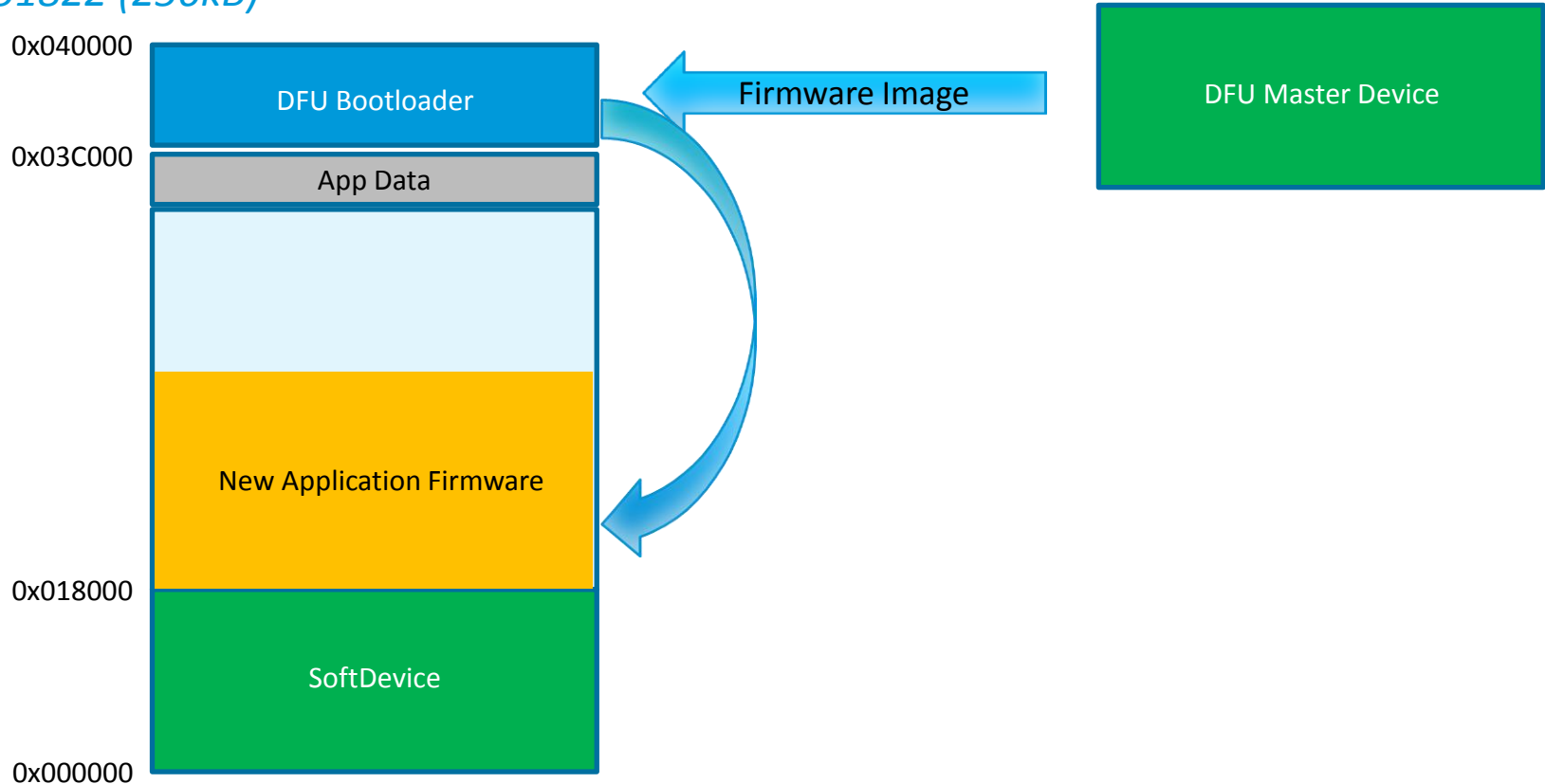
Application firmware single bank update

nRF51822 (256kB)



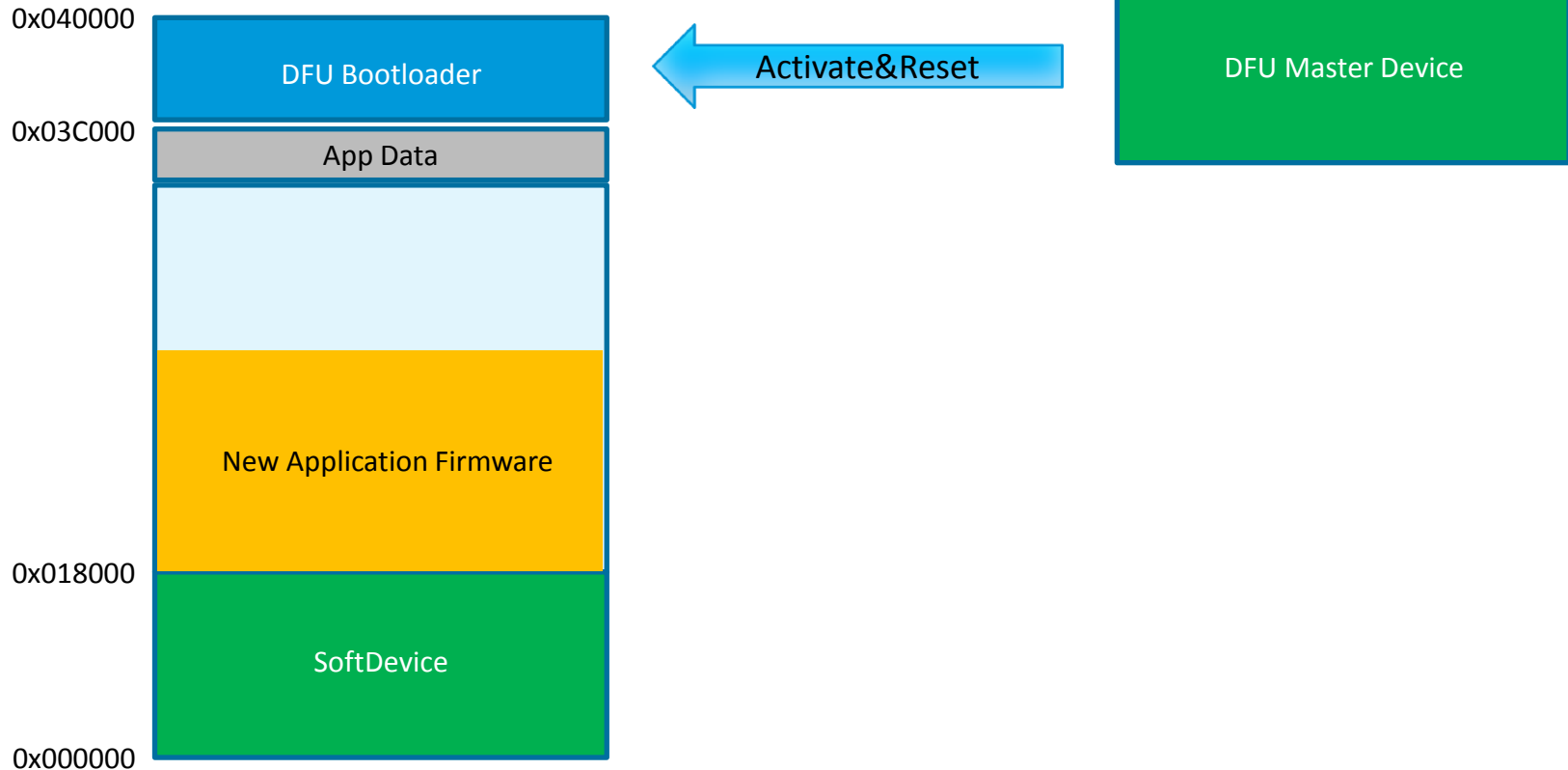
Application firmware single bank update

nRF51822 (256kB)



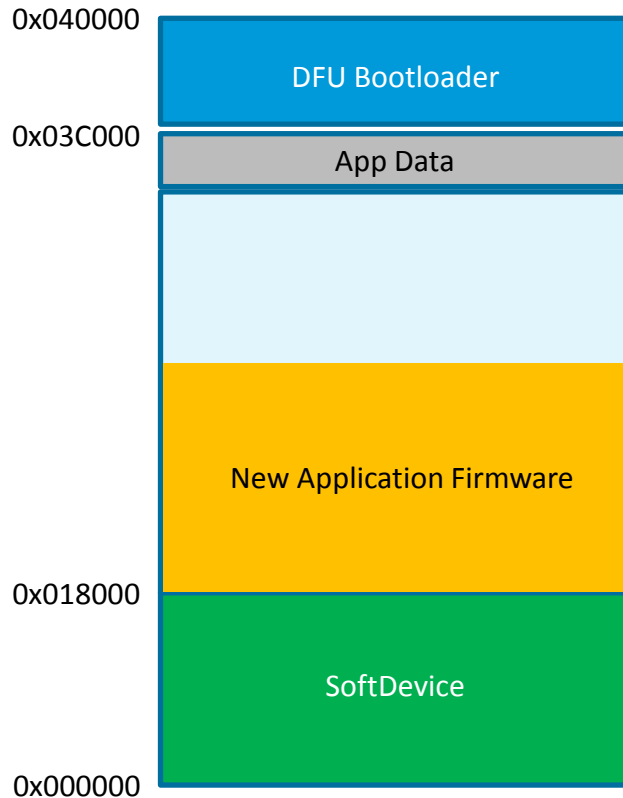
Application firmware single bank update

nRF51822 (256kB)



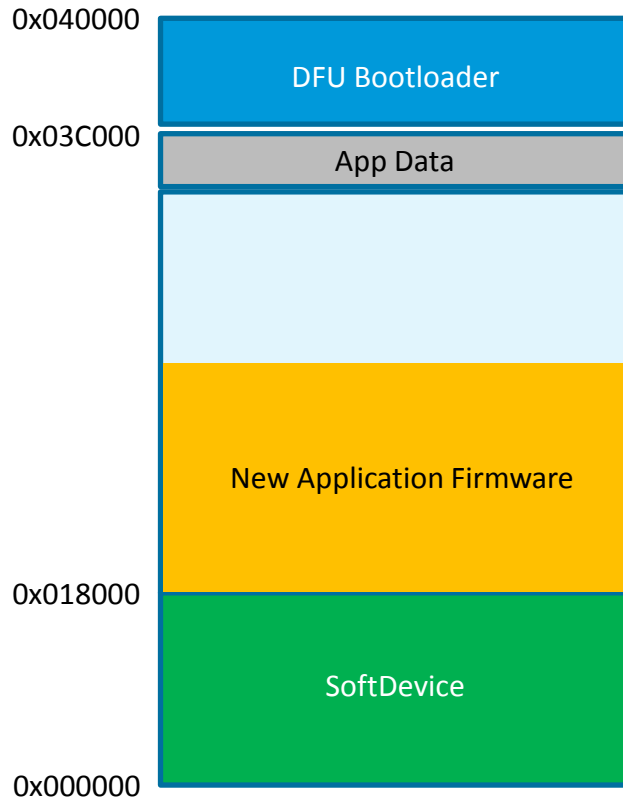
Application firmware single bank update

nRF51822 (256kB)



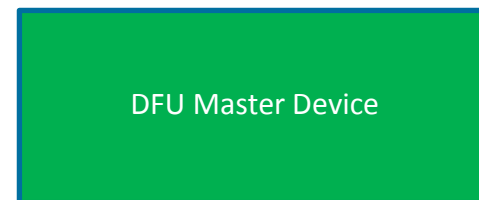
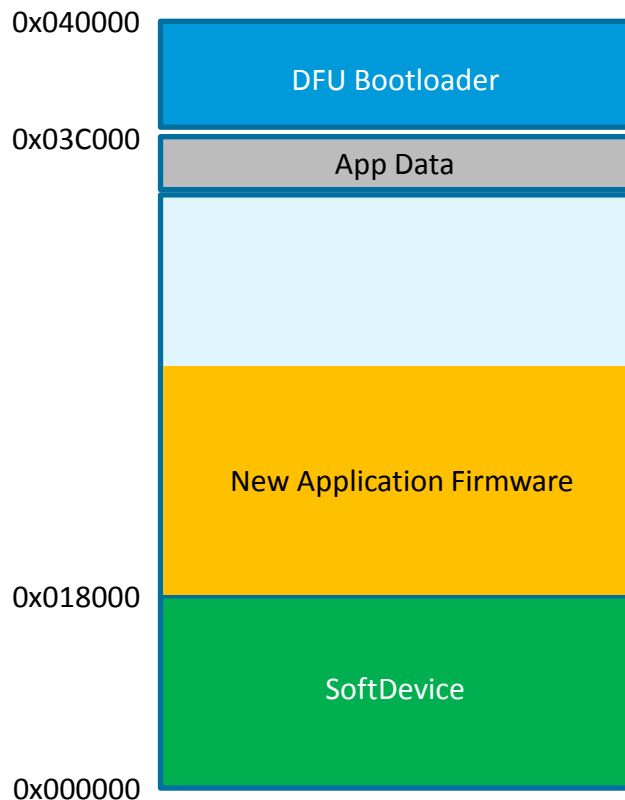
Application firmware single bank update

nRF51822 (256kB)



Application firmware single bank update

nRF51822 (256kB)

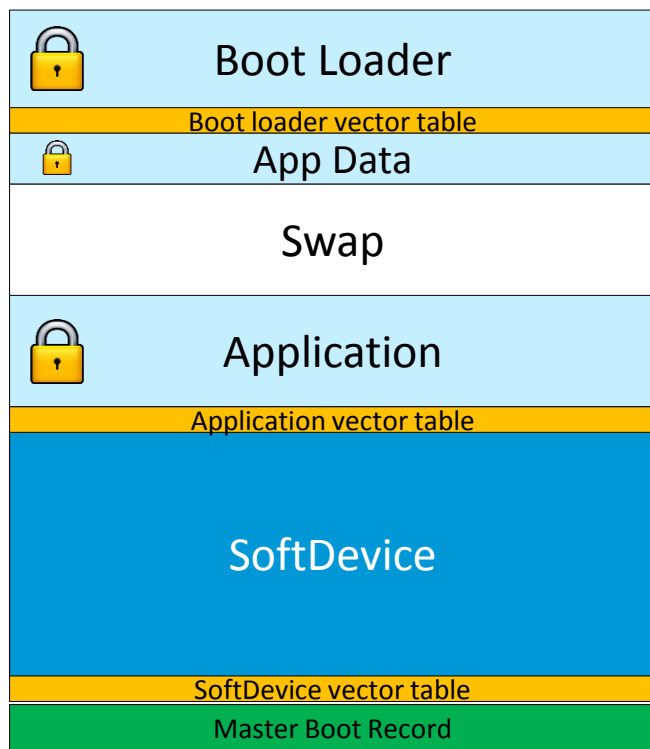


Memory Range*	Size	Usage
0x0003C000 - 0x00040000	16 kB	Bootloader
0x00018000 - 0x0003C000	144kB	Code Region 1: Application Code (BANK 0) and data
0x00000000 - 0x00018000	96 kB	Code Region 0: SoftDevice

*The memory layout is the layout from the DFU SDK example that is generic and can be improved

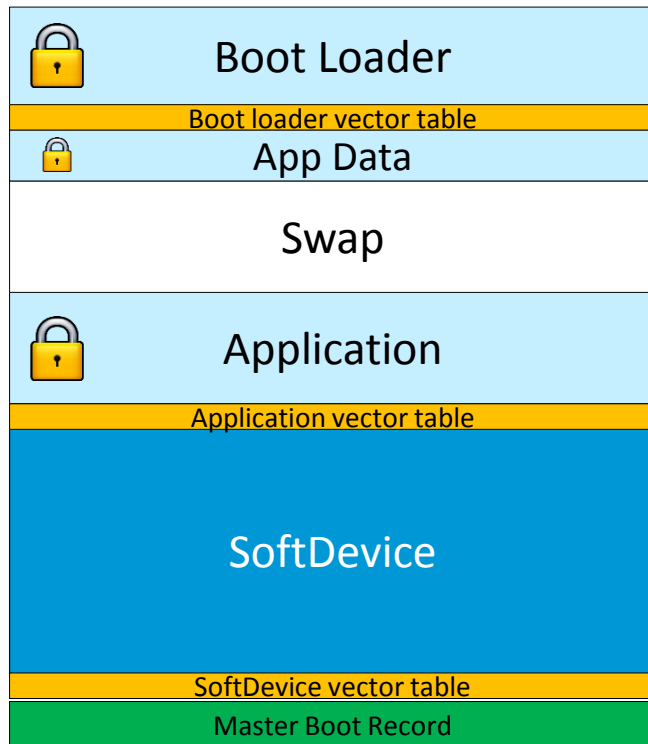
Dual Bank update

Application



Dual Bank update

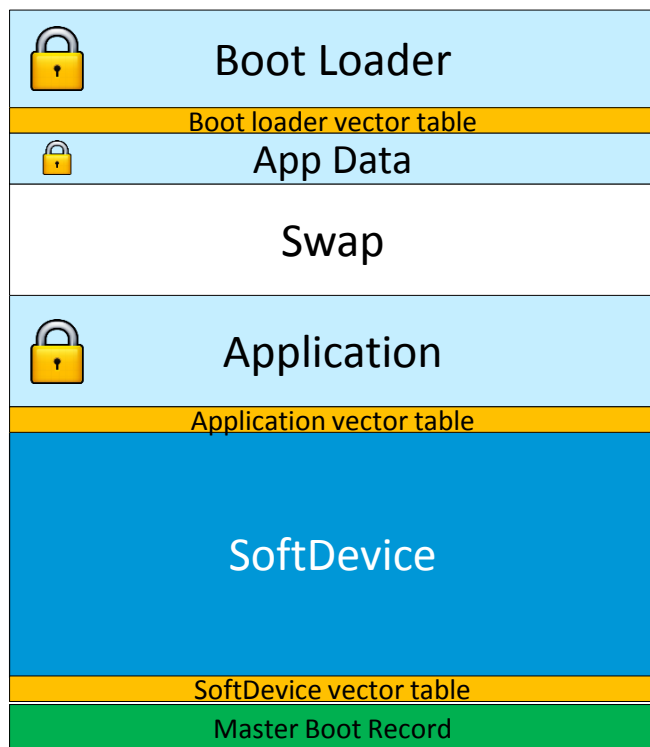
Application



- Dual Bank update:
 - UART interface or OTA via BLE
 - Bootloader writes new FW to swap space
 - When transfer is complete, new FW is copied to Application space and update is complete

Dual Bank update

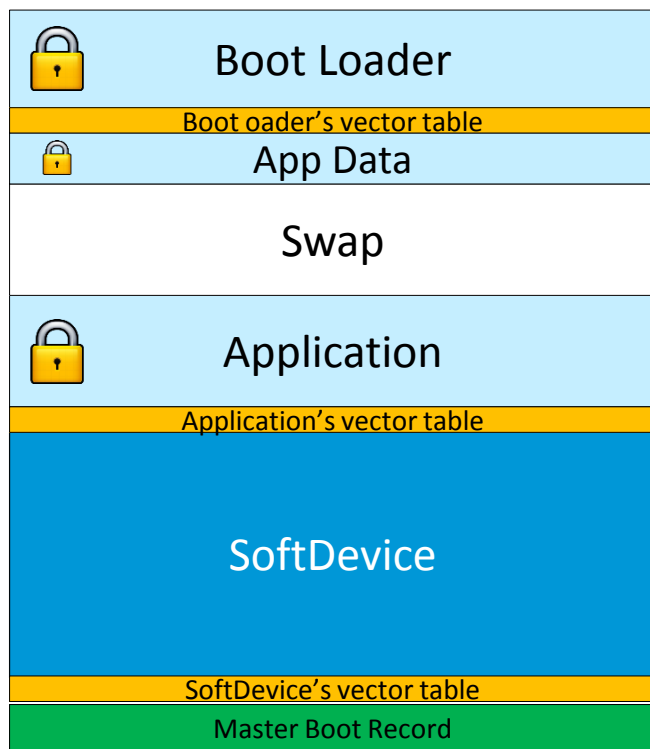
Application



- Dual Bank update:
 - UART interface or OTA via BLE
 - Bootloader writes new FW to swap space
 - When transfer is complete, new FW is copied to Application space and update is complete
- Characteristics:
 - 2x code space required for swap and app
 - If update fails, the boot loader can fall back to the original and still valid application
 - Device is never “bricked”
 - Maximum safety

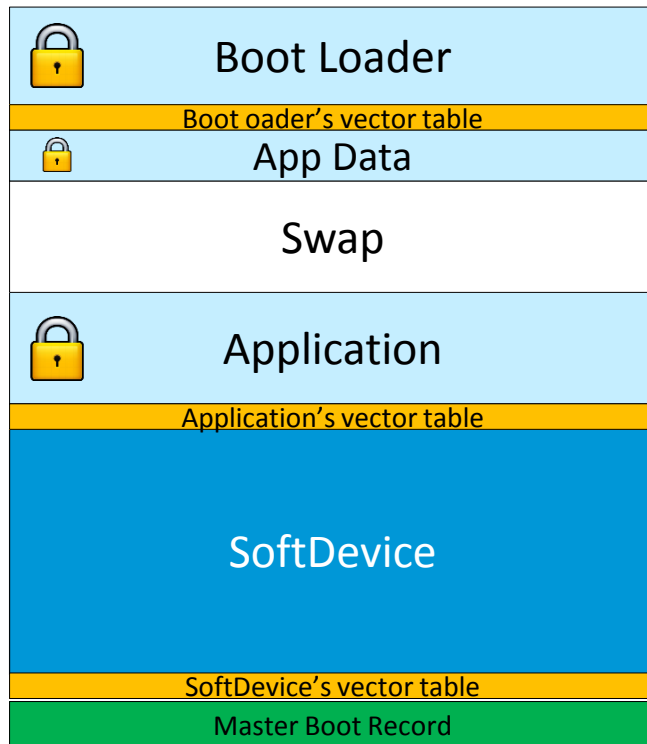
Dual Bank update

Via BLE



Dual Bank update

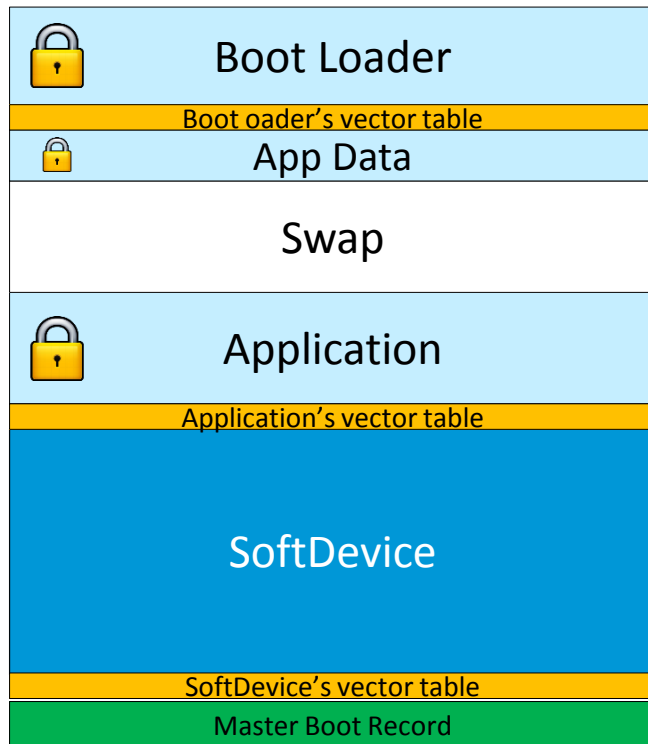
Via BLE



- Dual Bank update via BLE:
 - Swap enables reverting to old firmware if BLE link is permanently lost
- Supported from nRF51 SDK 4.4.0 (Aug 2013)
 - Code size for DFU Bootloader over BLE: 16 kB

Dual Bank update

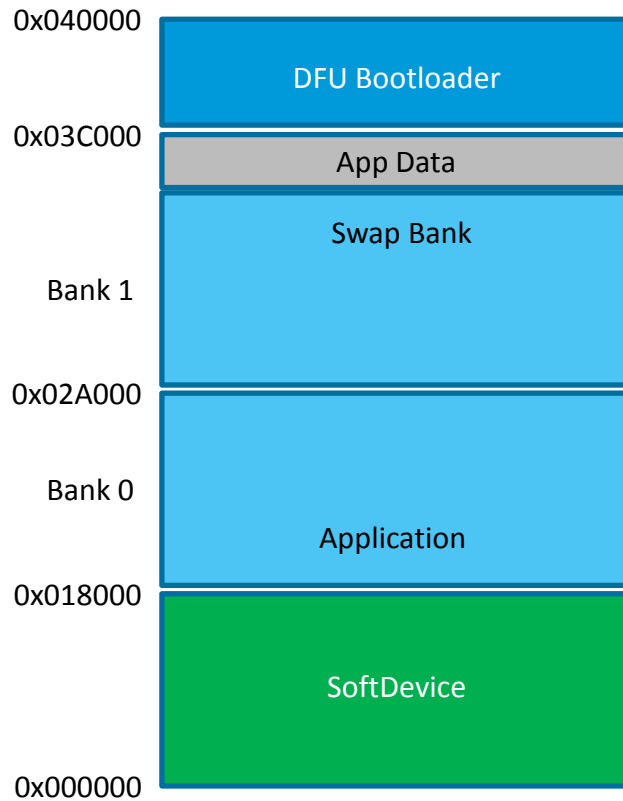
Via BLE



- Dual Bank update via BLE:
 - Swap enables reverting to old firmware if BLE link is permanently lost
- Supported from nRF51 SDK 4.4.0 (Aug 2013)
 - Code size for DFU Bootloader over BLE: 16 kB
- Max App size dual bank update:
 - App size = (flash memory – SD – boot loader) / 2
 - nRF51822 xxAA (256 kB): 72 kB
 - nRF51822 QFAB (128 kB): 8 kB

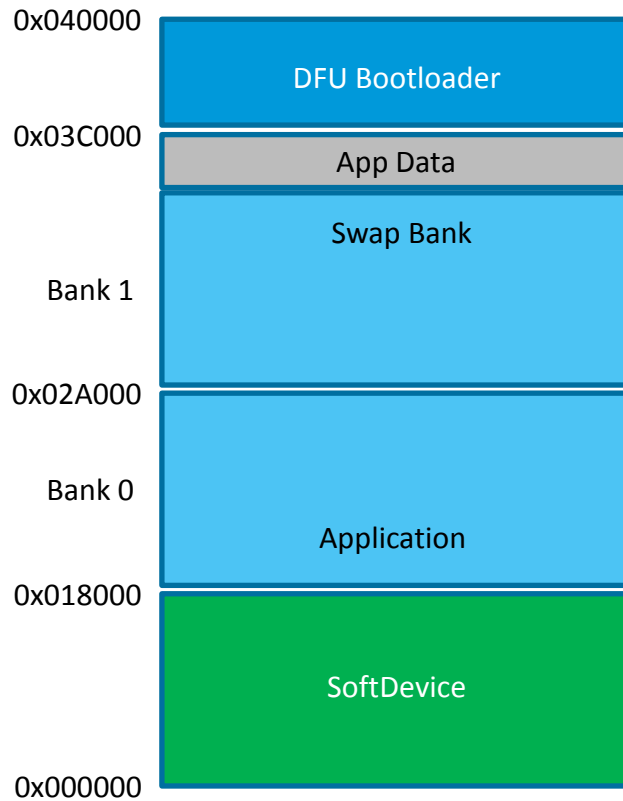
Dual Bank update

Application firmware update on a 256KB nRF51822



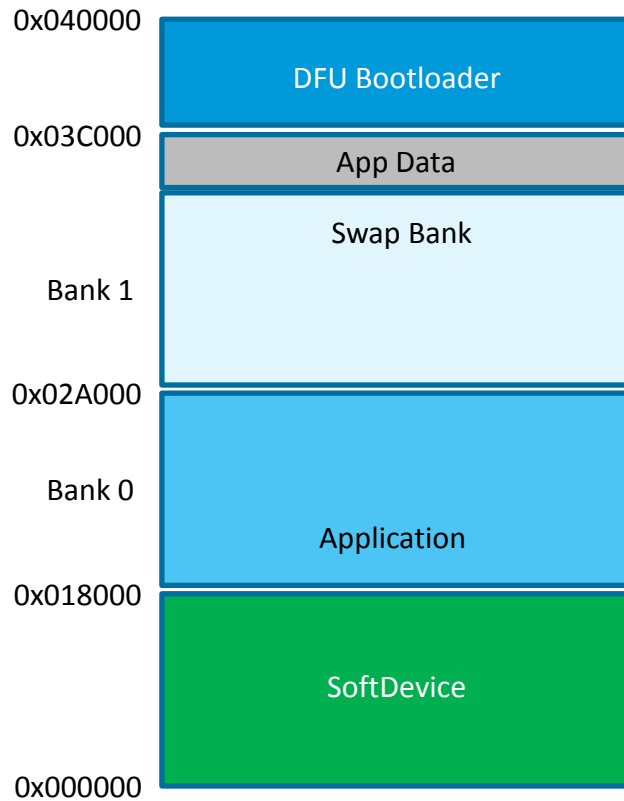
Dual Bank update

Application firmware update on a 256KB nRF51822



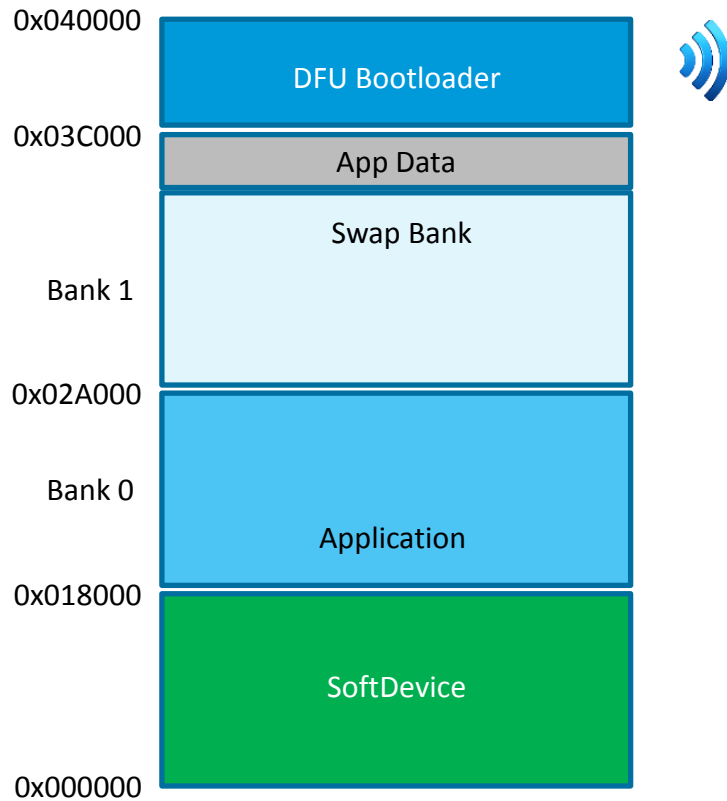
Dual Bank update

Application firmware update on a 256KB nRF51822



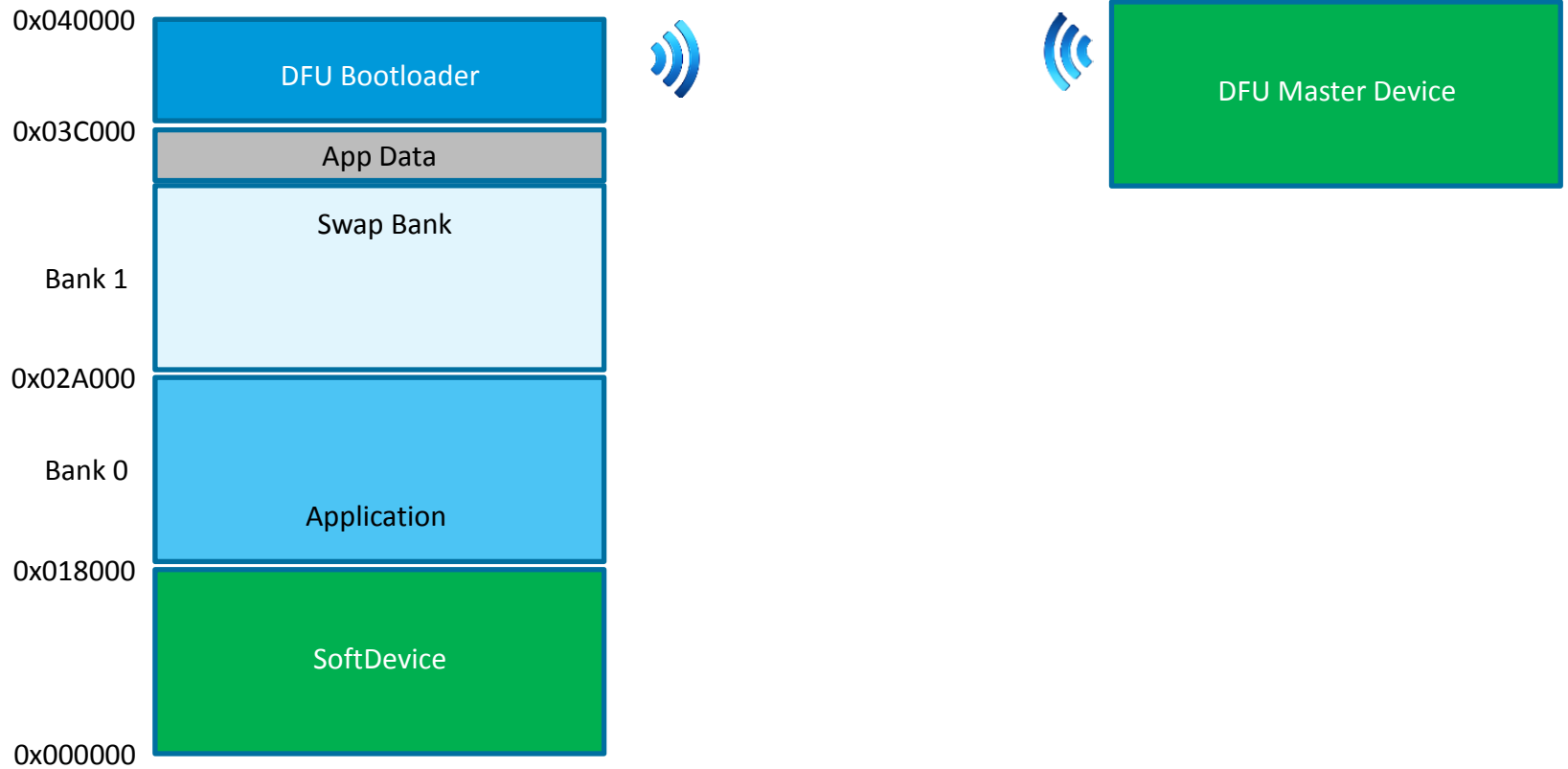
Dual Bank update

Application firmware update on a 256KB nRF51822



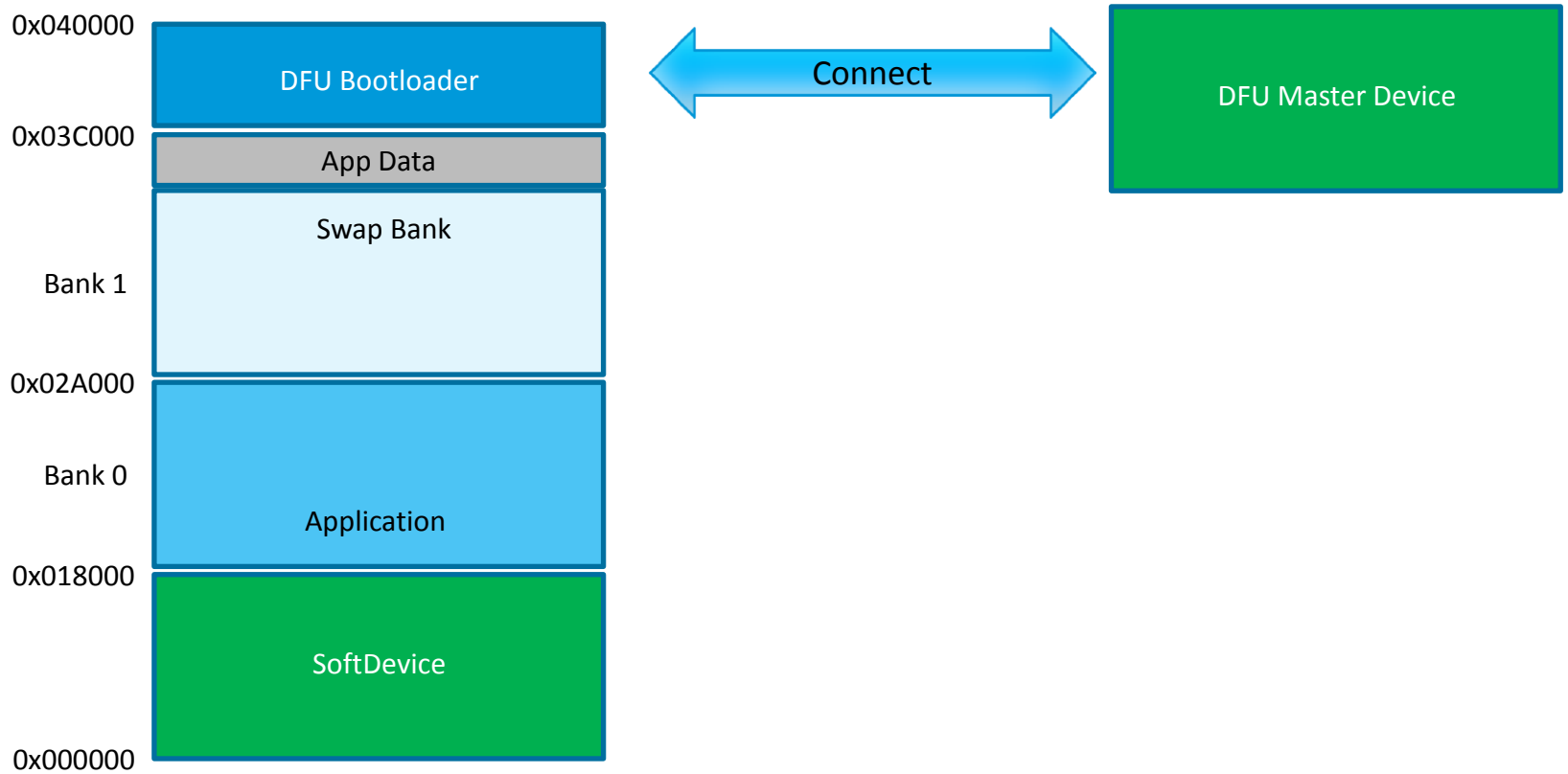
Dual Bank update

Application firmware update on a 256KB nRF51822



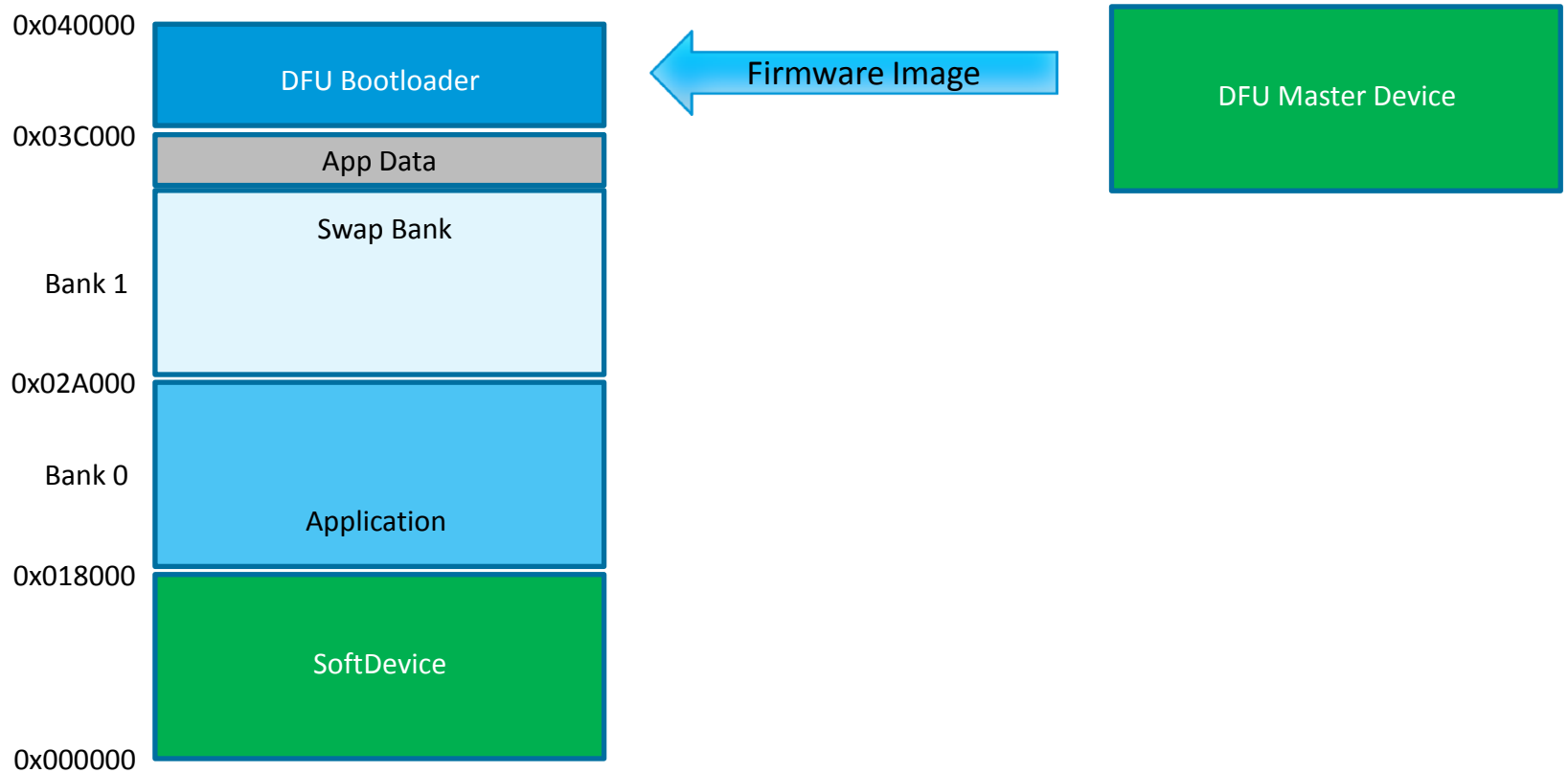
Dual Bank update

Application firmware update on a 256KB nRF51822



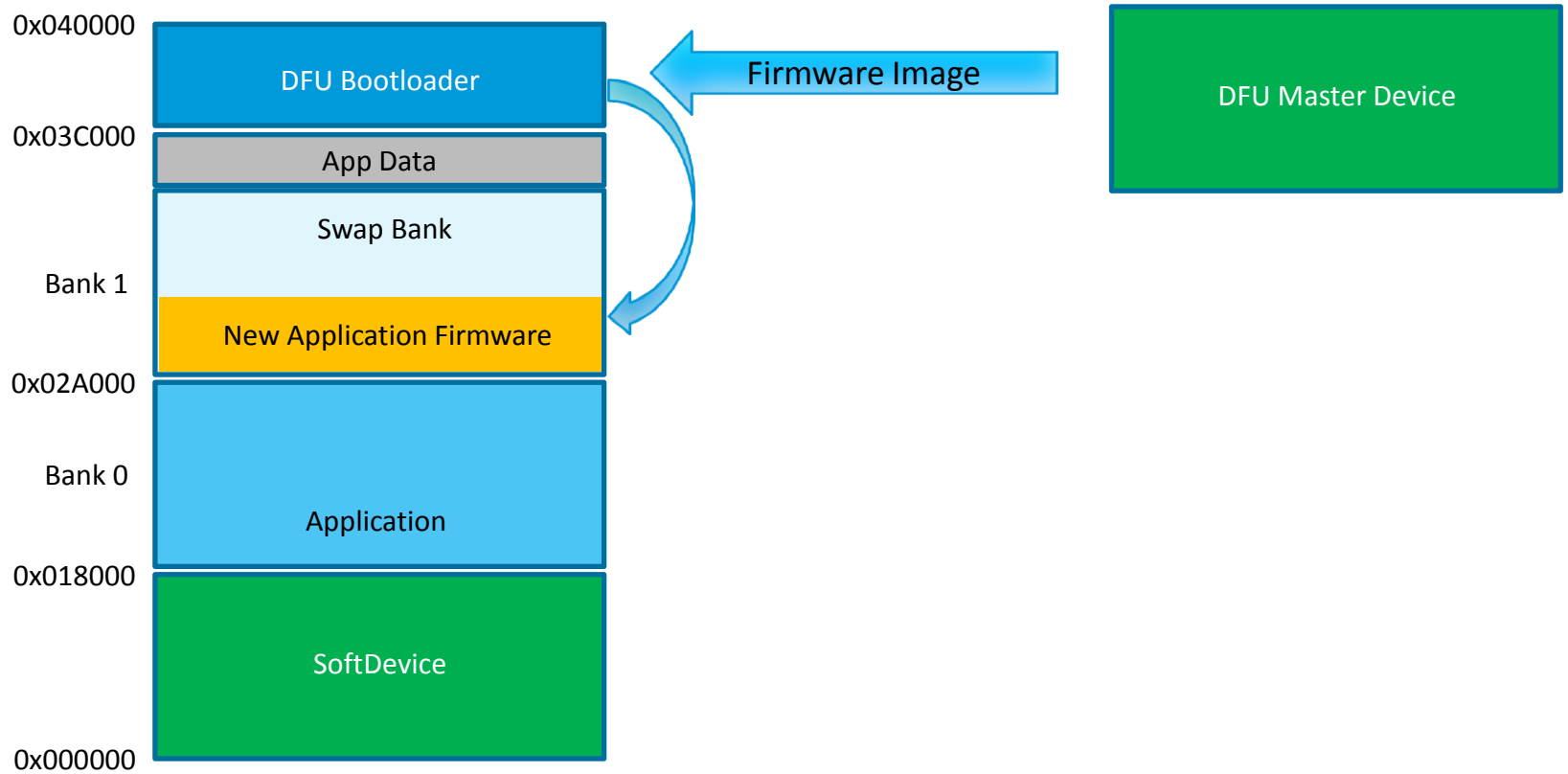
Dual Bank update

Application firmware update on a 256KB nRF51822



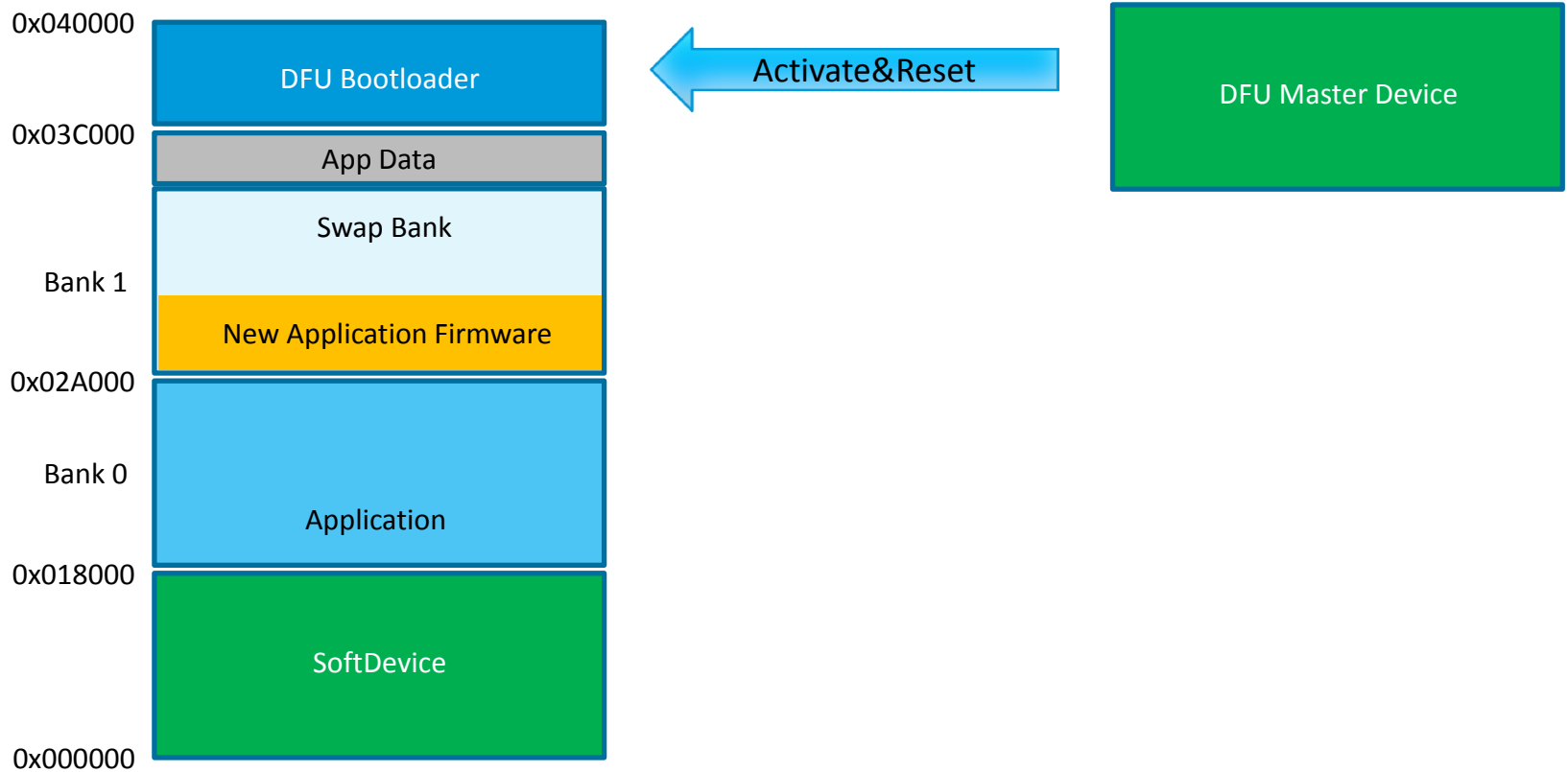
Dual Bank update

Application firmware update on a 256KB nRF51822



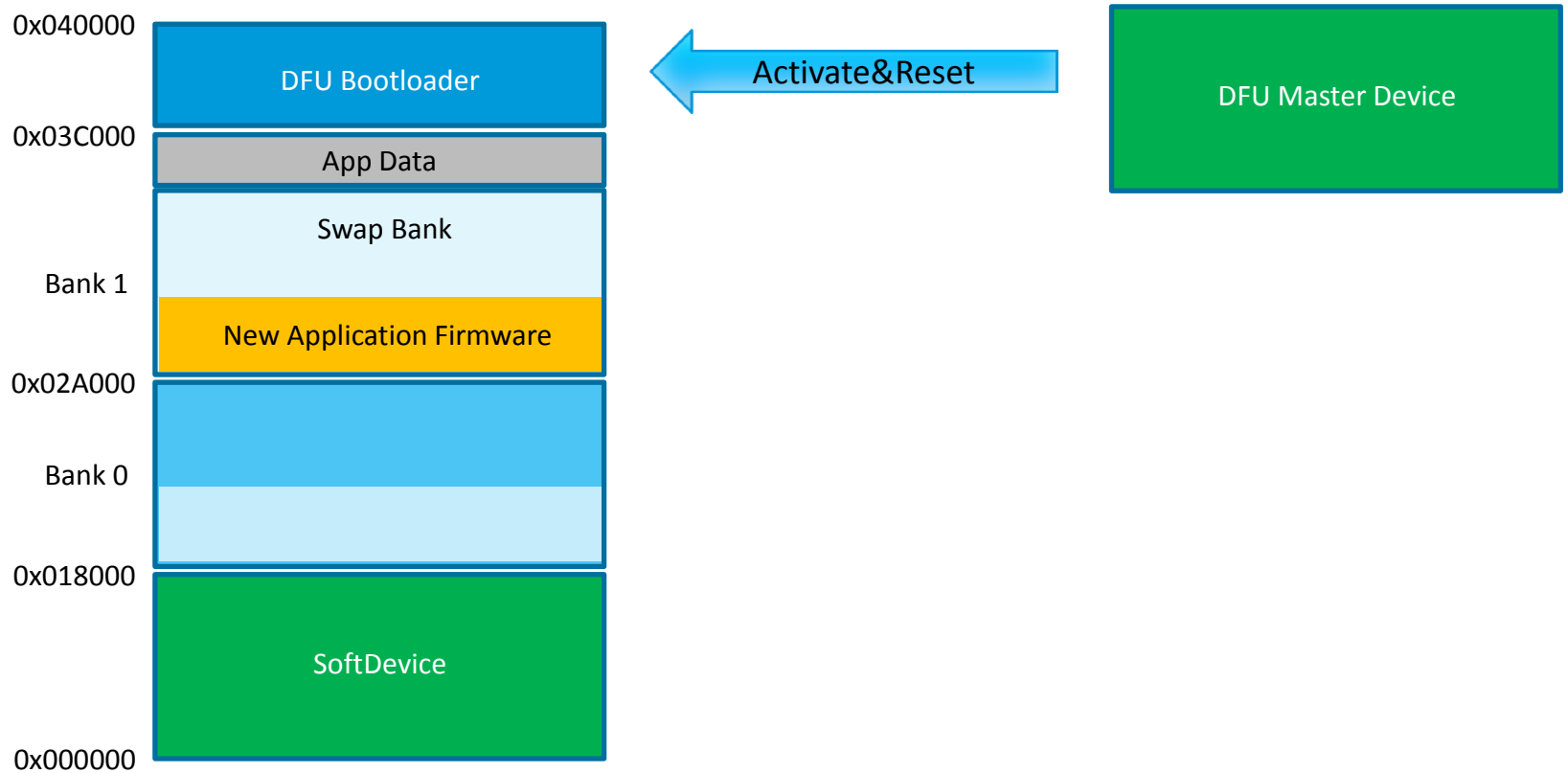
Dual Bank update

Application firmware update on a 256KB nRF51822



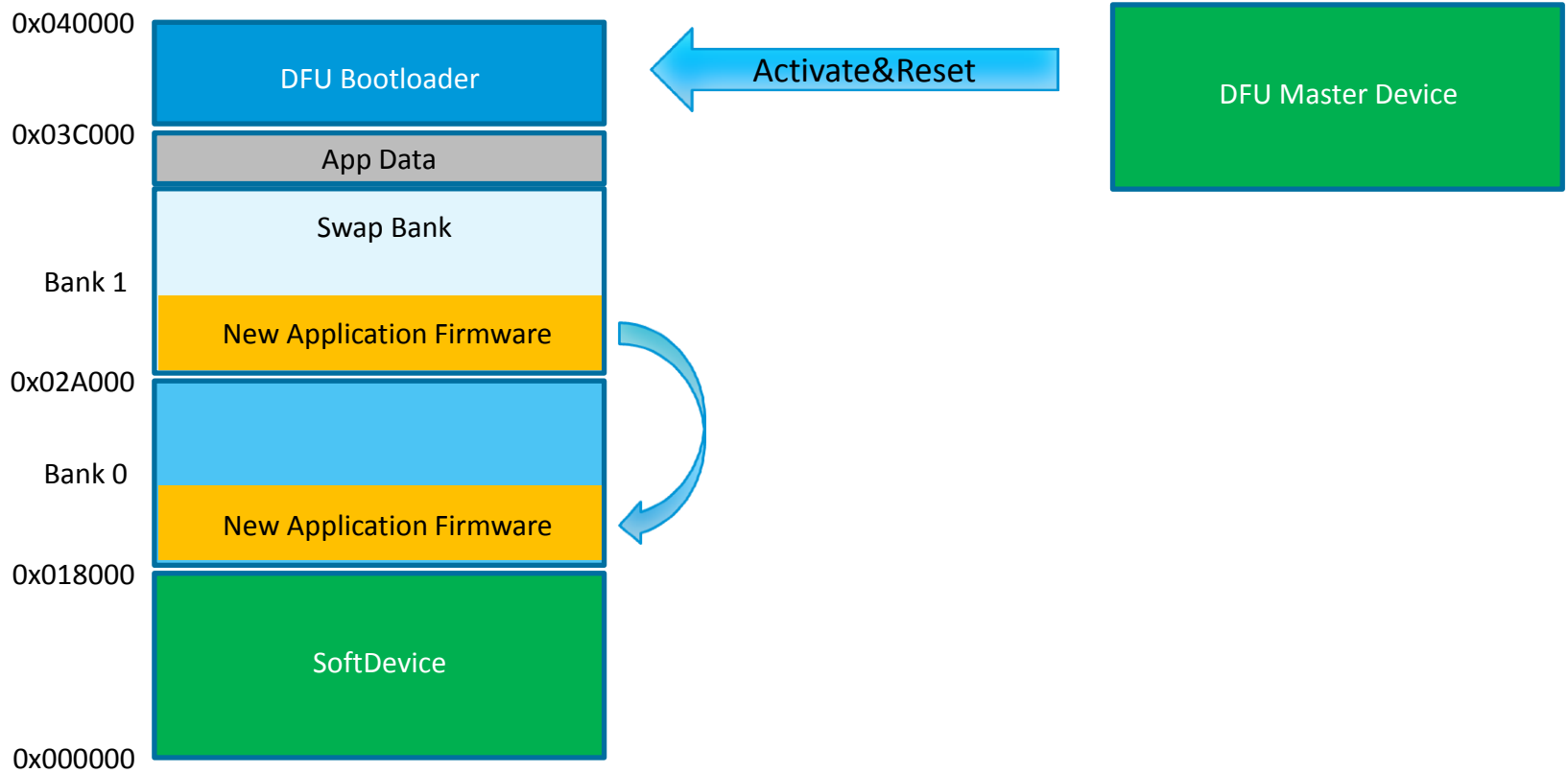
Dual Bank update

Application firmware update on a 256KB nRF51822



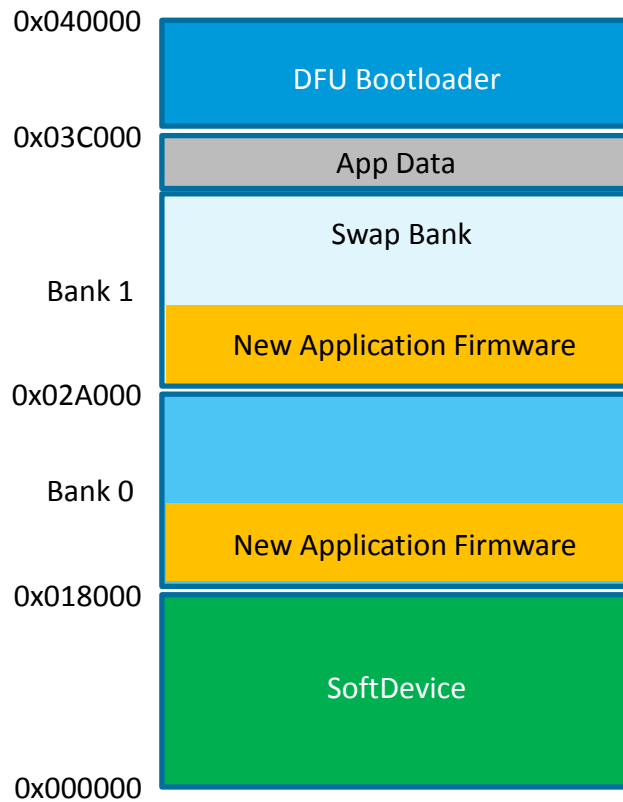
Dual Bank update

Application firmware update on a 256KB nRF51822



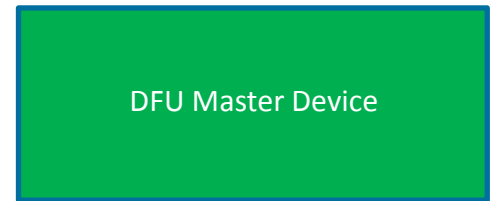
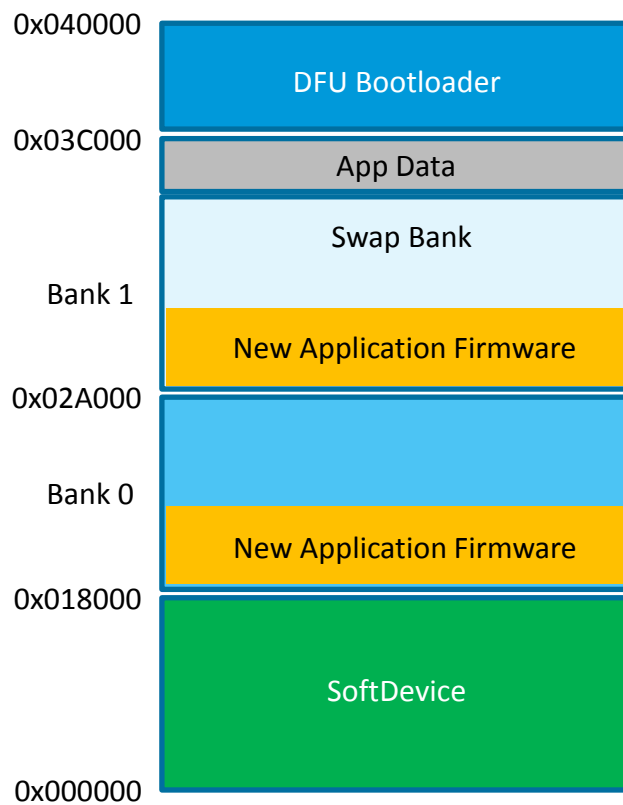
Dual Bank update

Application firmware update on a 256KB nRF51822



Dual Bank update

Application firmware update on a 256KB nRF51822

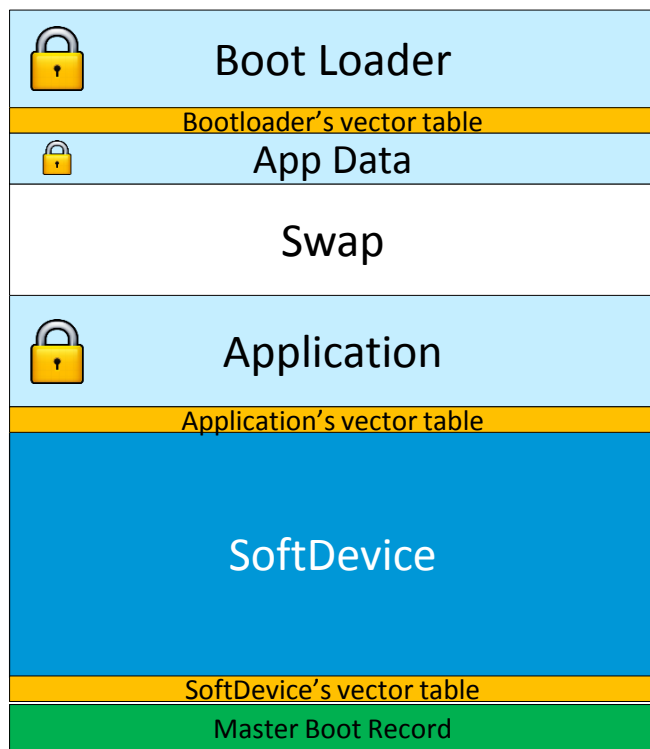


Memory Range*	Size	Usage
0x0003C000 - 0x00040000	16 kB	Bootloader
0x00018000 - 0x0003C000	144kB	Code Region 1: Application Code (BANK 0), Swap (BANK 1), and data
0x00000000 - 0x00018000	96 kB	Code Region 0: SoftDevice

*The memory layout is the layout from the DFU SDK example

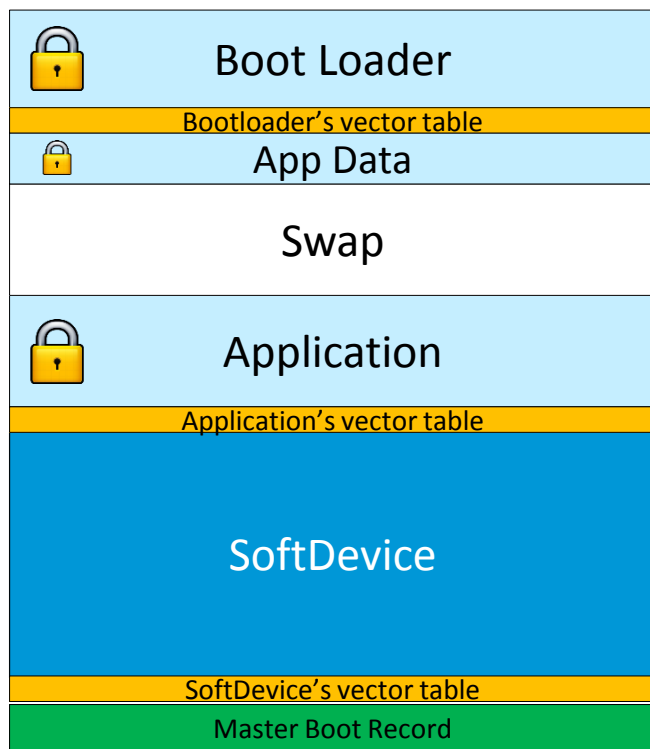
Dual Bank Update in nRF51 SDK

Performance



Dual Bank Update in nRF51 SDK

Performance

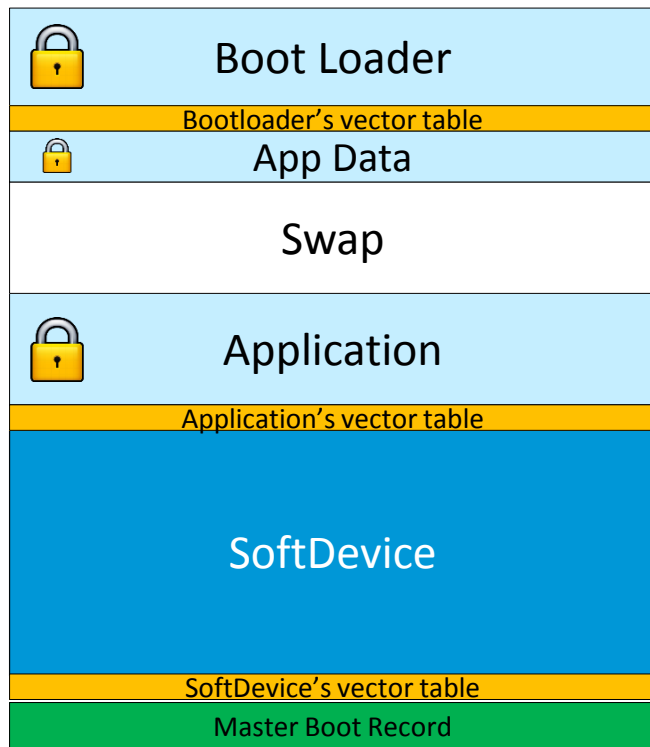


- BLE DFU service details:
 - Connection Interval (SDK 6.1.0): 15 ms
 - Maximum 6 packets pr connection interval
 - 20 bytes / packet
- Update times*:
 - nRF Master Emulator (Peer as fast as SDK):
 - 20 kB application (eg. Heart Rate): **5 sec**
 - 76 kB application: 14 sec
 - Android 4.4: **20 sec** (20 kB)*
 - iOS7: **07 sec** (20 kB)

*Varied between Android phones

Dual Bank Update in nRF51 SDK

SAFETY



- Link Loss Recovery Mechanism
 - Upon re-connection Peer issue a «Request Received Image Size»
 - Continue data transfer if remote and local image size doesn't match

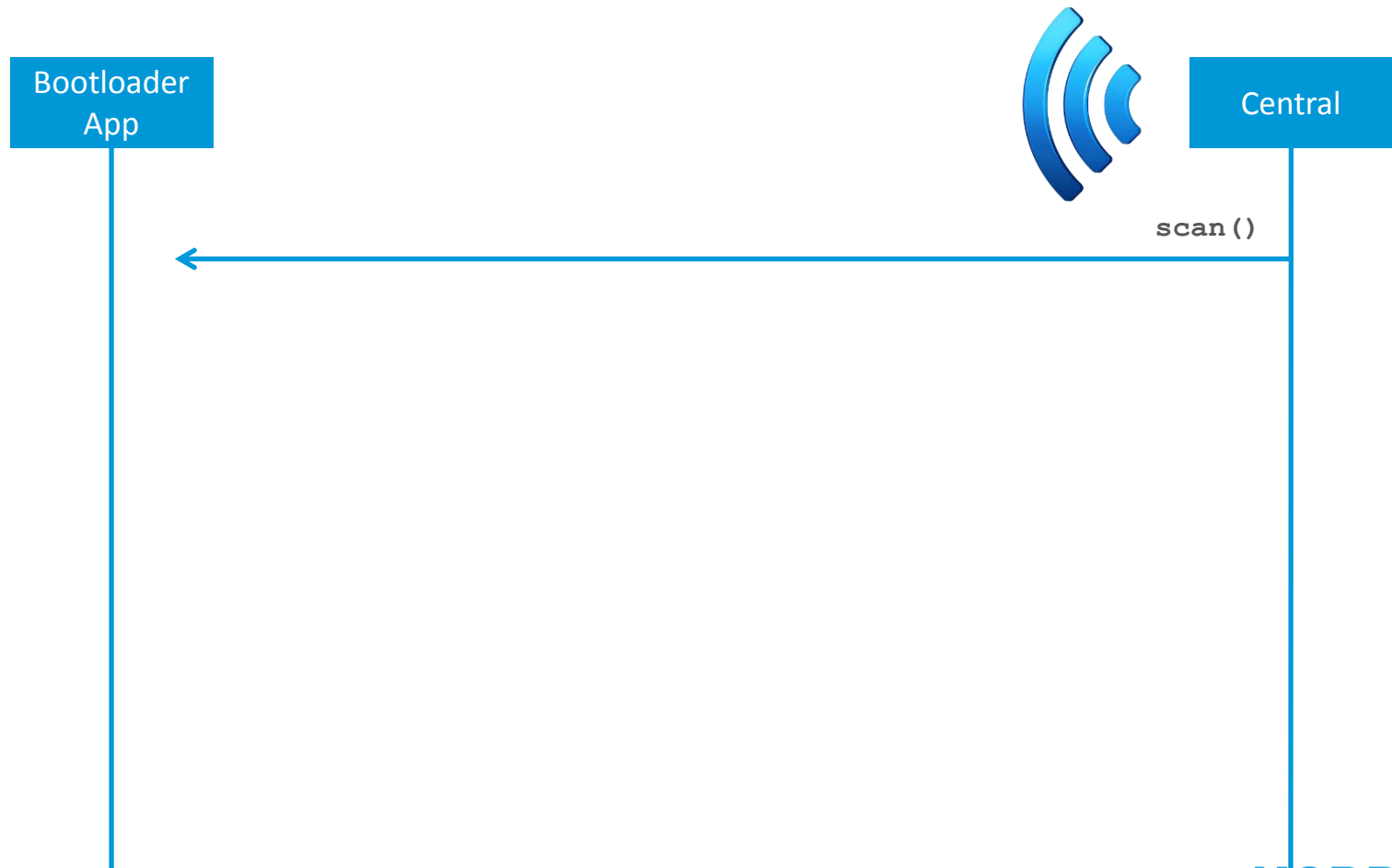
- Recovery from incomplete/failed Firmware Update
 - Old application preserved until successful Firmware Update completed
 - Old application always is fallback image
 - Old application will be re-started after 60s time out

Over the air firmware update sequence

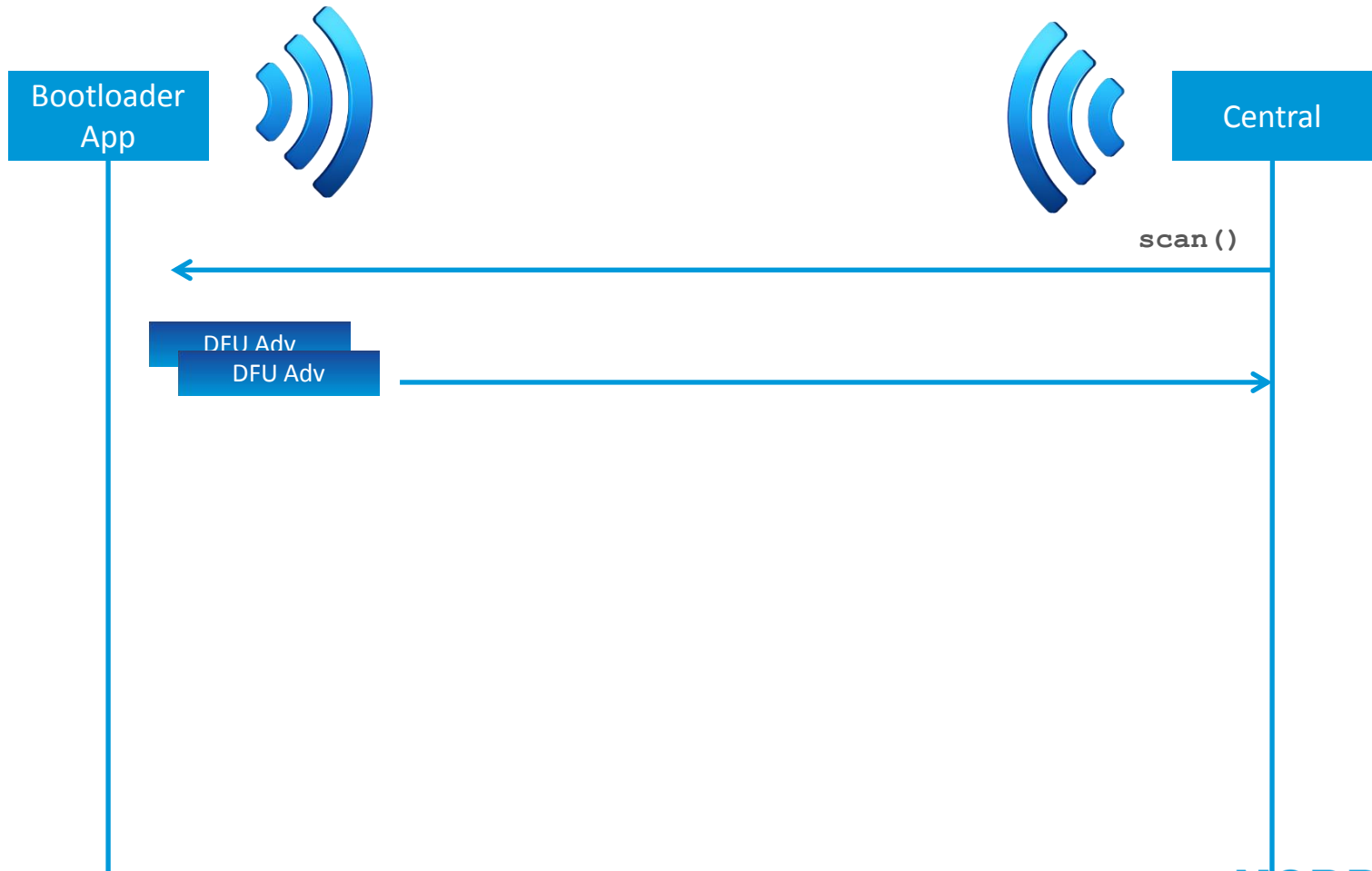
Bootloader
App

Central

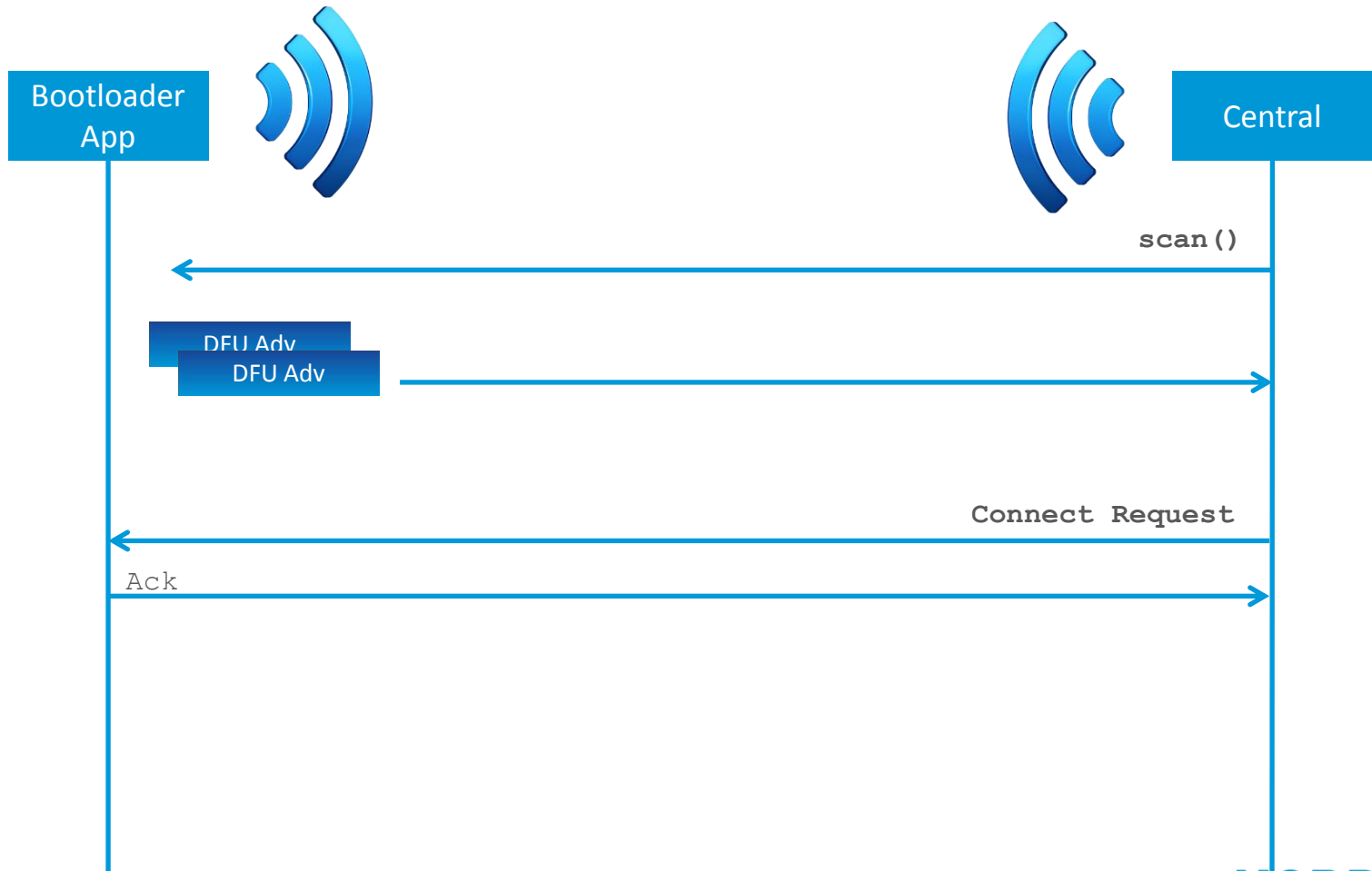
Over the air firmware update sequence



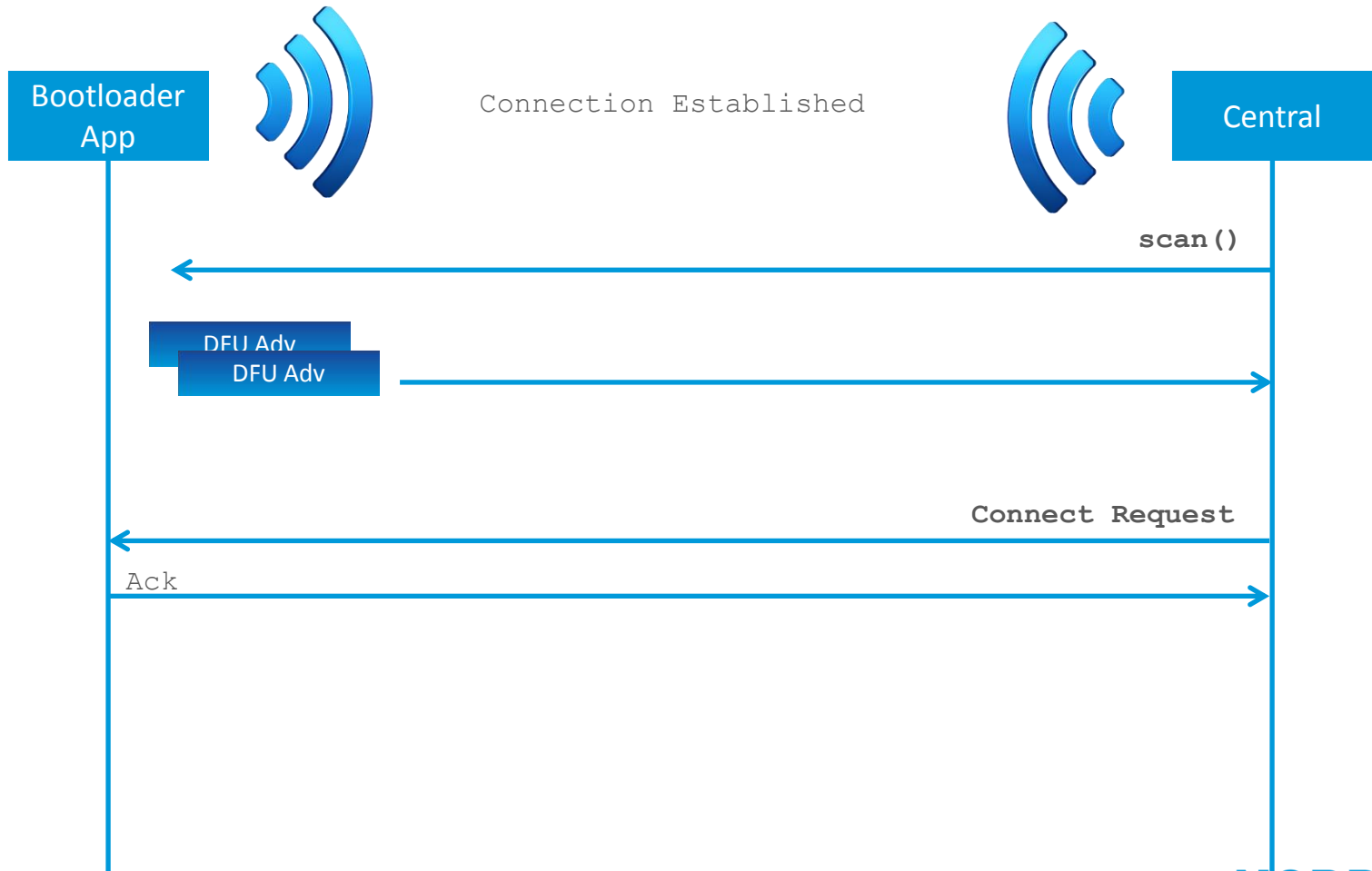
Over the air firmware update sequence



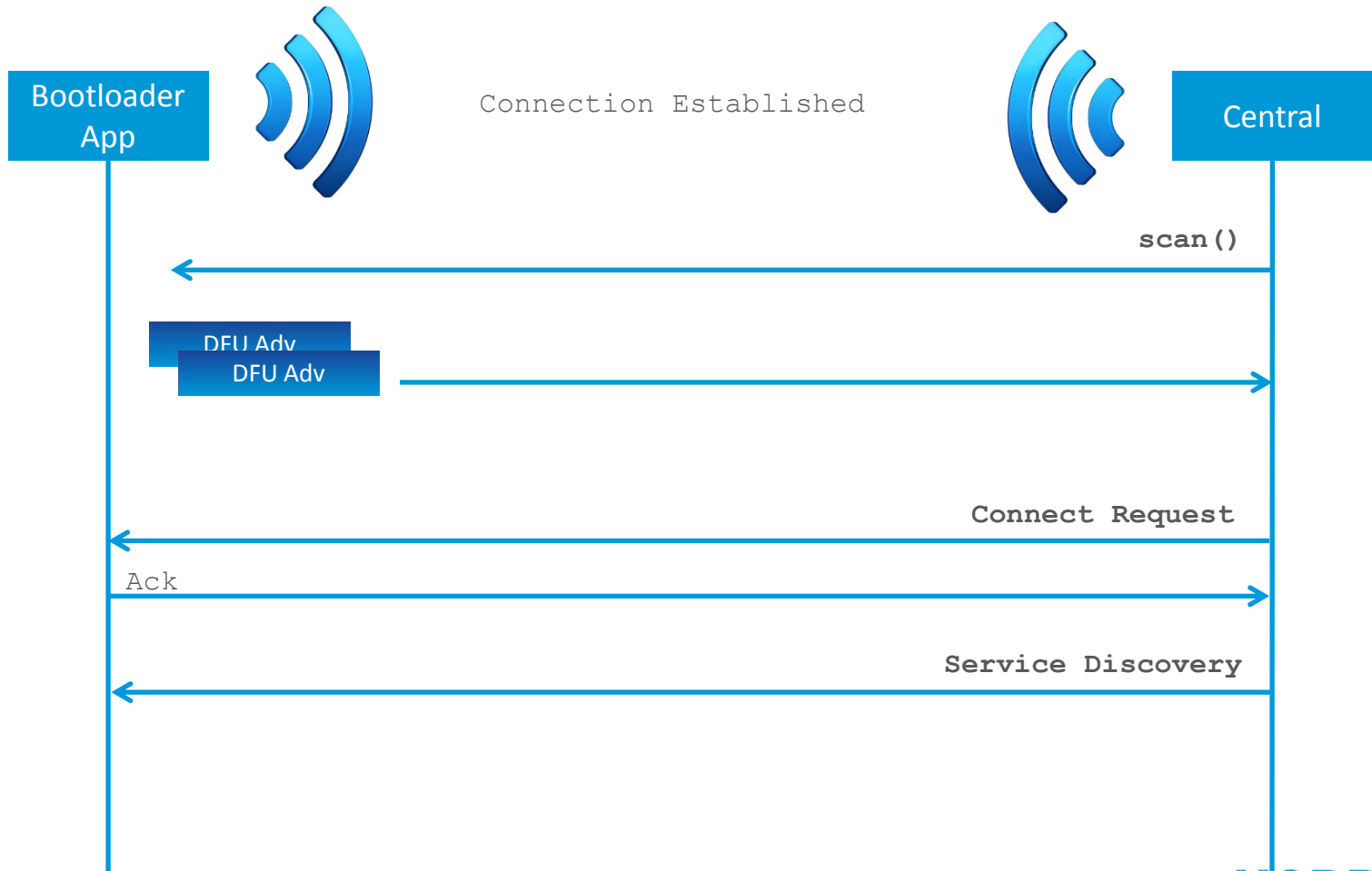
Over the air firmware update sequence



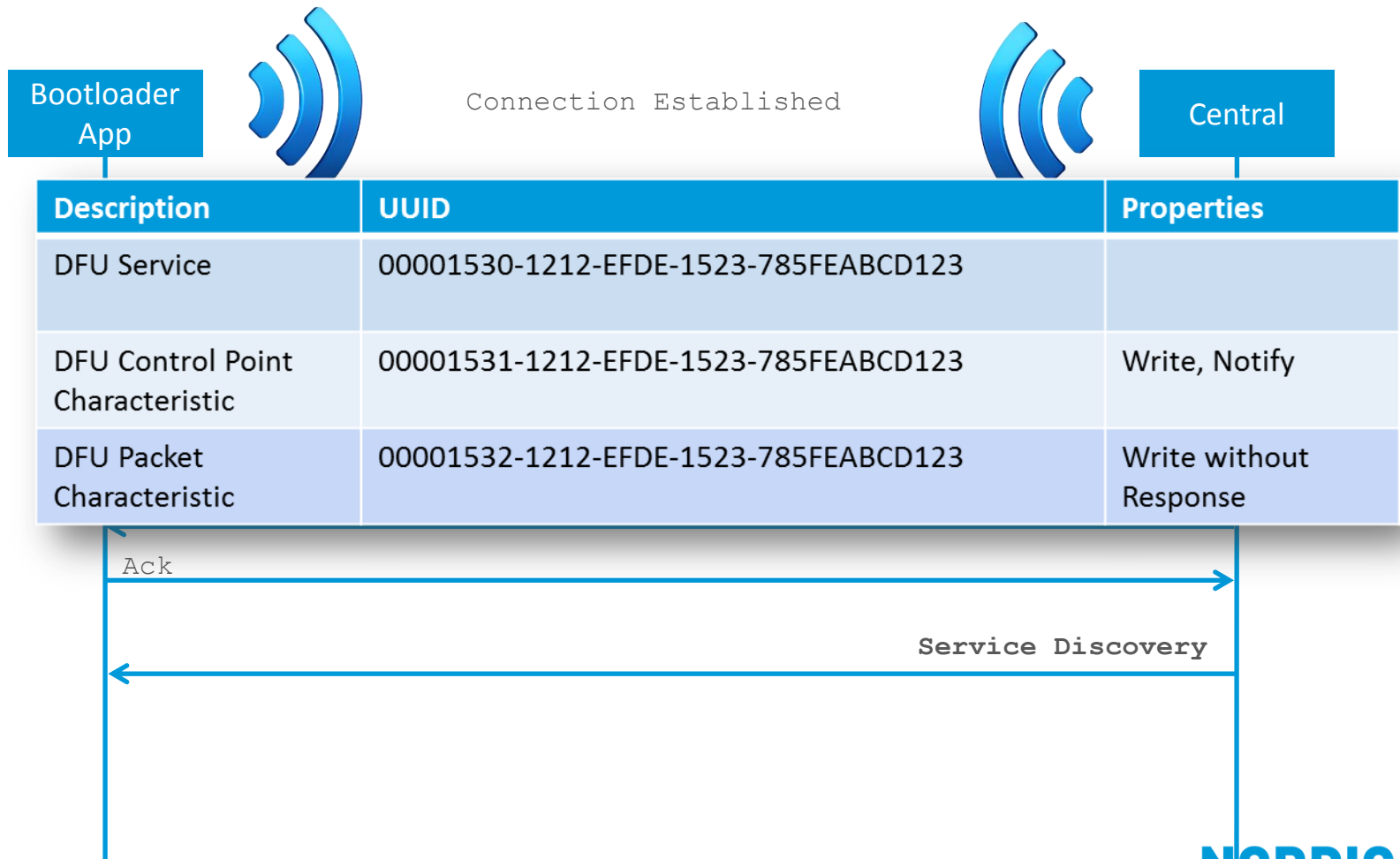
Over the air firmware update sequence



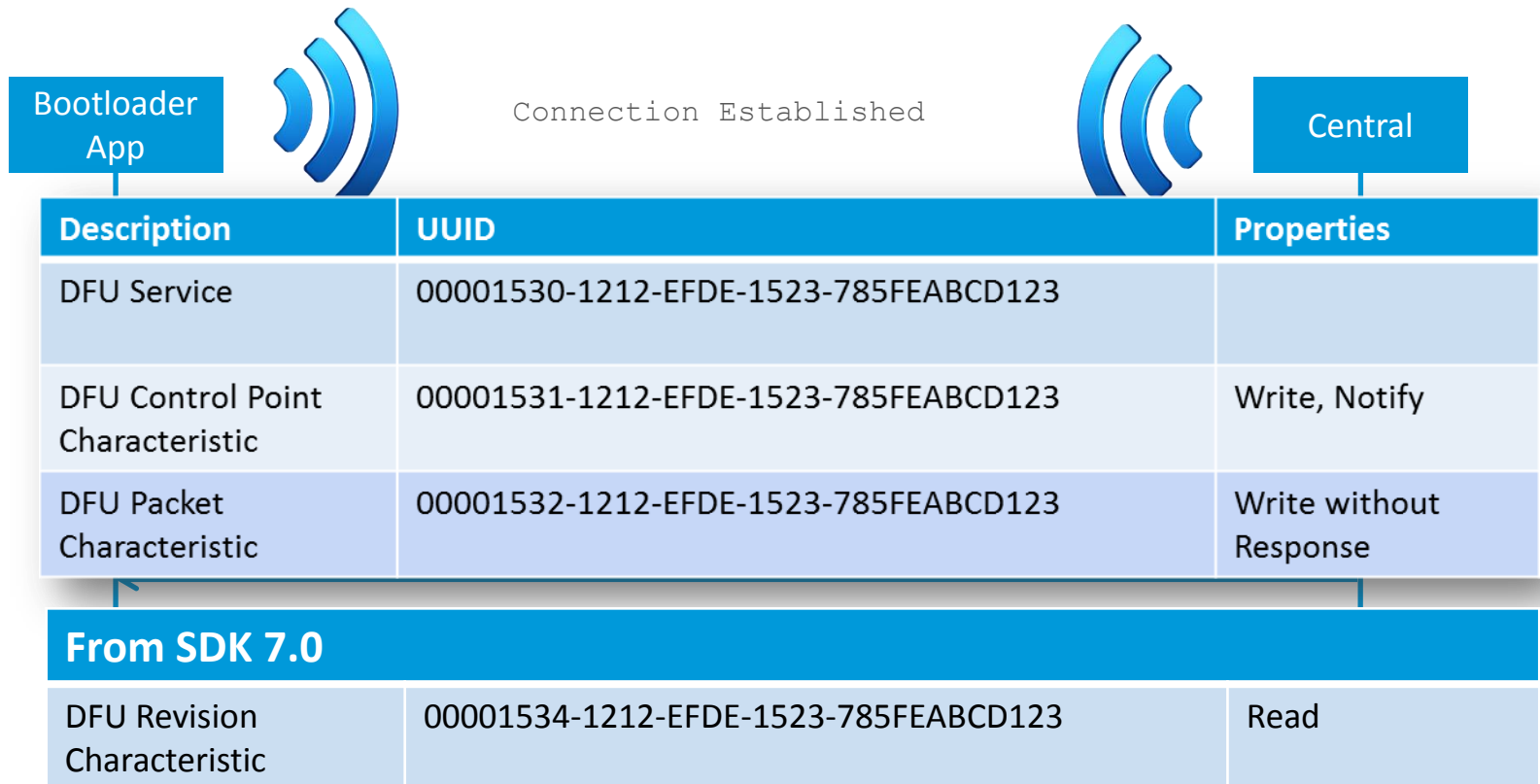
Over the air firmware update sequence



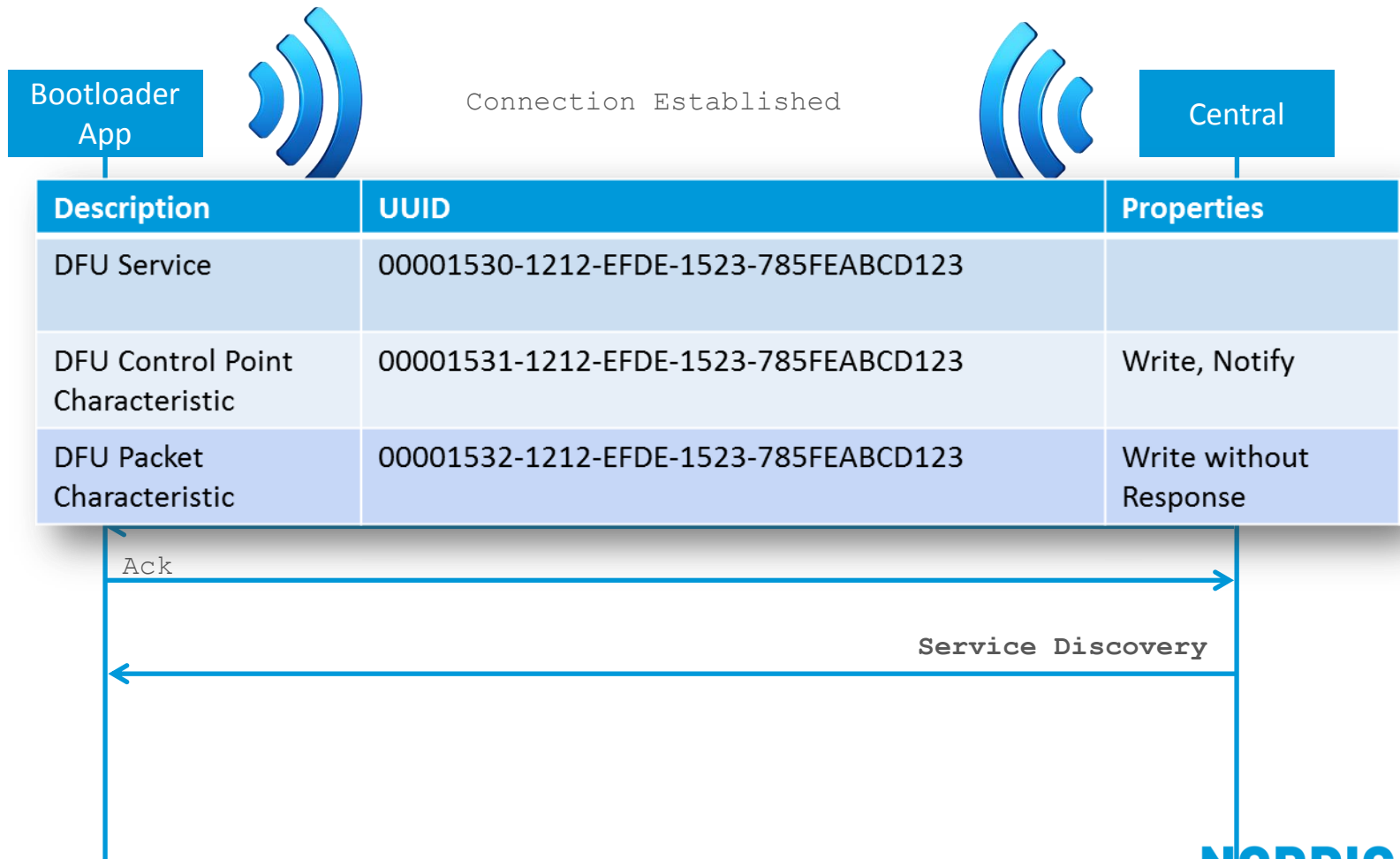
Over the air firmware update sequence



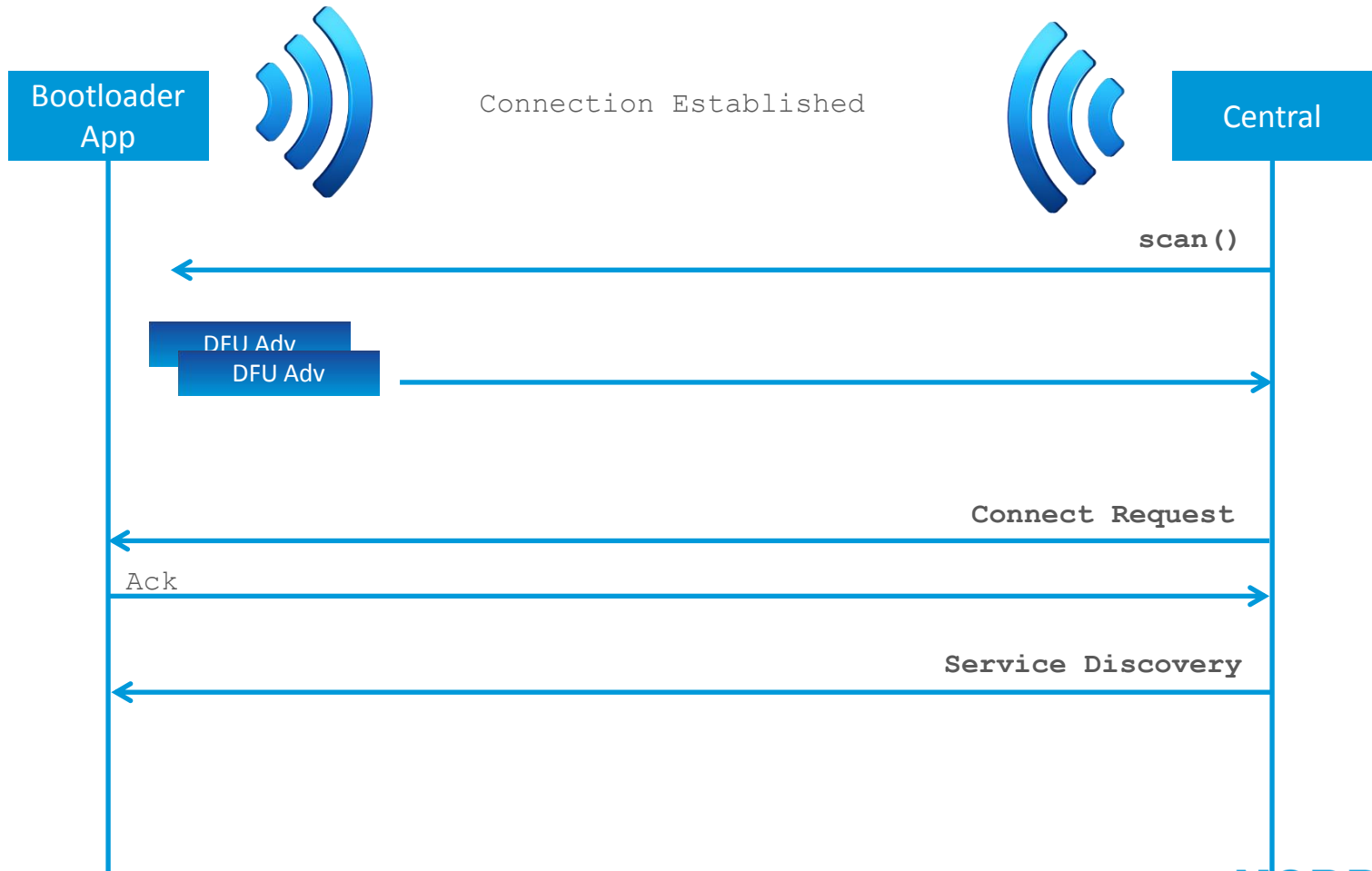
Over the air firmware update sequence



Over the air firmware update sequence



Over the air firmware update sequence



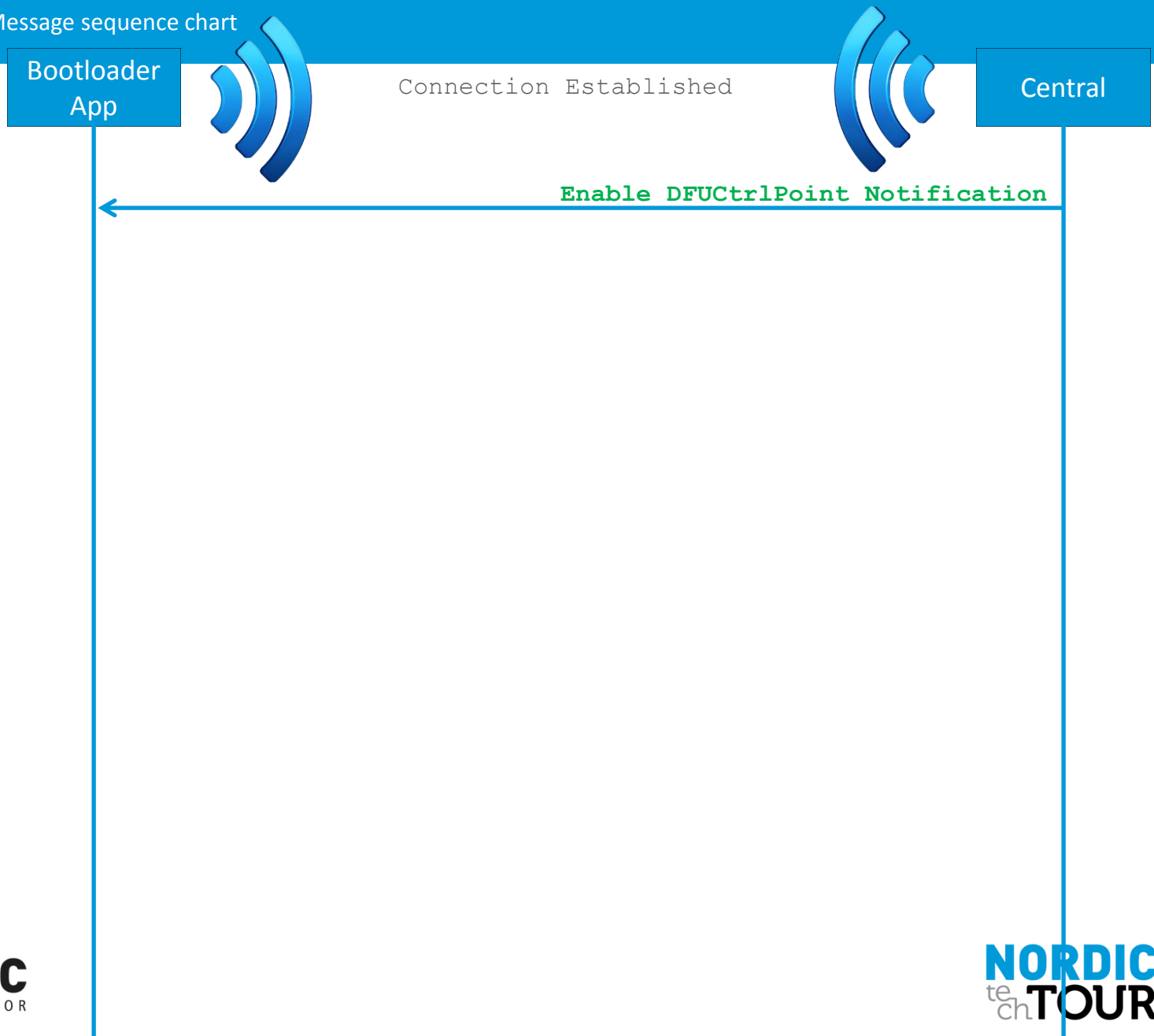
Bootloader
App

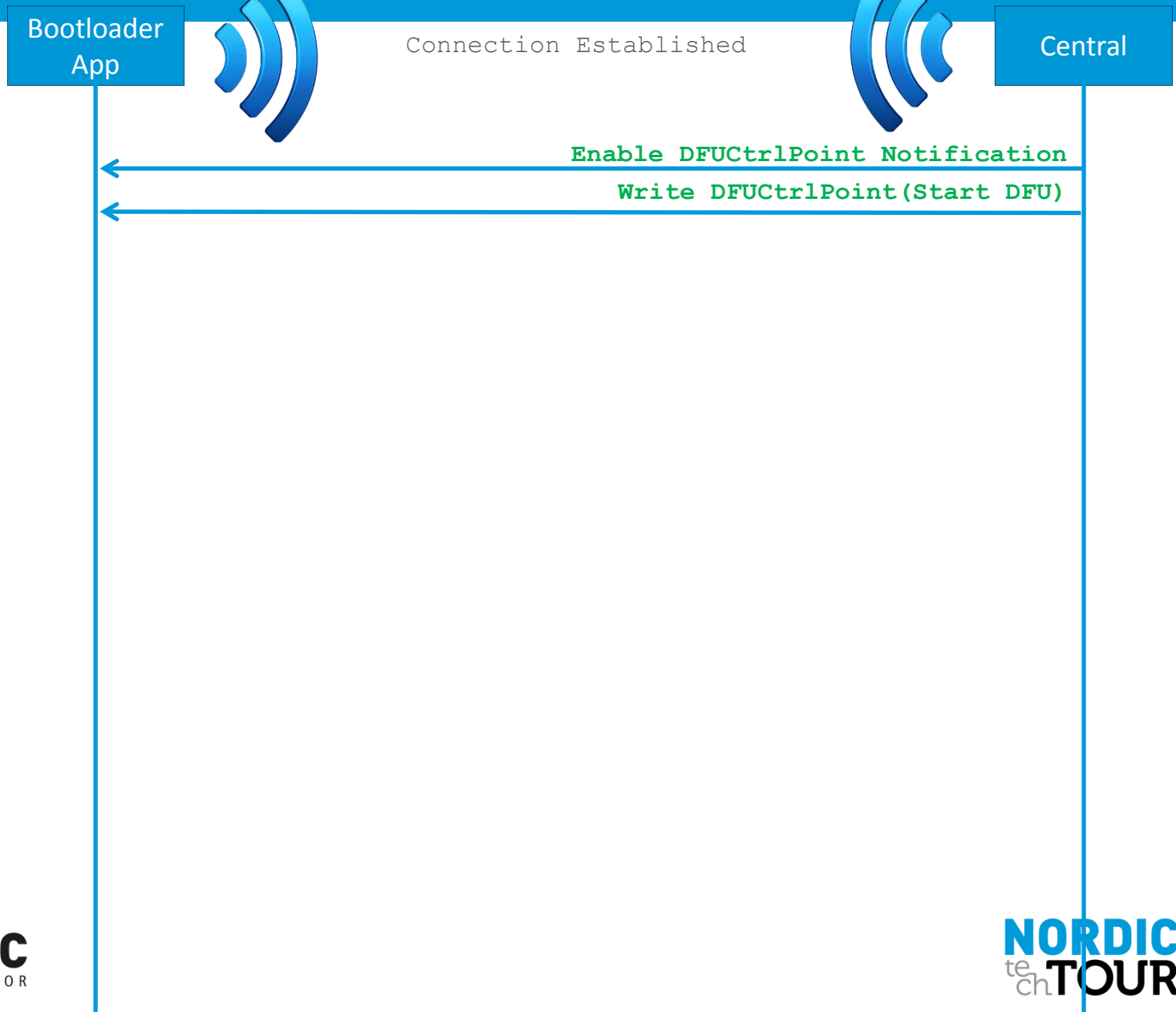


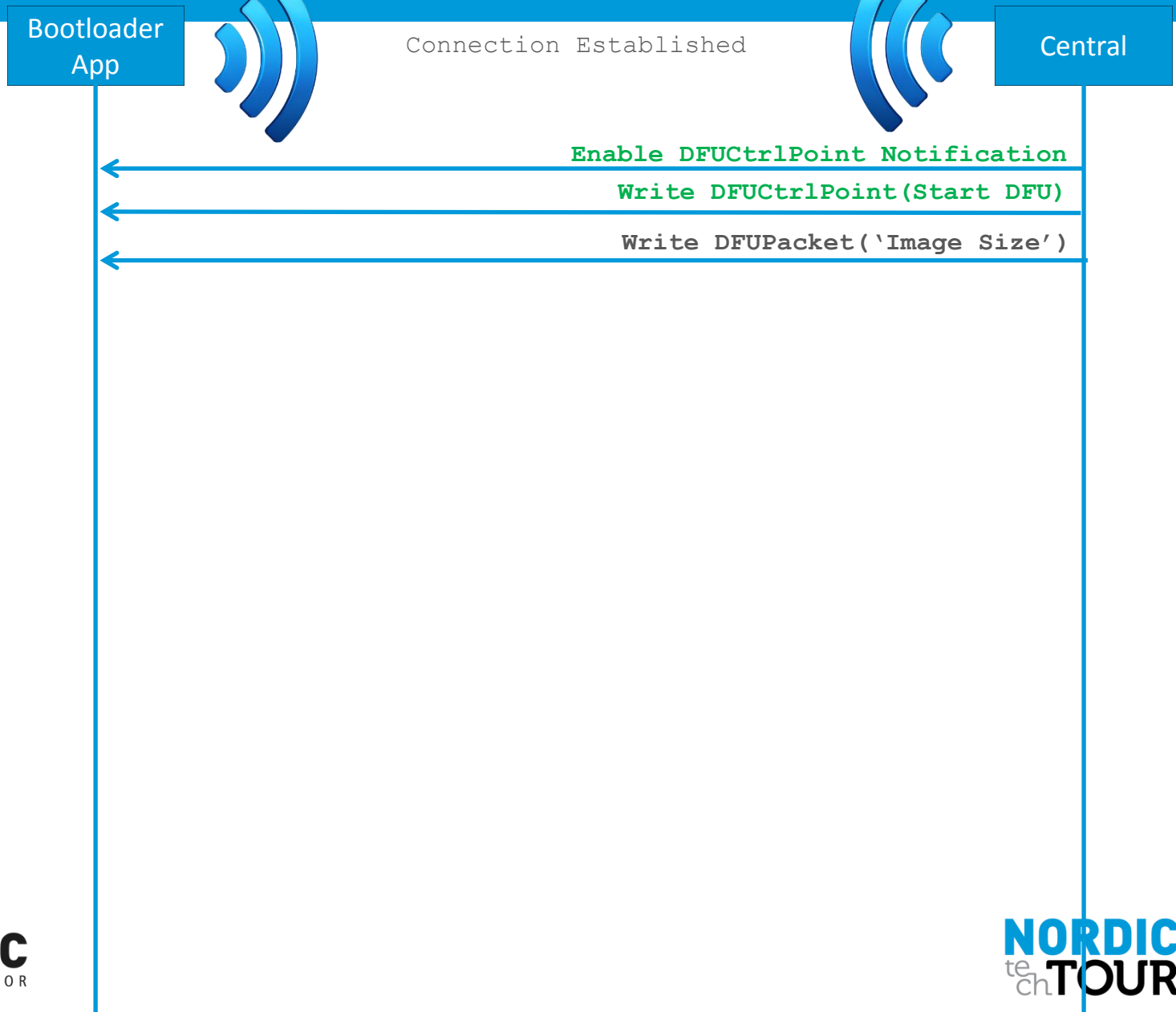
Connection Established

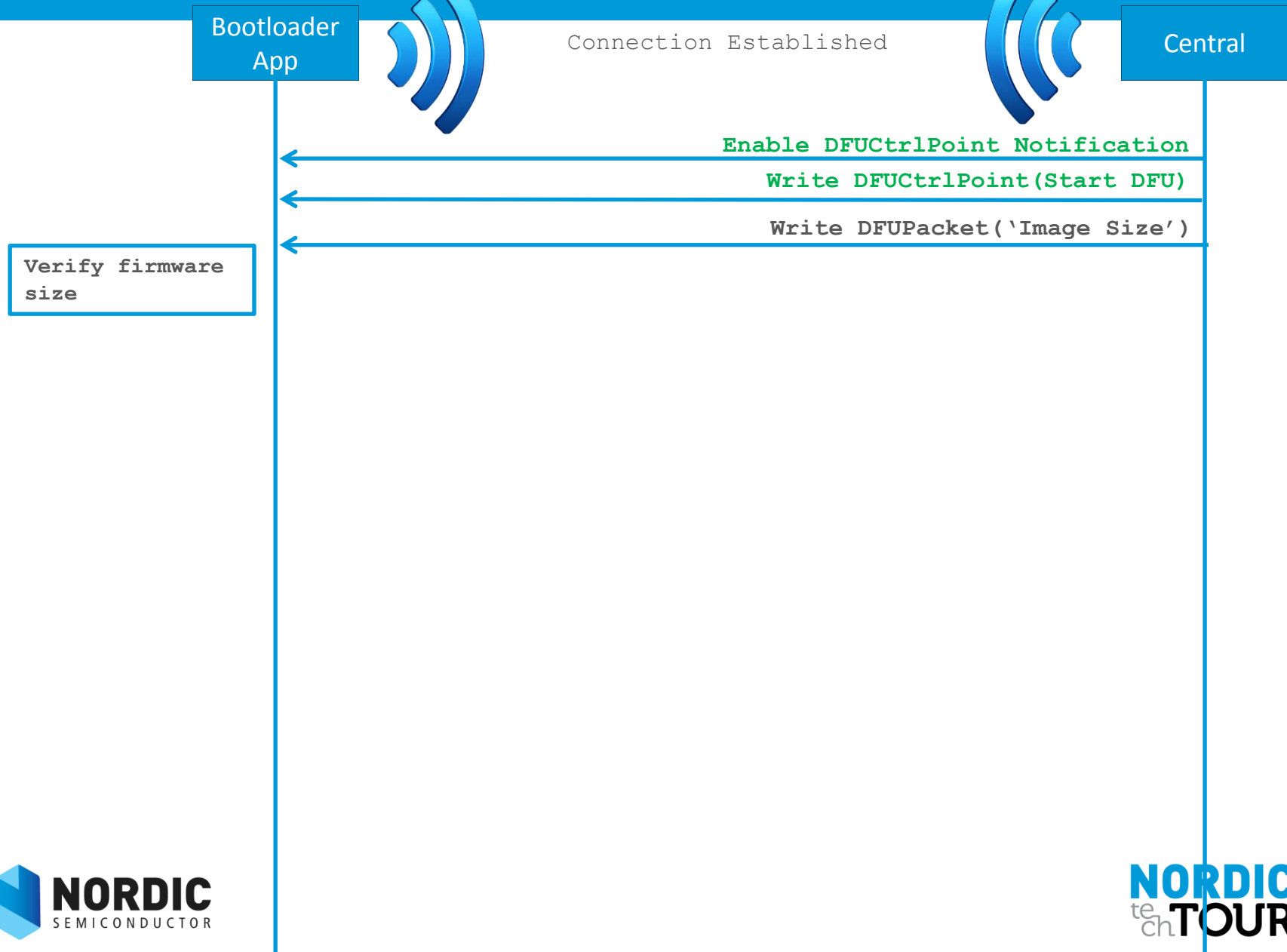


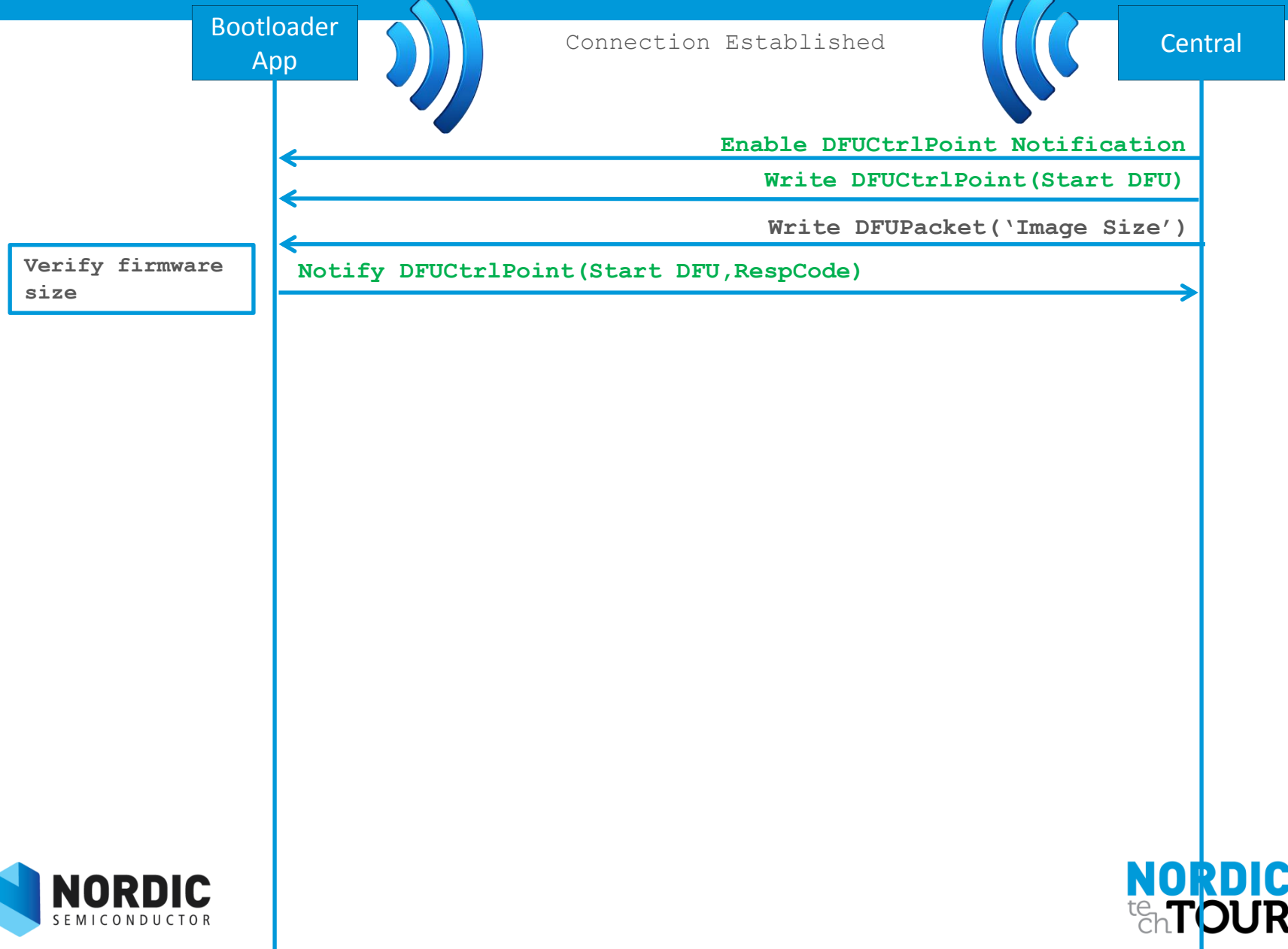
Central

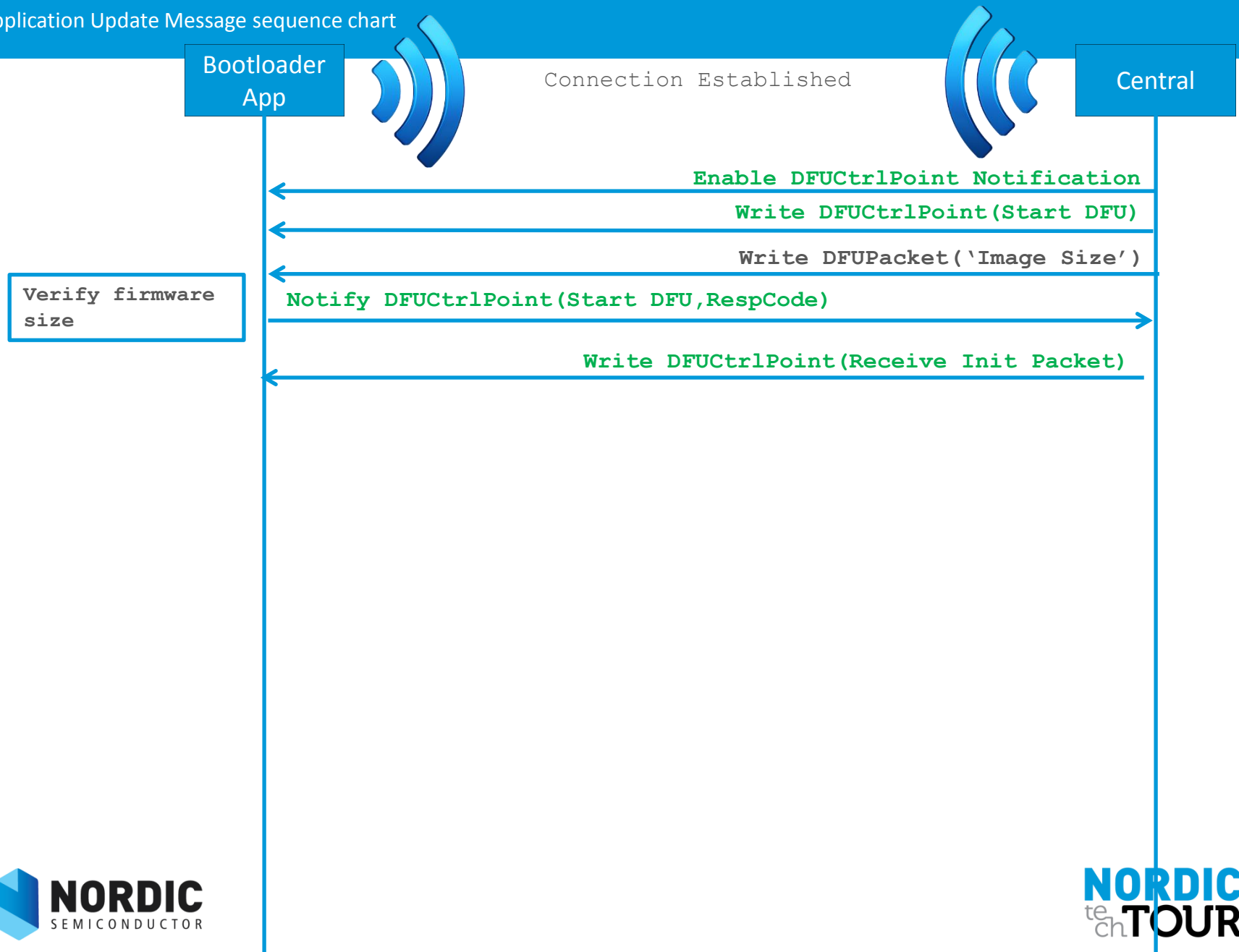


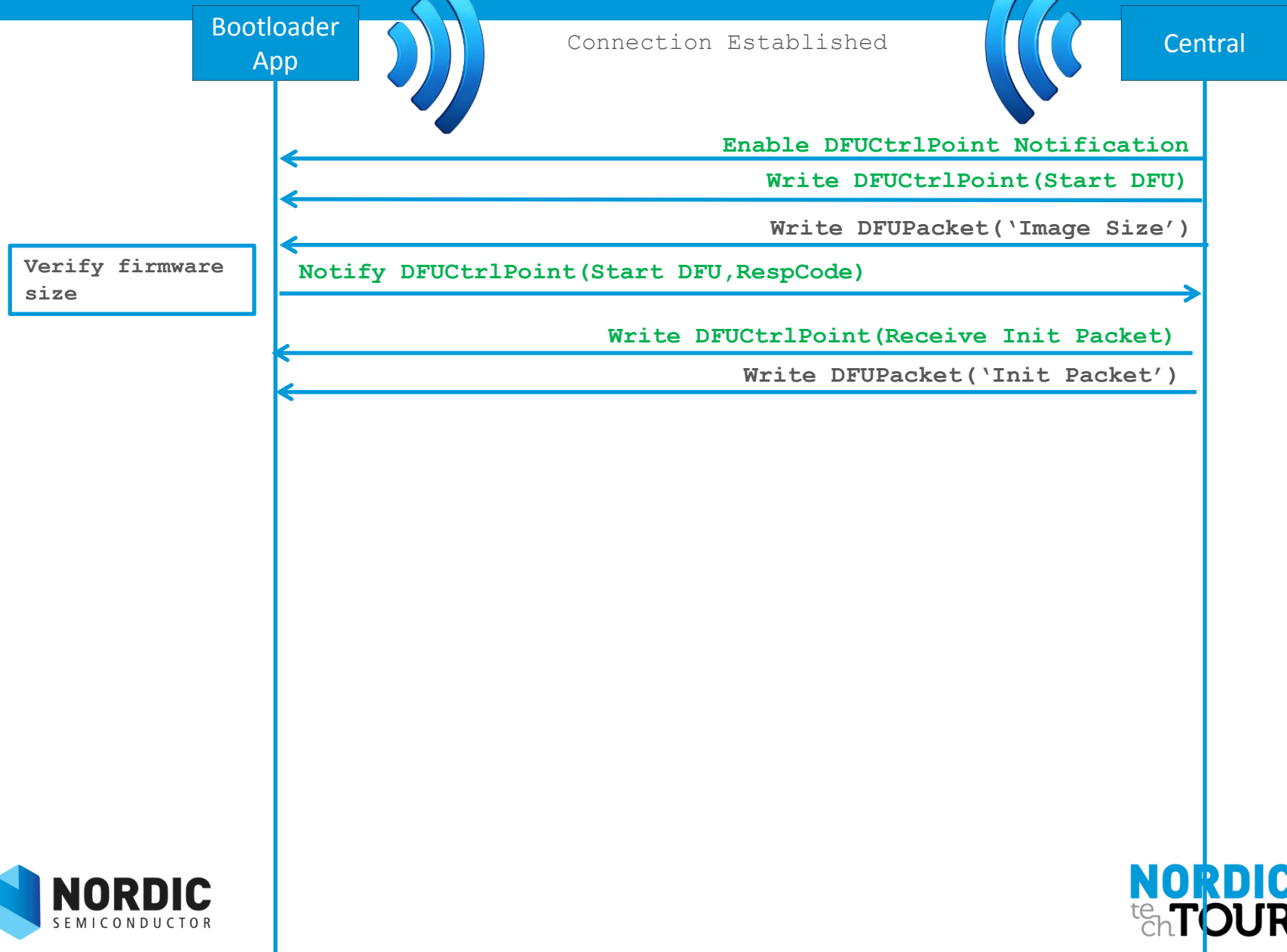


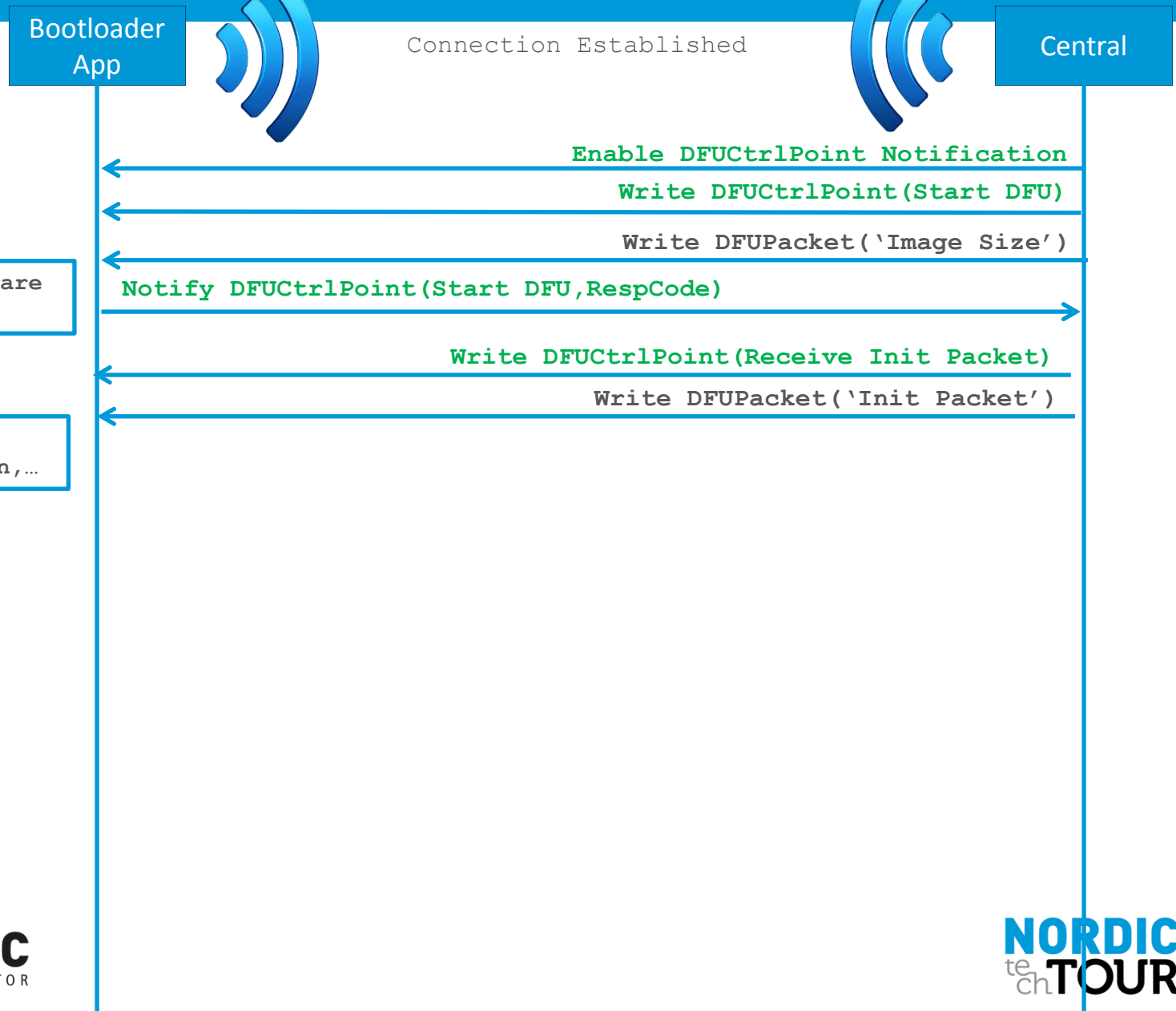


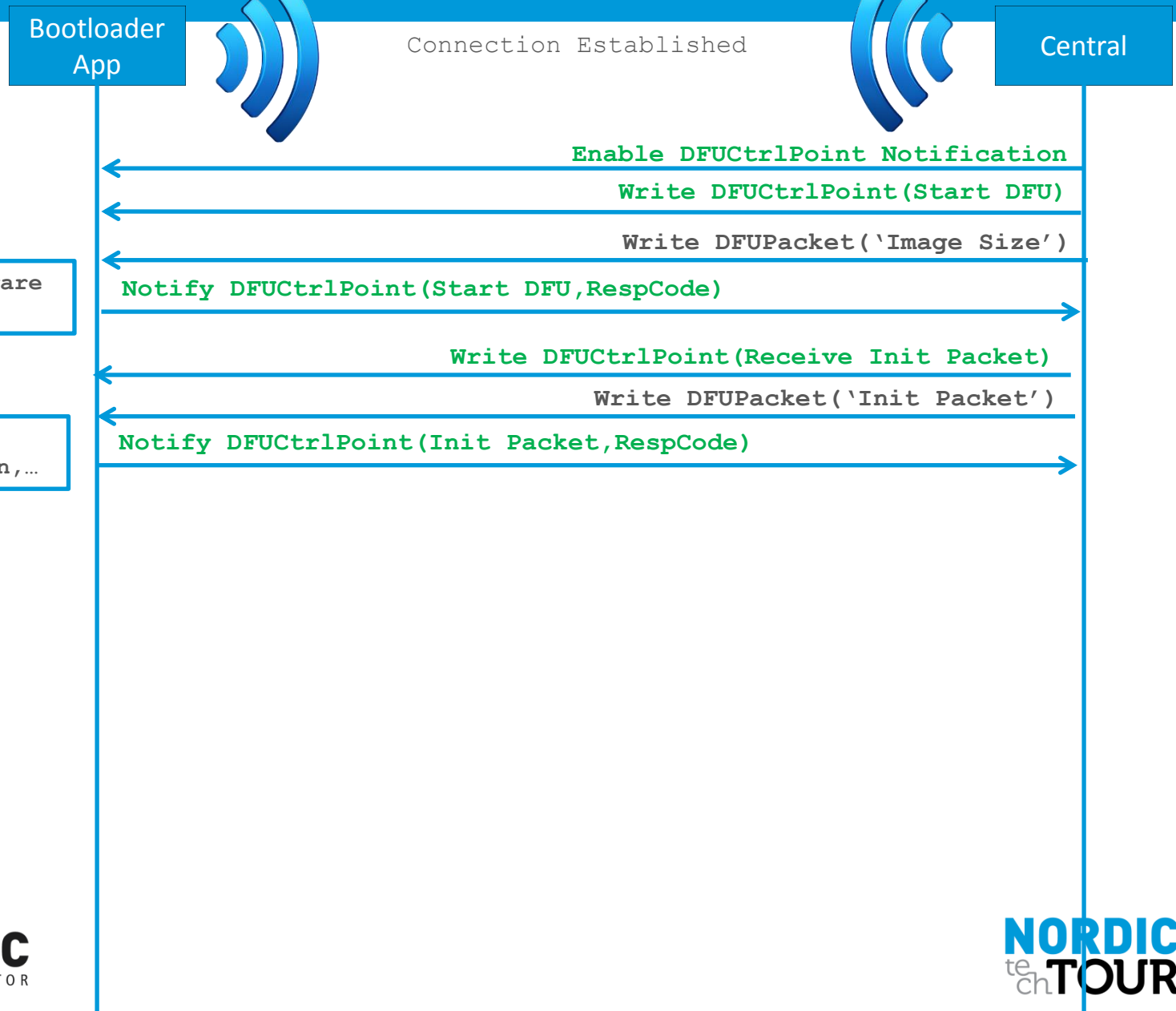


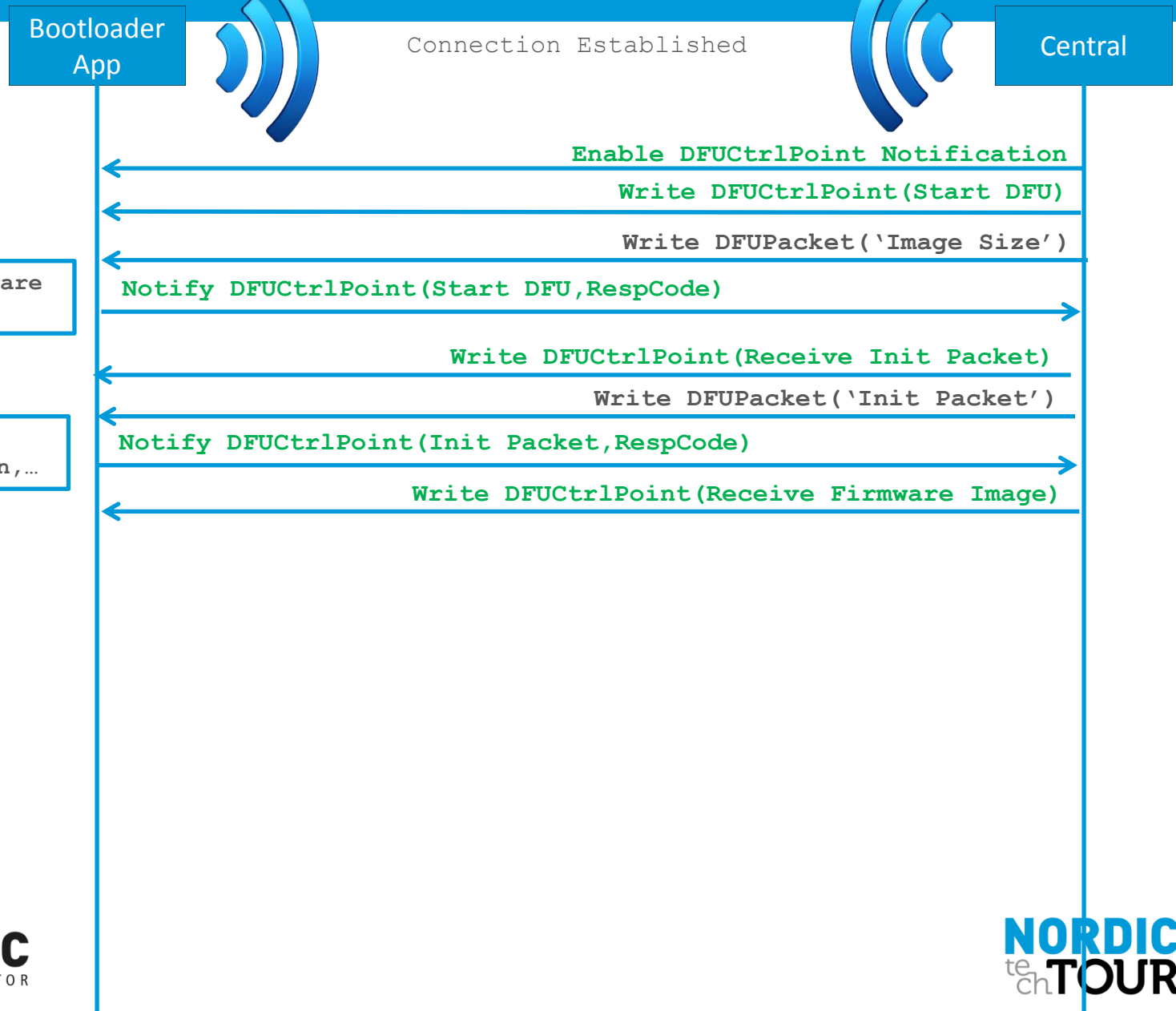


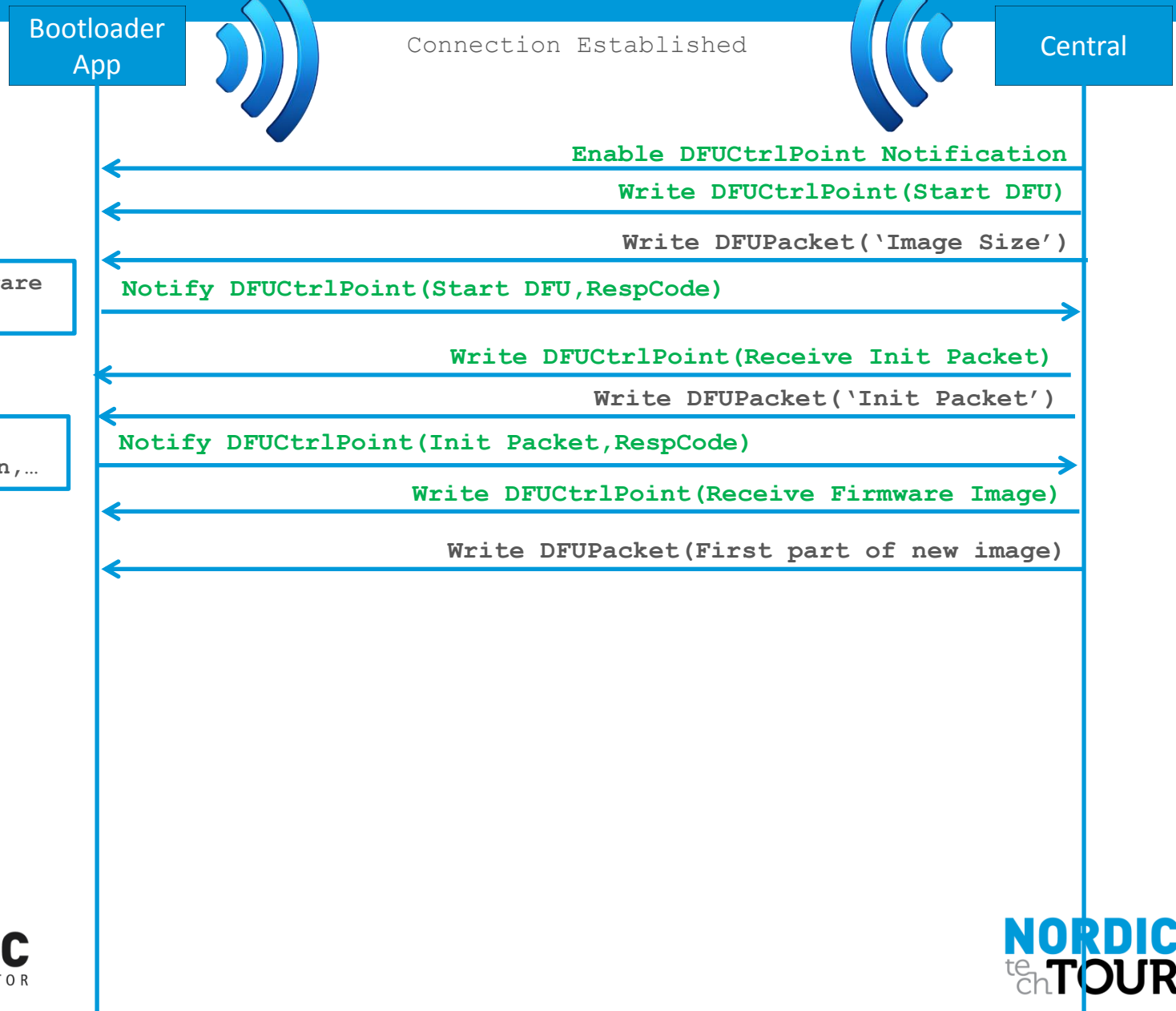


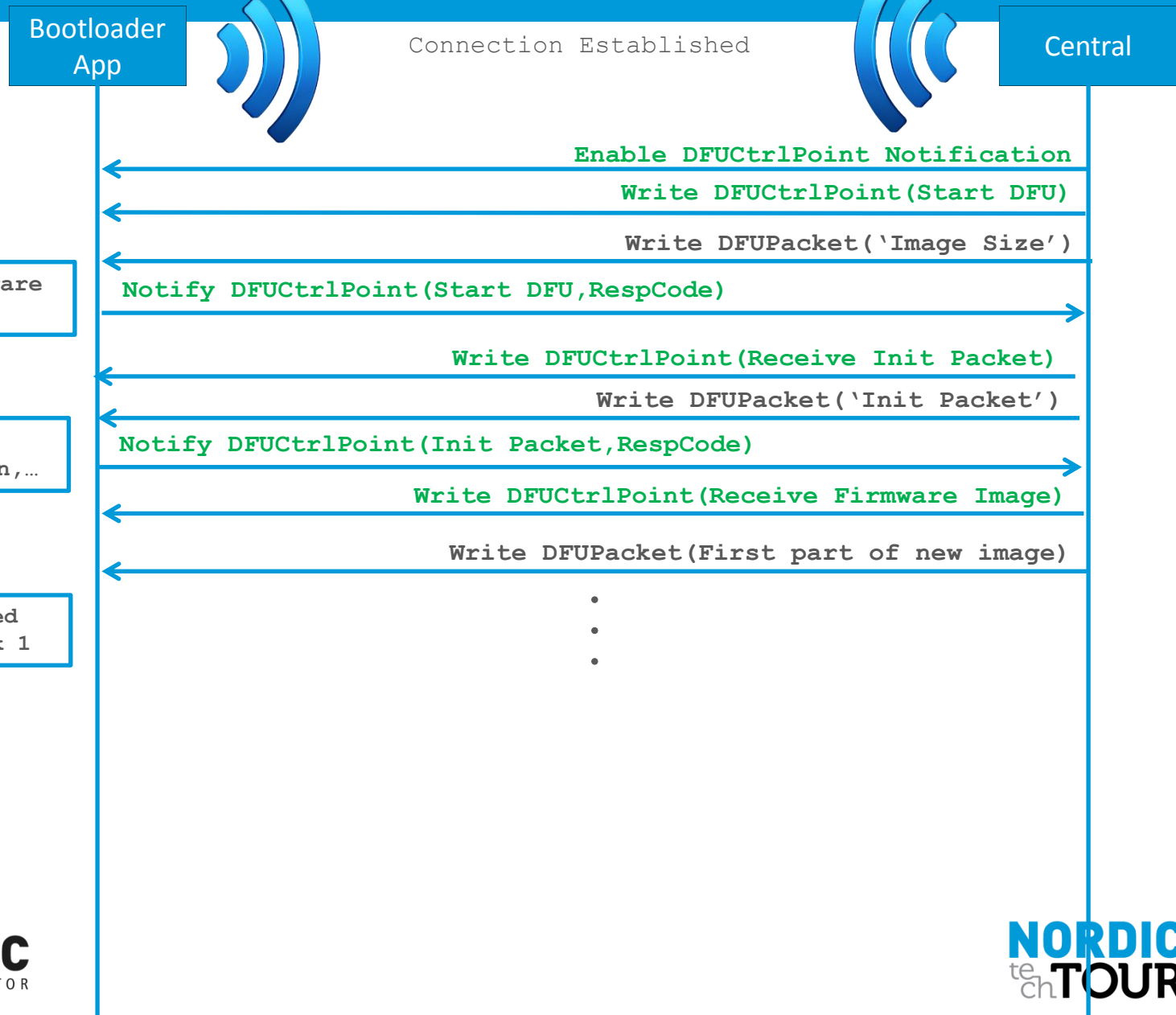


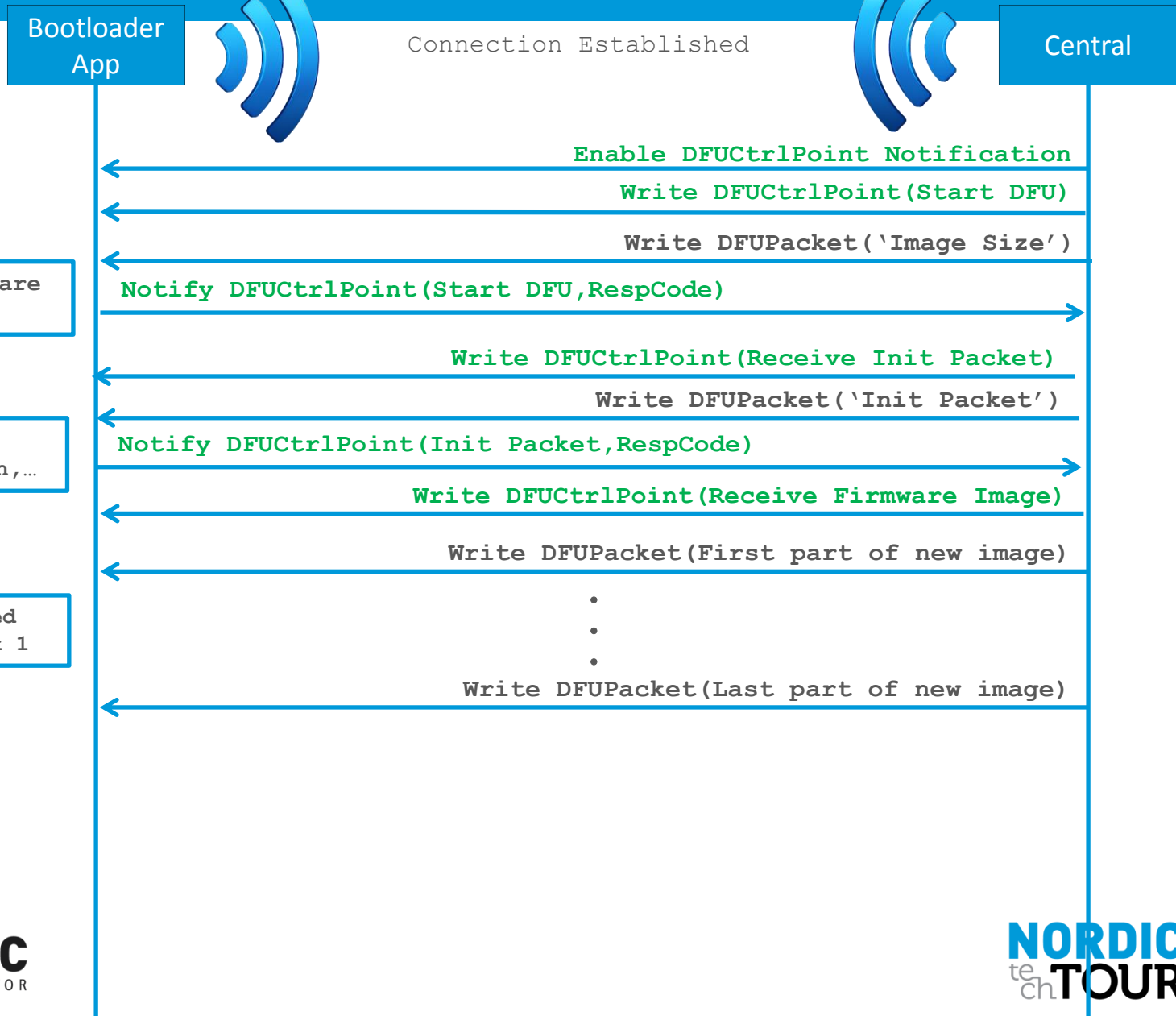


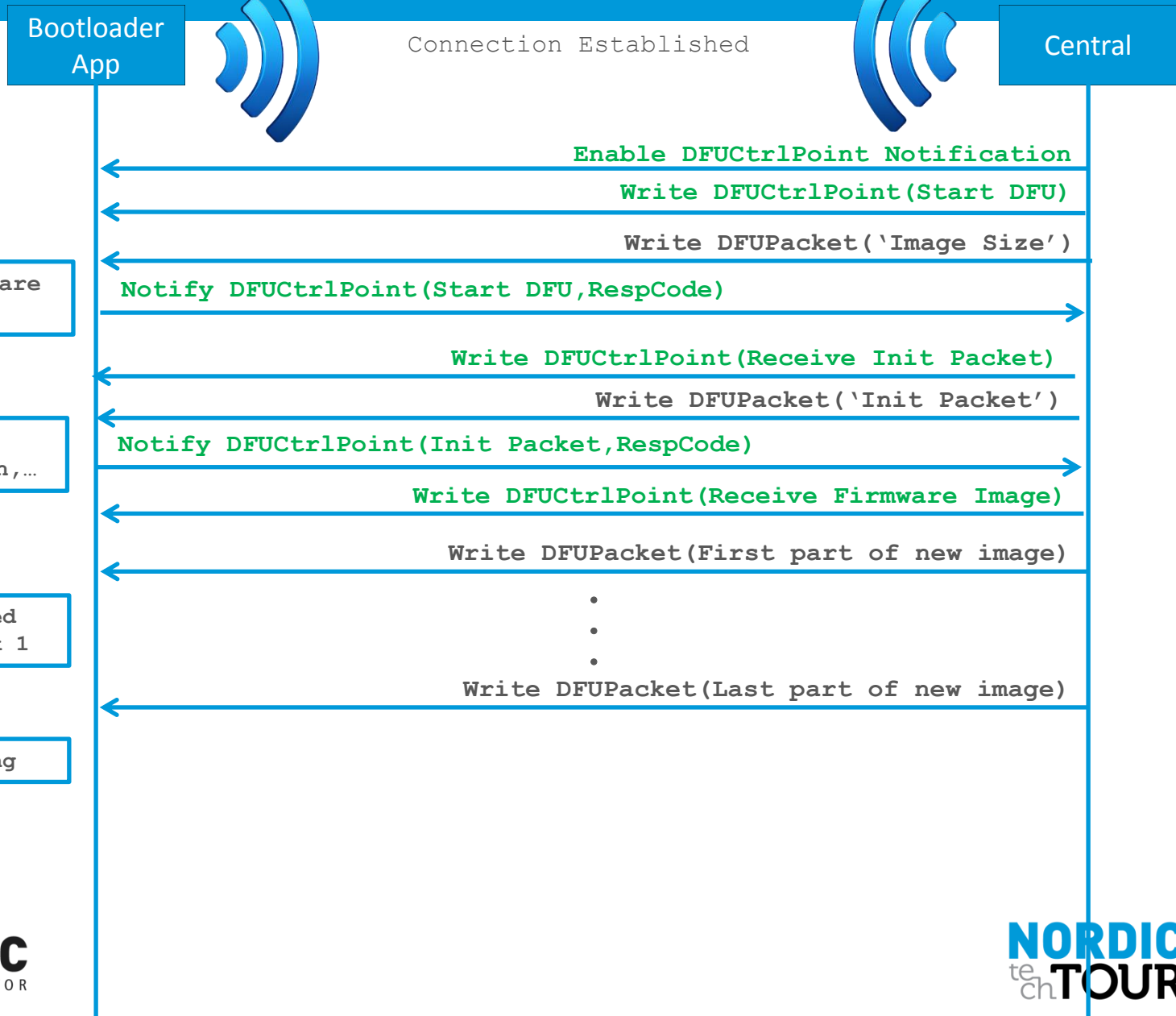


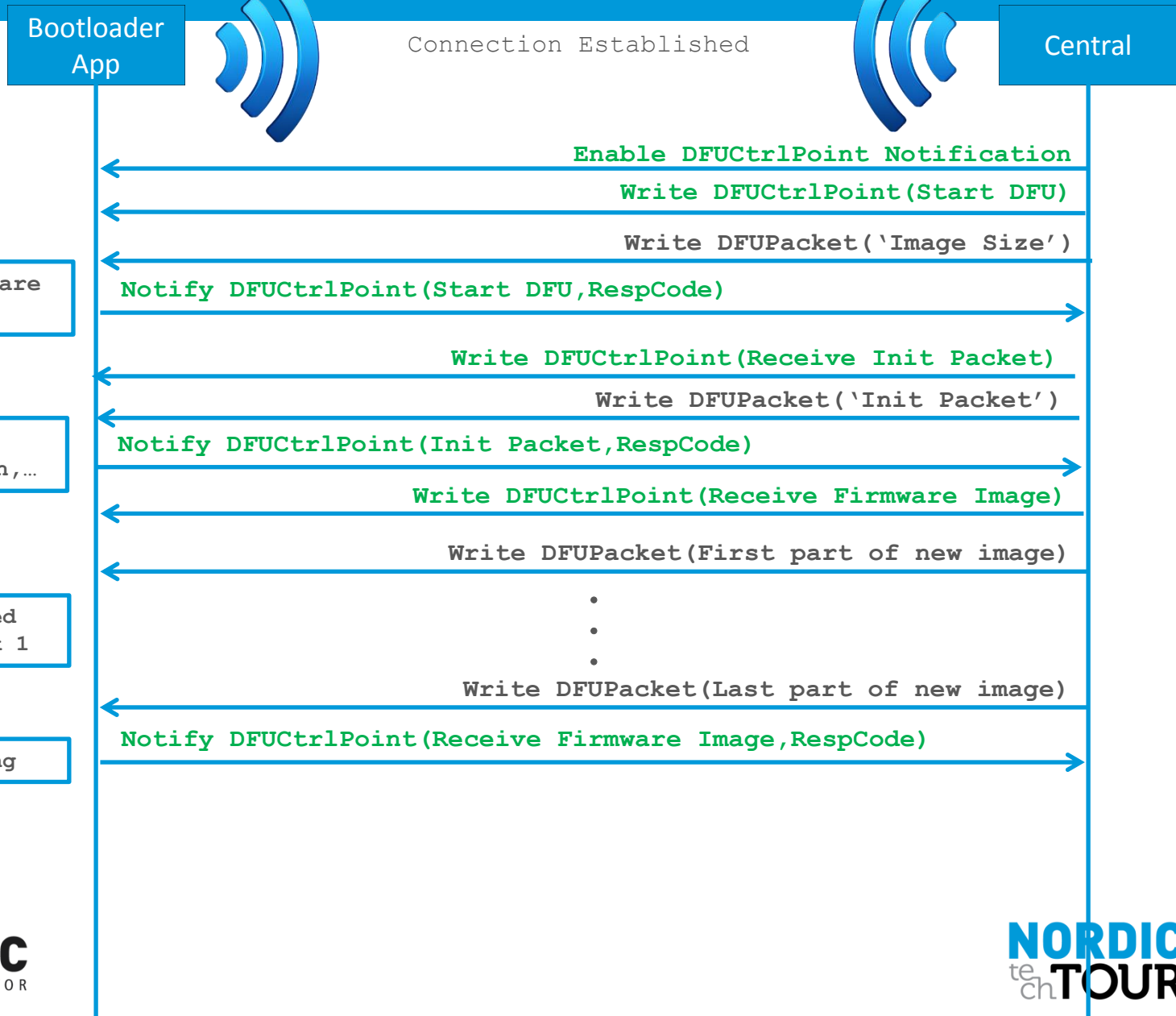


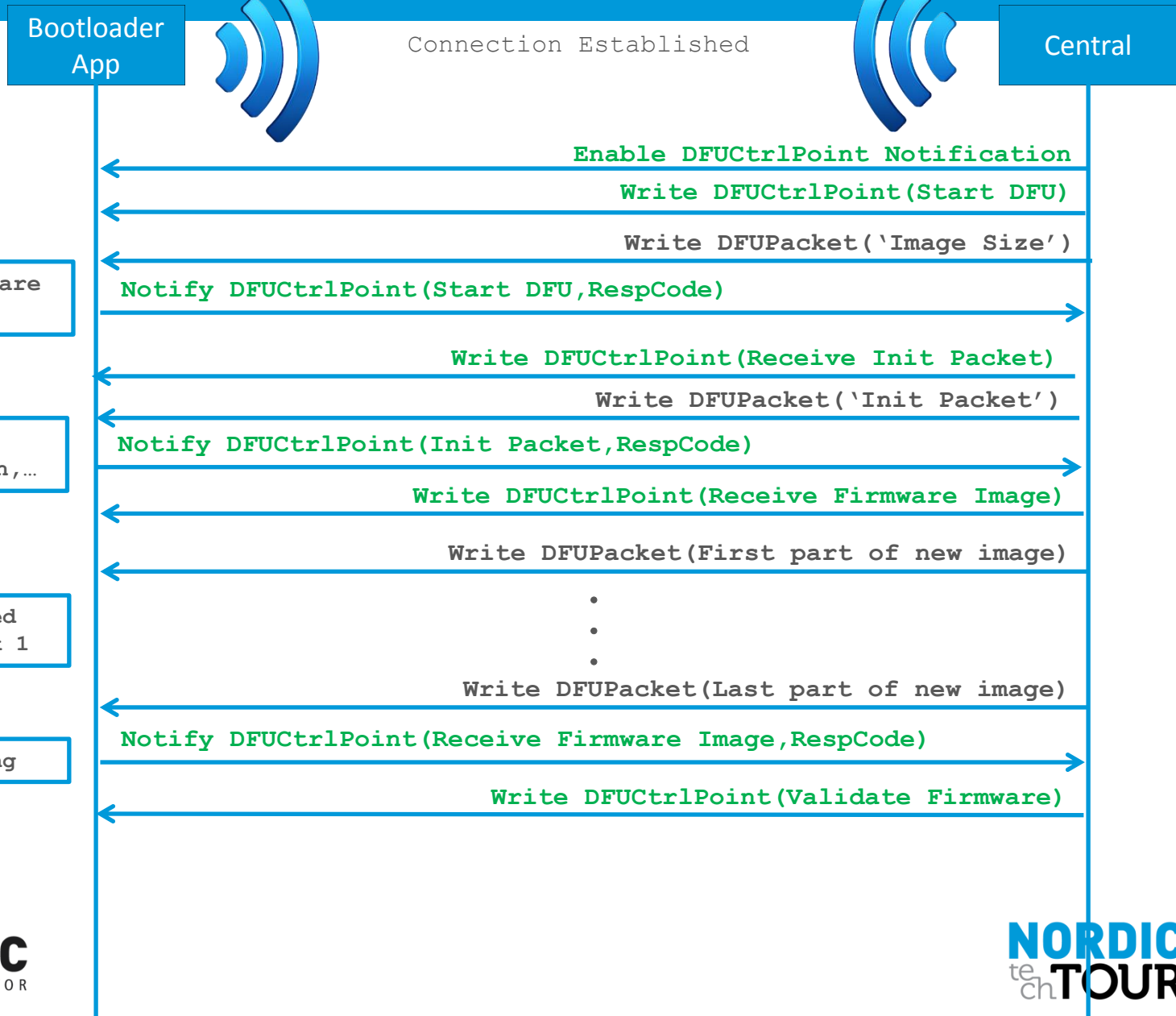


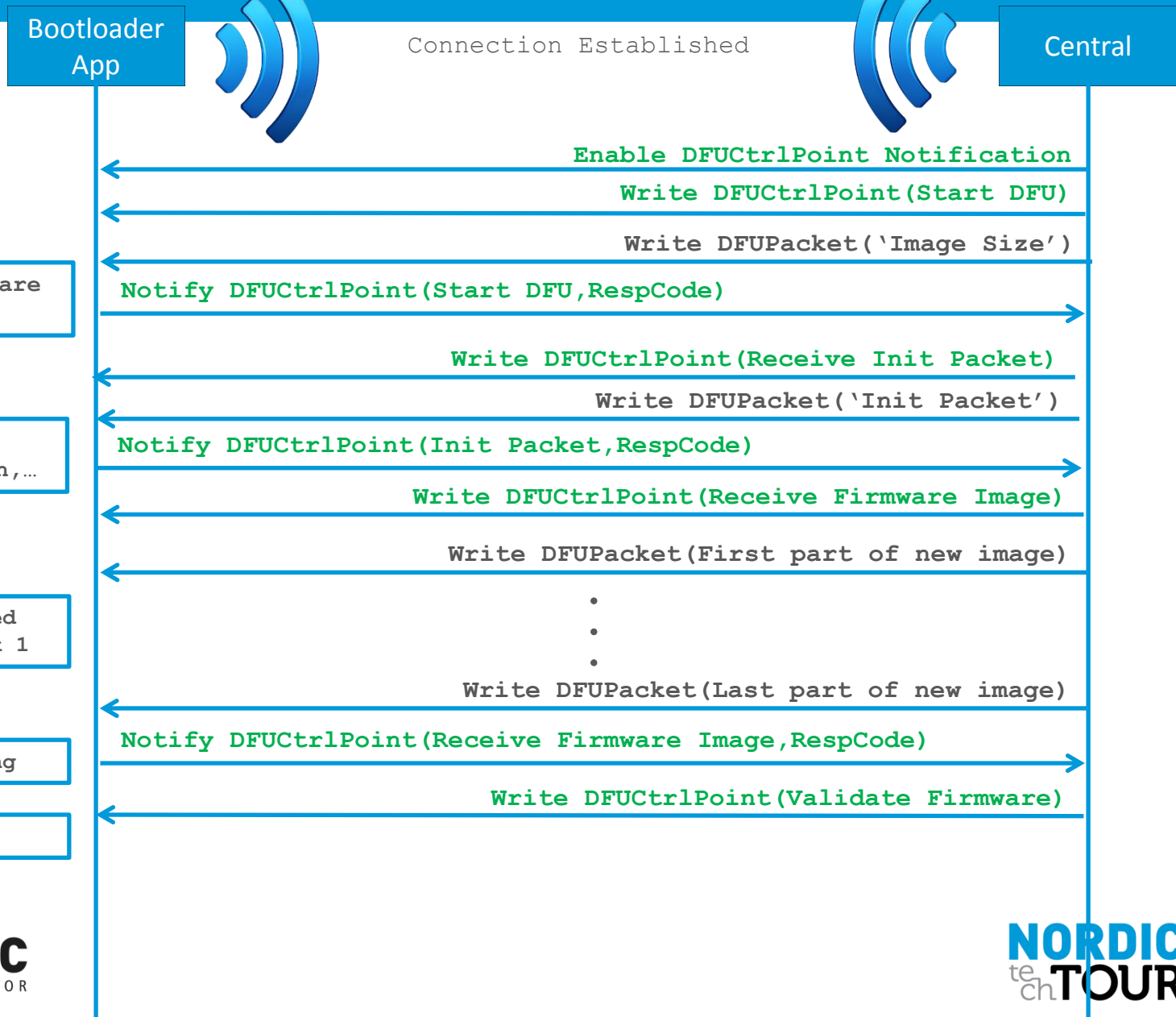


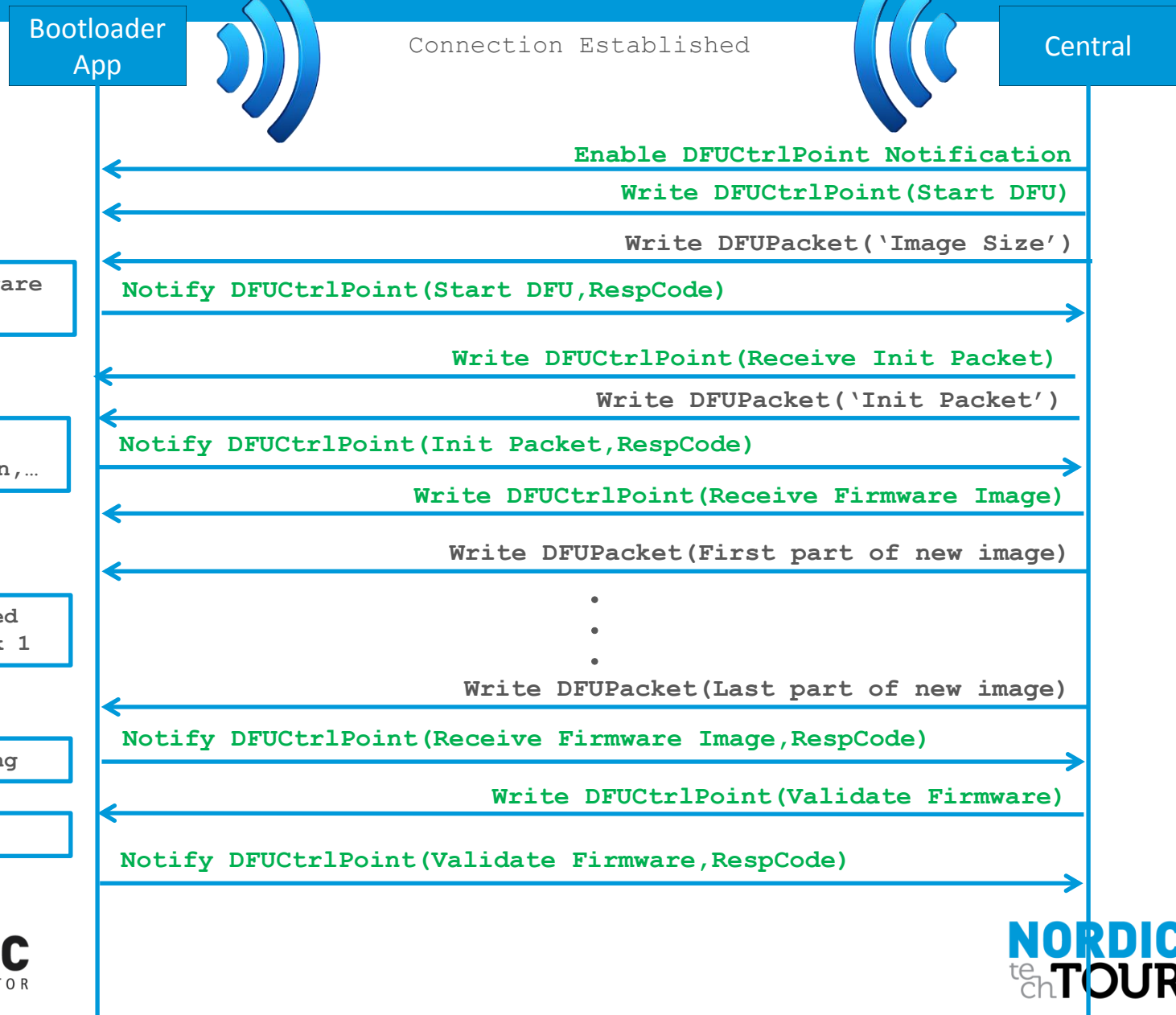


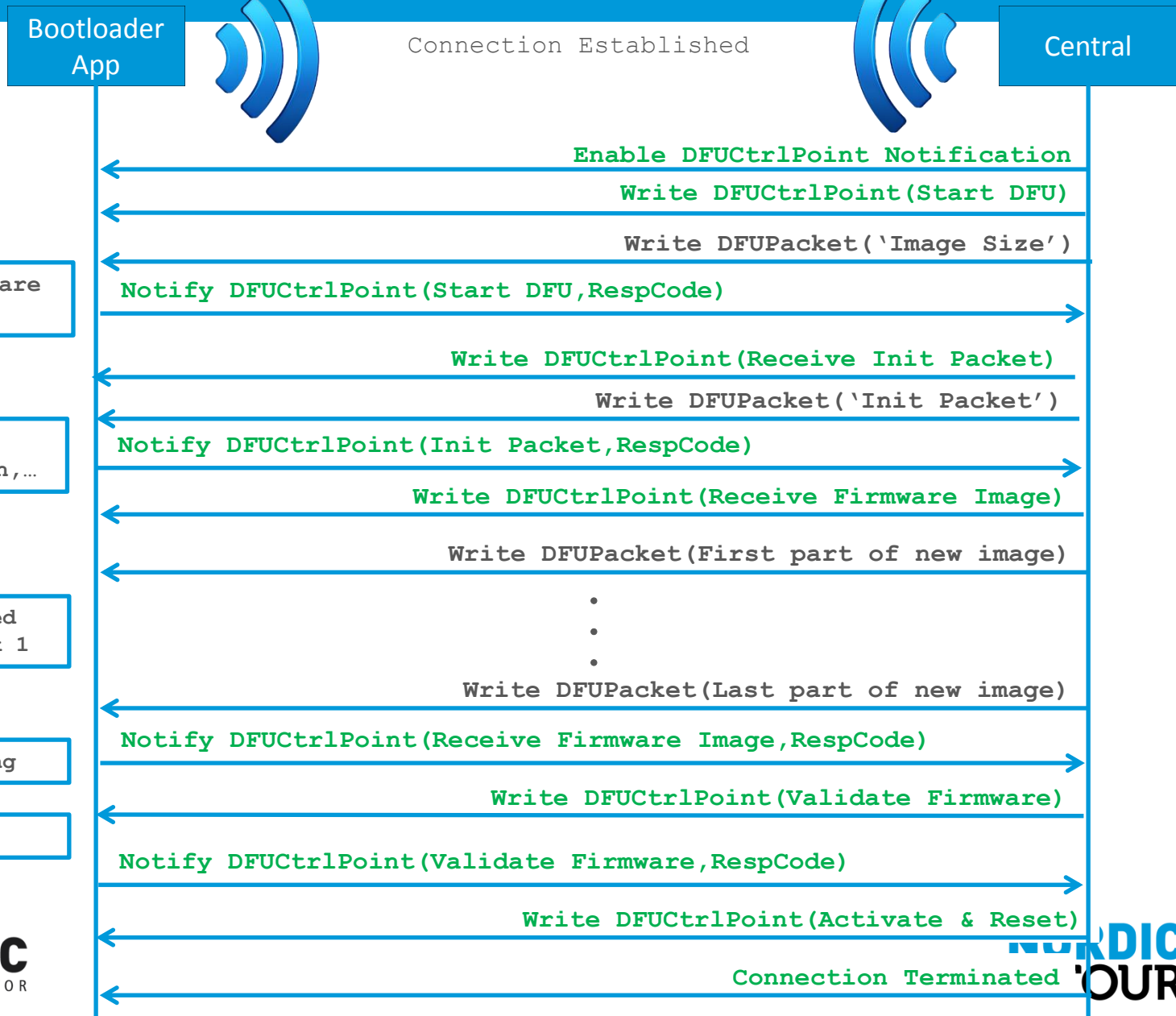


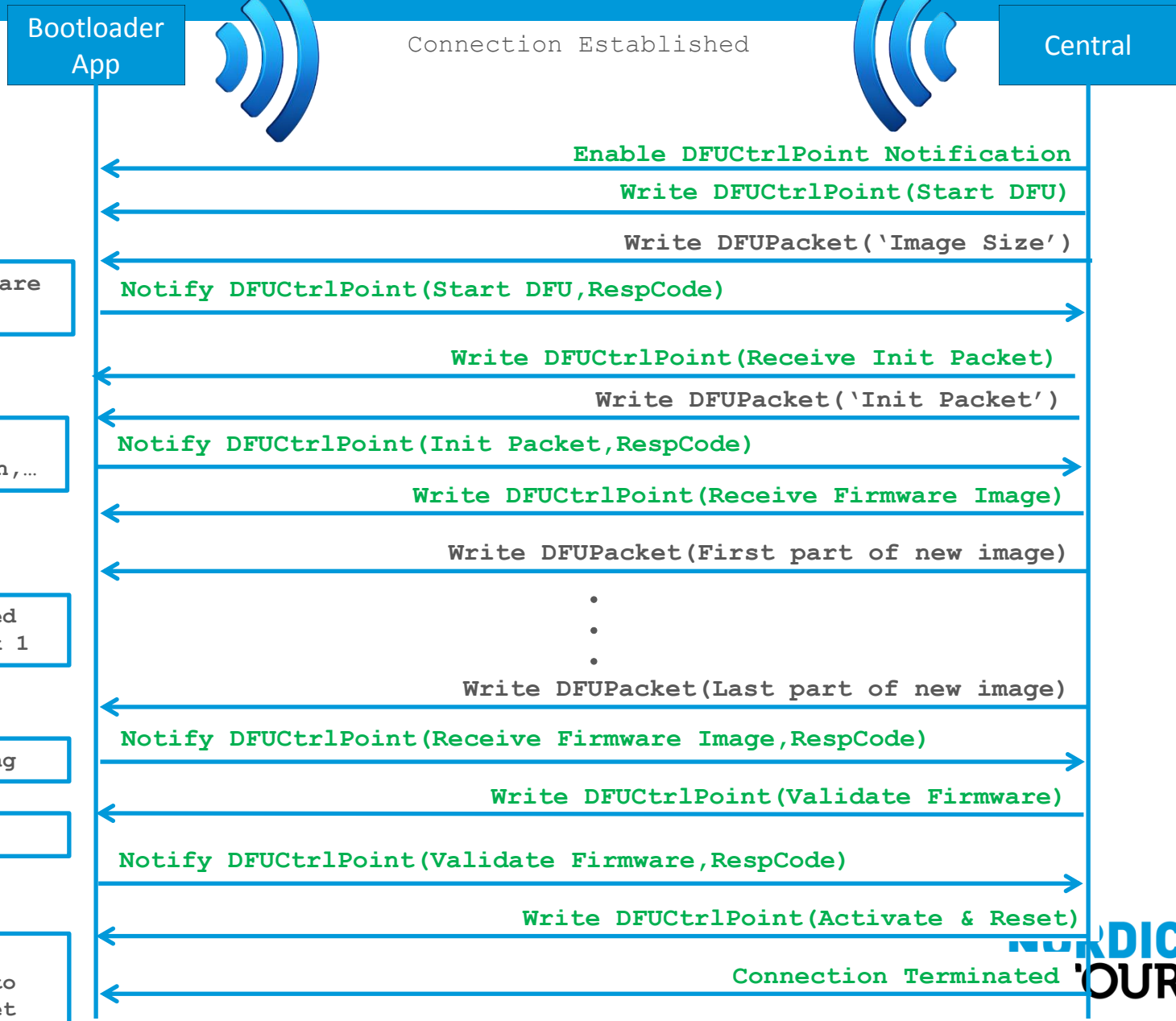












What is Init Packet

What is Init Packet

Device Type	Device Revision
Application Version	
SoftDevice array length (n)	SoftDevice - 1
SoftDevice - 2
SoftDevice - n	CRC - 16 - CCITT
← 2 bytes →	← 2 bytes →

What is Init Packet

Device Type	Device Revision
Application Version	
SoftDevice array length (n)	SoftDevice - 1
SoftDevice - 2
SoftDevice - n	CRC - 16 - CCITT
← 2 bytes →	← 2 bytes →

- Init packet contain extra data for the bootloader to verify:
 - Compatible applications are going to be installed on the device
 - Image integrity with CRC check

What is Init Packet

Device Type	Device Revision
Application Version	
SoftDevice array length (n)	SoftDevice - 1
SoftDevice - 2
SoftDevice - n	CRC - 16 - CCITT
← 2 bytes →	← 2 bytes →

- Init packet contain extra data for the bootloader to verify:
 - Compatible applications are going to be installed on the device
 - Image integrity with CRC check
- Init packet is obligated when doing DFU from SDK v7.0

What is Init Packet

One example:

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

LittleEndian

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)
- Softdevice supported = 0x0002(two softdevices)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)
- Softdevice supported = 0x0002(two softdevices)
- Softdevice 1 = 0x004F (S110 v7.0.0)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)
- Softdevice supported = 0x0002(two softdevices)
- Softdevice 1 = 0x004F (S110 v7.0.0)
- Softdevice 2 = 0x005a (S110 v7.1.0)

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)
- Softdevice supported = 0x0002(two softdevices)
- Softdevice 1 = 0x004F (S110 v7.0.0)
- Softdevice 2 = 0x005a (S110 v7.1.0)
- **If Softdevice = 0xFFFE => Any softdevice**

What is Init Packet

One example:

FFFF FFFF FFFF FFFF 0200 4F00 5a00 74CF

Decode :

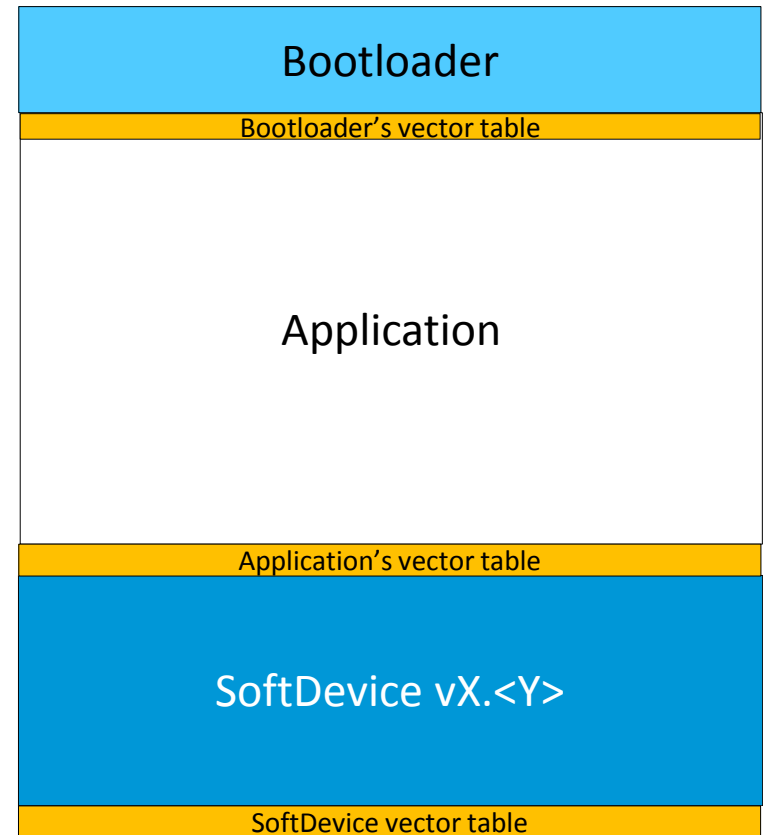
LittleEndian

- Device Type = 0xFFFF (any type)
- Device Revision = 0xFFFF (any revision)
- Application Version = 0xFFFF FFFF (any version)
- Softdevice supported = 0x0002(two softdevices)
- Softdevice 1 = 0x004F (S110 v7.0.0)
- Softdevice 2 = 0x005a (S110 v7.1.0)
- **If Softdevice = 0xFFFE => Any softdevice**
- CRC = 0xCF74

SoftDevice and Bootloader Update

New feature from S110 v7.0 and later

Application can be updated, but what about updating softdevice and also the bootloader itself ?

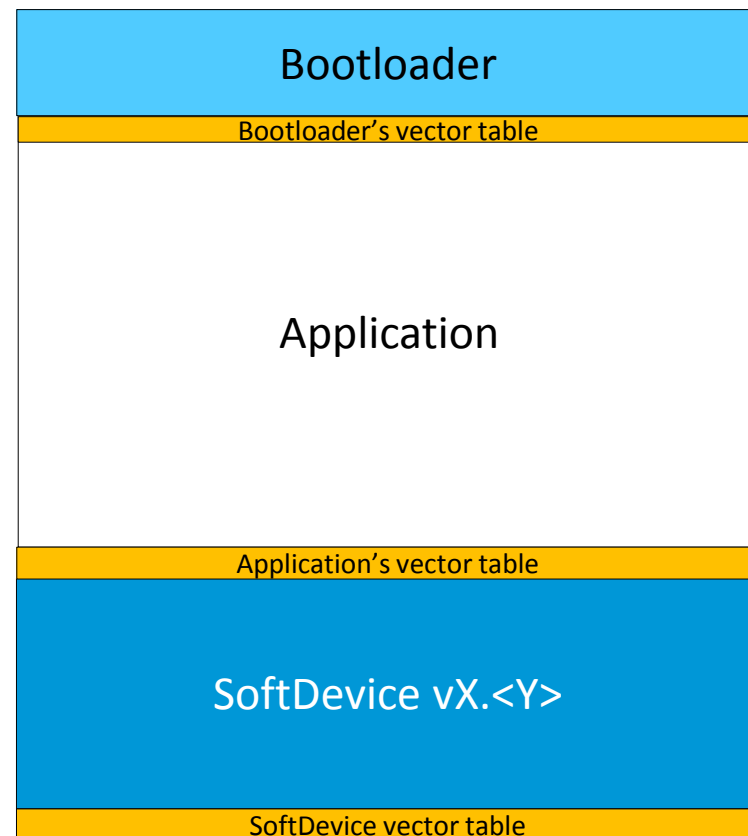


SoftDevice and Bootloader Update

New feature from S110 v7.0 and later

Application can be updated, but what about updating softdevice and also the bootloader itself ?

From SDK v6.1, softdevice update is supported by adding a “Master boot record” under the Softdevice in code region 0 to handle flash updating for softdevice and bootloader.

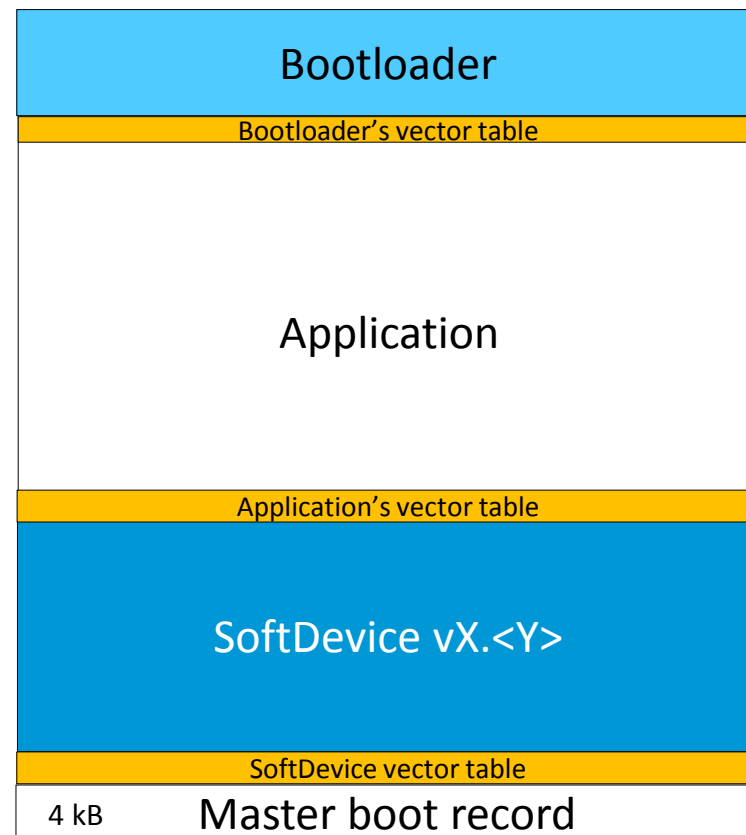


SoftDevice and Bootloader Update

New feature from S110 v7.0 and later

Application can be updated, but what about updating softdevice and also the bootloader itself ?

From SDK v6.1, softdevice update is supported by adding a “Master boot record” under the Softdevice in code region 0 to handle flash updating for softdevice and bootloader.



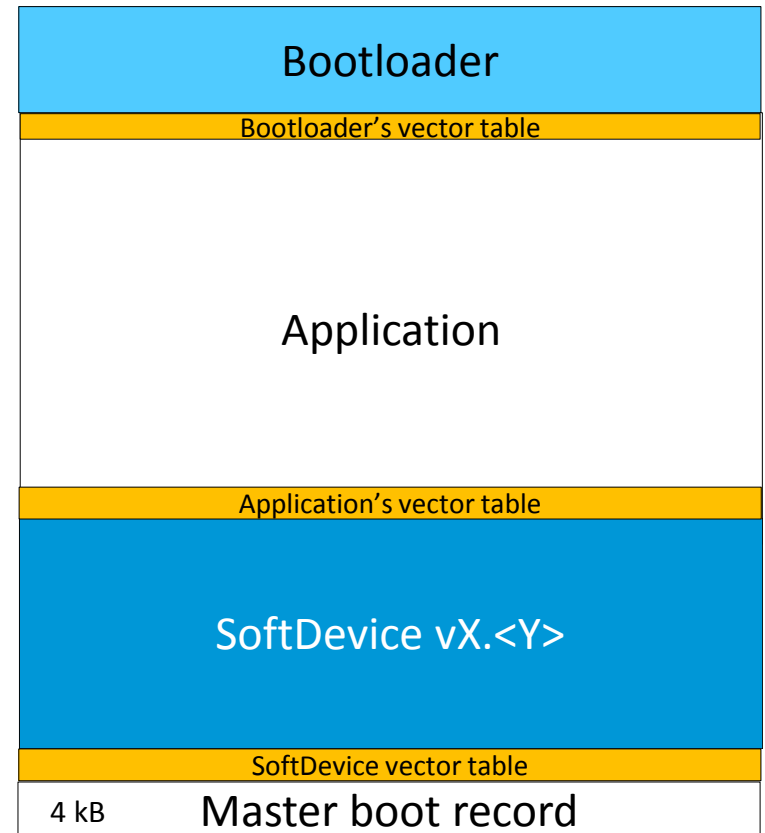
SoftDevice and Bootloader Update

New feature from S110 v7.0 and later

Application can be updated, but what about updating softdevice and also the bootloader itself ?

From SDK v6.1, softdevice update is supported by adding a “Master boot record” under the Softdevice in code region 0 to handle flash updating for softdevice and bootloader.

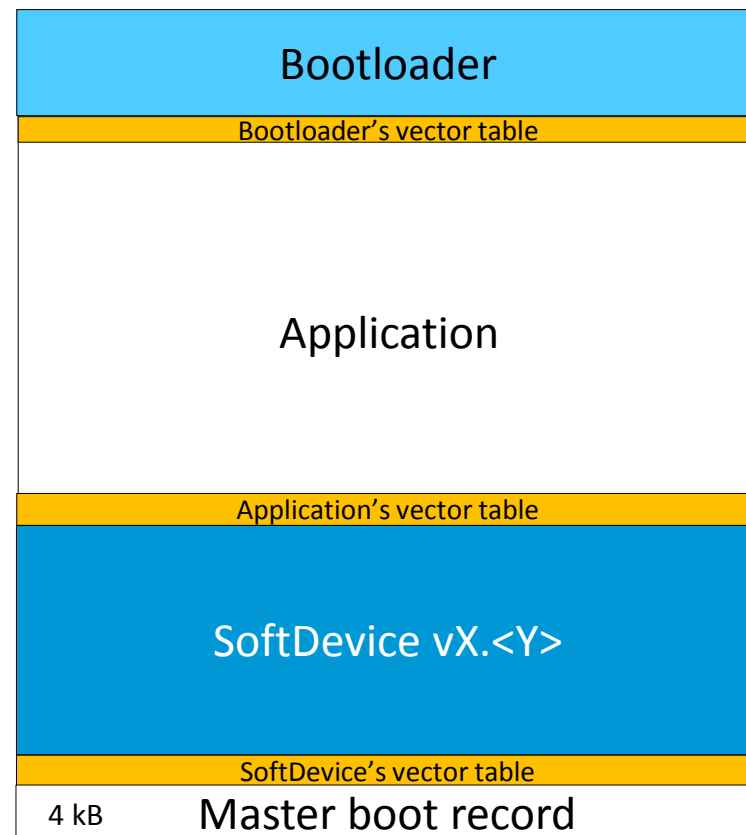
- Four options:
 - Application update
 - Bootloader update
 - Softdevice update
 - Softdevice + bootloader update



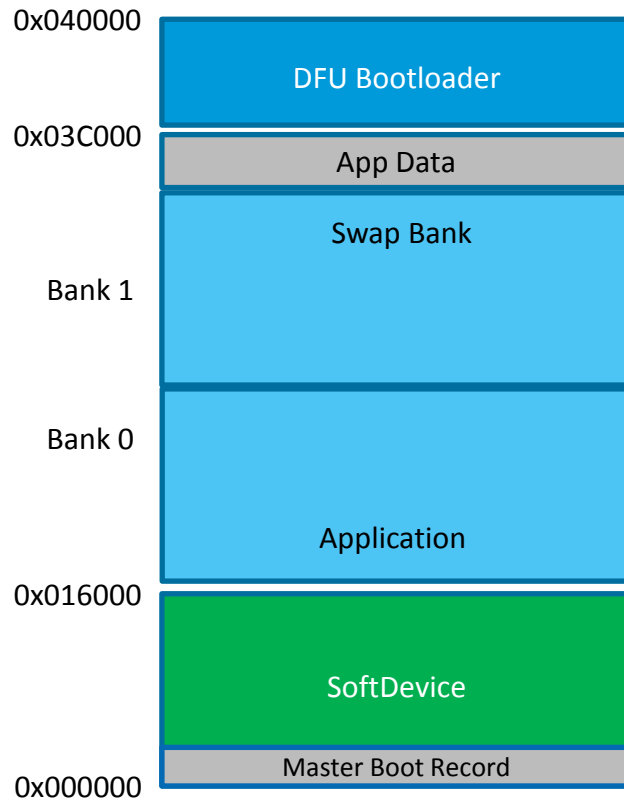
SoftDevice and Bootloader Update

New feature from S110 v7.0 and later

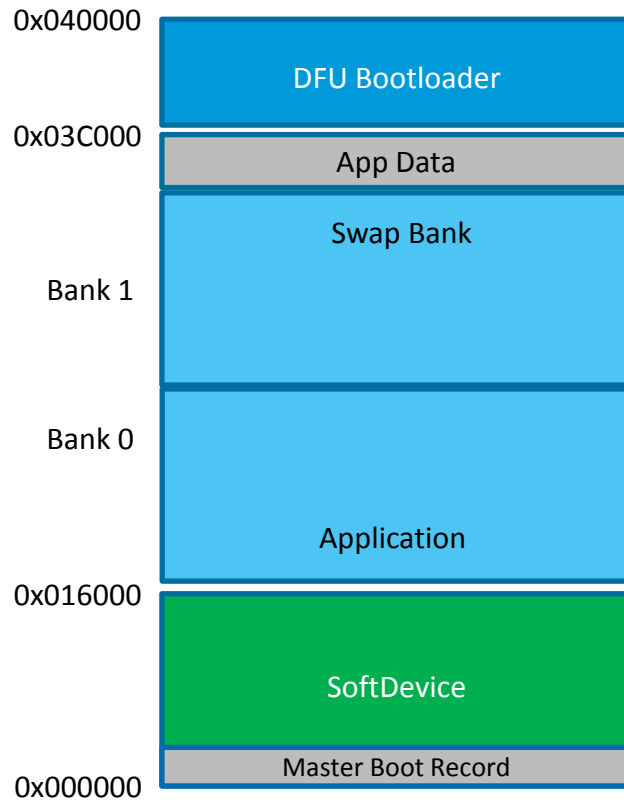
- **PROs:**
 - Full SoftDevice update
 - Bug fixes **as well as** feature updates!
 - Full application and bootloader upgrade
 - Can use new SD features!
- **CONs:**
 - 4kB flash used for Bottom part to handle flash updating



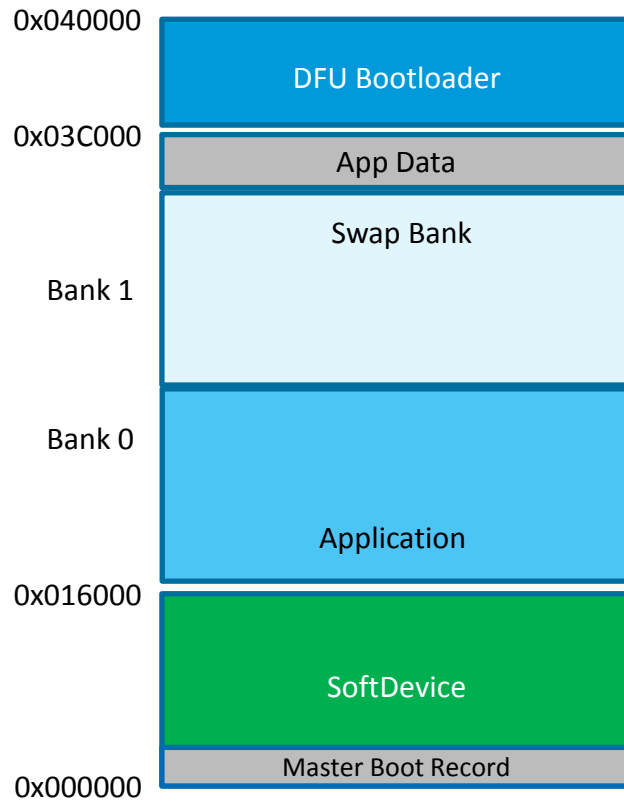
Softdevice and bootloader update



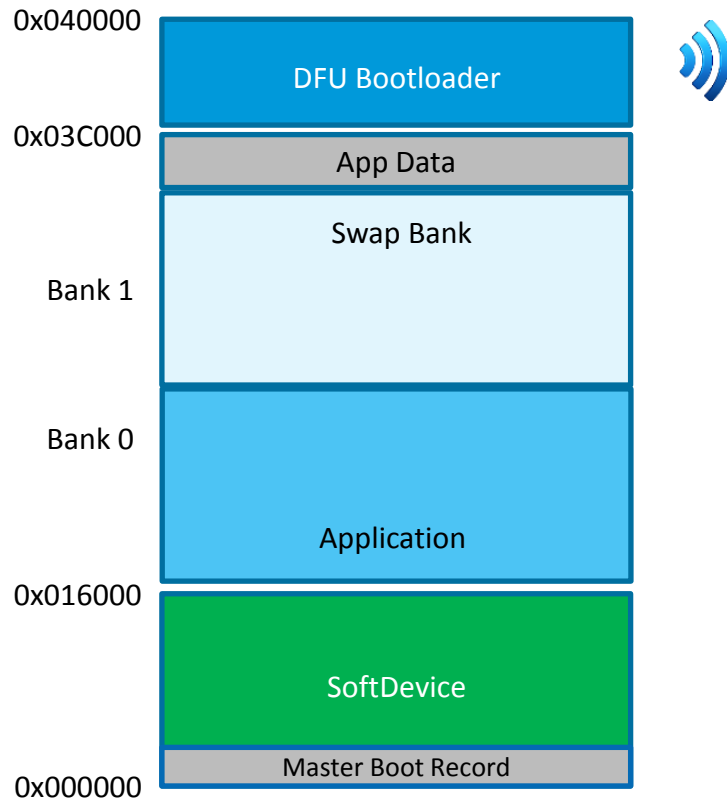
Softdevice and bootloader update



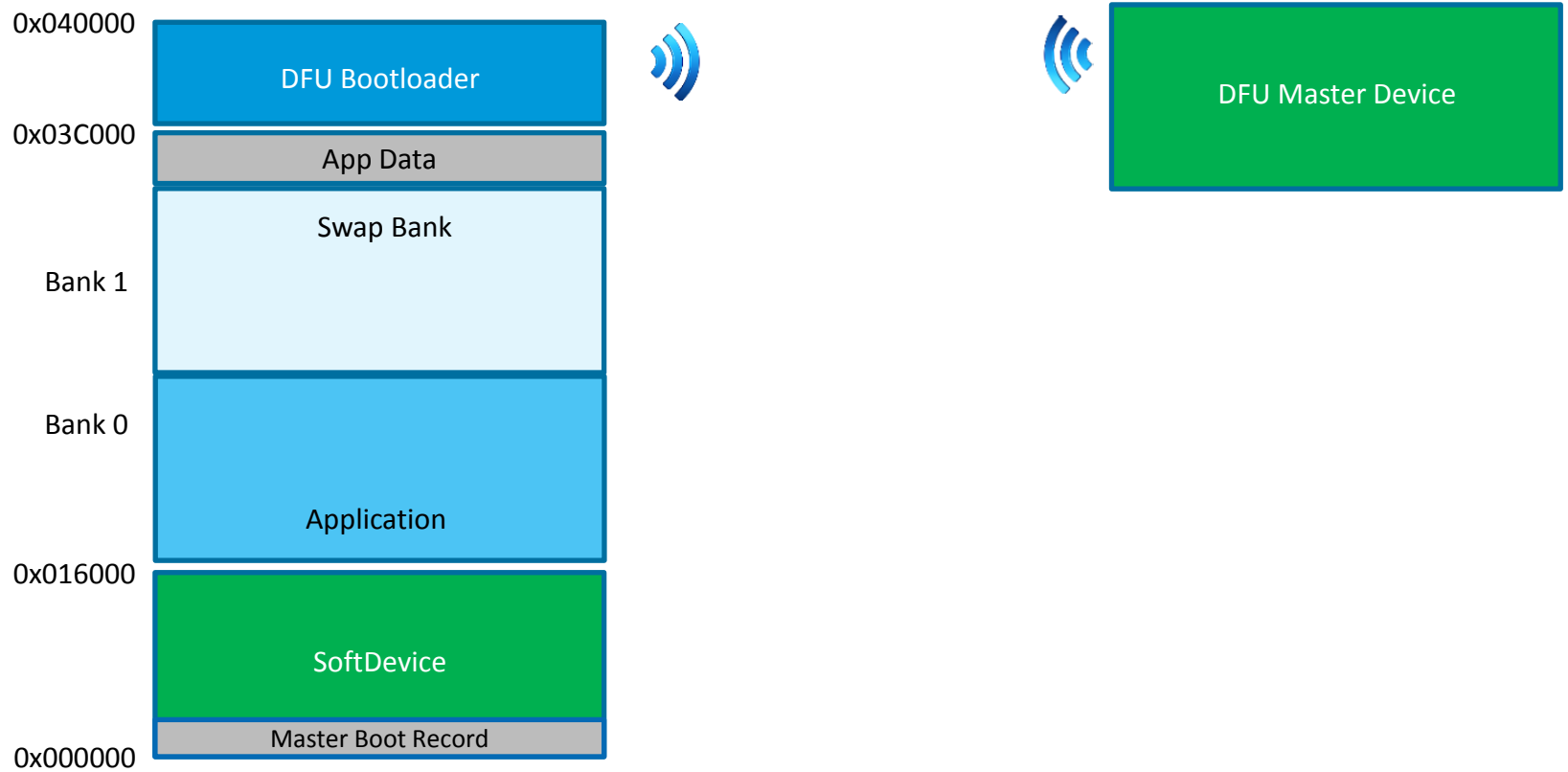
Softdevice and bootloader update



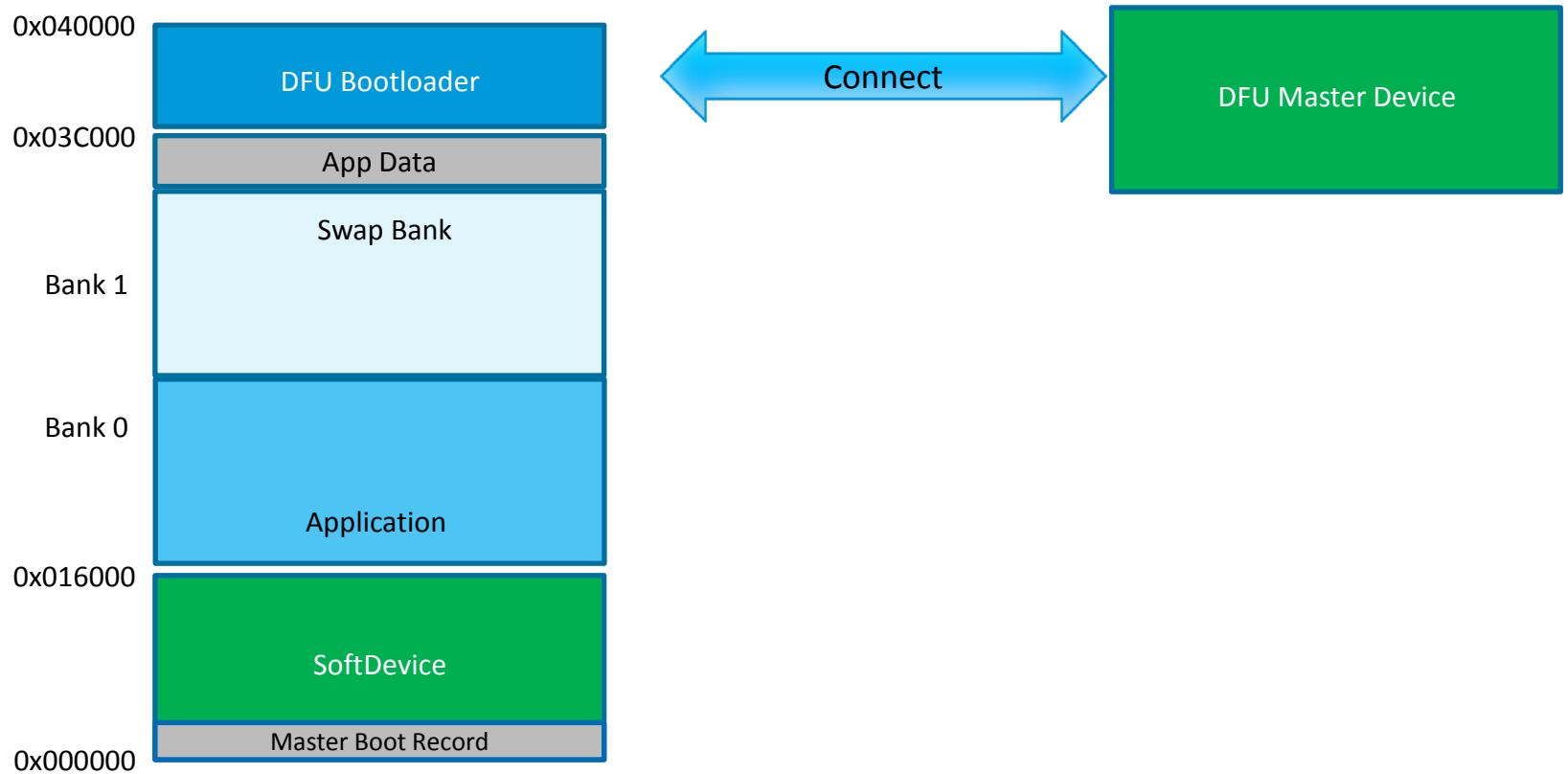
Softdevice and bootloader update



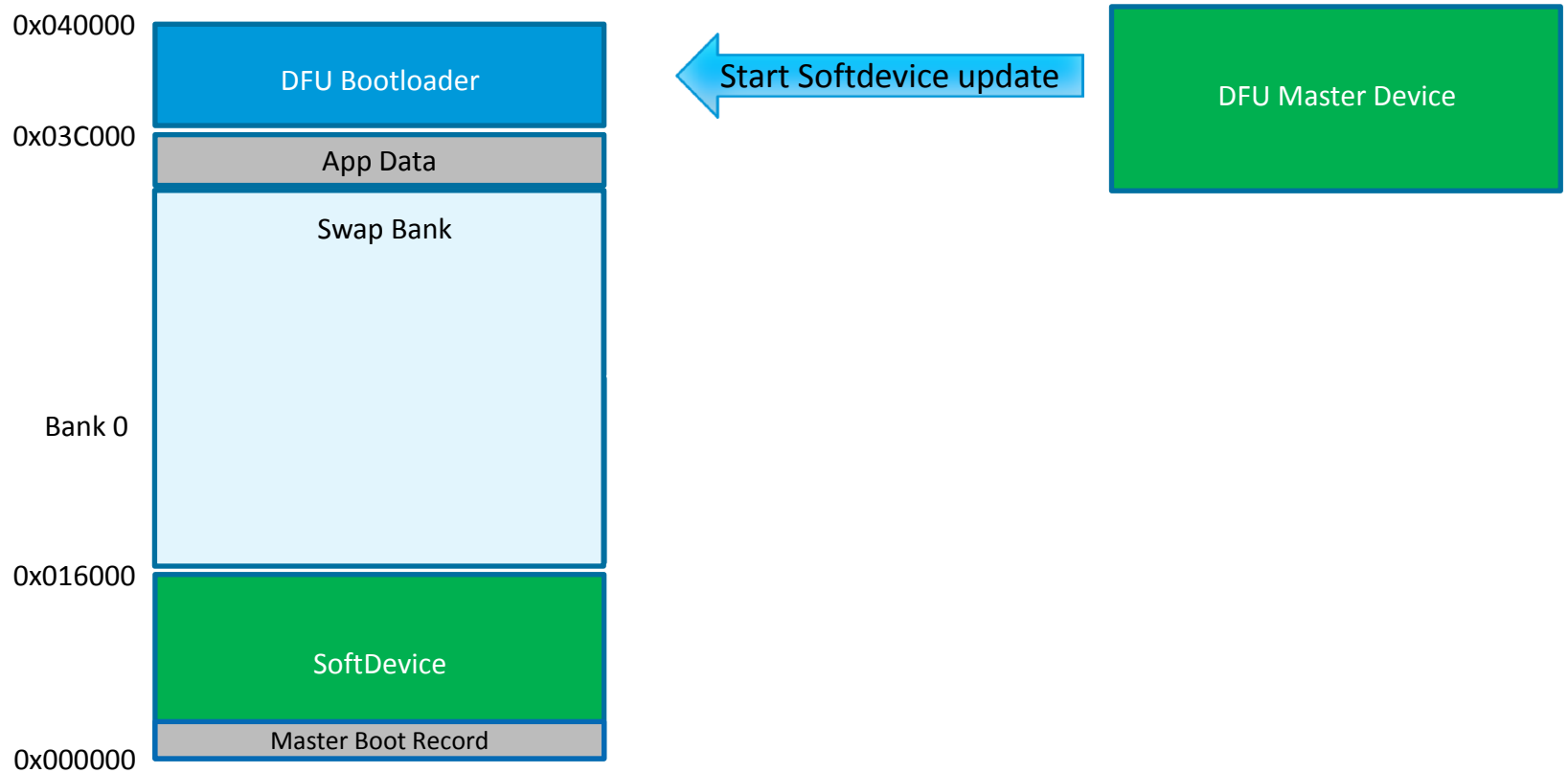
Softdevice and bootloader update



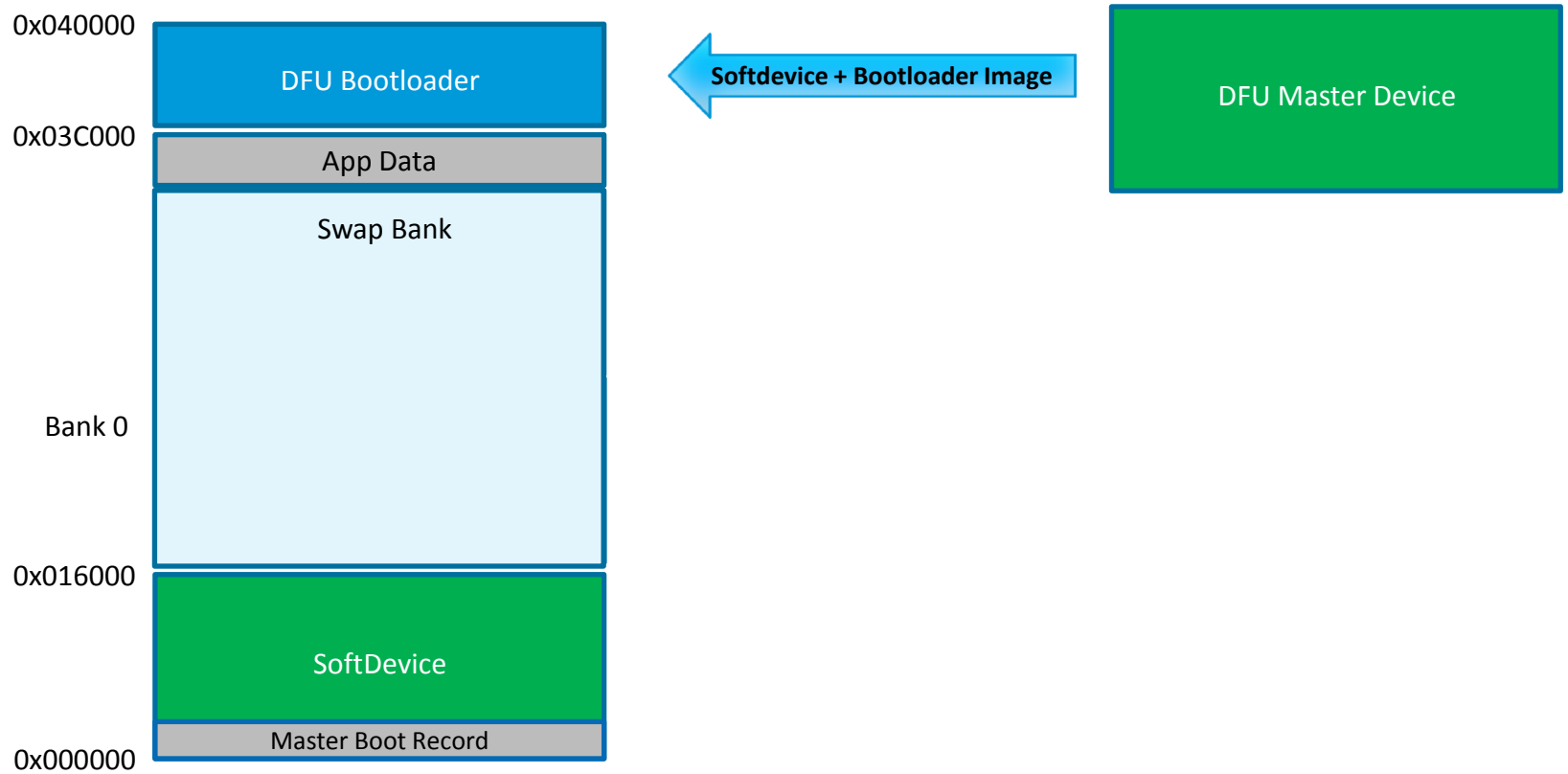
Softdevice and bootloader update



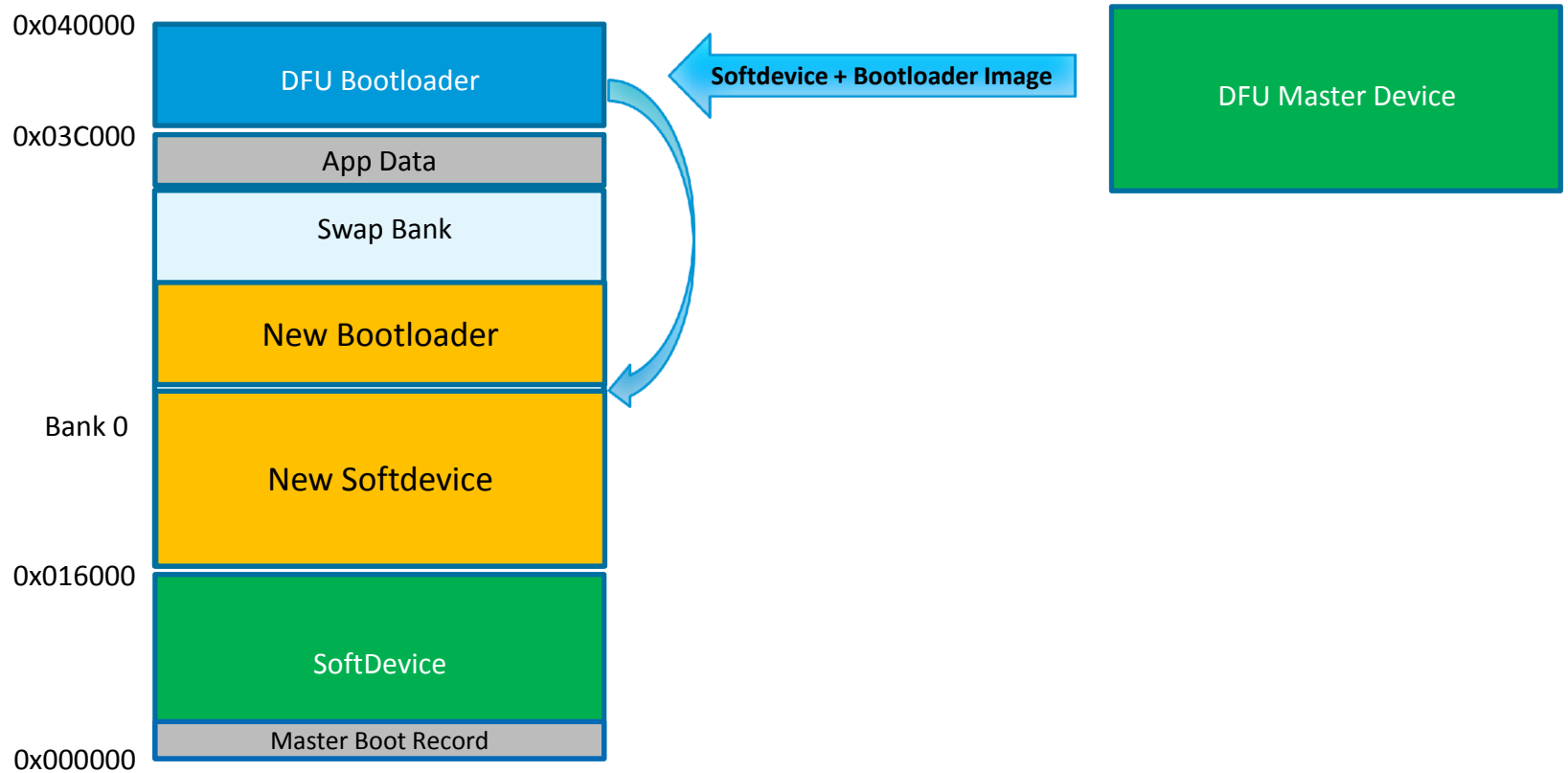
Softdevice and bootloader update



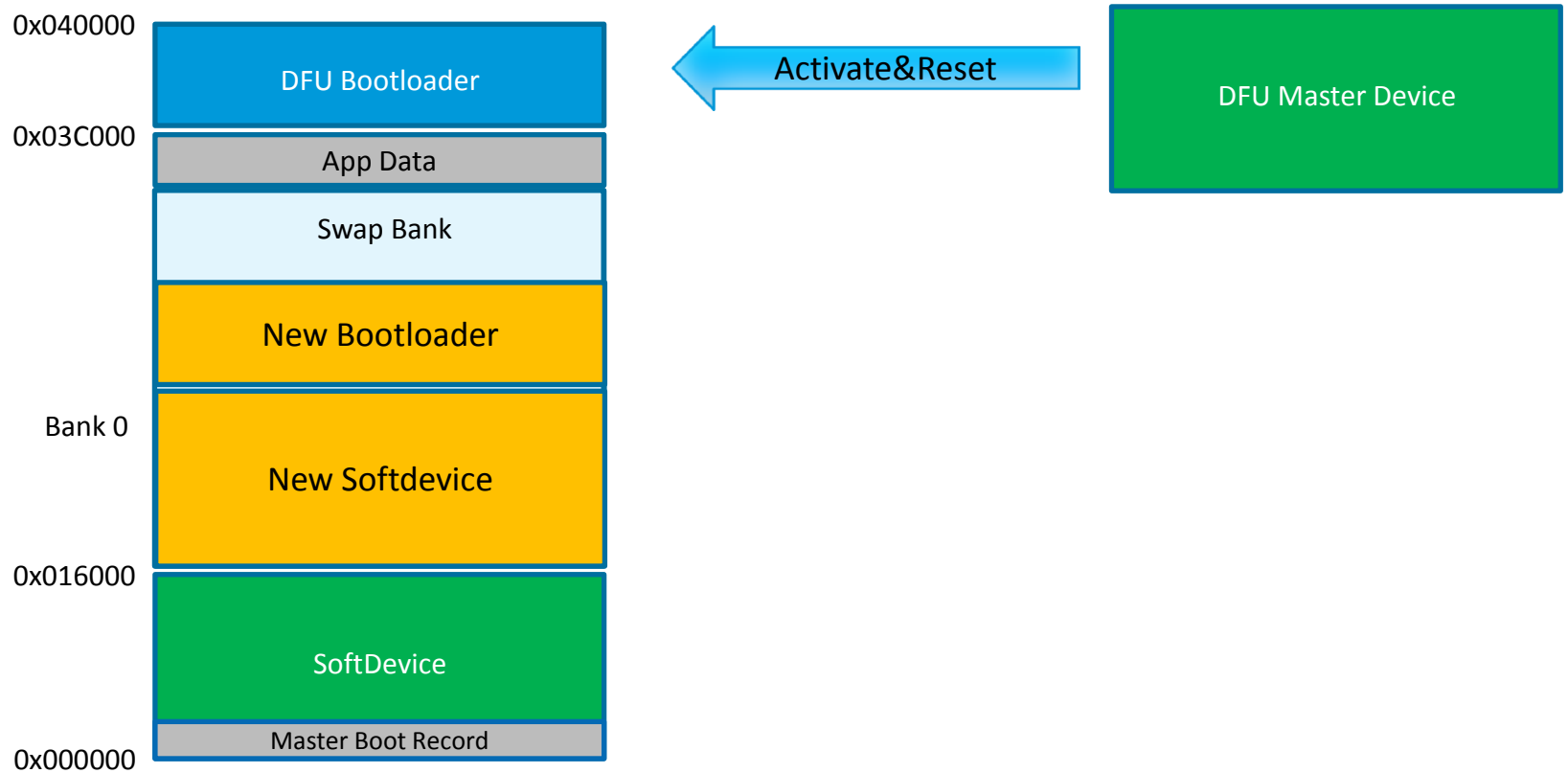
Softdevice and bootloader update



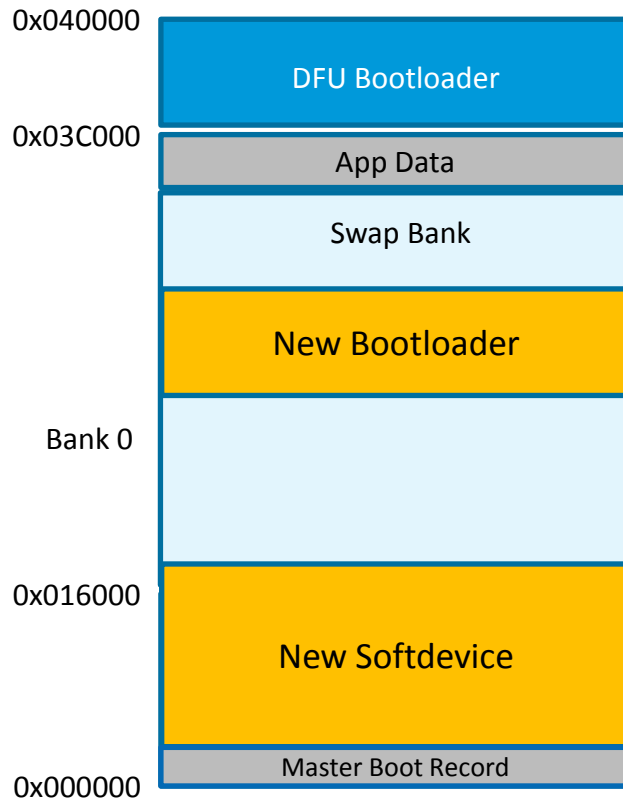
Softdevice and bootloader update



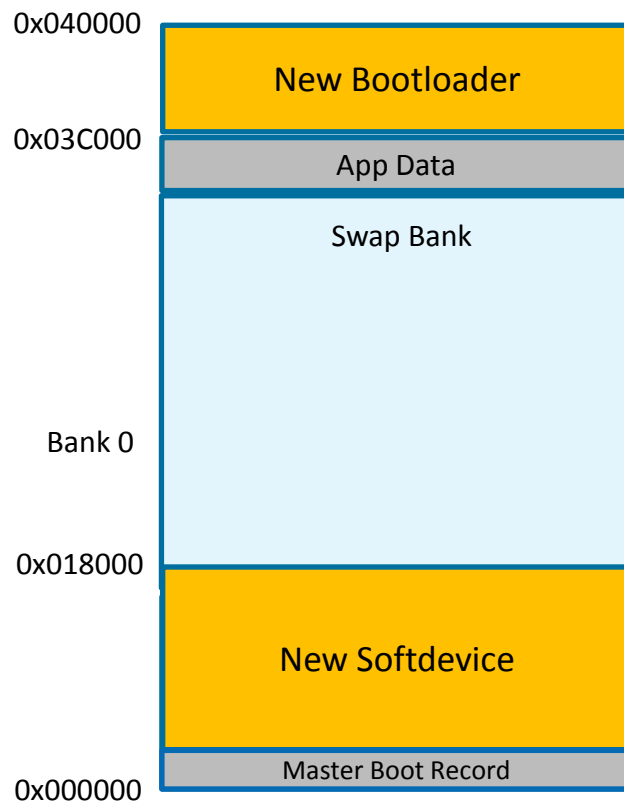
Softdevice and bootloader update



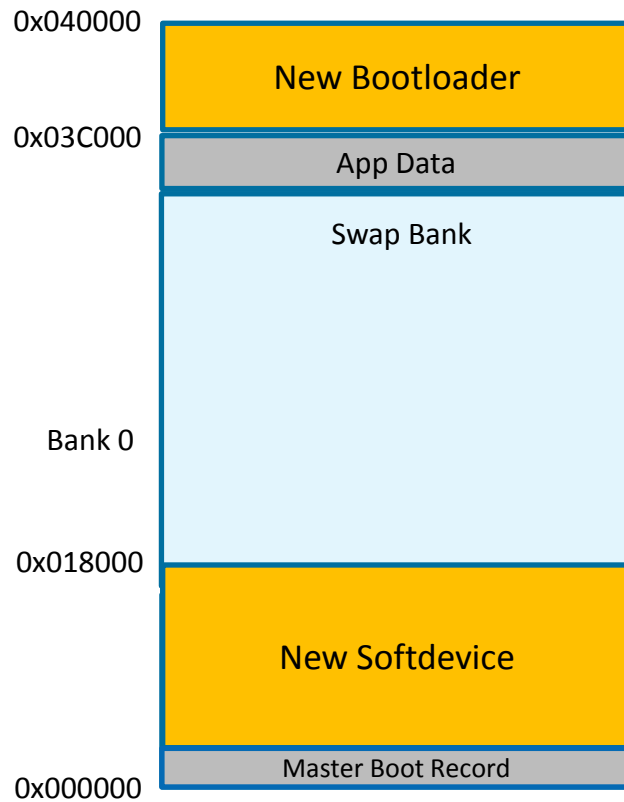
Softdevice and bootloader update



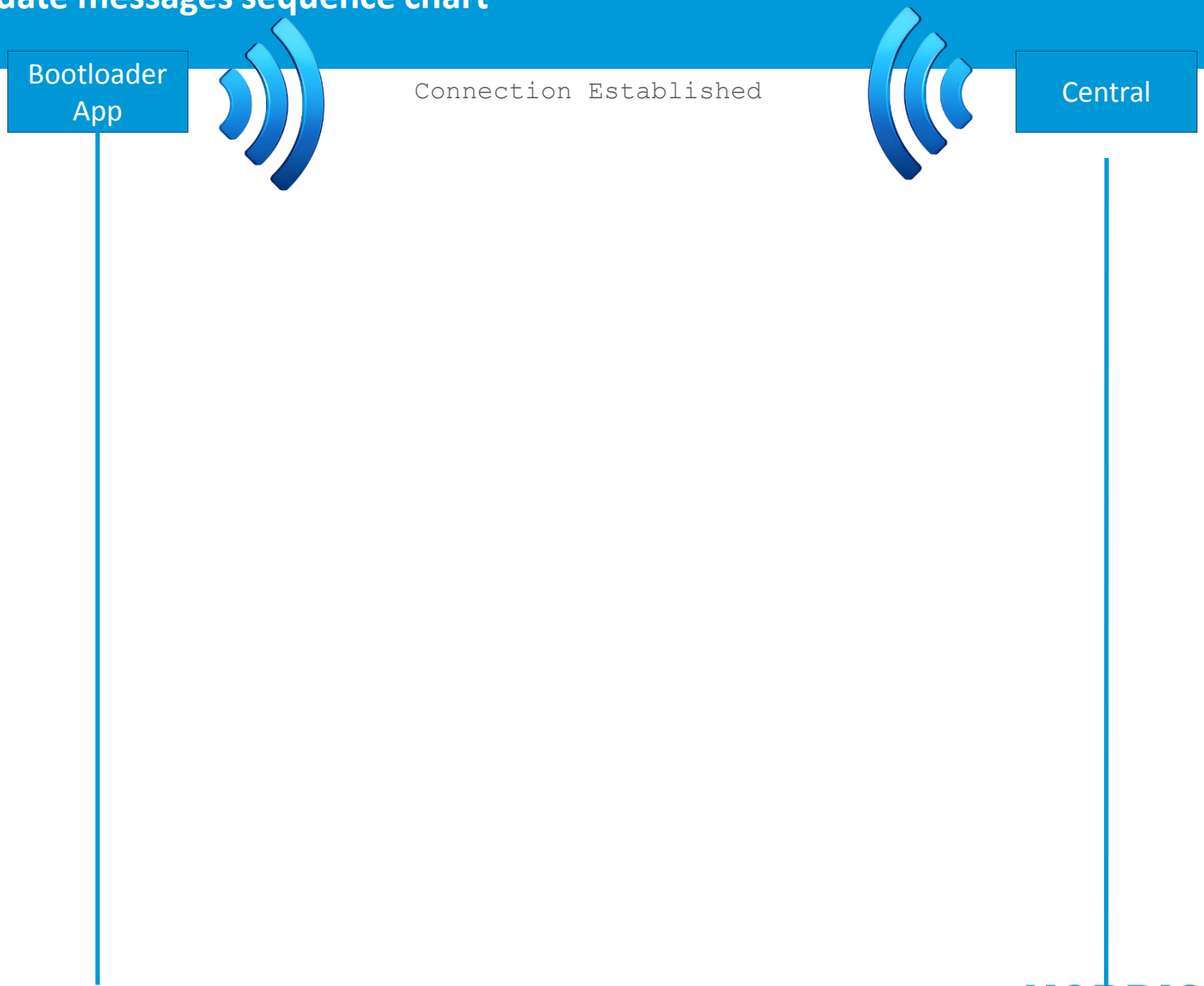
Softdevice and bootloader update



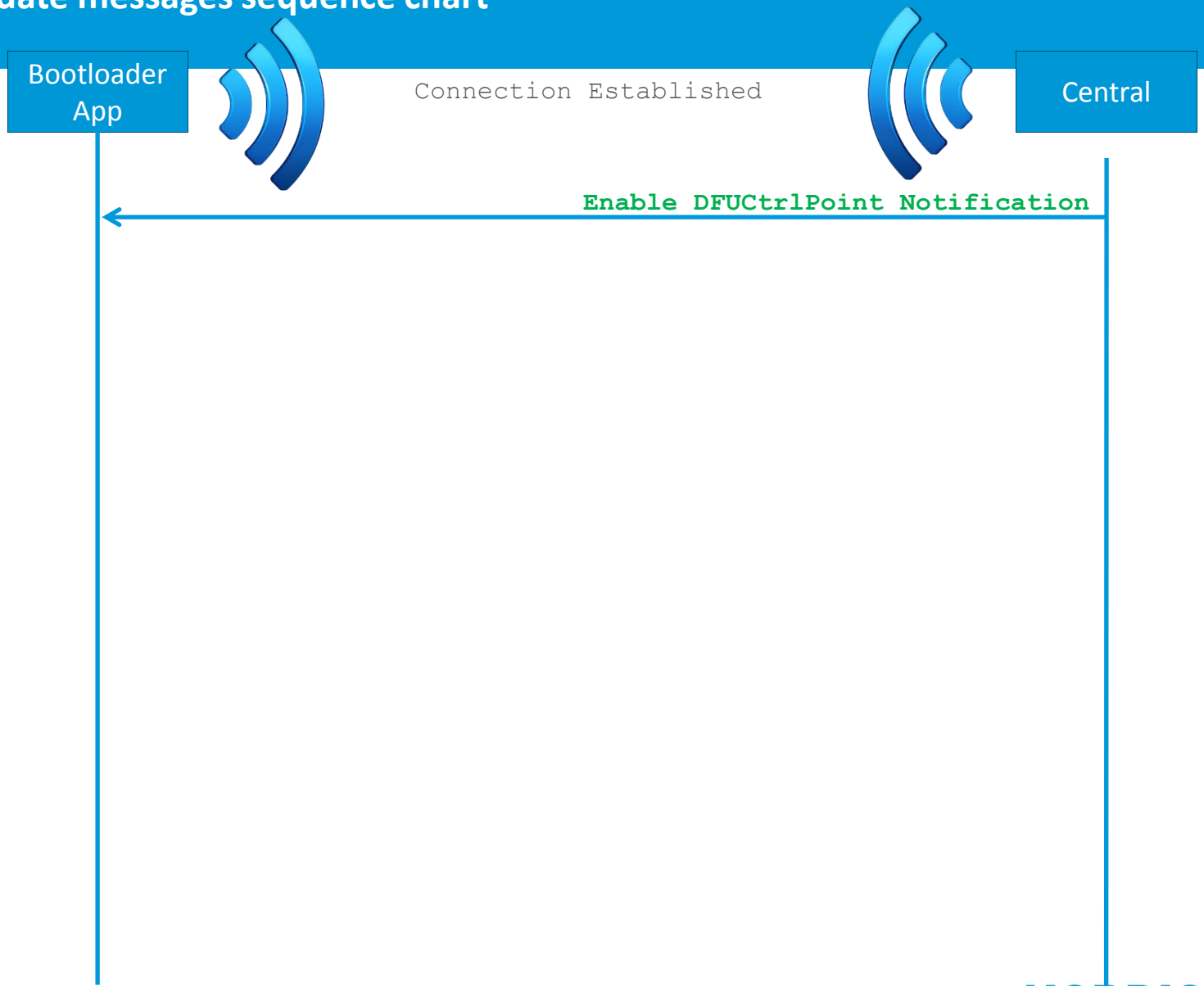
Softdevice and bootloader update



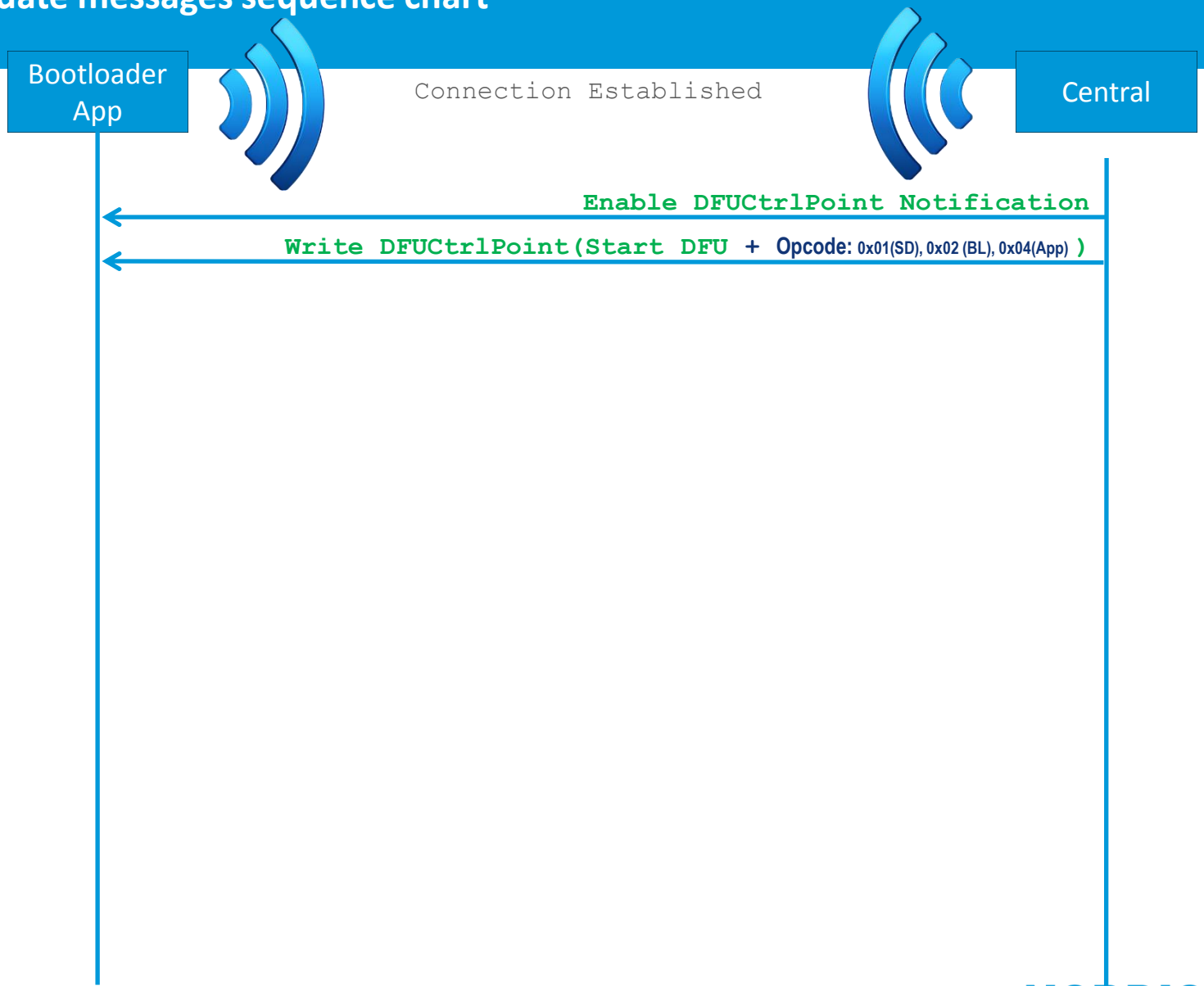
Softdevice update messages sequence chart



Softdevice update messages sequence chart



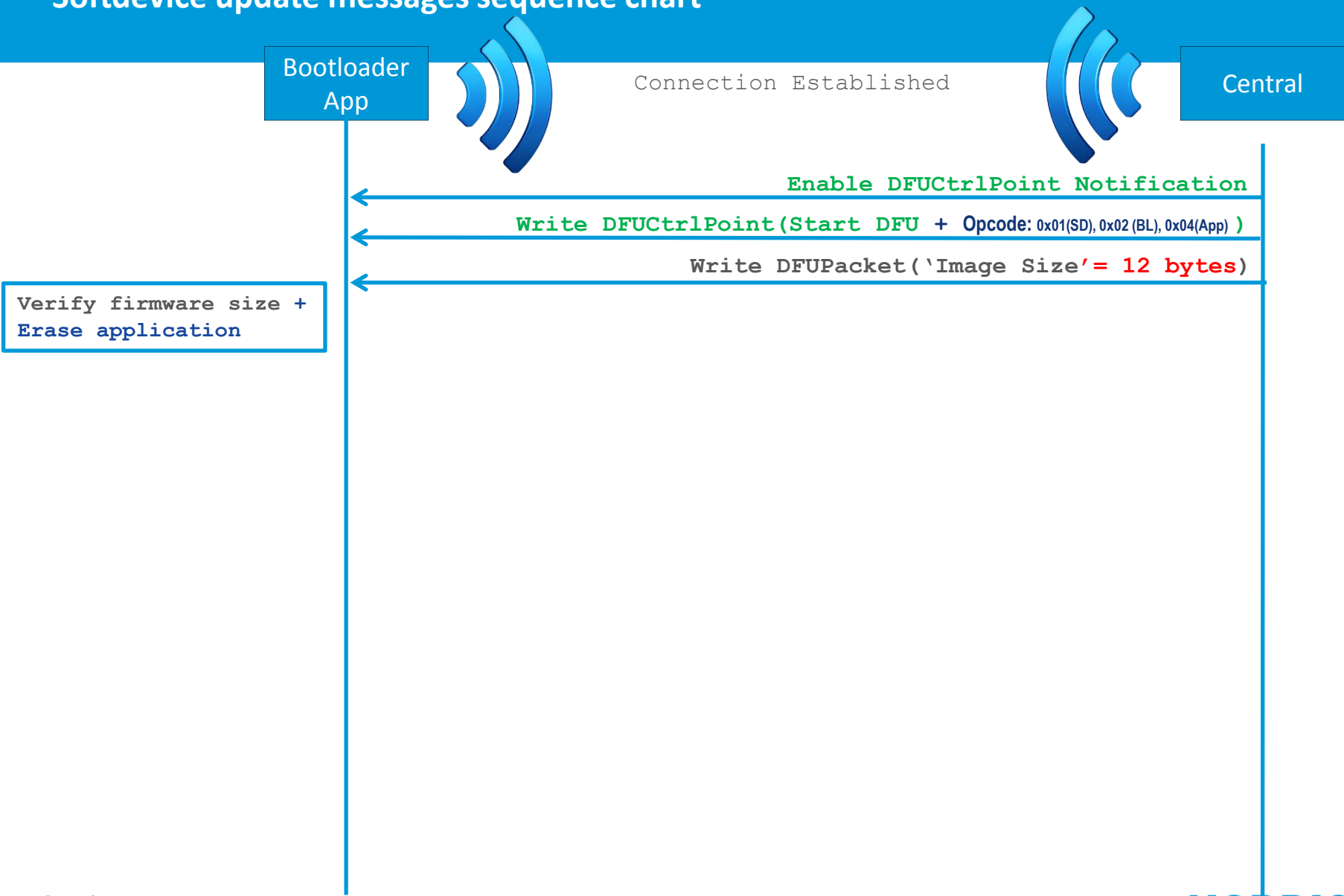
Softdevice update messages sequence chart



Softdevice update messages sequence chart



Softdevice update messages sequence chart



Verify firmware size + Erase application

Softdevice update messages sequence chart

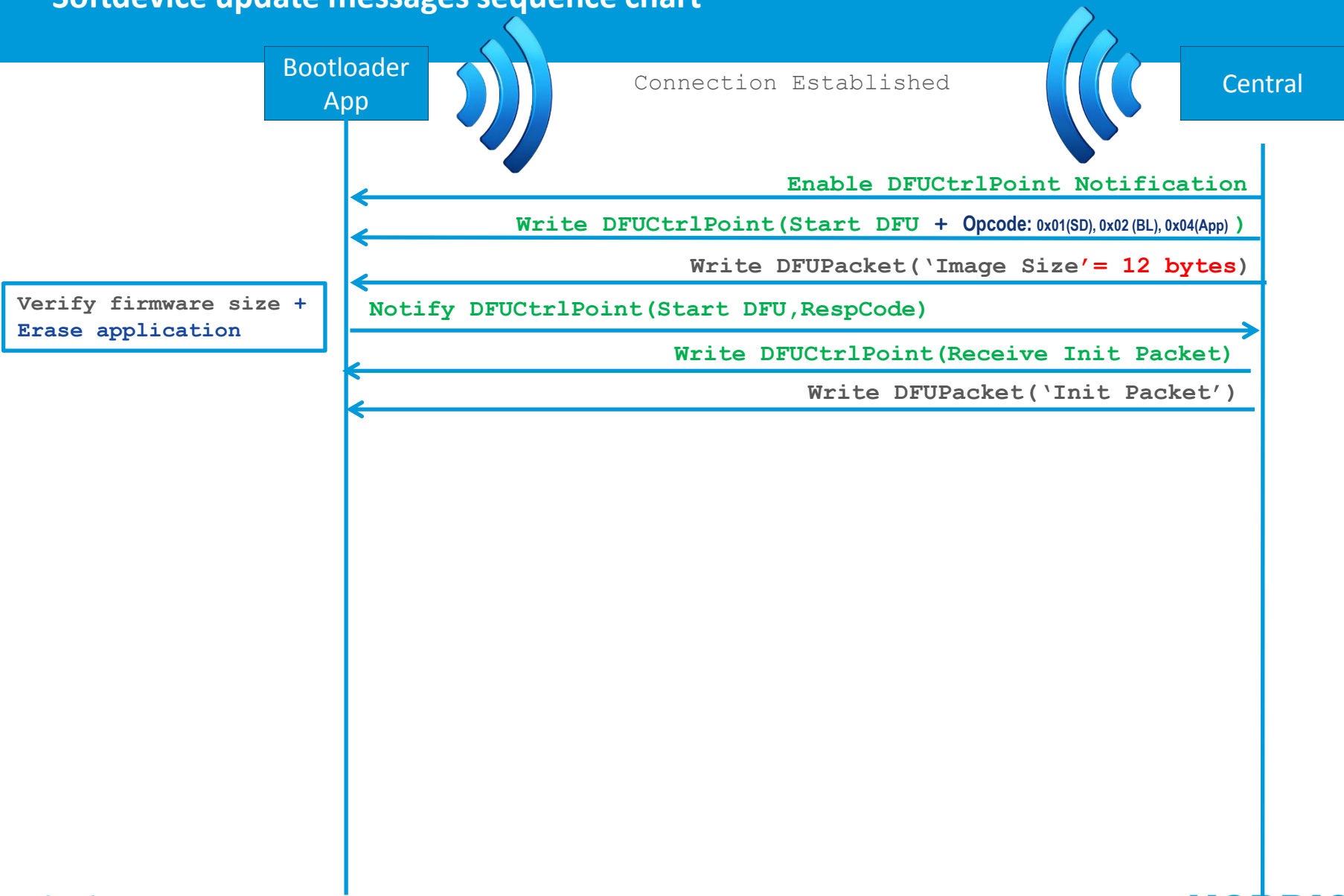


Verify firmware size + Erase application

Softdevice update messages sequence chart



Softdevice update messages sequence chart



Softdevice update messages sequence chart



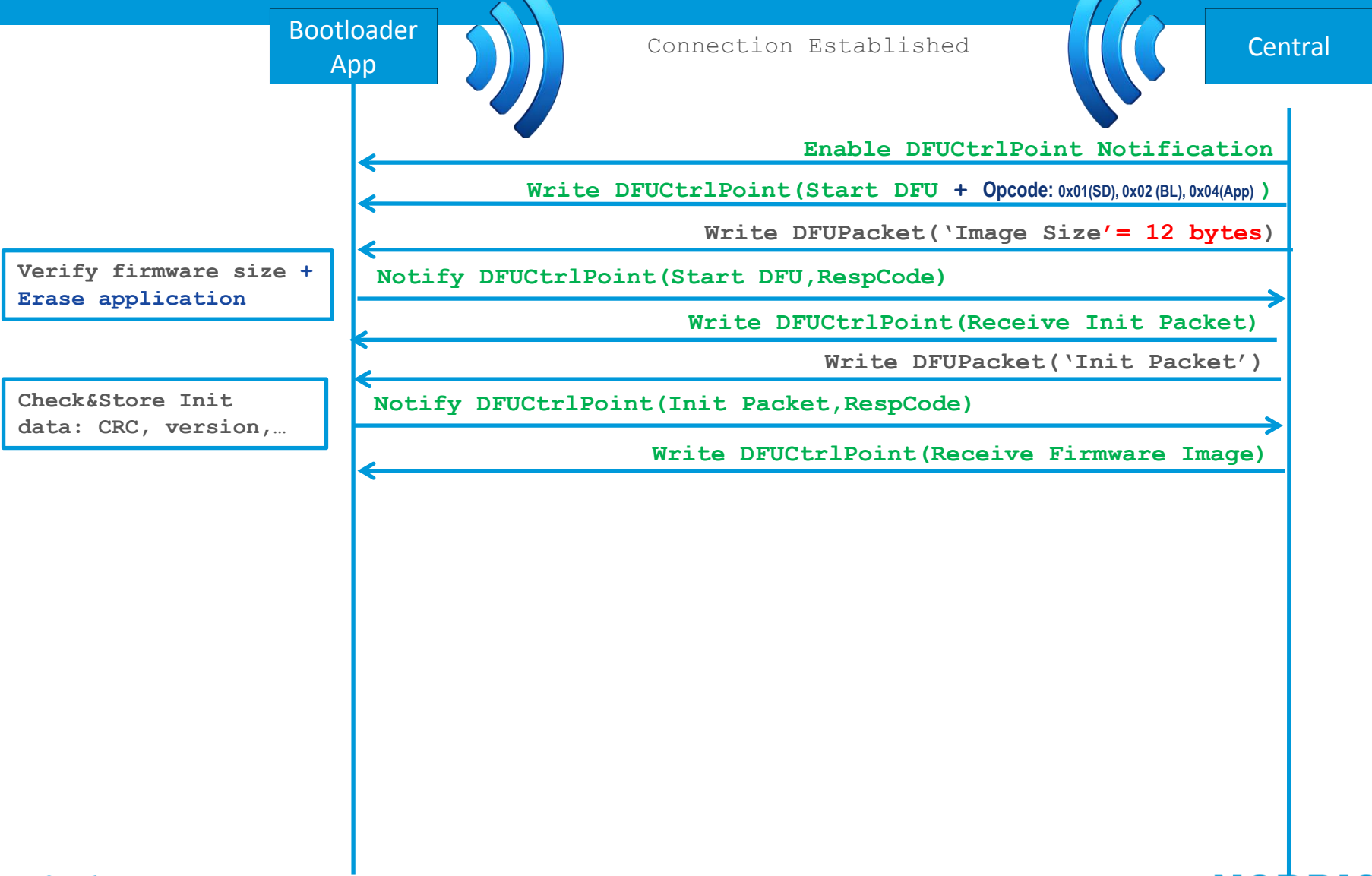
Softdevice update messages sequence chart



Verify firmware size + Erase application

Check & Store Init data: CRC, version, ...

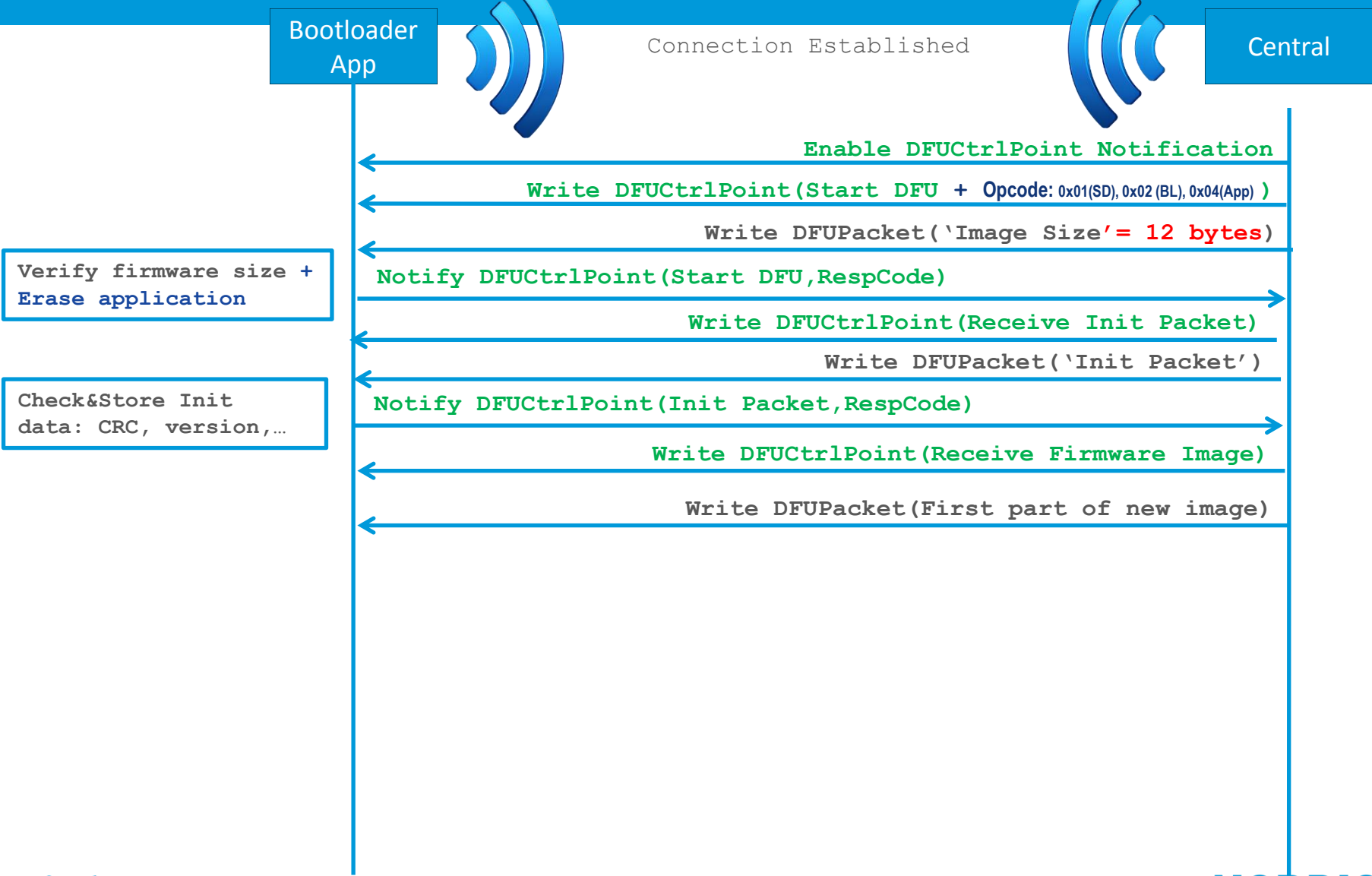
Softdevice update messages sequence chart



Verify firmware size + Erase application

Check & Store Init data: CRC, version, ...

Softdevice update messages sequence chart



Verify firmware size +
Erase application

Check&Store Init
data: CRC, version,...

Softdevice update messages sequence chart

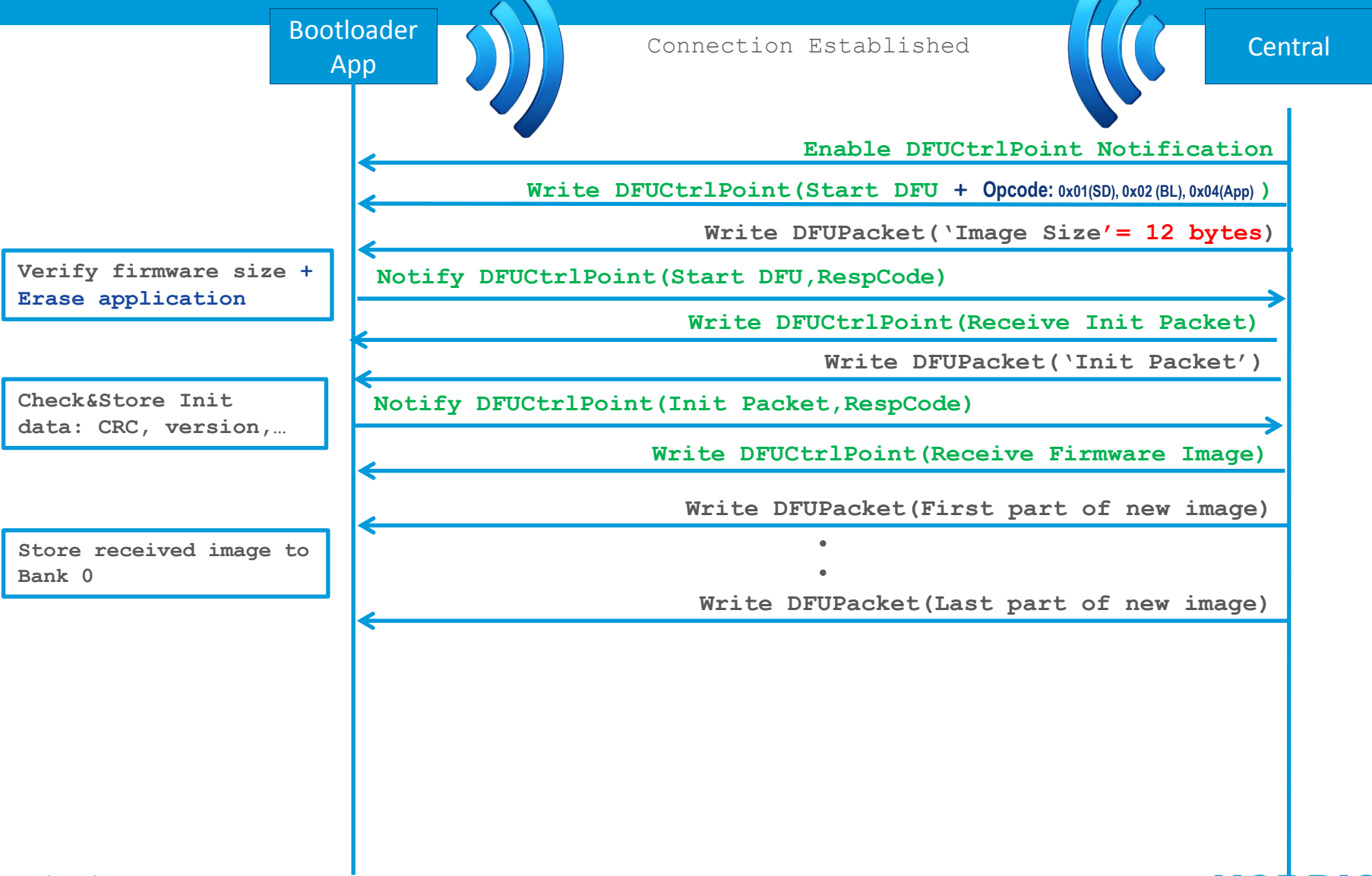


Verify firmware size +
Erase application

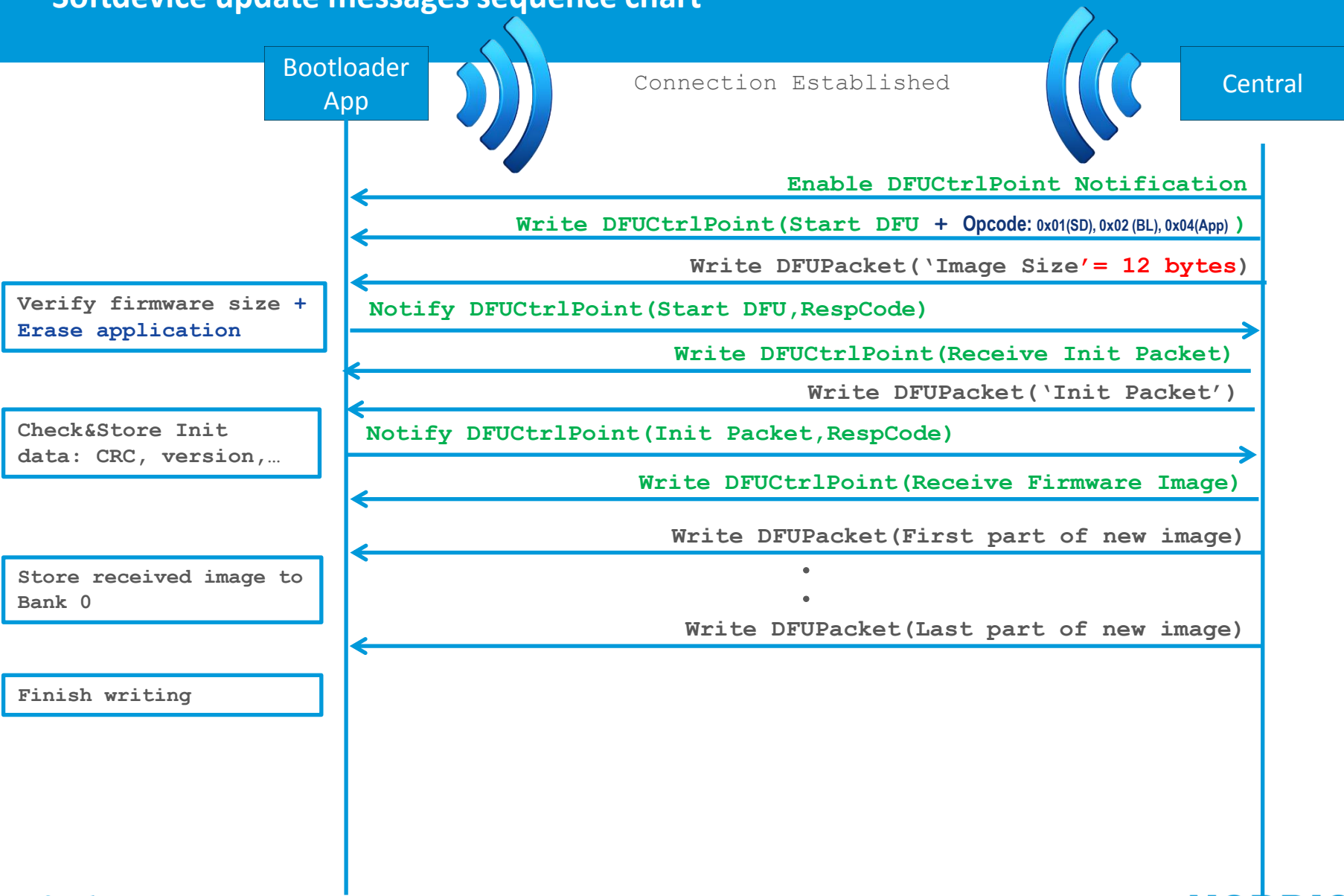
Check&Store Init
data: CRC, version,...

Store received image to
Bank 0

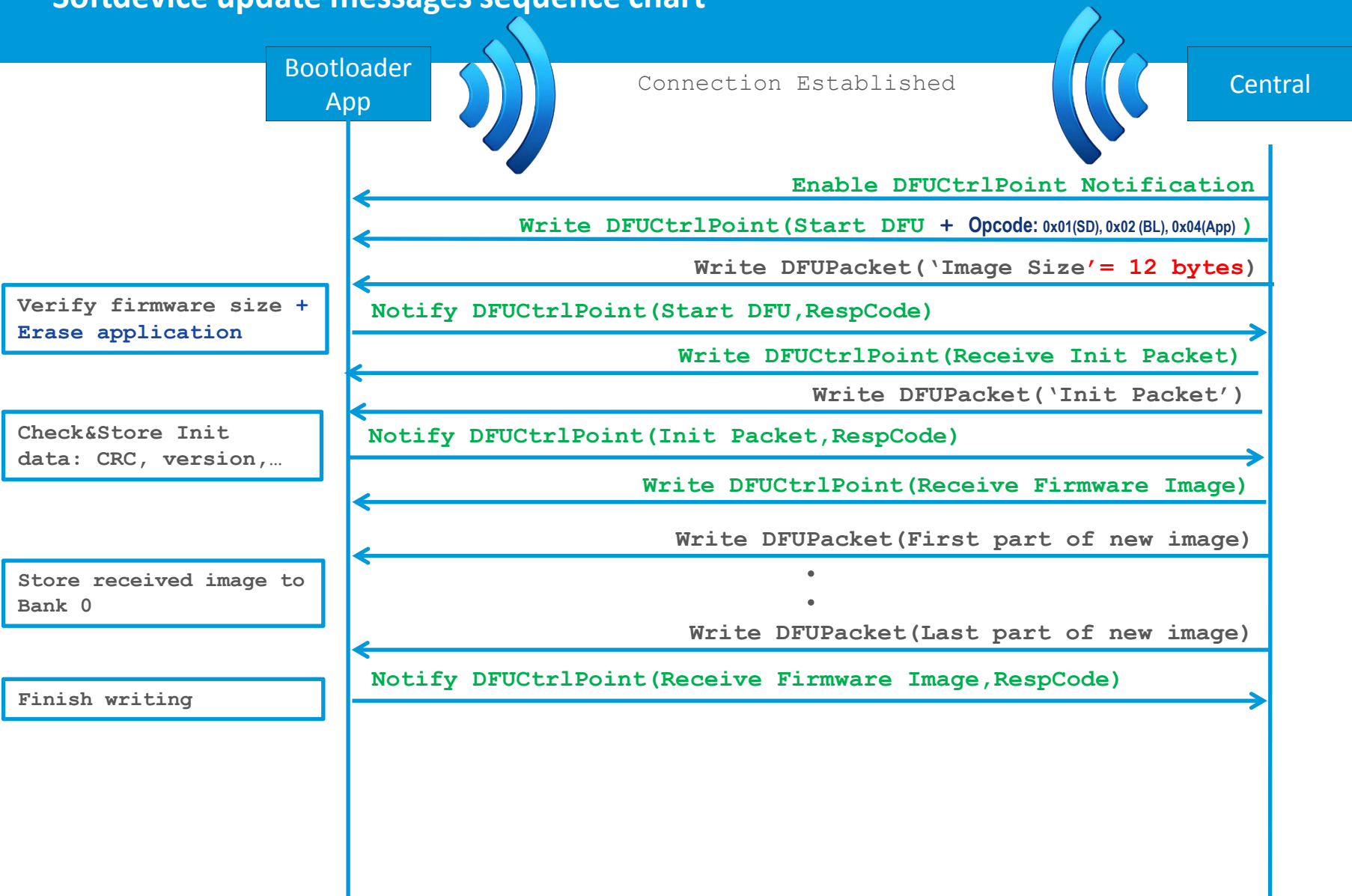
Softdevice update messages sequence chart



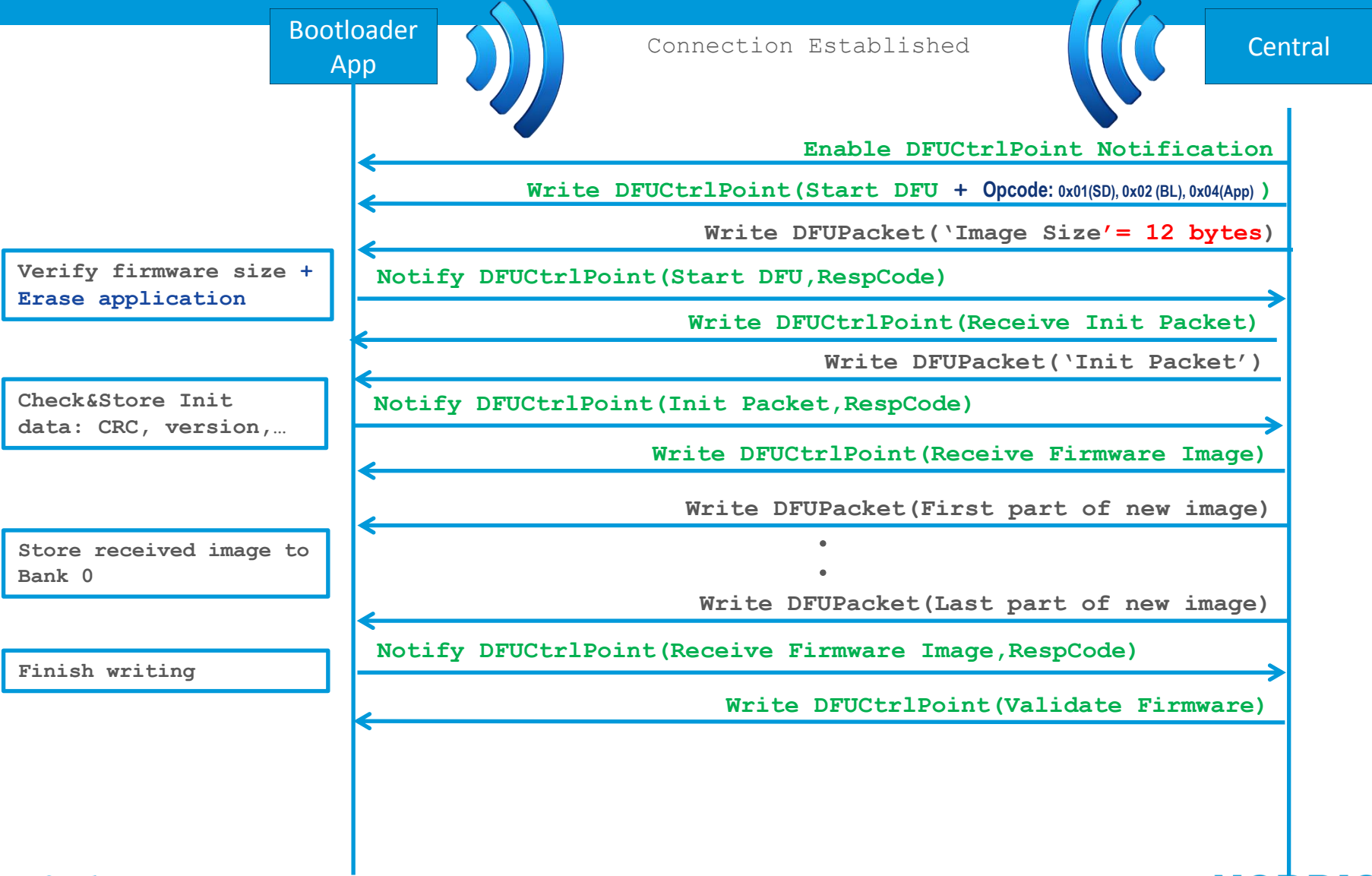
Softdevice update messages sequence chart



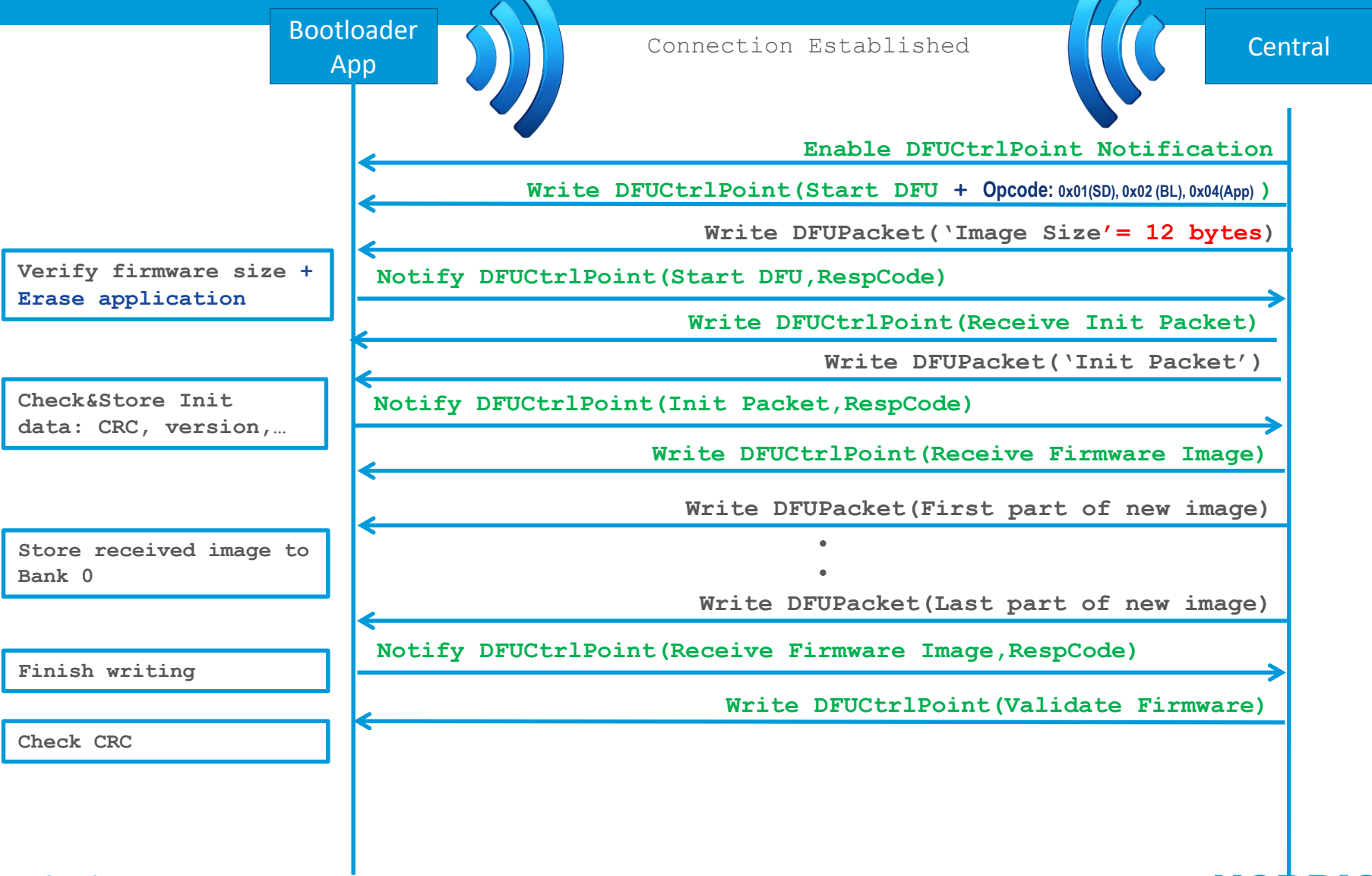
Softdevice update messages sequence chart



Softdevice update messages sequence chart



Softdevice update messages sequence chart



Verify firmware size + Erase application

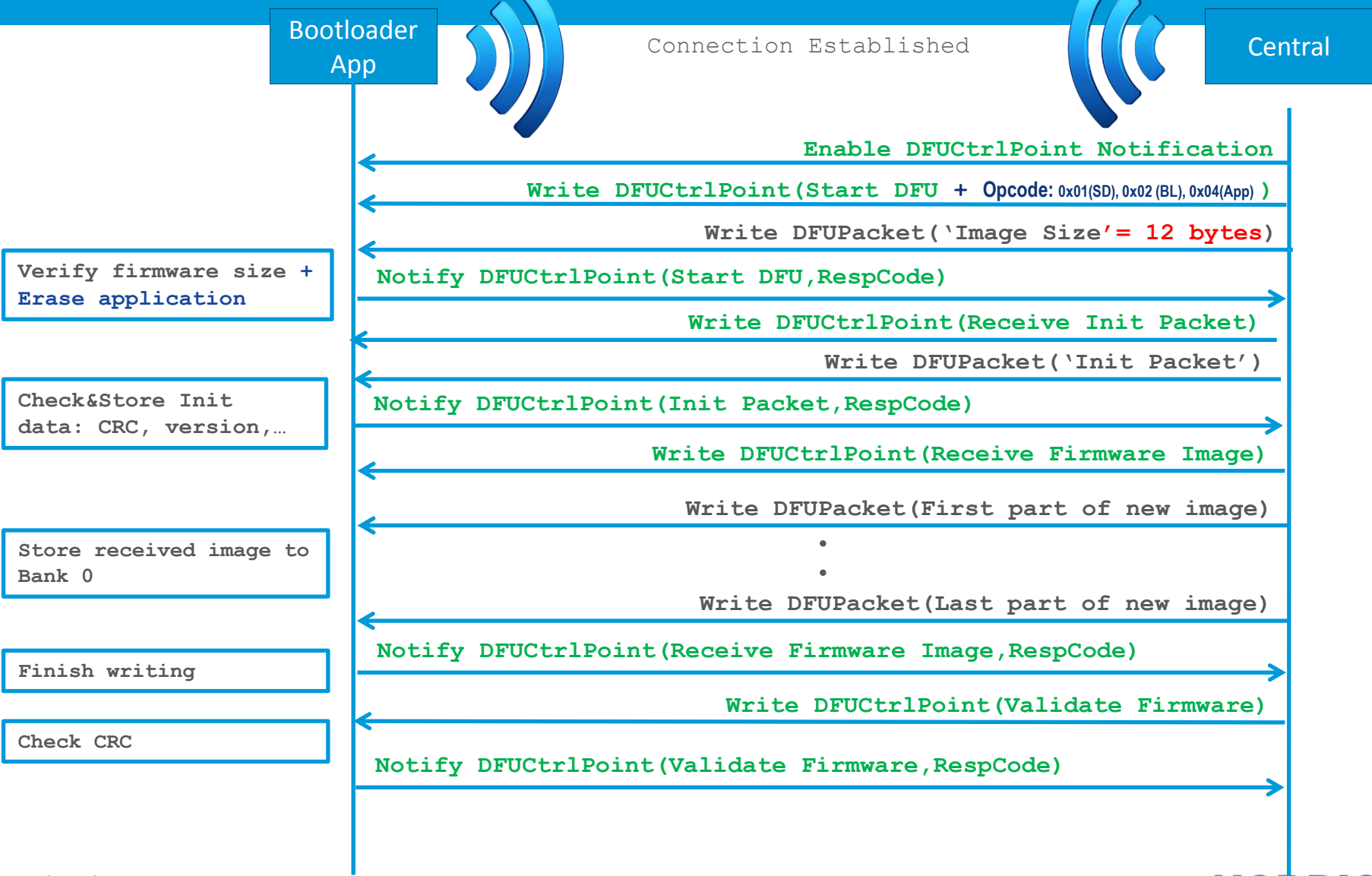
Check & Store Init data: CRC, version, ...

Store received image to Bank 0

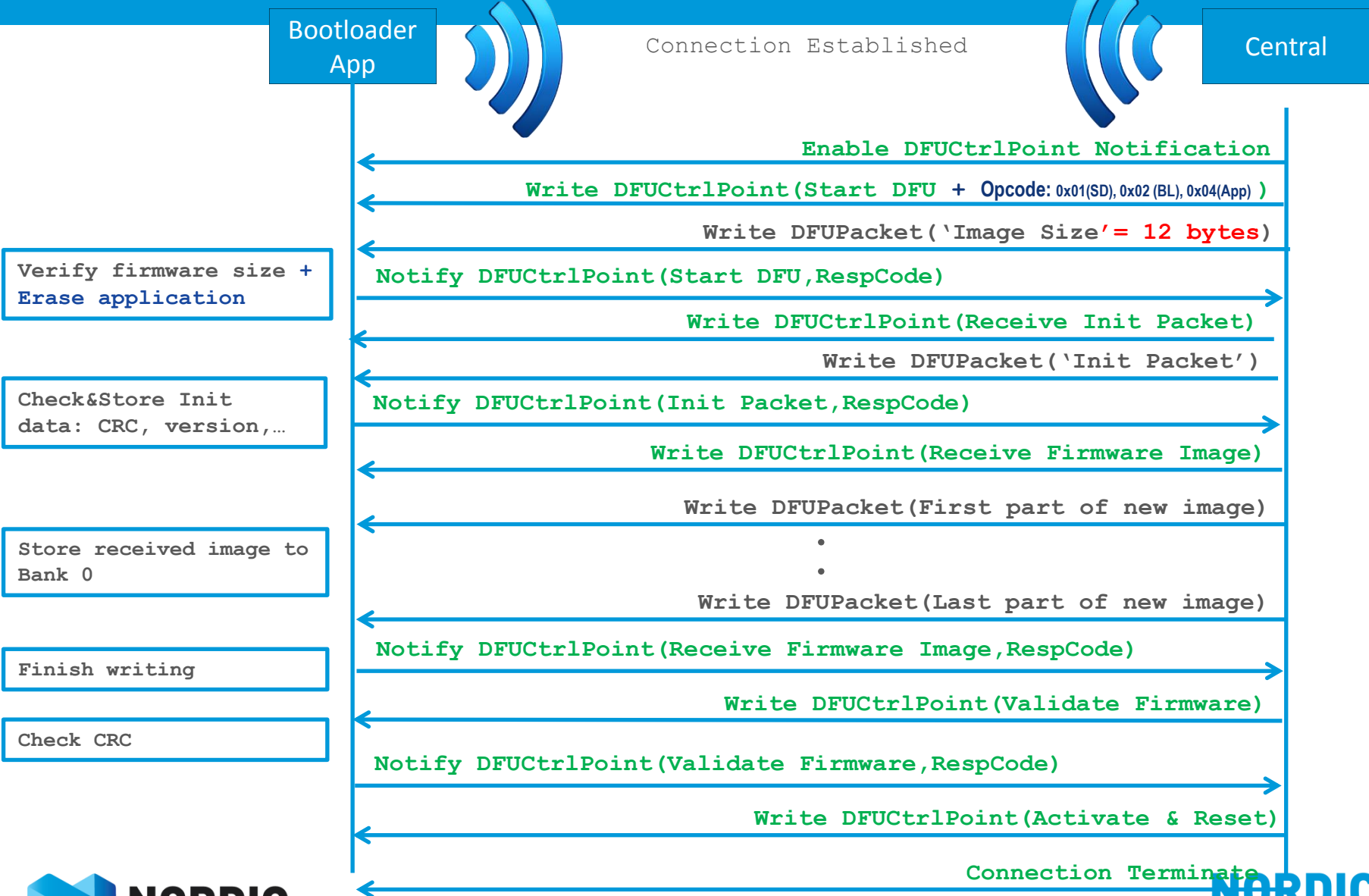
Finish writing

Check CRC

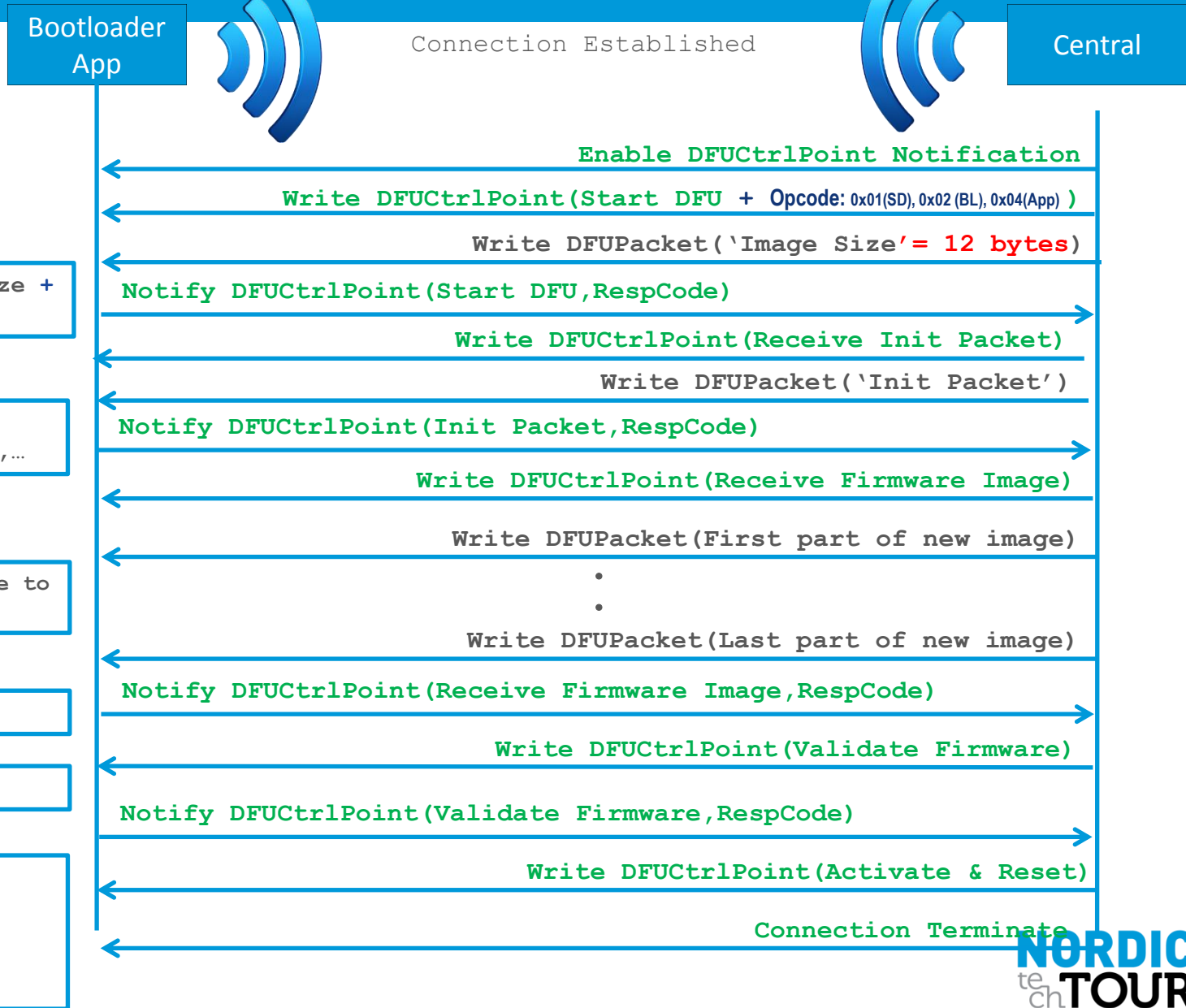
Softdevice update messages sequence chart



Softdevice update messages sequence chart



Softdevice update messages sequence chart



NORDIC tech TOUR

*nRF51 SoC Device Firmware Update
(DFU)*