Nordic Tech support

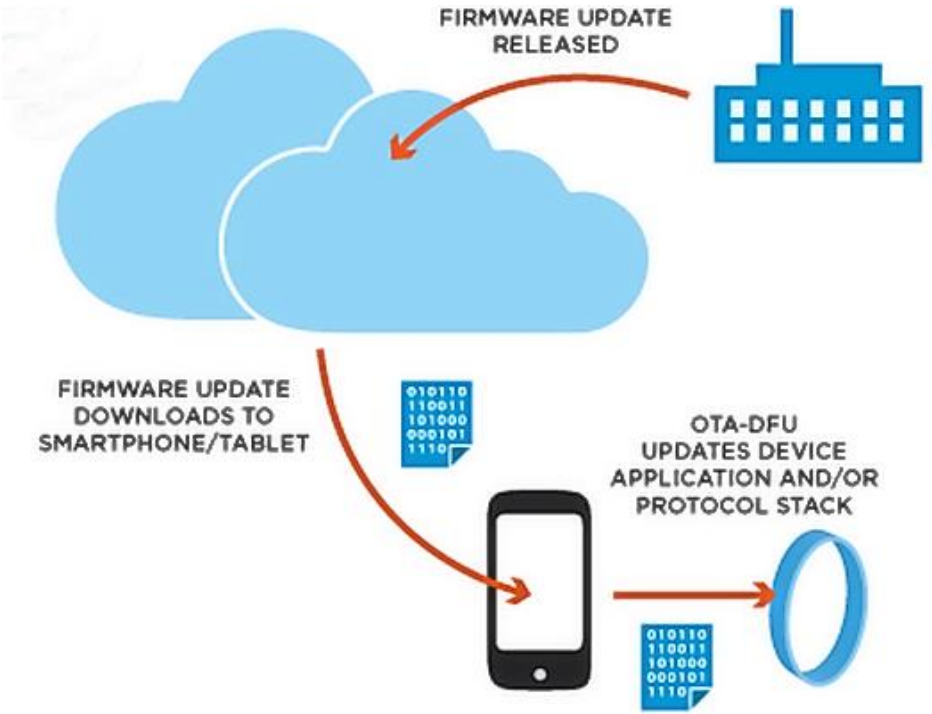
October, 2014

nRF51 SoC Device Firmware Update (DFU)

NORDIC SEMICONDUCTOR

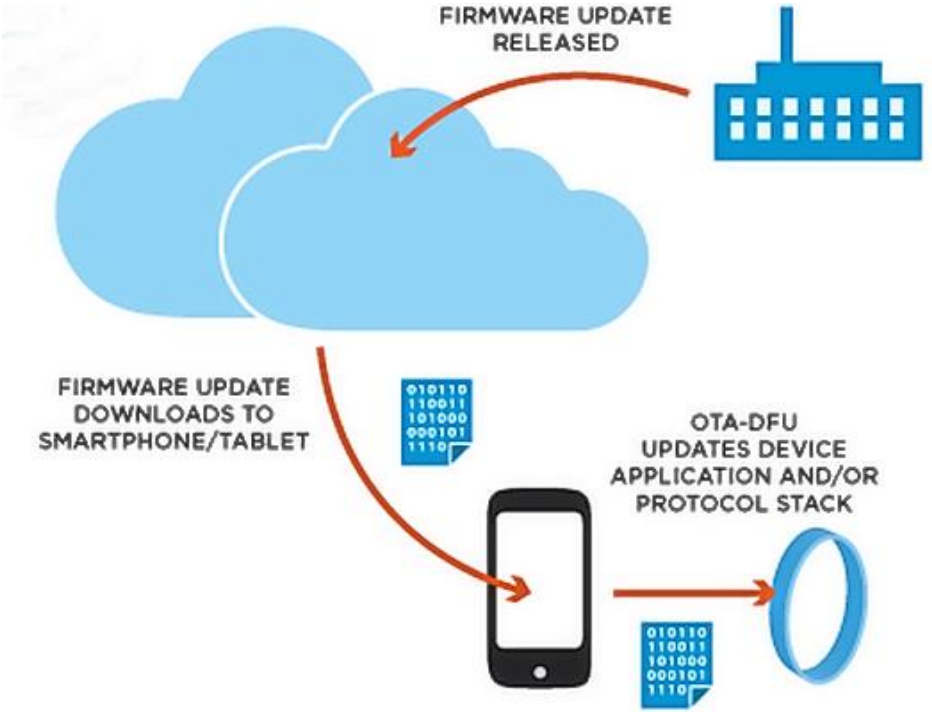
Device Firmware Updates (DFU)

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

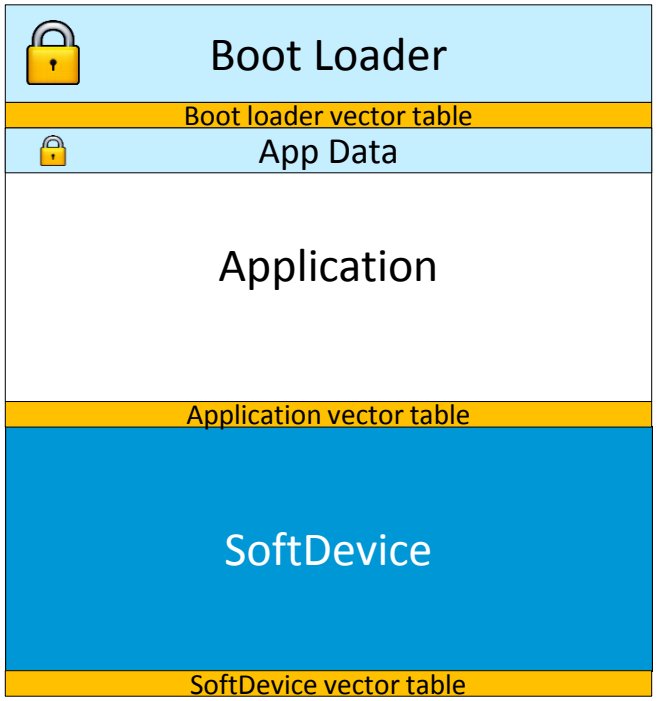


Device Firmware Updates (DFU)

- Nordic have been 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, in SDK v5.0
- Bluetooth stack (SoftDevice) update, released Aug 2014, in SDK v6.1
- Continuous development and improving

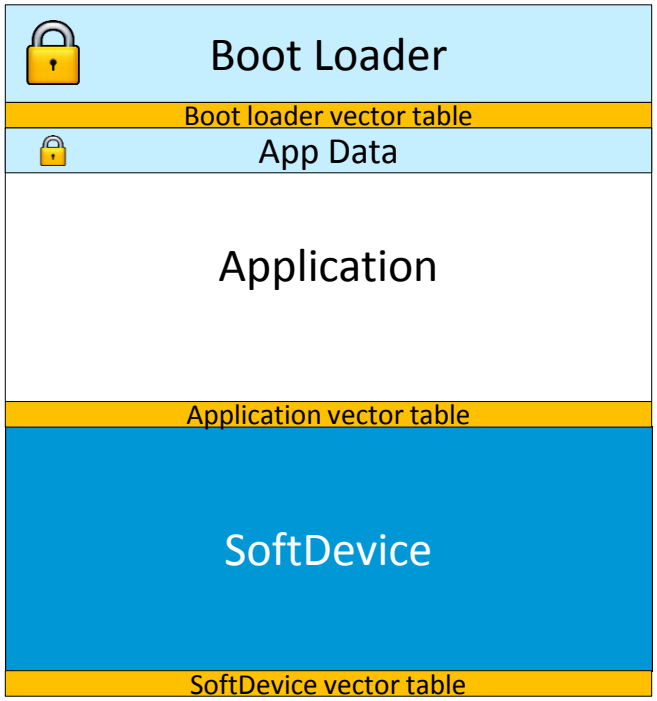


Boot Loader



- A small program that's always executed first when device booting up.
  - Write and erase protected (PROTENSET)
- Functions:
  - Controls 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 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 file size = application
    - SoftDevice only
      - DFU file size = SoftDevice + application

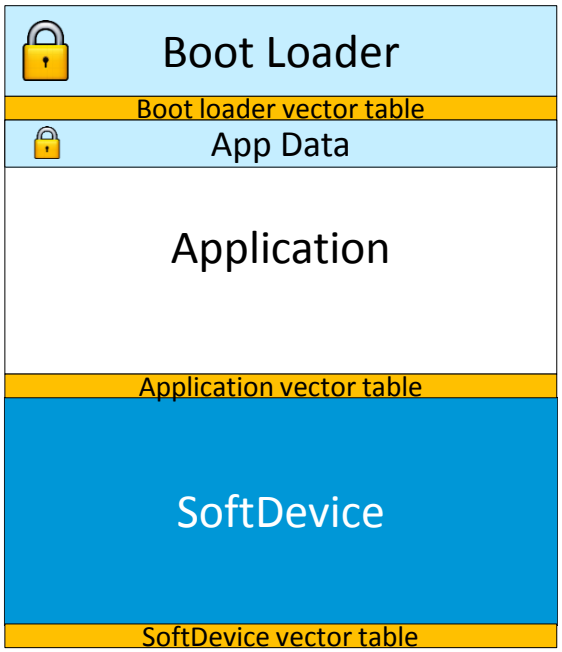
Application Update

(Supported from S110 4.4 and later)

Two methods of update

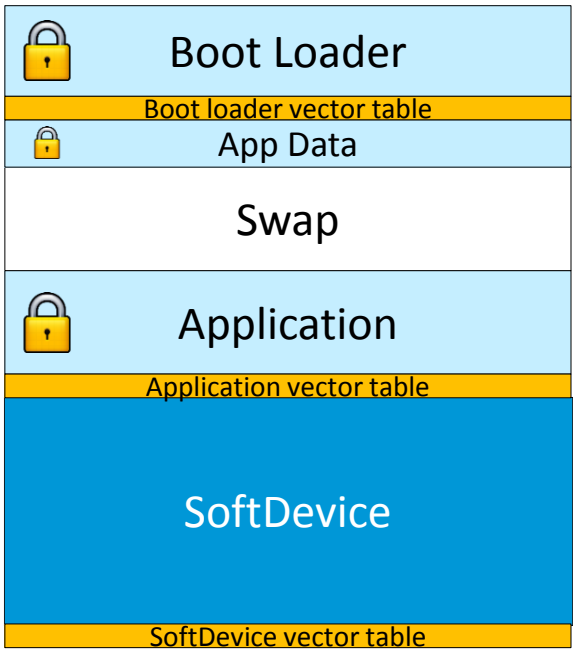
Single Bank Update

Minimum code space overhead



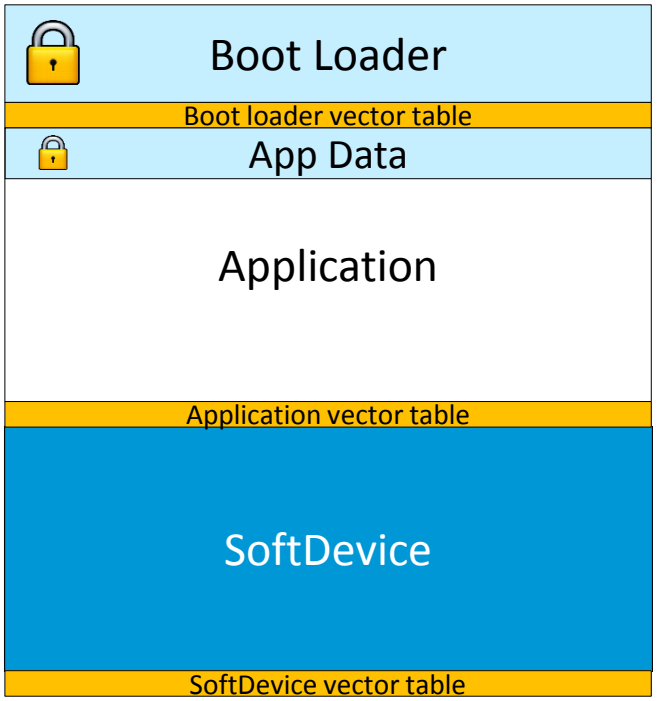
Dual Bank update

Maximum safety



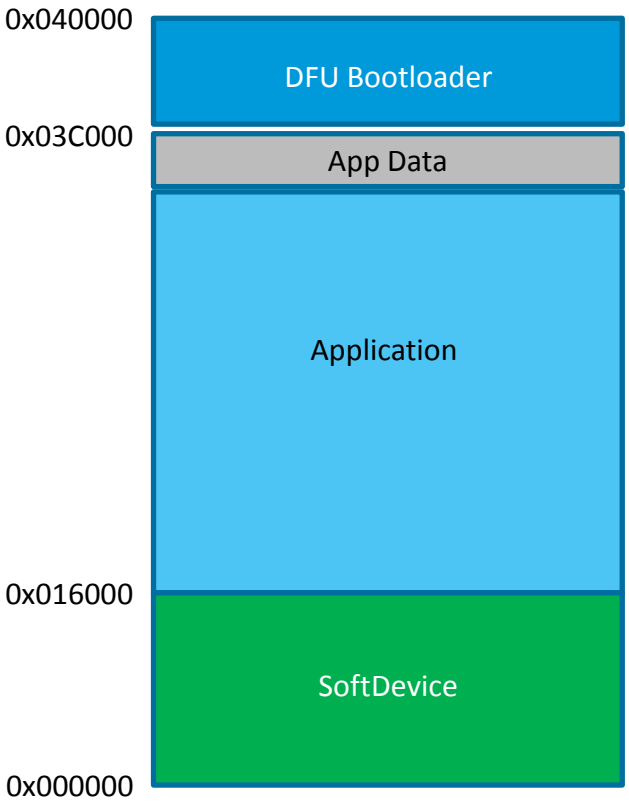
Single Bank update

Application



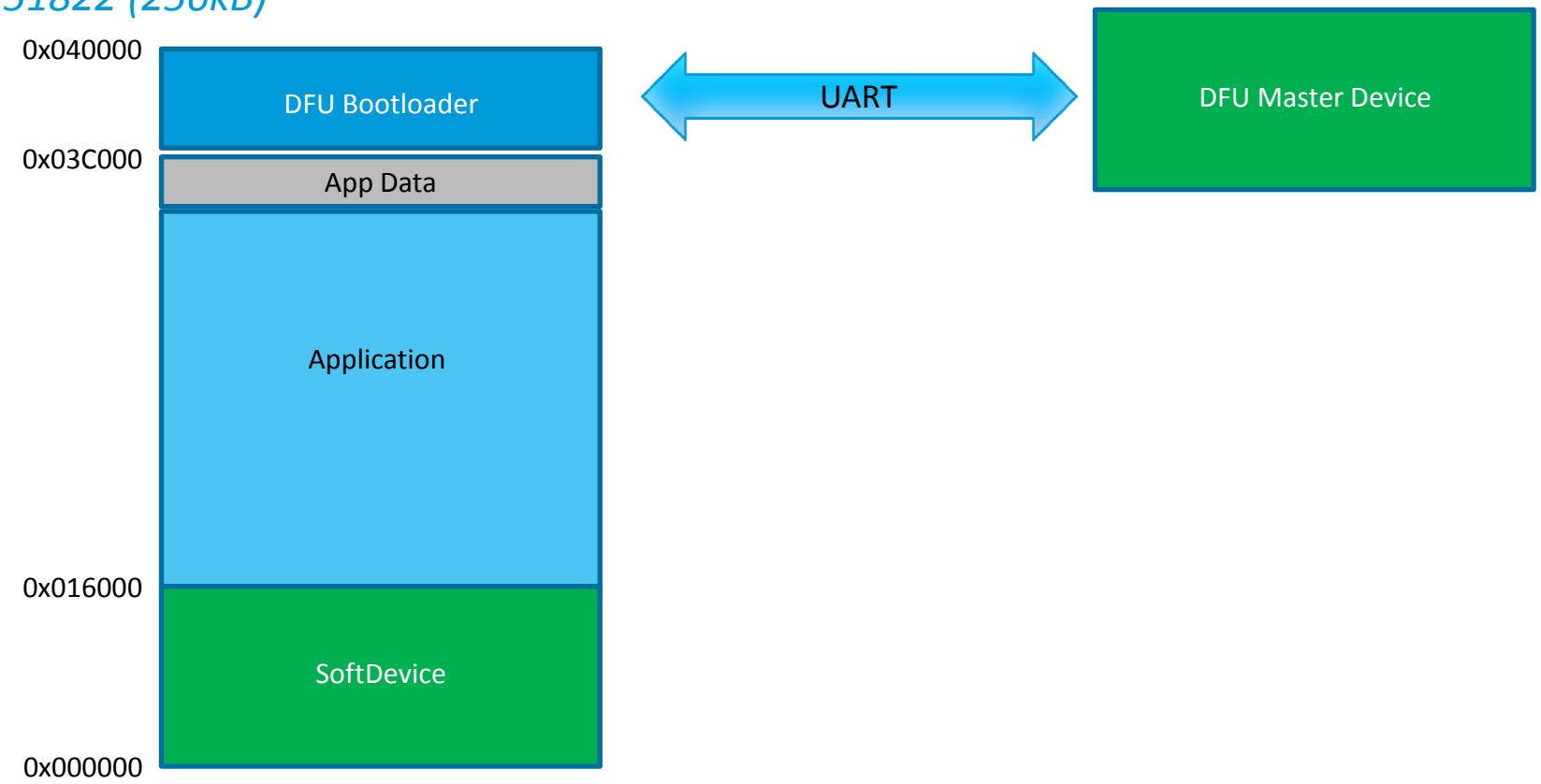
Application firmware single bank update

nRF51822 (256kB)



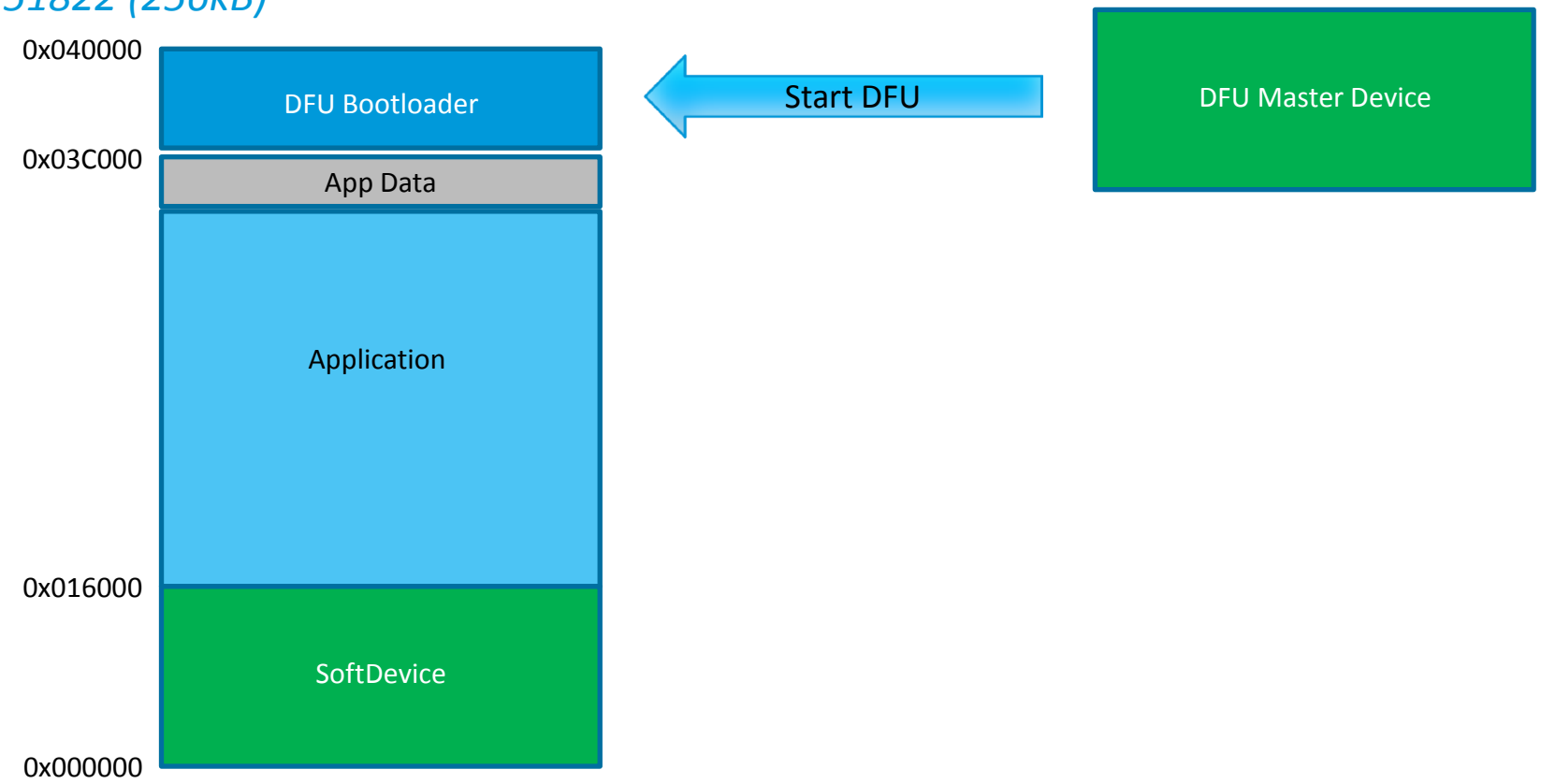
Application firmware single bank update

nRF51822 (256kB)



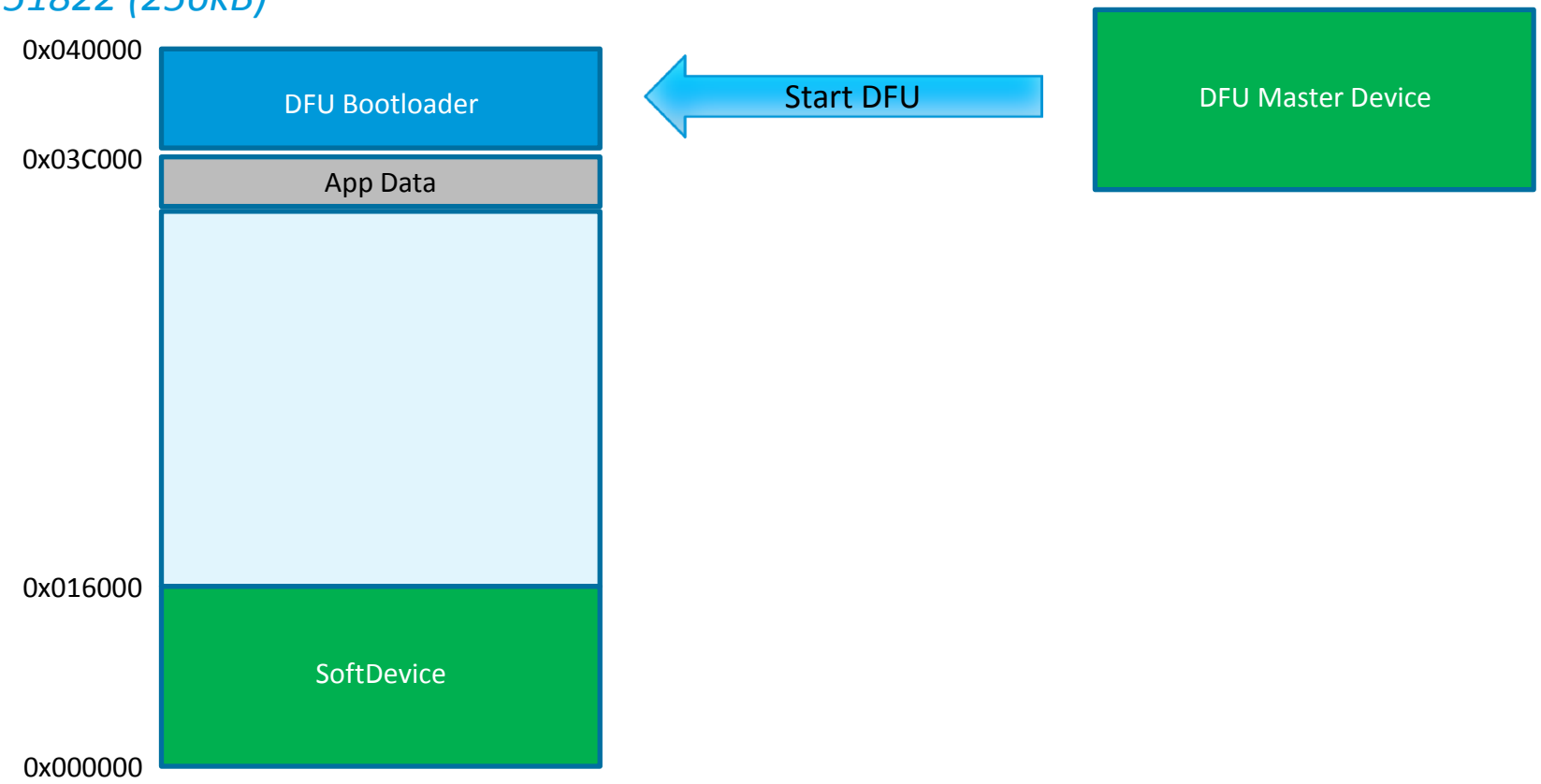
Application firmware single bank update

nRF51822 (256kB)



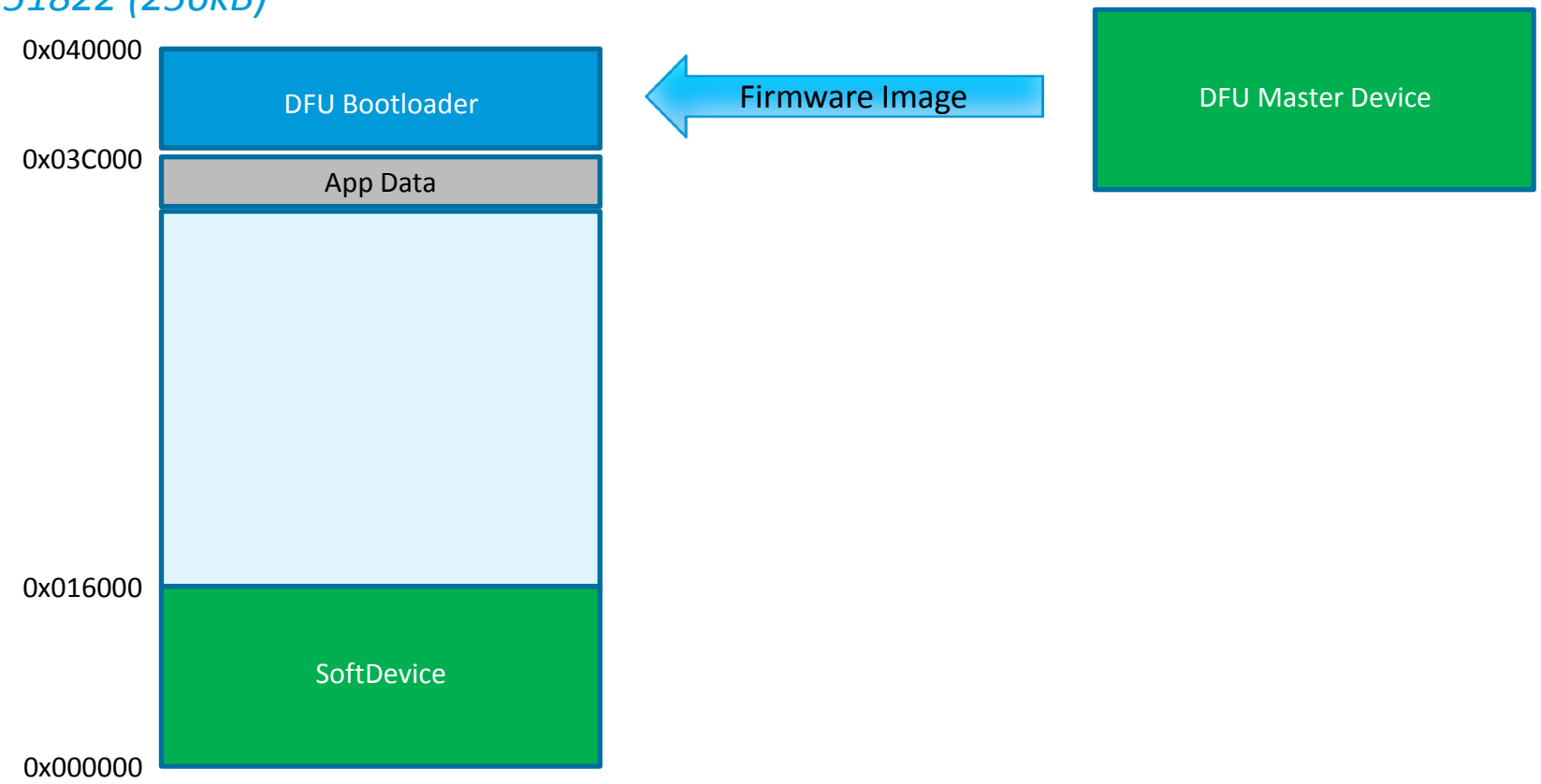
Application firmware single bank update

nRF51822 (256kB)



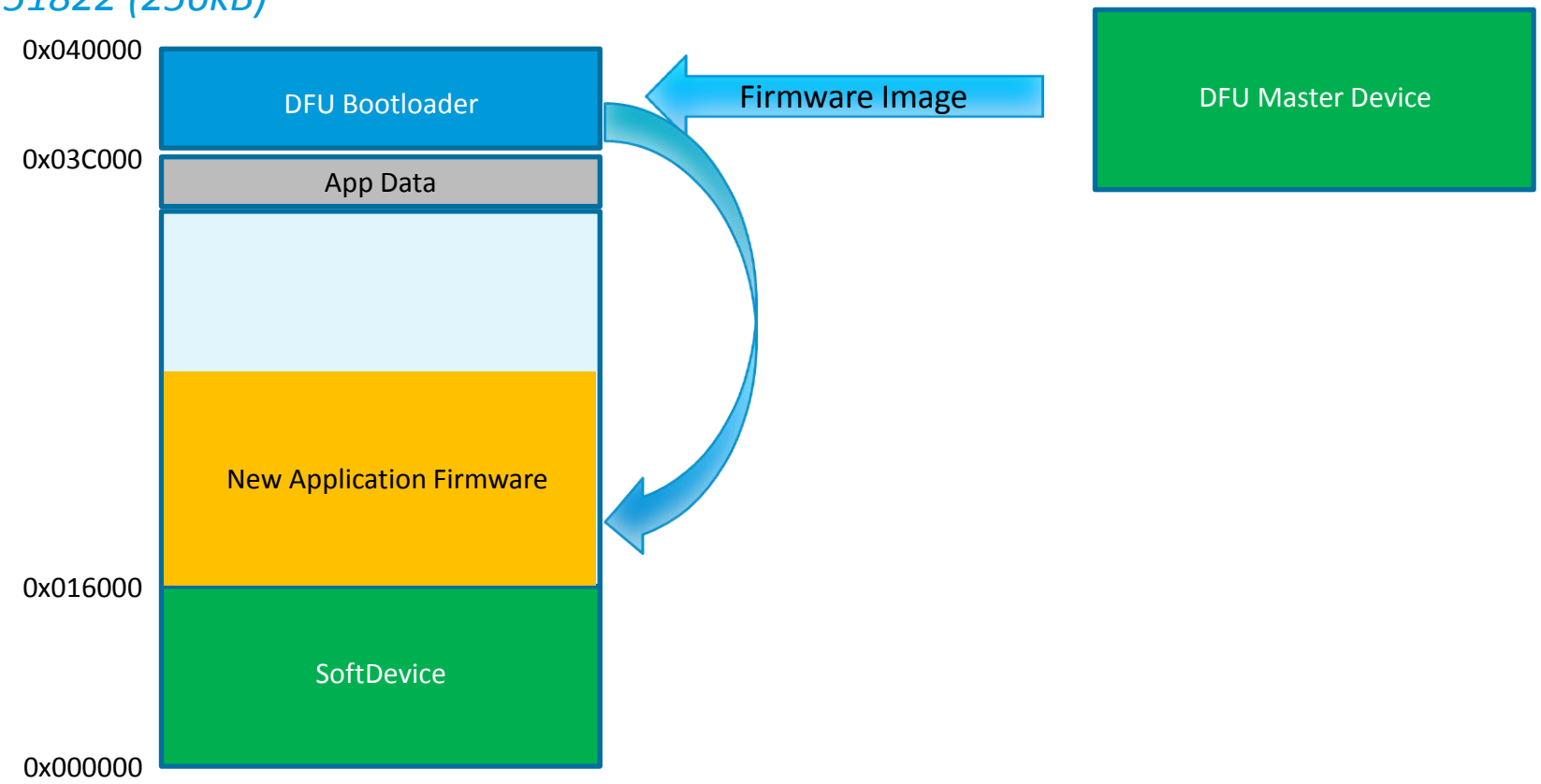
Application firmware single bank update

nRF51822 (256kB)



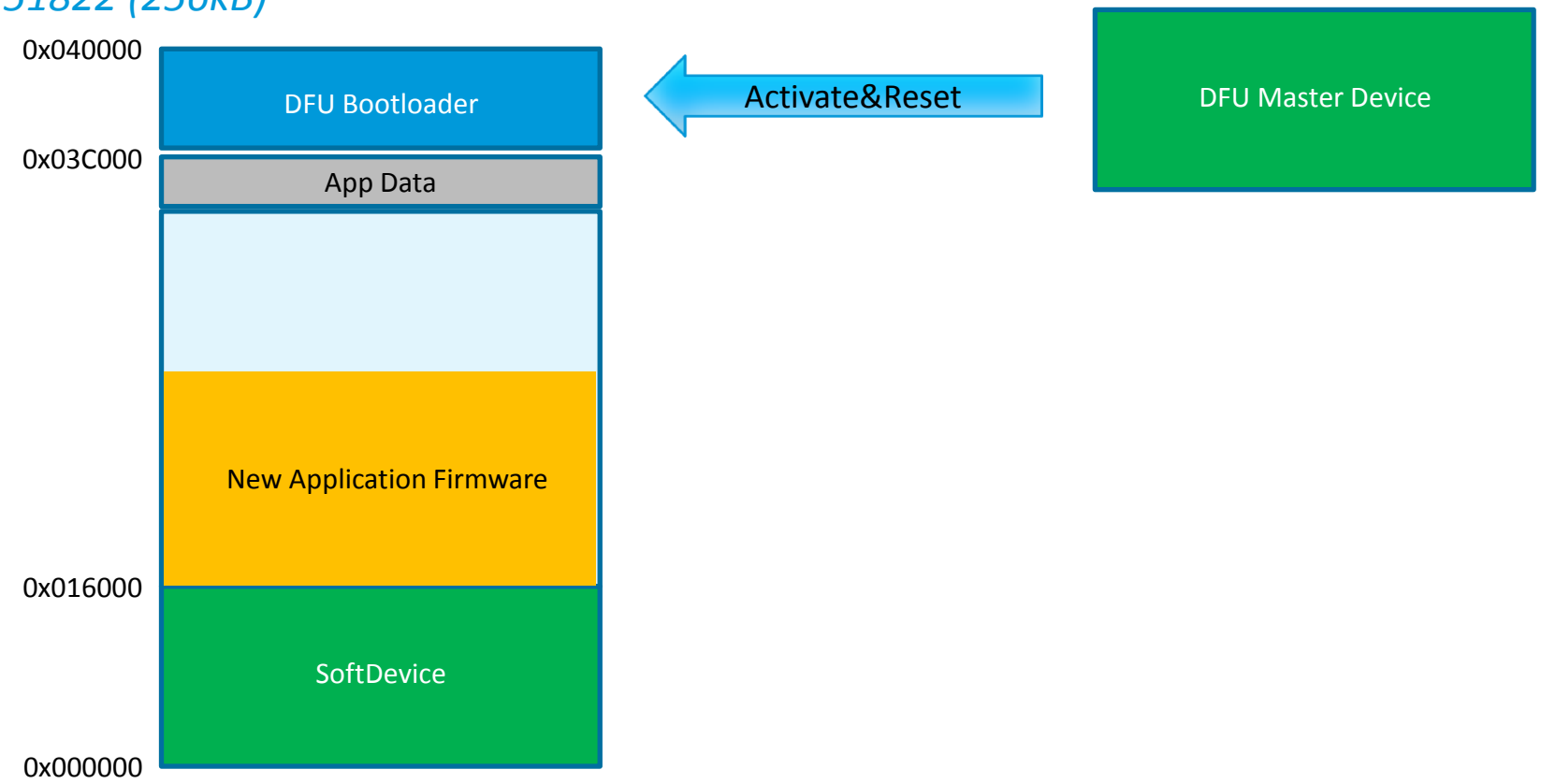
Application firmware single bank update

nRF51822 (256kB)



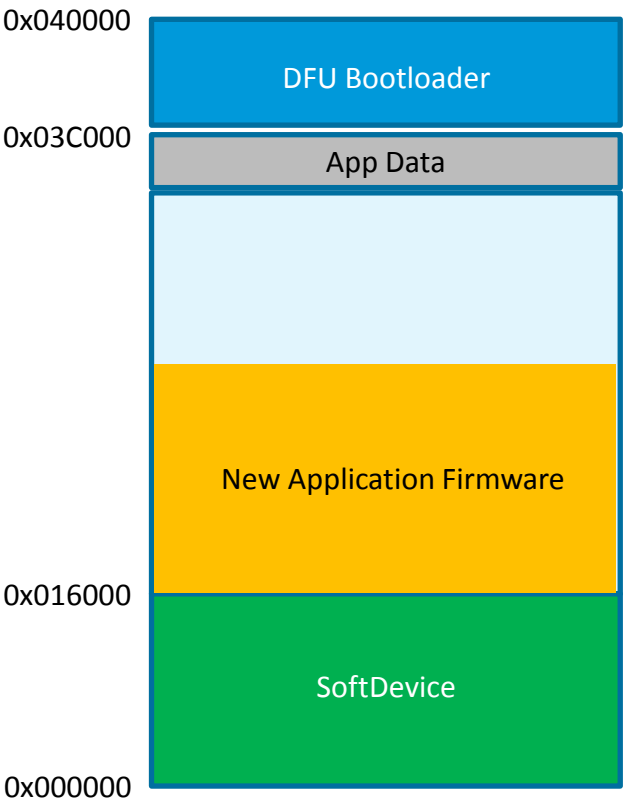
Application firmware single bank update

nRF51822 (256kB)



Application firmware single bank update

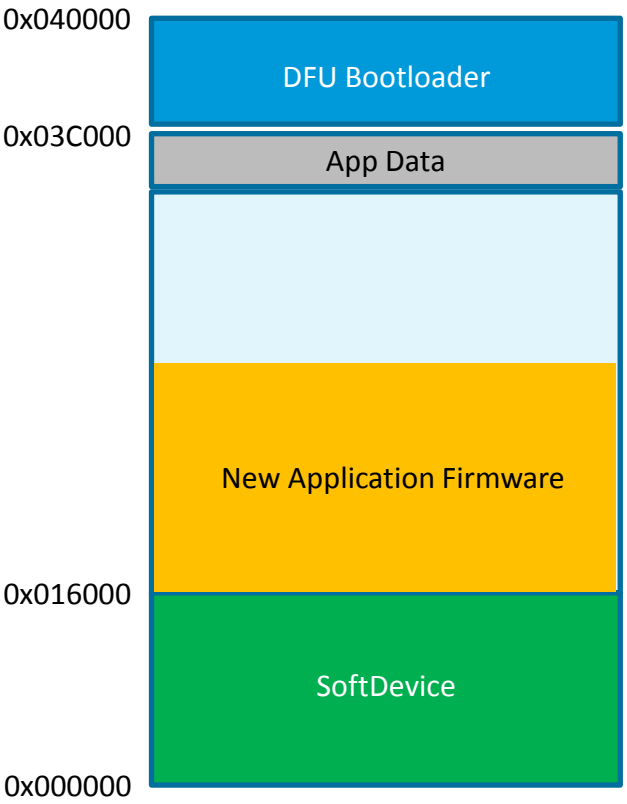
nRF51822 (256kB)



DFU Master Device

Application firmware single bank update

nRF51822 (256kB)

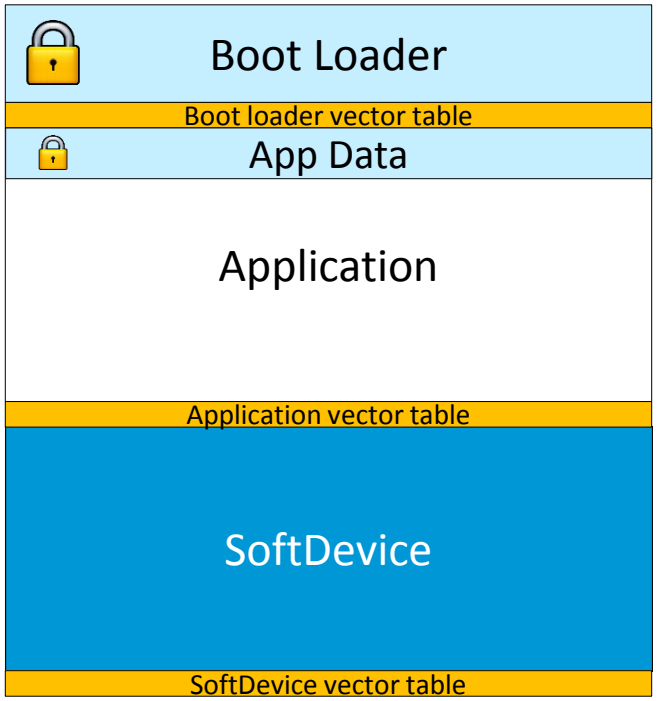


Memory Range*	Size	Usage
0x0003C000 - 0x00040000	16 kB	Bootloader
0x00016000 - 0x0003C000	152kB	Code Region 1: Application Code (BANK 0) and data
0x00000000 - 0x00016000	88 kB	Code Region 0: SoftDevice

*The memory layout based on the DFU example in SDK v7.1

Single Bank update

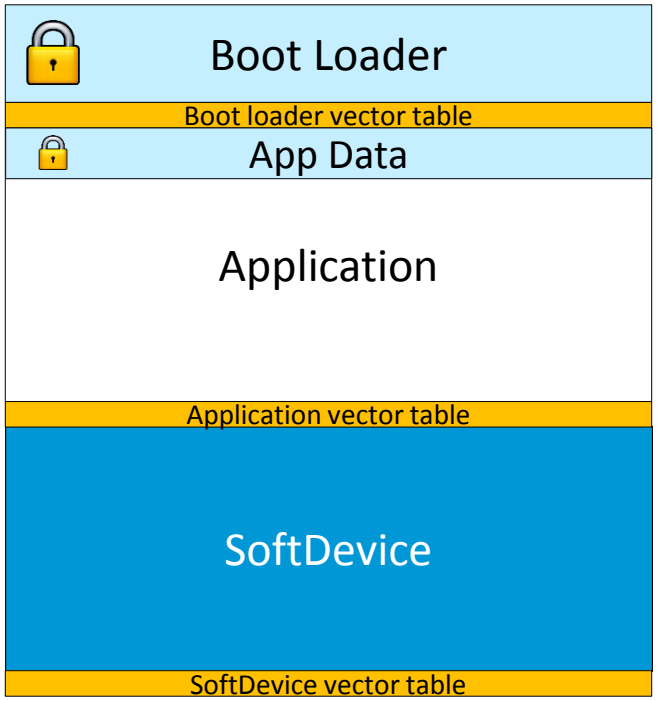
Application



- Single bank update:
  - Use Serial interface 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

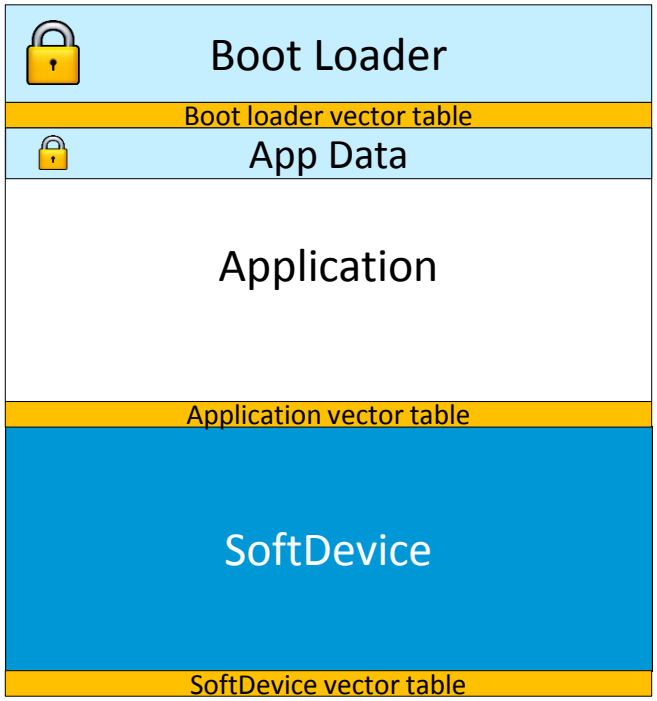


- Current single bank bootloader:
  - Using UART transportation
  - Supported from nRF51 SDK v4.4.0
  - Code size of bootloader 13kB
- Max application size:
  - App = Flash size – Softdevice - Bootloader
    - nRF51822 XXAA (256 kB): 155 kB*
    - nRF51822 QFAB (128 kB): 27 kB*

**Calculated base on S110 v7.0 size = 88kB*

Single Bank update

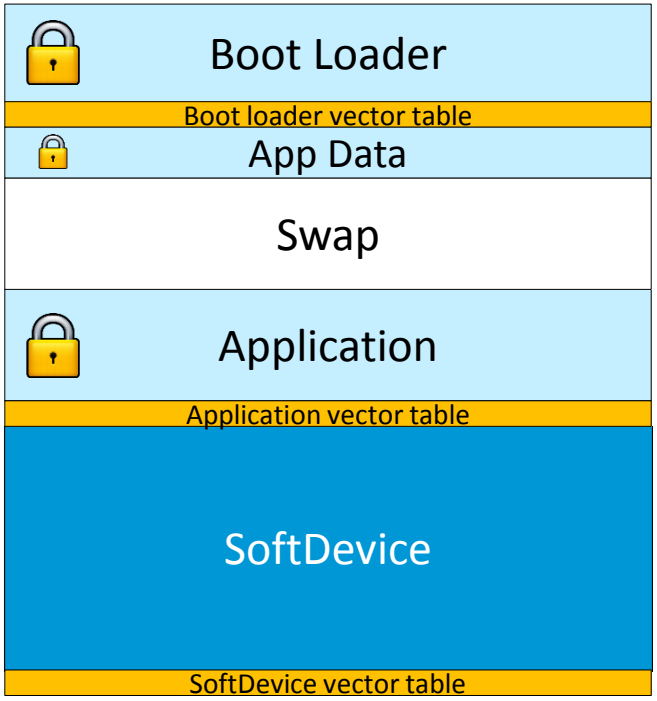
Performance



- 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

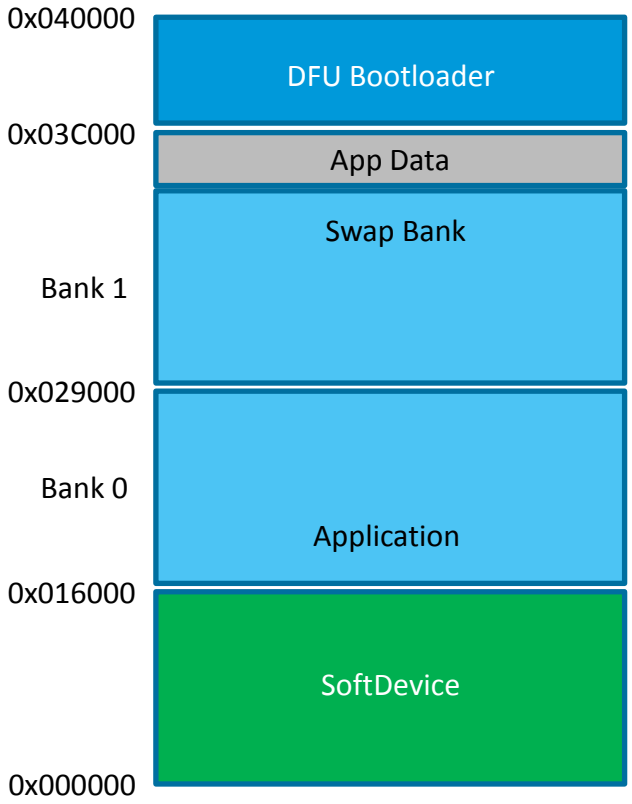
Dual Bank update

Application



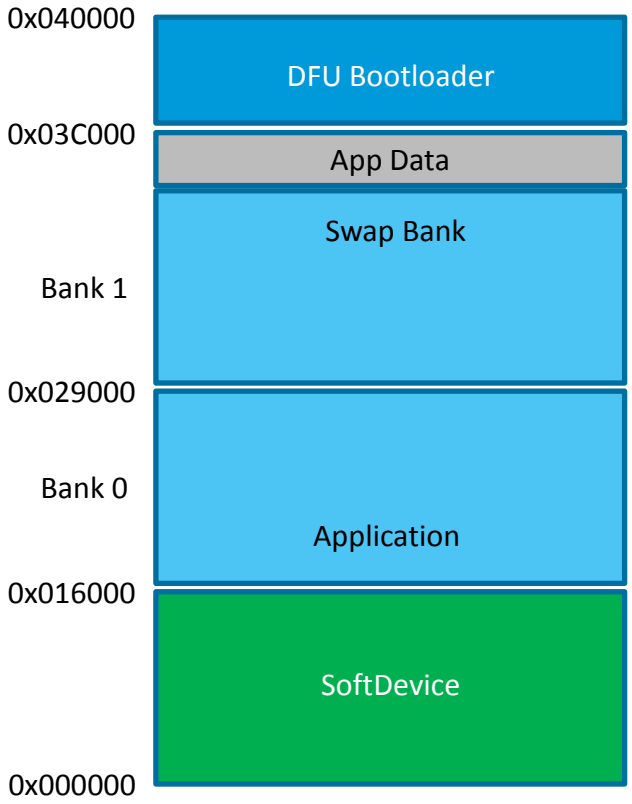
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



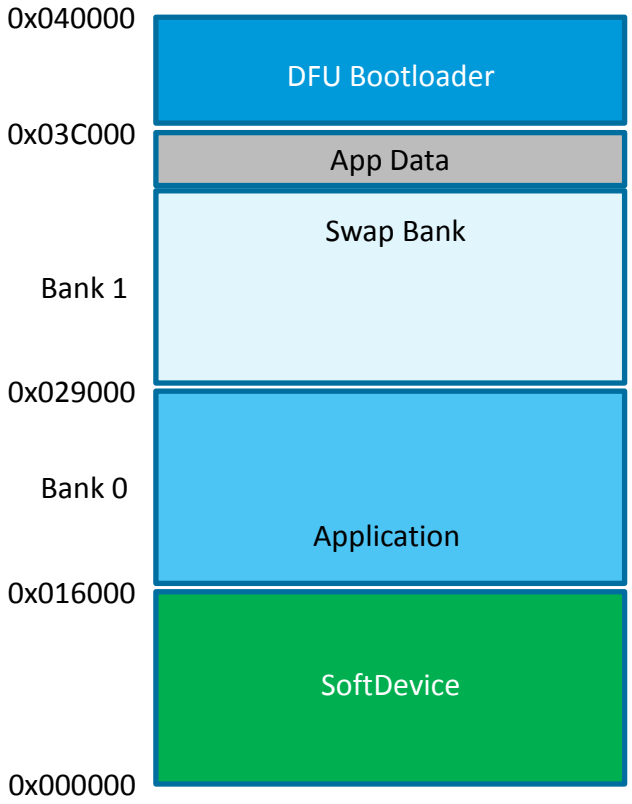
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



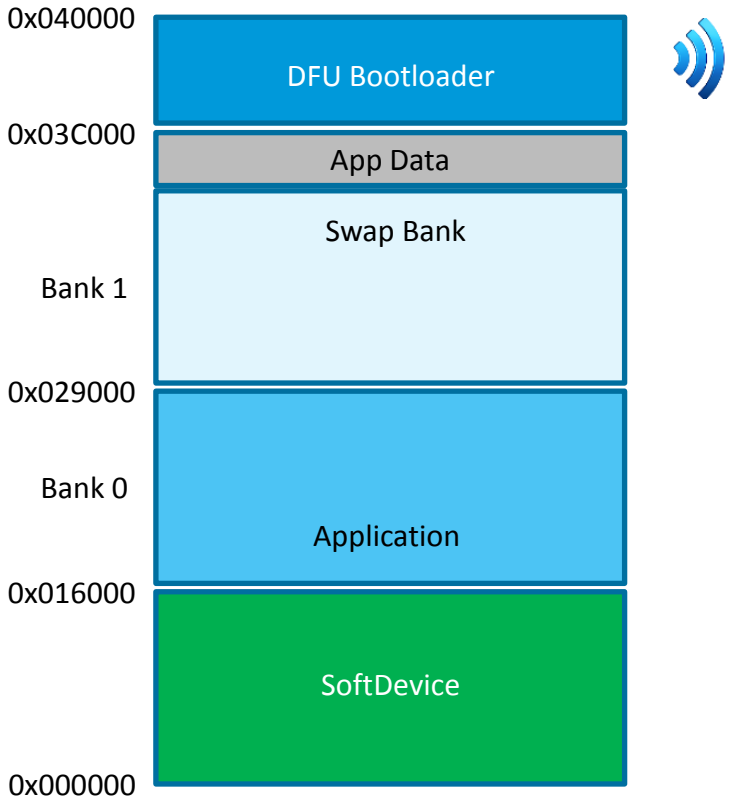
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



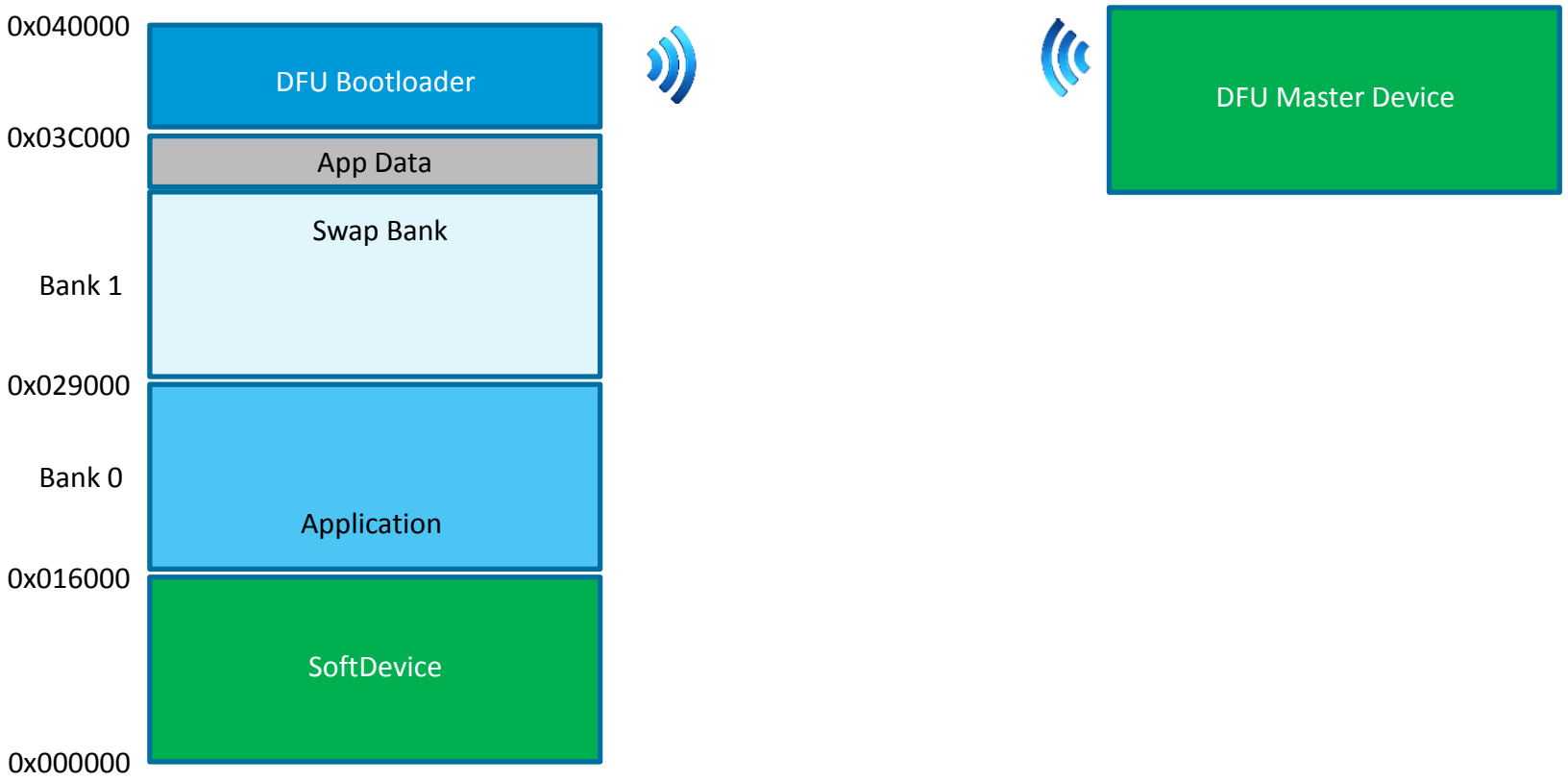
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



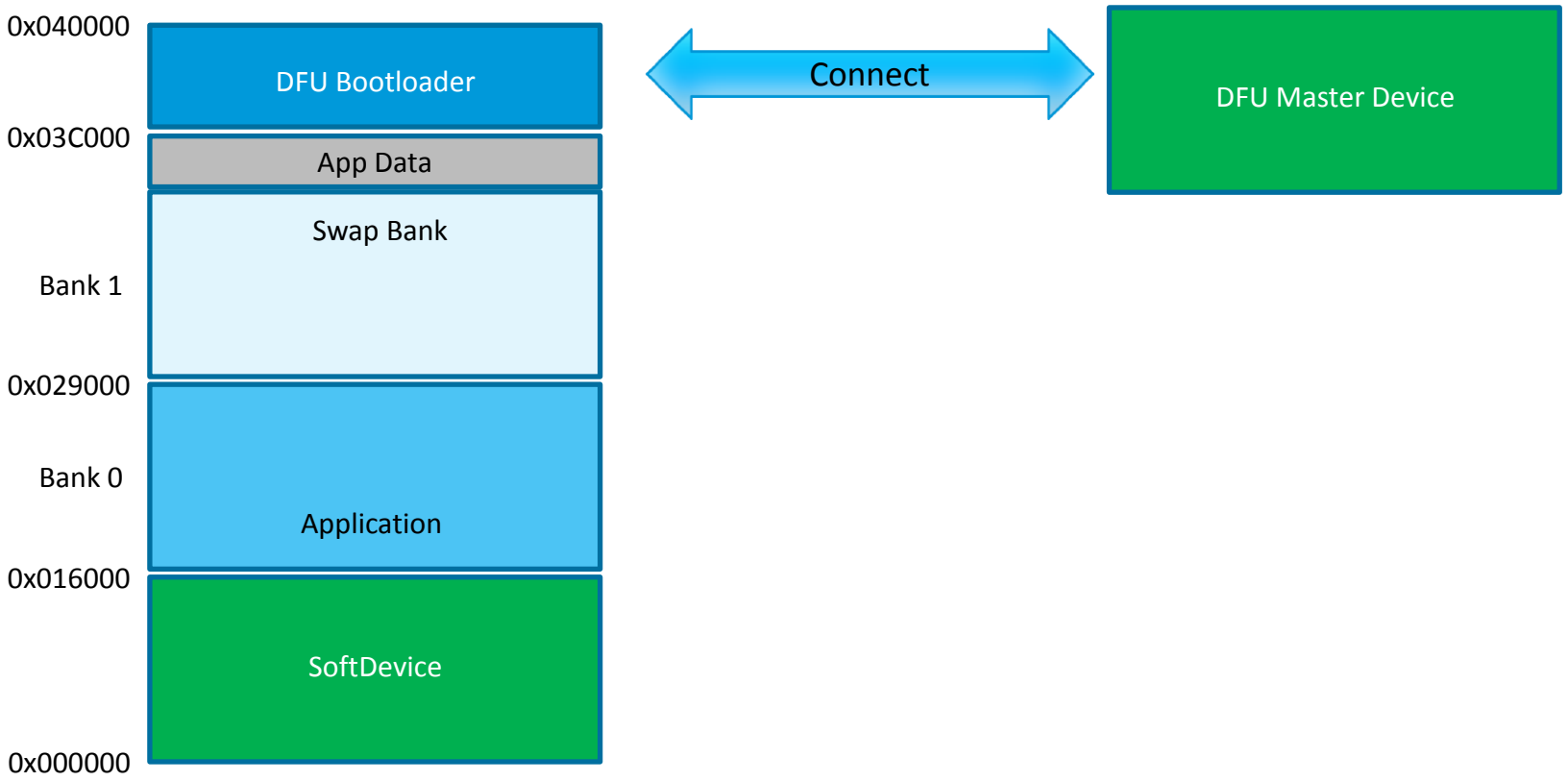
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



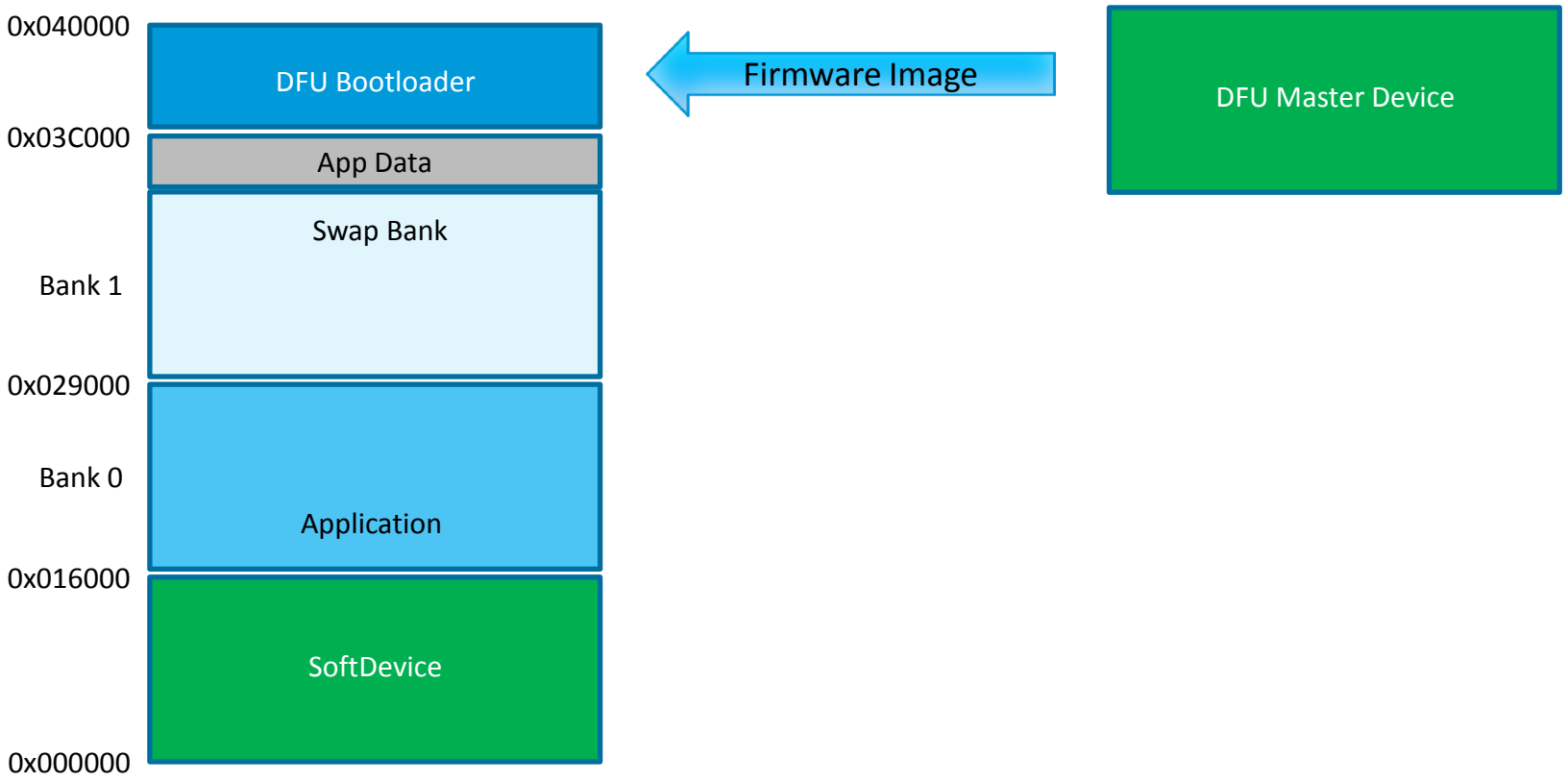
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



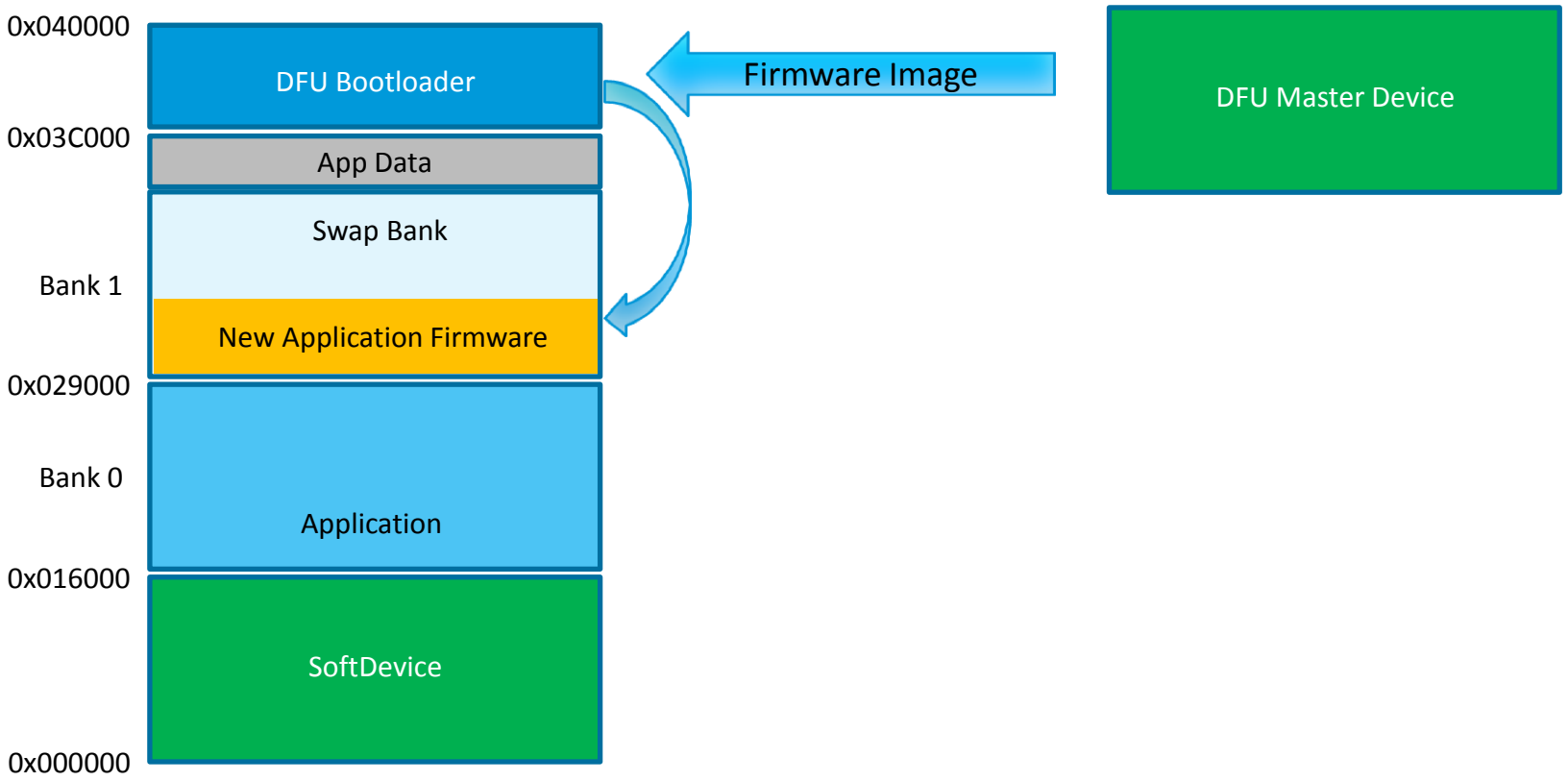
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



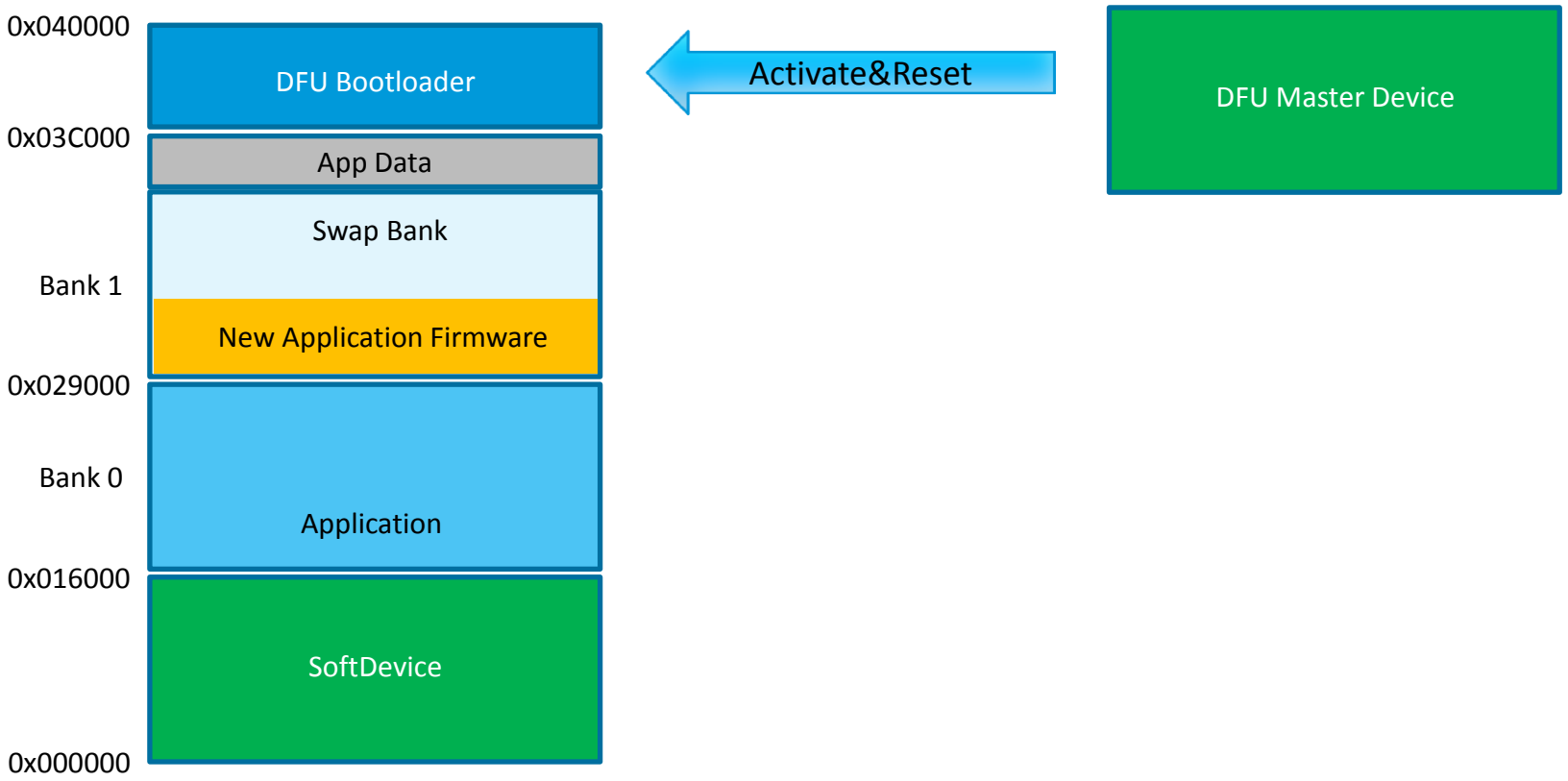
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



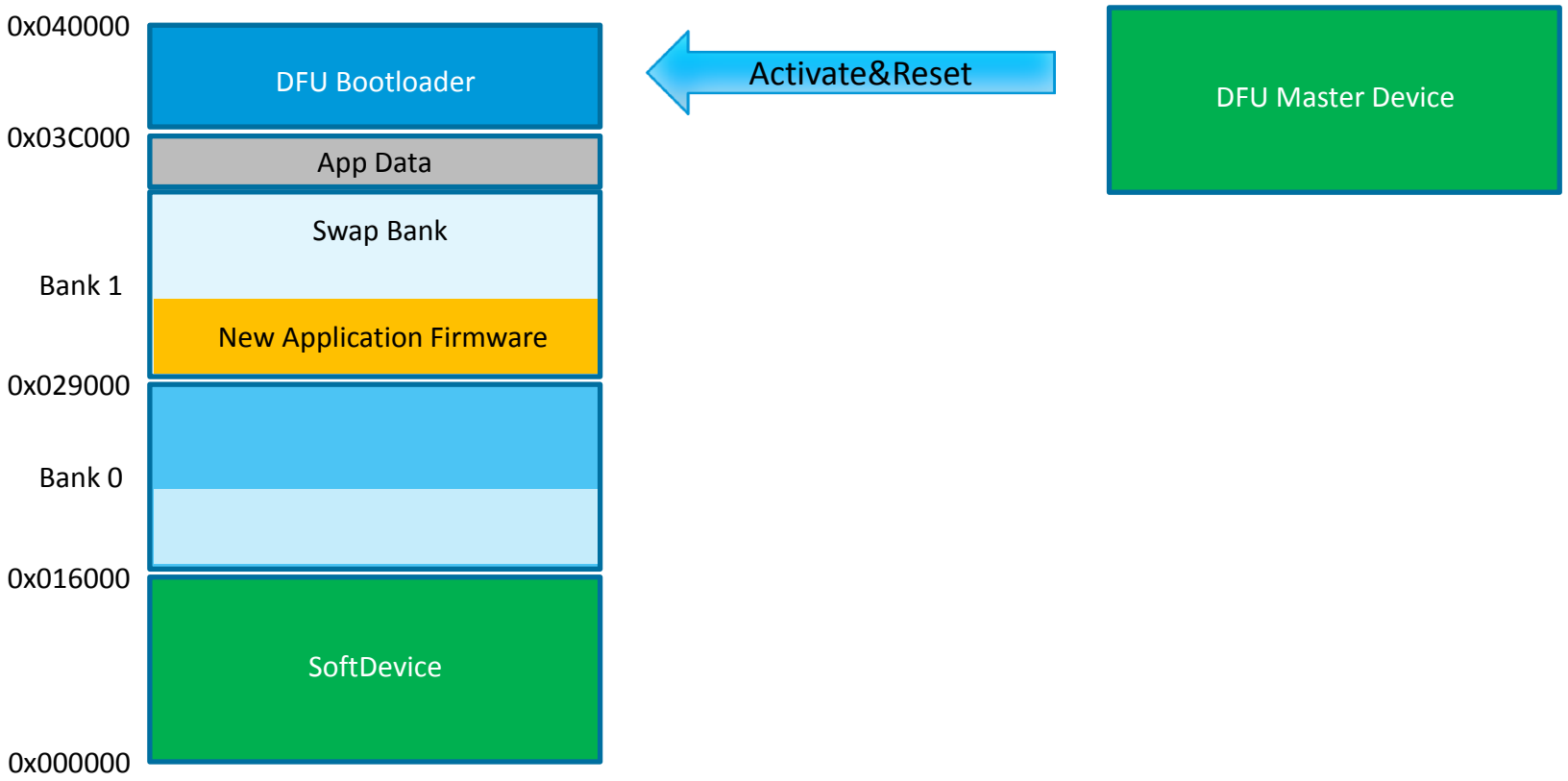
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



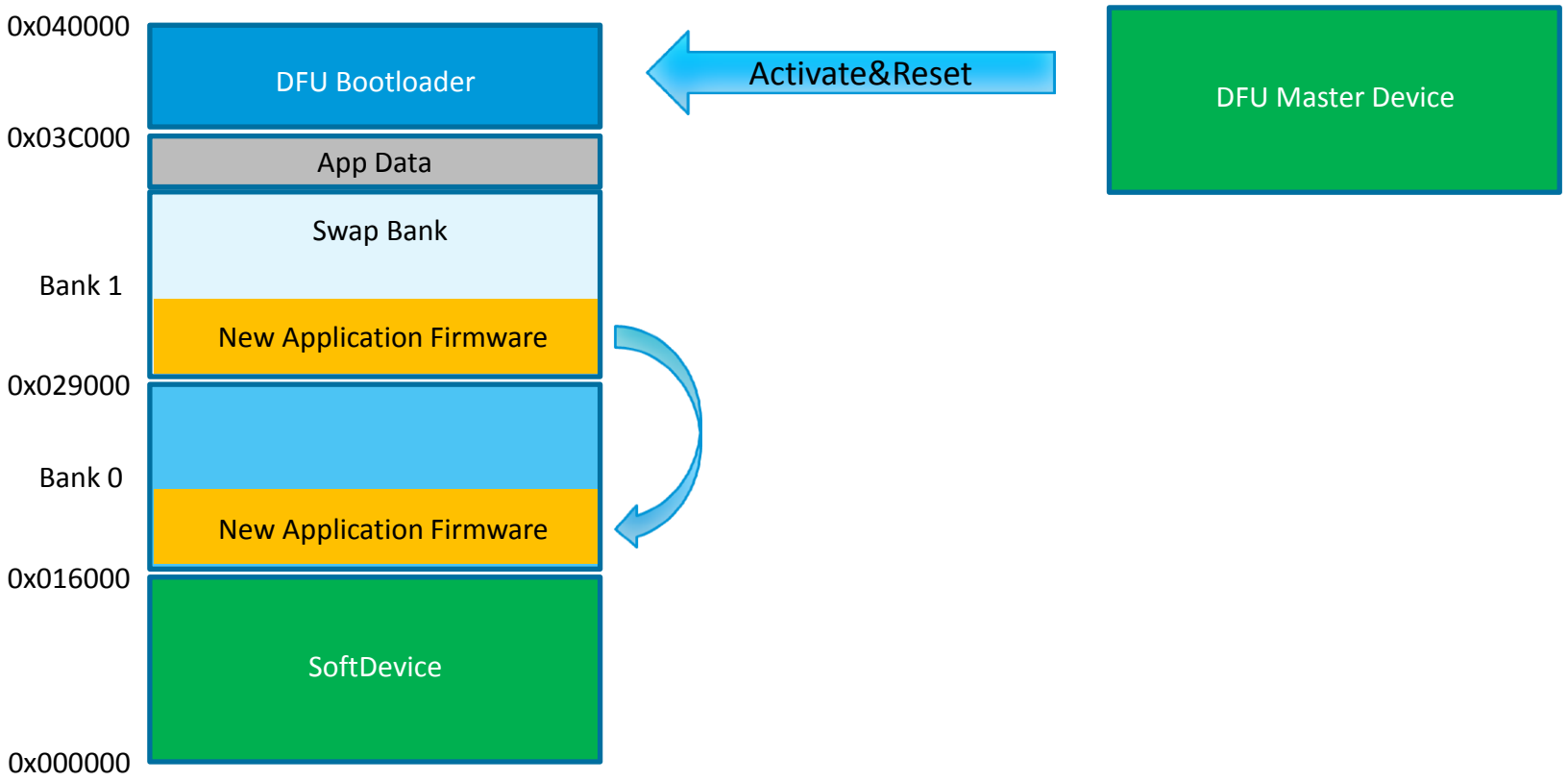
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



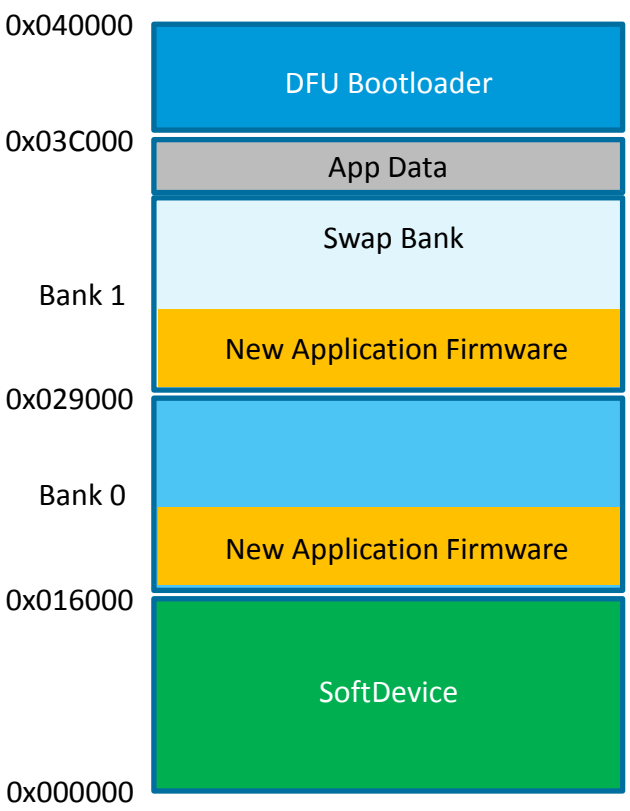
OTA Dual Bank update

Application firmware update on a 256kB nRF51822



OTA Dual Bank update

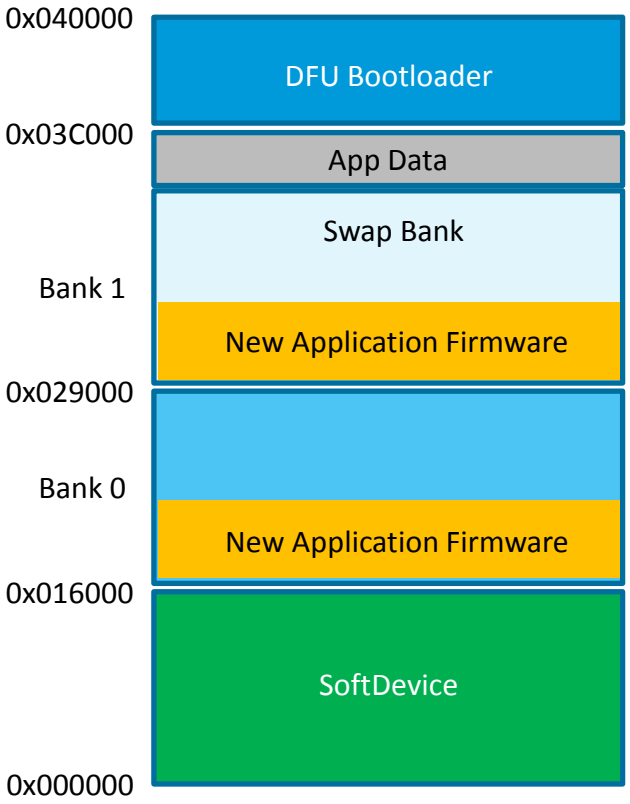
Application firmware update on a 256kB nRF51822



DFU Master Device

OTA Dual Bank update

Application firmware update on a 256kB nRF51822

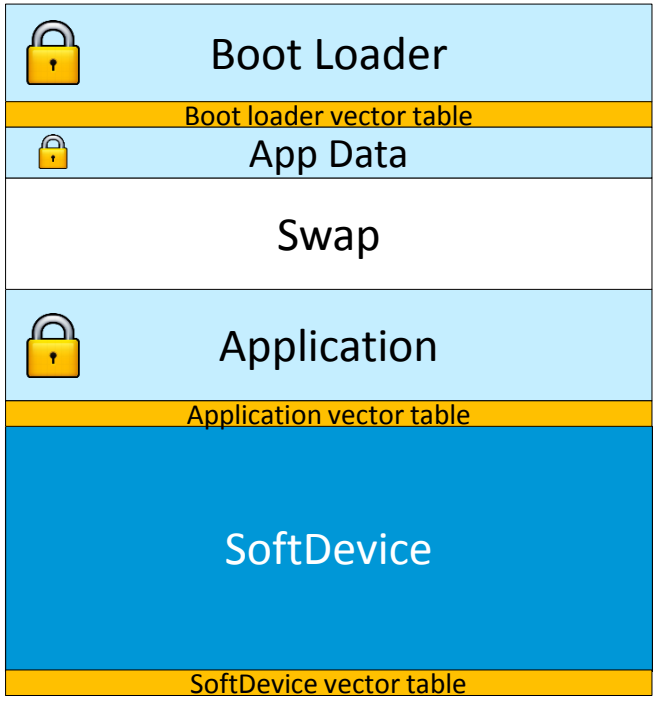


Memory Range*	Size	Usage
0x0003C000 - 0x00040000	16 kB	Bootloader
0x00016000 - 0x0003C000	152kB	Code Region 1: Application Code (BANK 0), Swap (BANK 1), and data
0x00000000 - 0x00016000	88 kB	Code Region 0 –SoftDevice

*The memory layout based on the DFU example in SDK v7.1

Dual Bank update

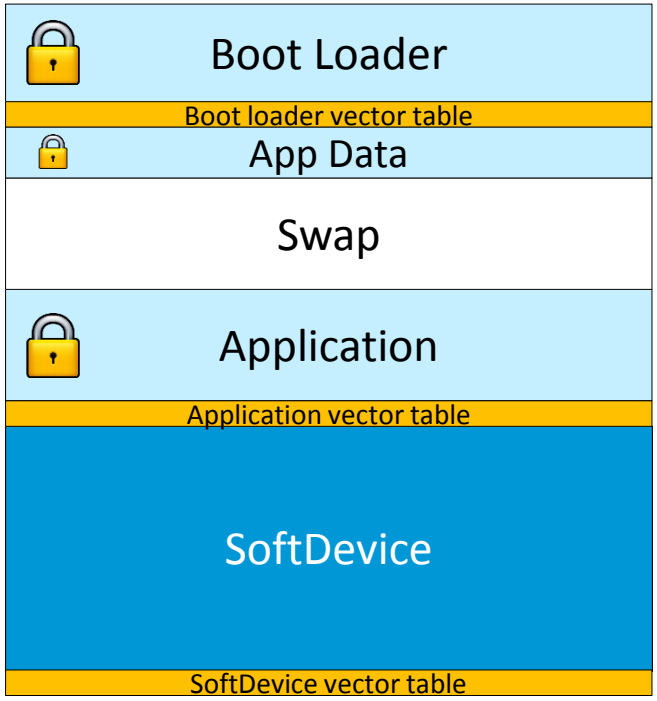
Application



- 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

OTA Dual Bank update

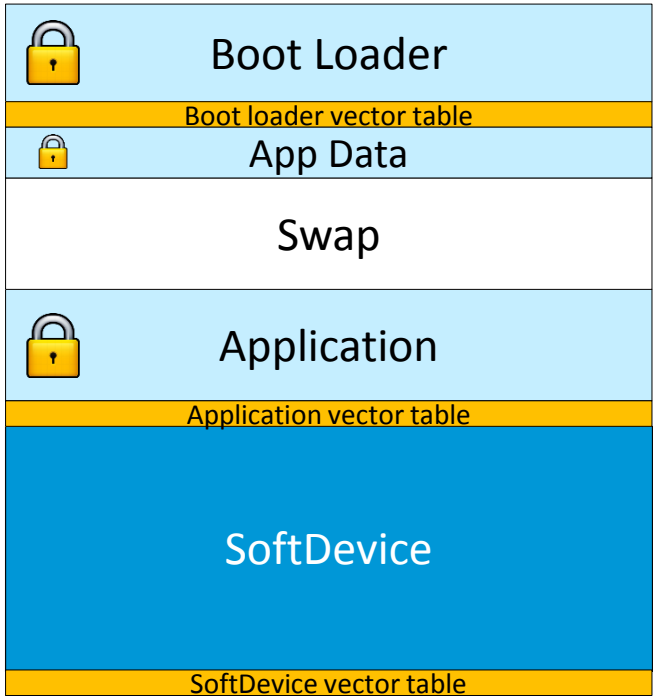
Via BLE



- Dual Bank update via BLE:
  - Swap enables reverting to old firmware if BLE link is permanantly 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): 76 kB
    - nRF51822 QFAB (128 kB): 12 kB

Dual Bank Update in nRF51 SDK

Performance

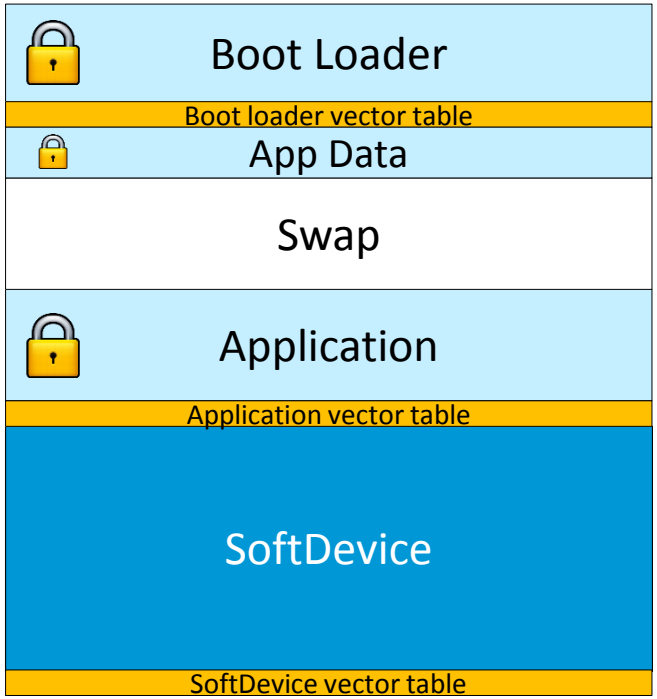


- BLE DFU service details:
  - Connection Interval (SDK 6.1.0): 15 ms
  - Maximum 6 packets pr connection interval
  - 20 bytes / packet
- Update times:
  - nRF Master Emulator (Peer as fast as SDK):
    - 20 kB application (eg. Heart Rate): **5 sec**
    - 76 kB application: 14 sec
  - Android 4.4: **20 sec** (20 kB)*
  - iOS7: 0**7 sec** (20 kB)

*Varied between Android phones

Dual Bank Update in nRF51 SDK

SAFETY

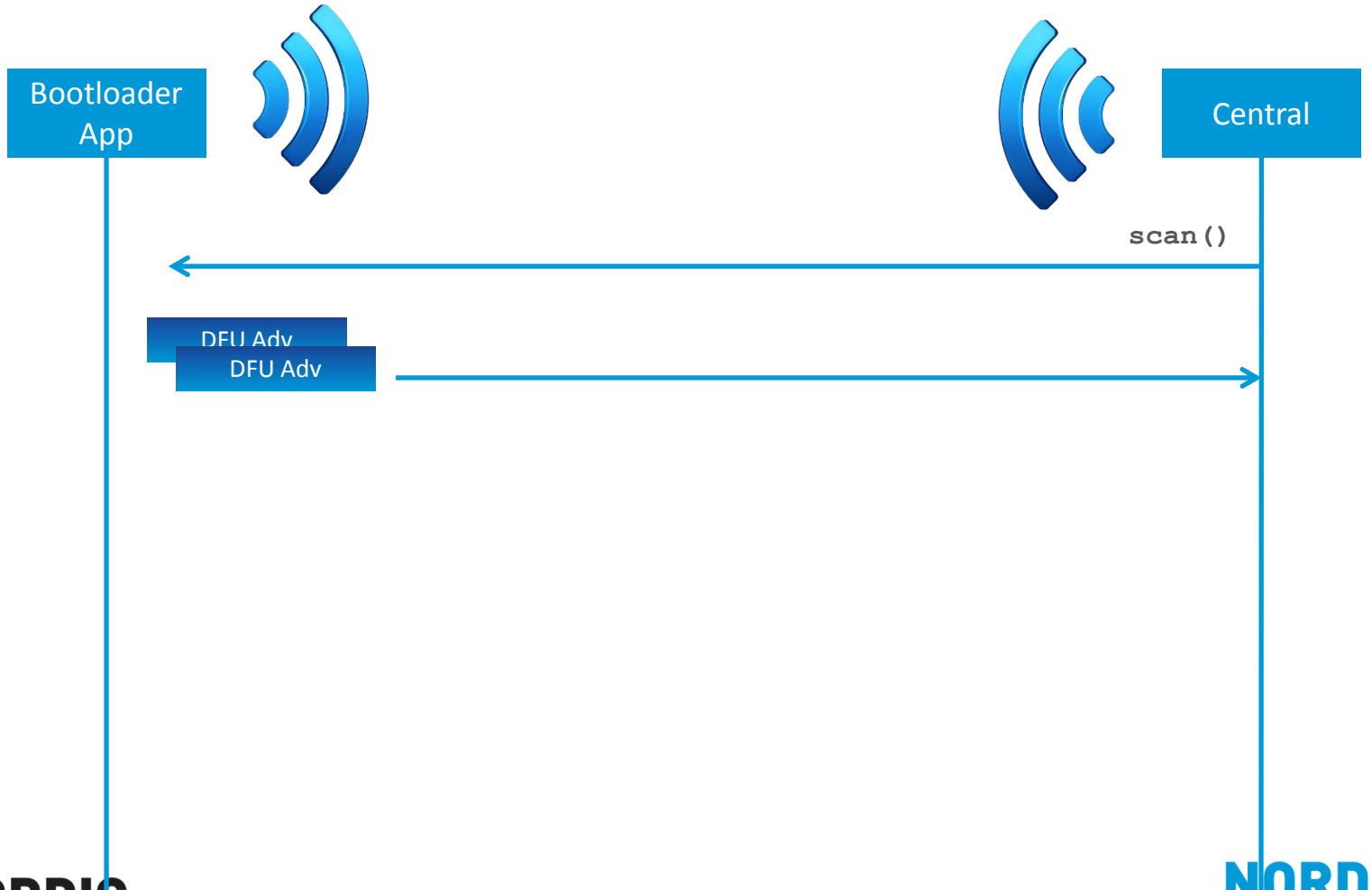


- Link Loss Recovery Mechanism
  - Upon re-connection Peer issue a «Request Received Image Size»
  - Continue data transfer if remote and local image size doesn't match
- Recovery from incomplete/failed Firmware Update
  - Old application preserved until successful Firmware Update completed
  - Old application always is fallback image
  - Old application will be re-started after 60s time out

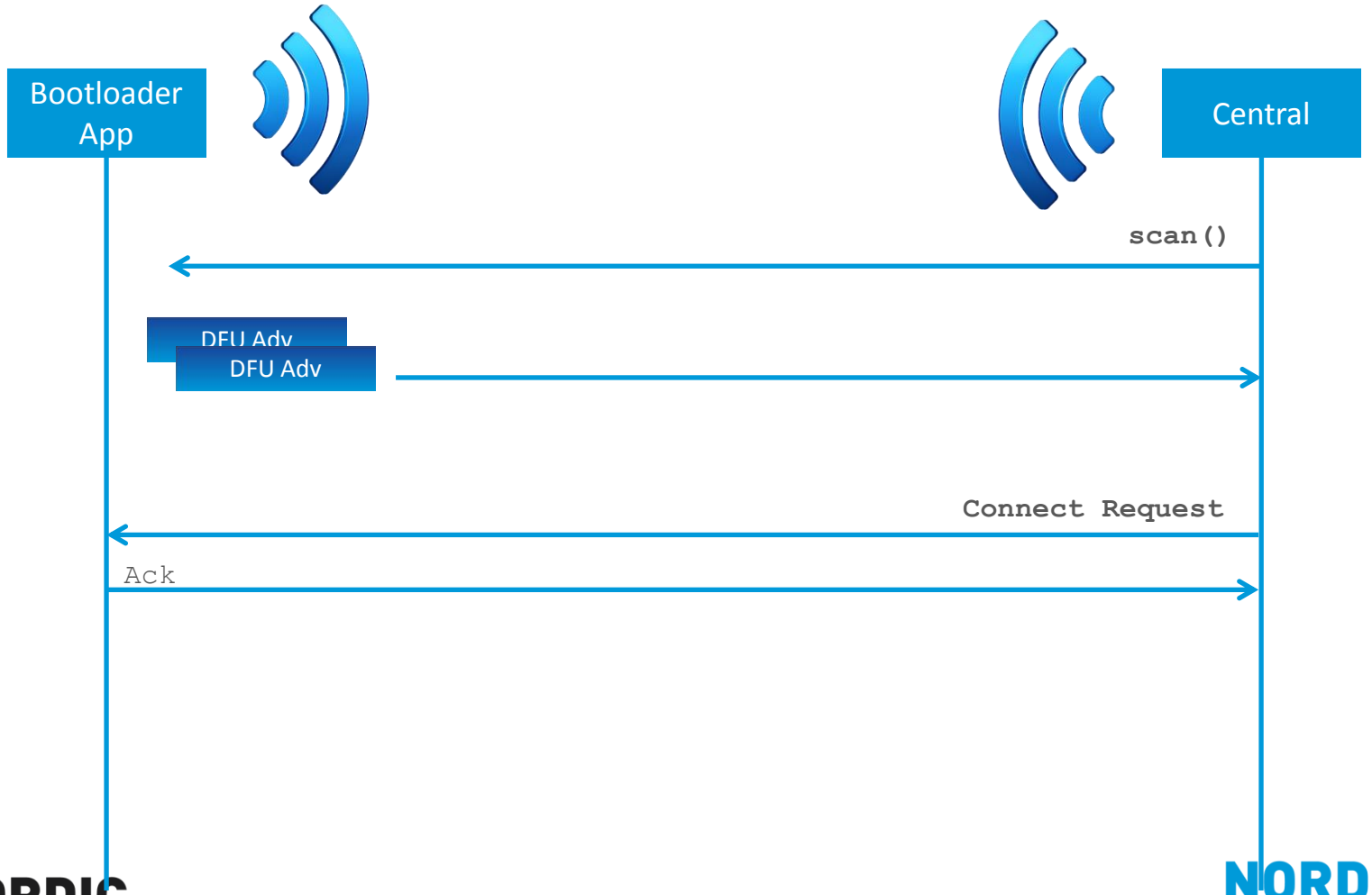
Over the air firmware update sequence

Over the air firmware update sequence

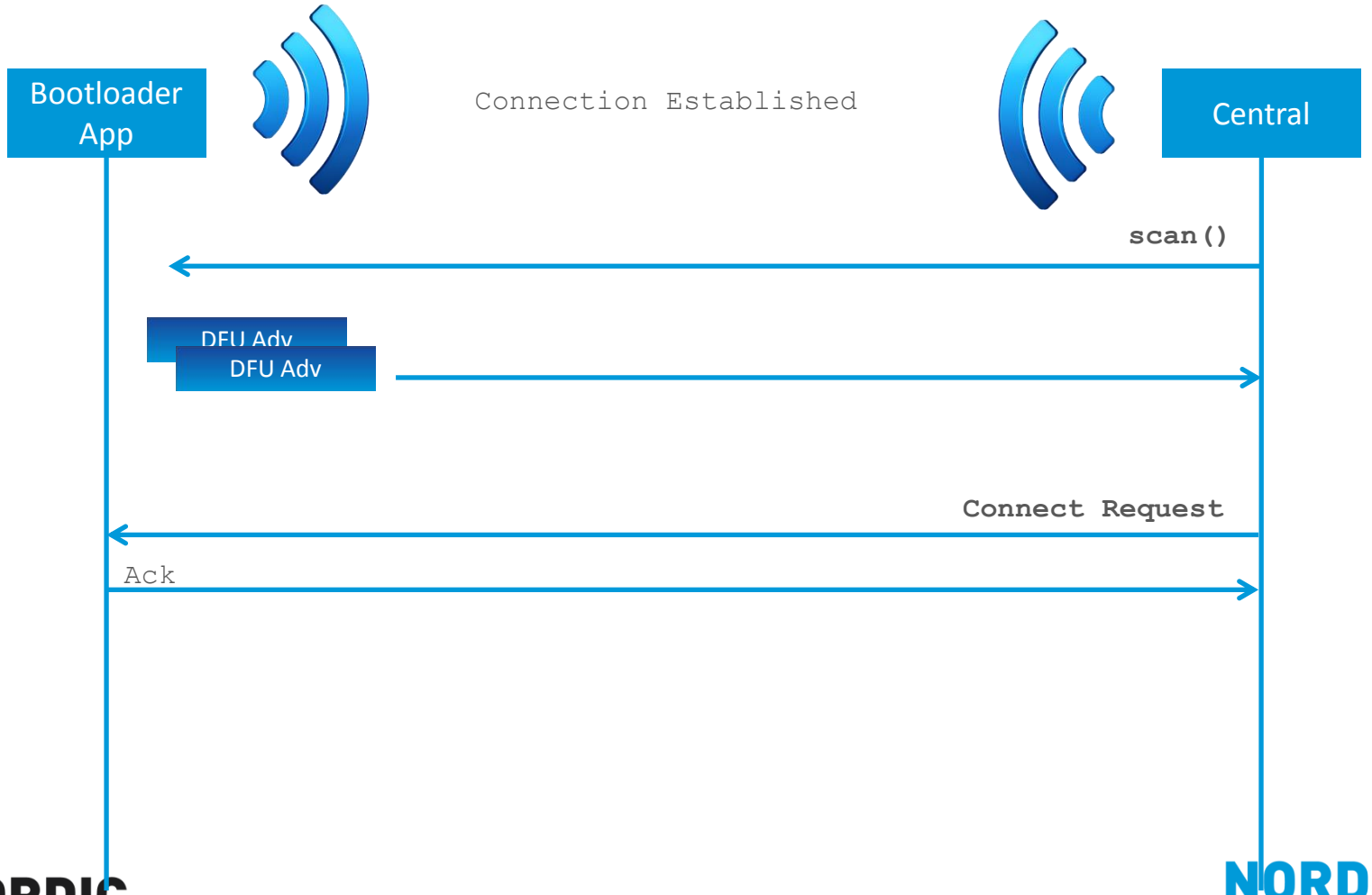
Over the air firmware update sequence



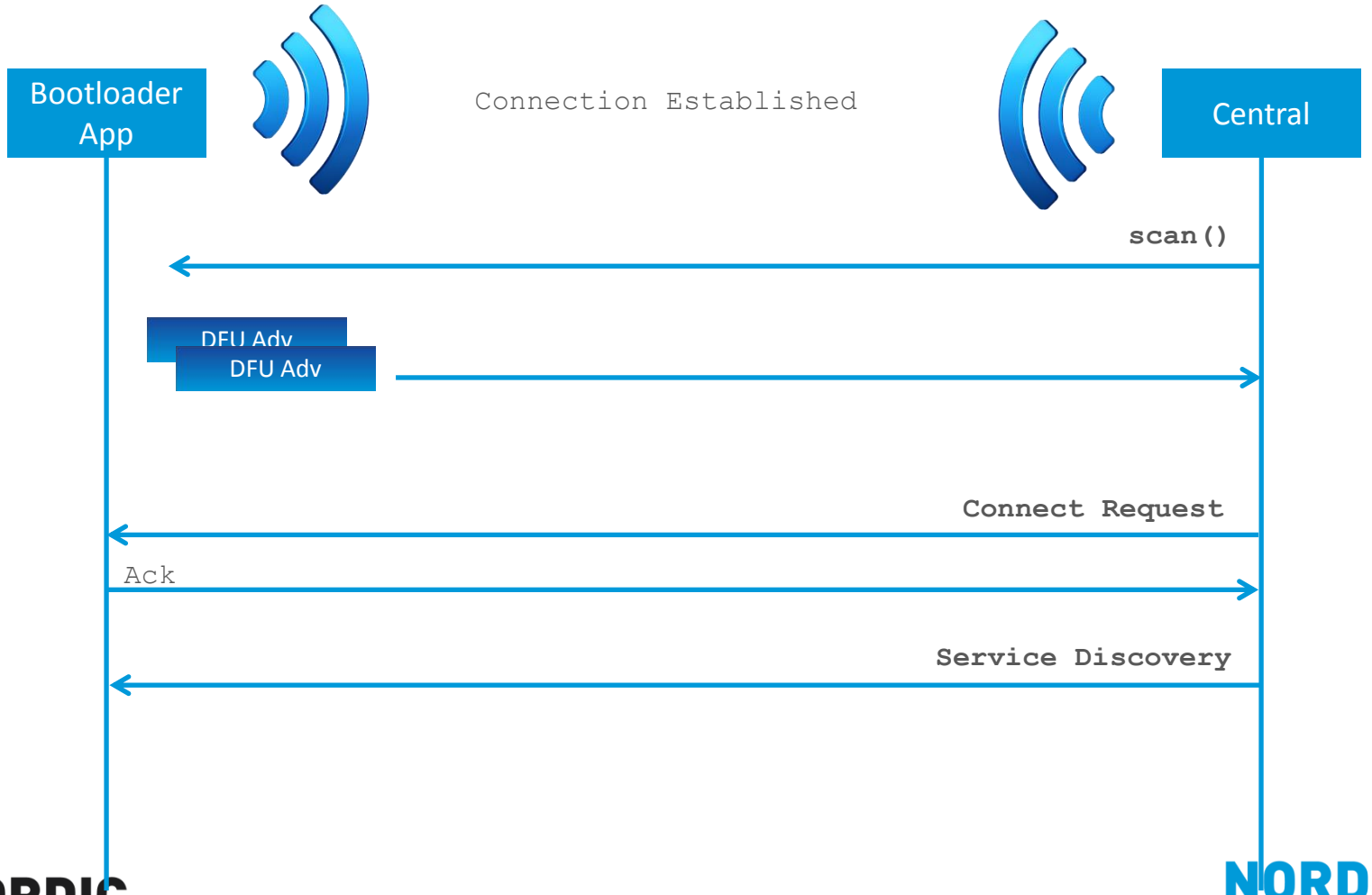
Over the air firmware update sequence



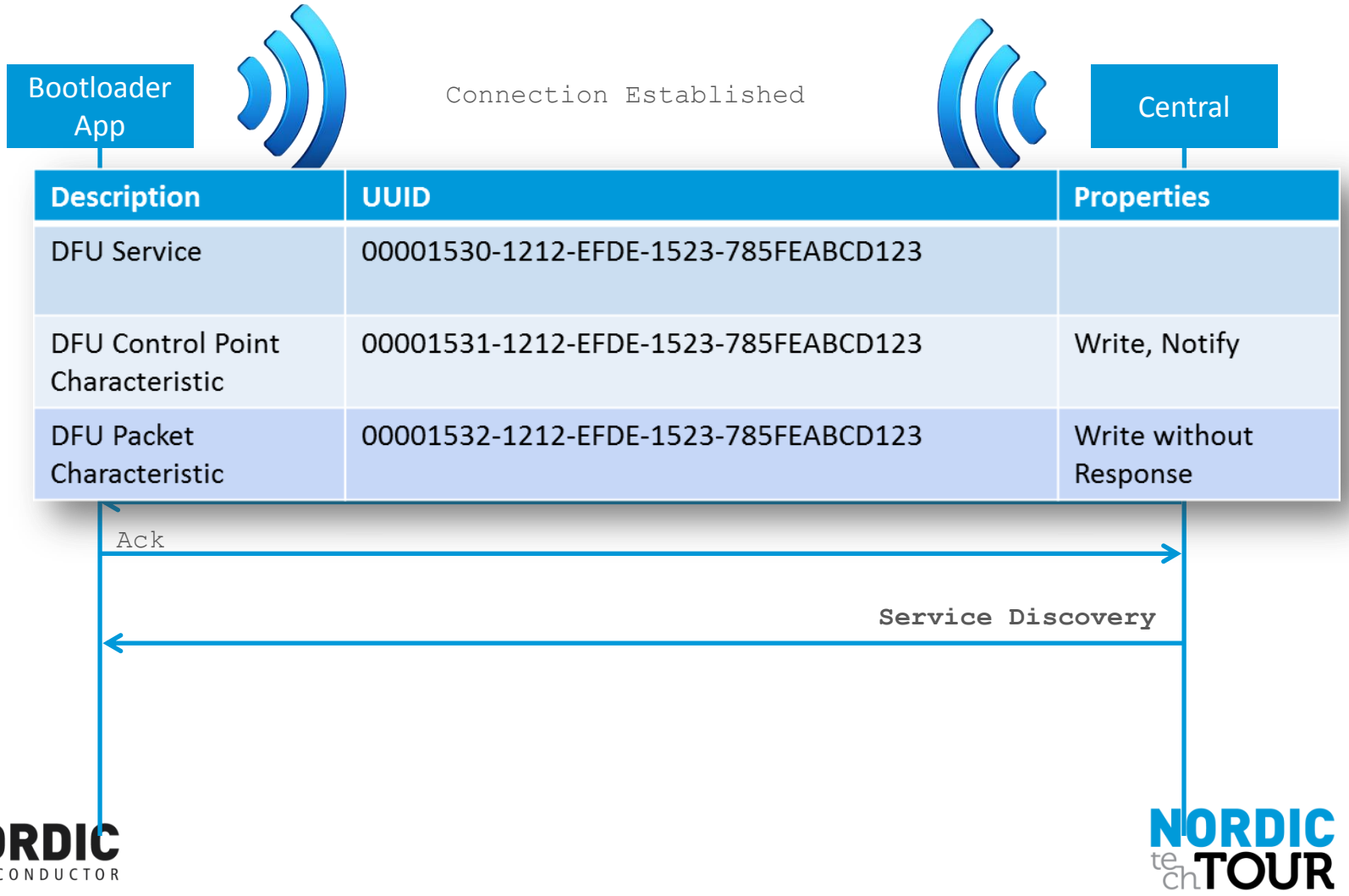
Over the air firmware update sequence



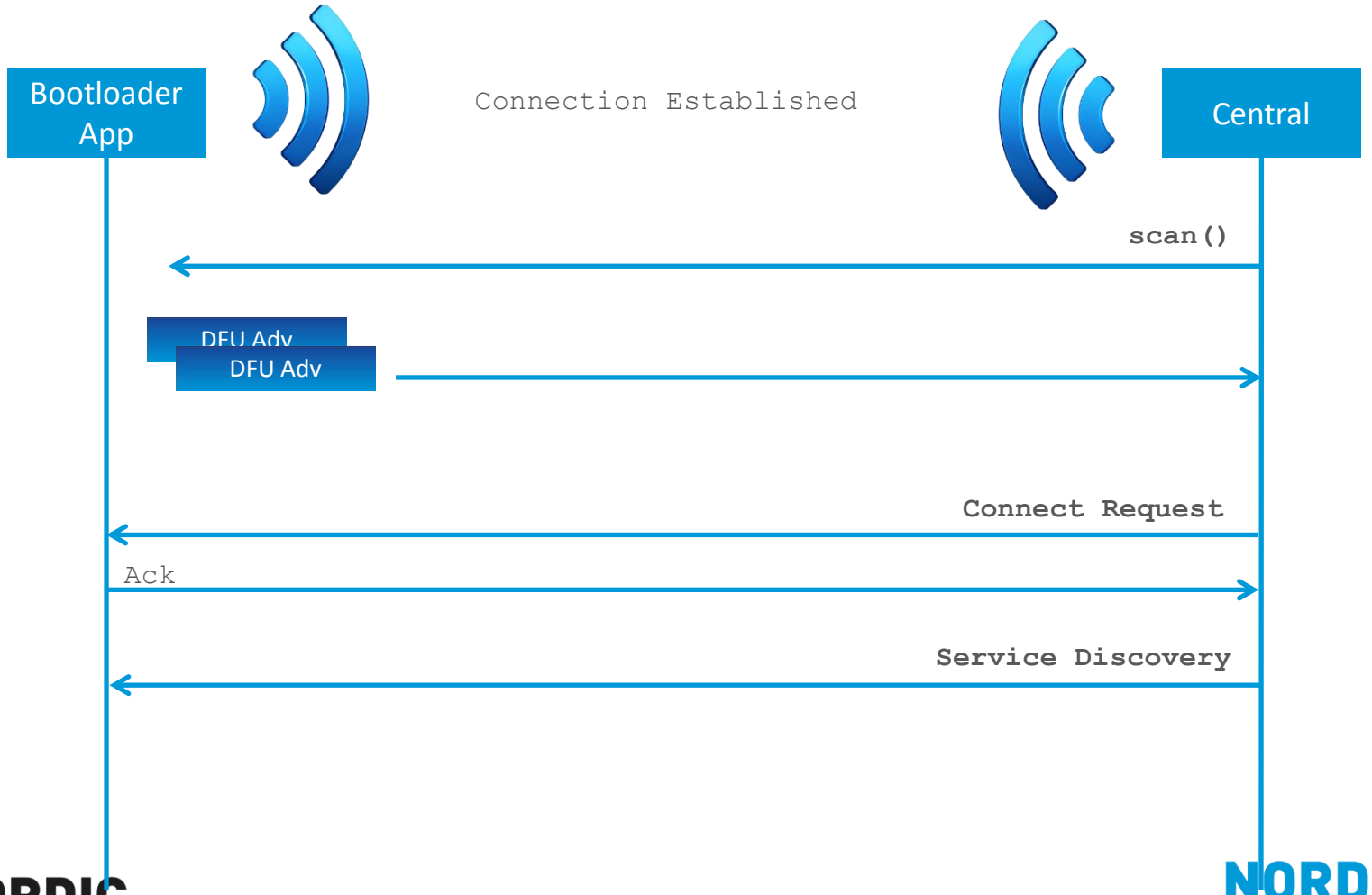
Over the air firmware update sequence

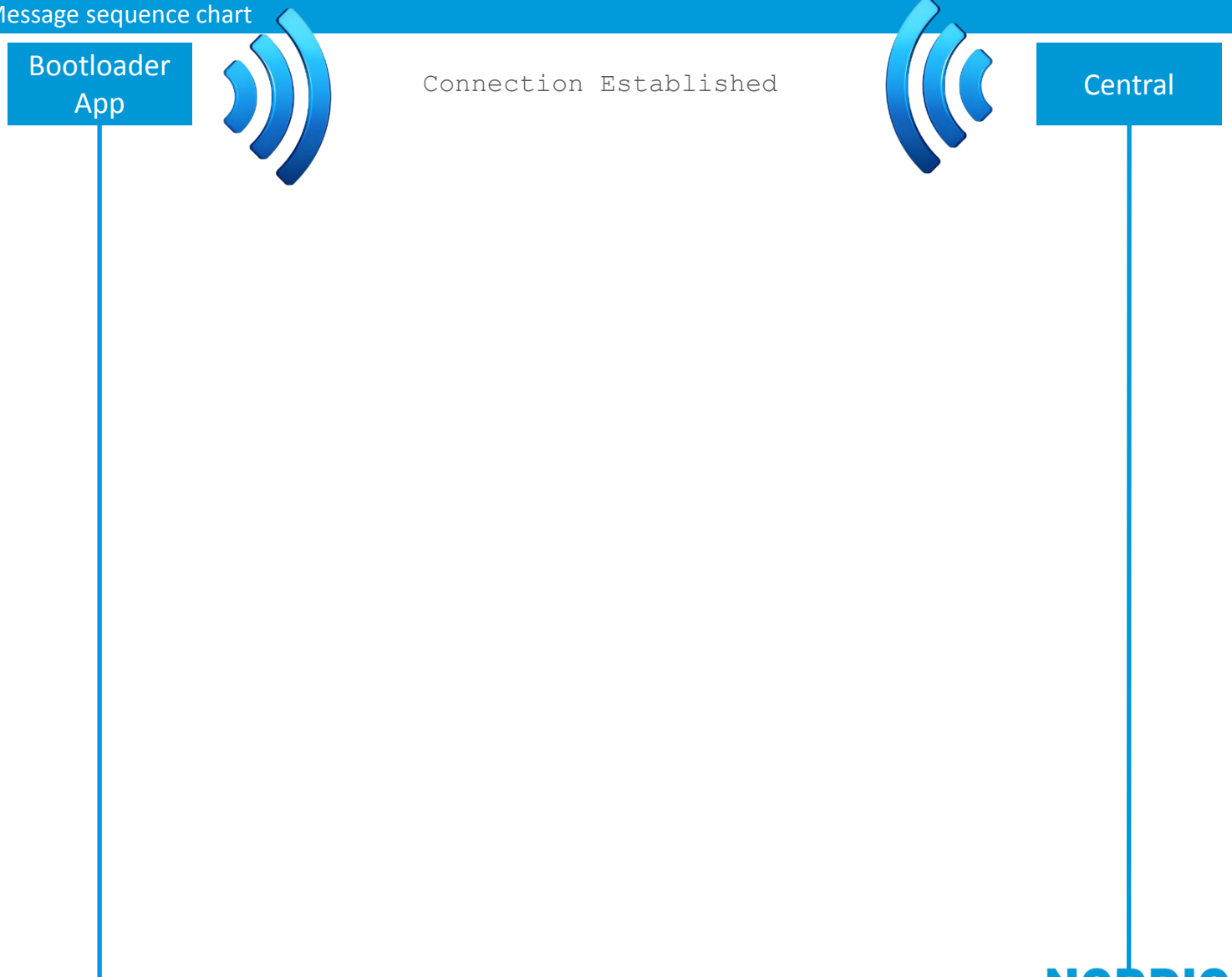


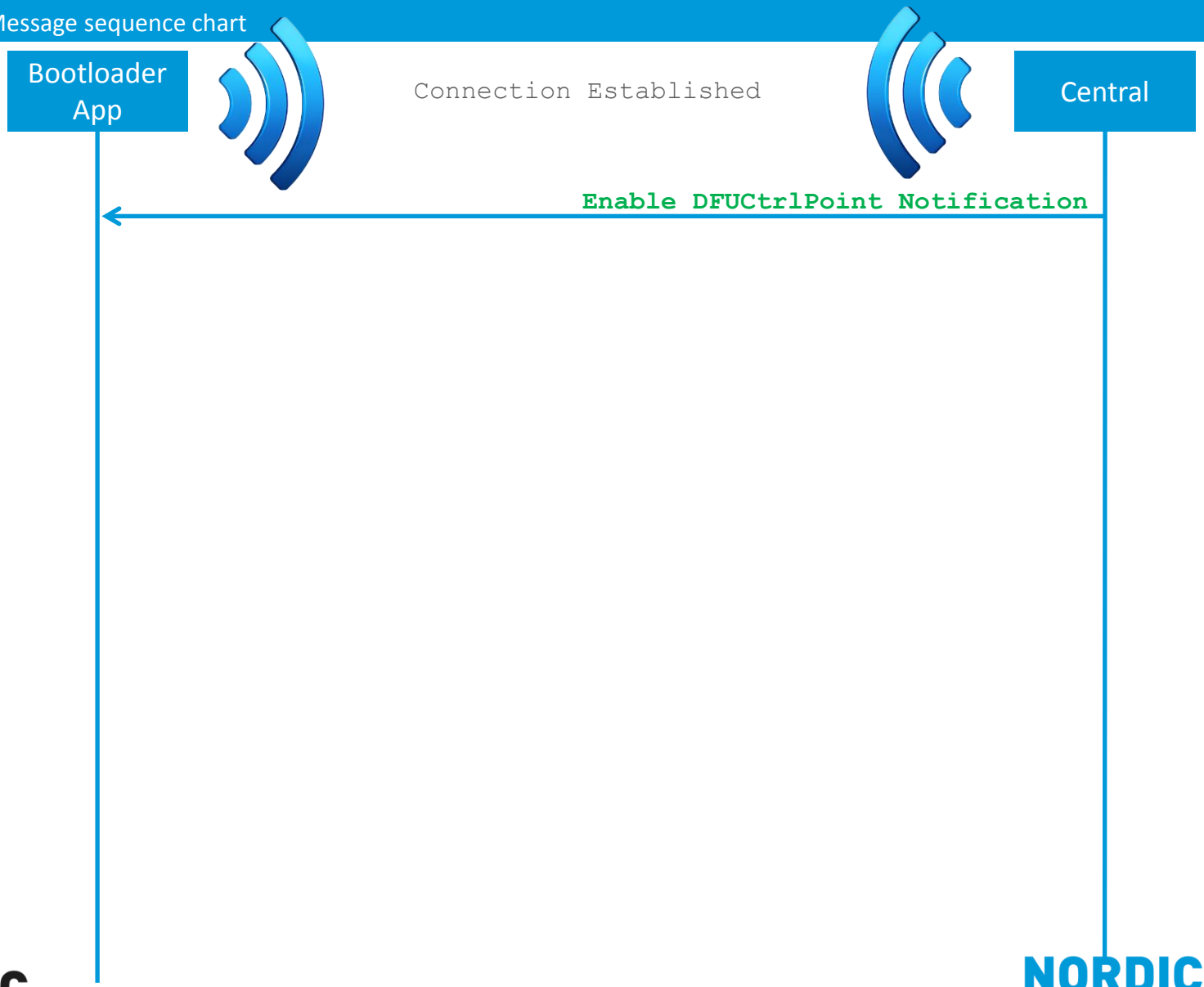
Over the air firmware update sequence



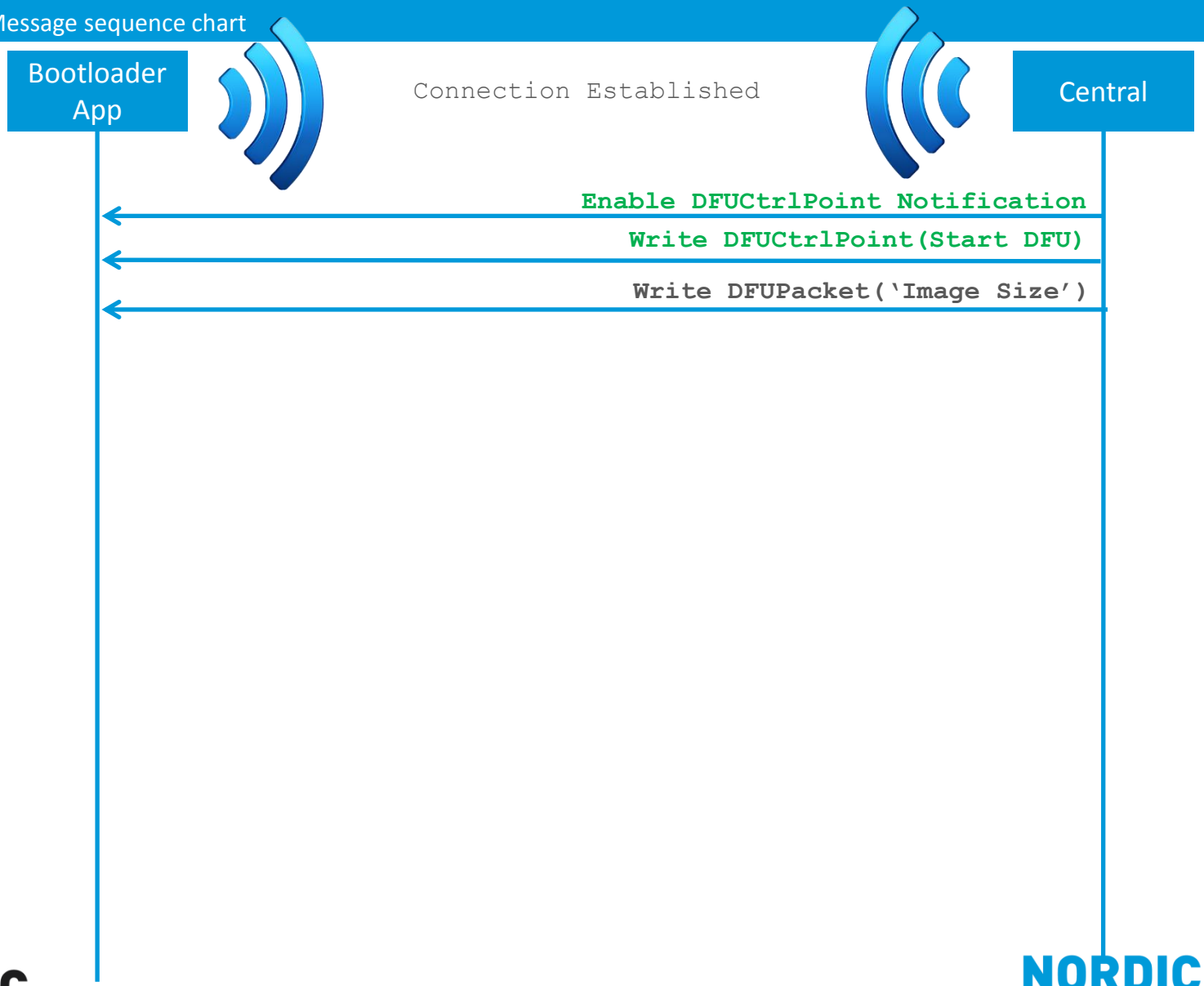
Over the air firmware update sequence

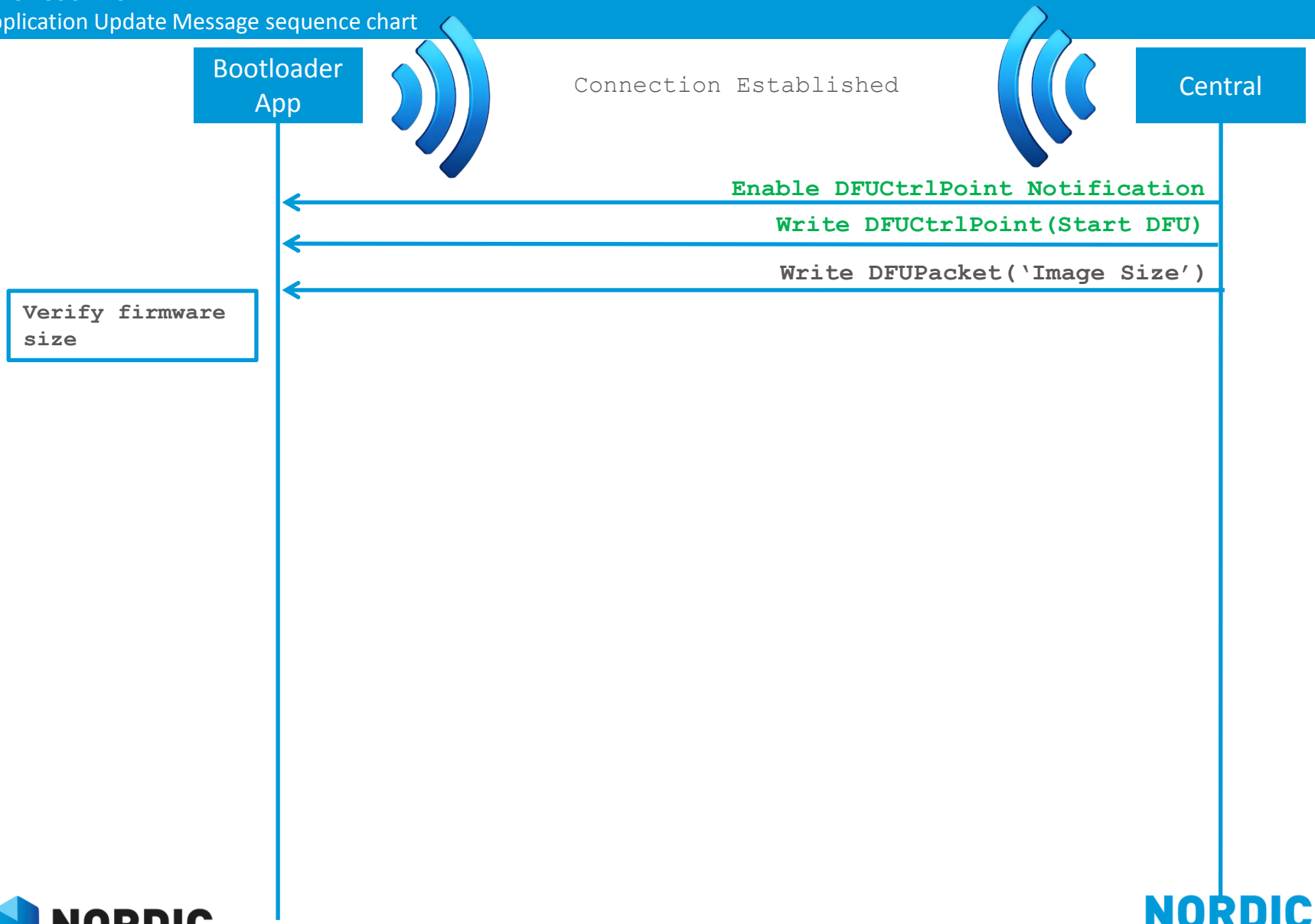


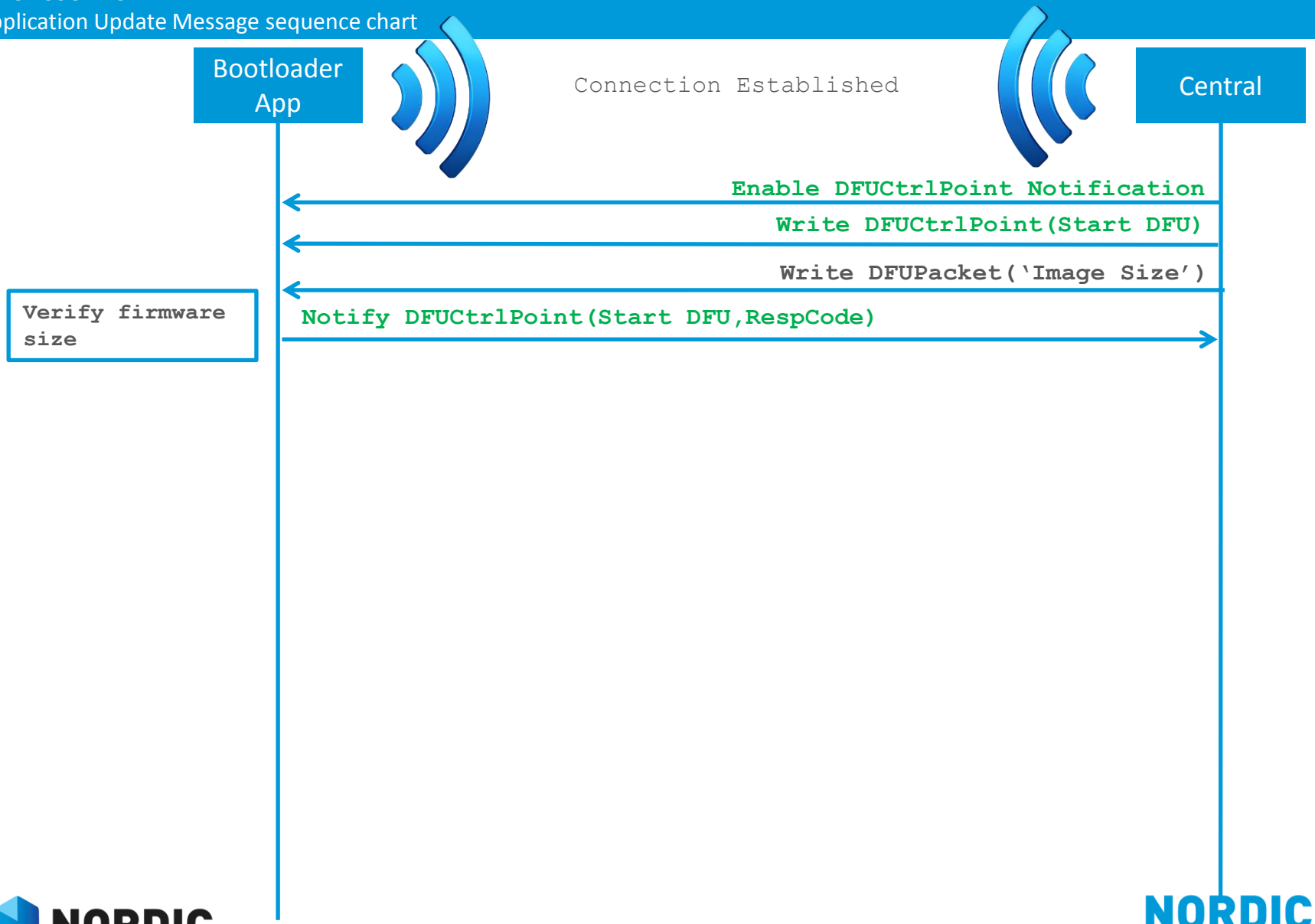

Bootloader App
Connection Established
Central

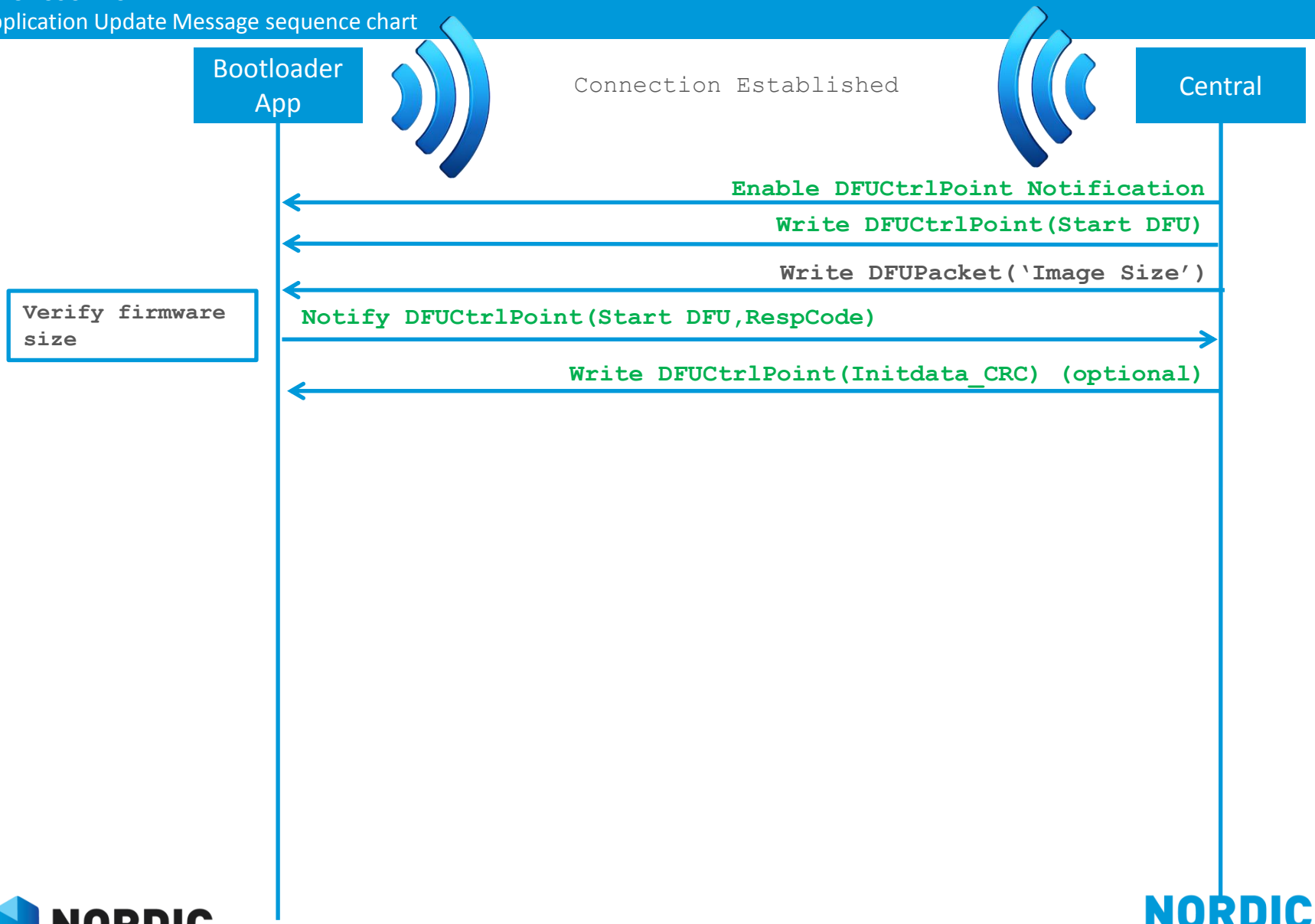

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification

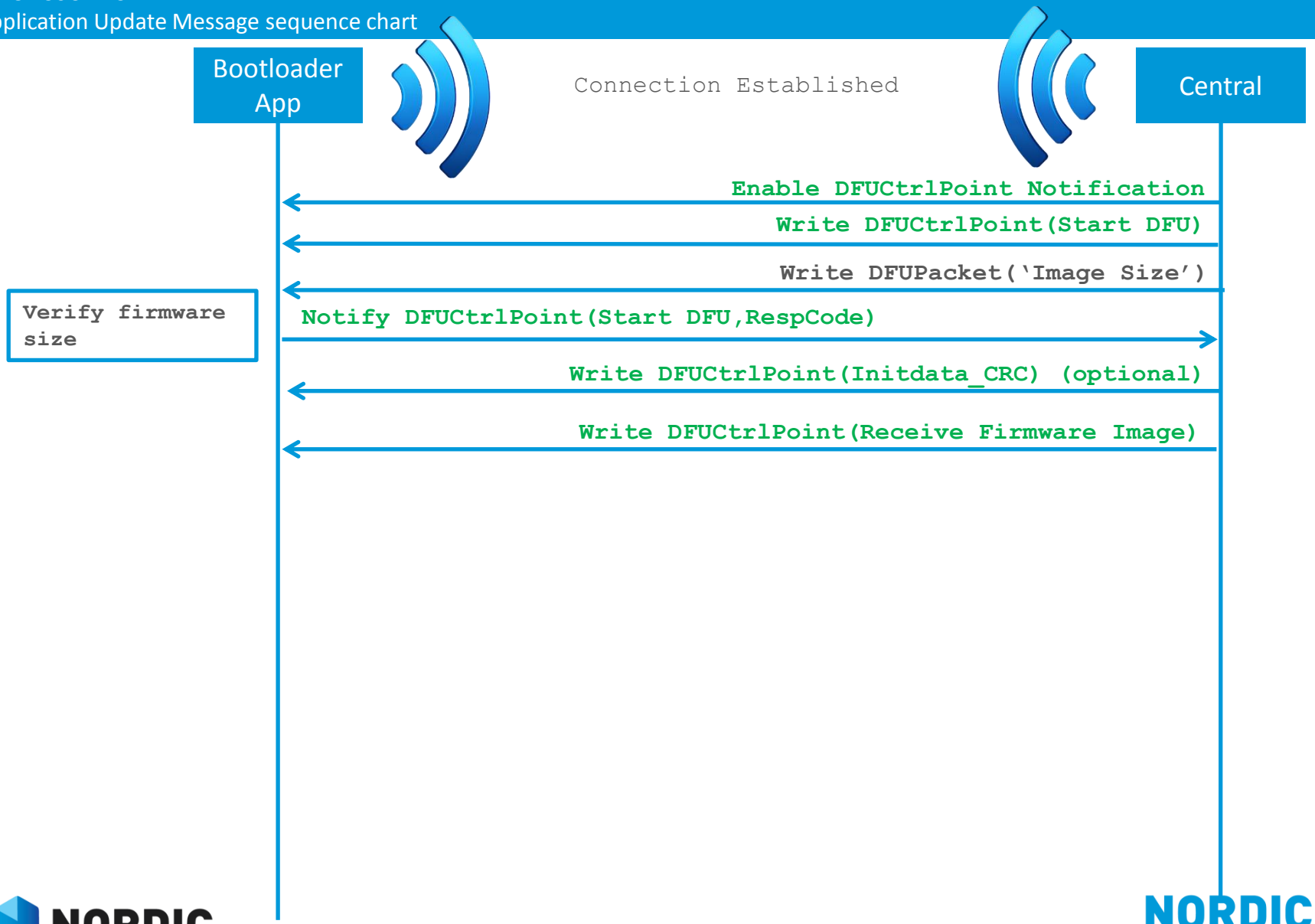

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)

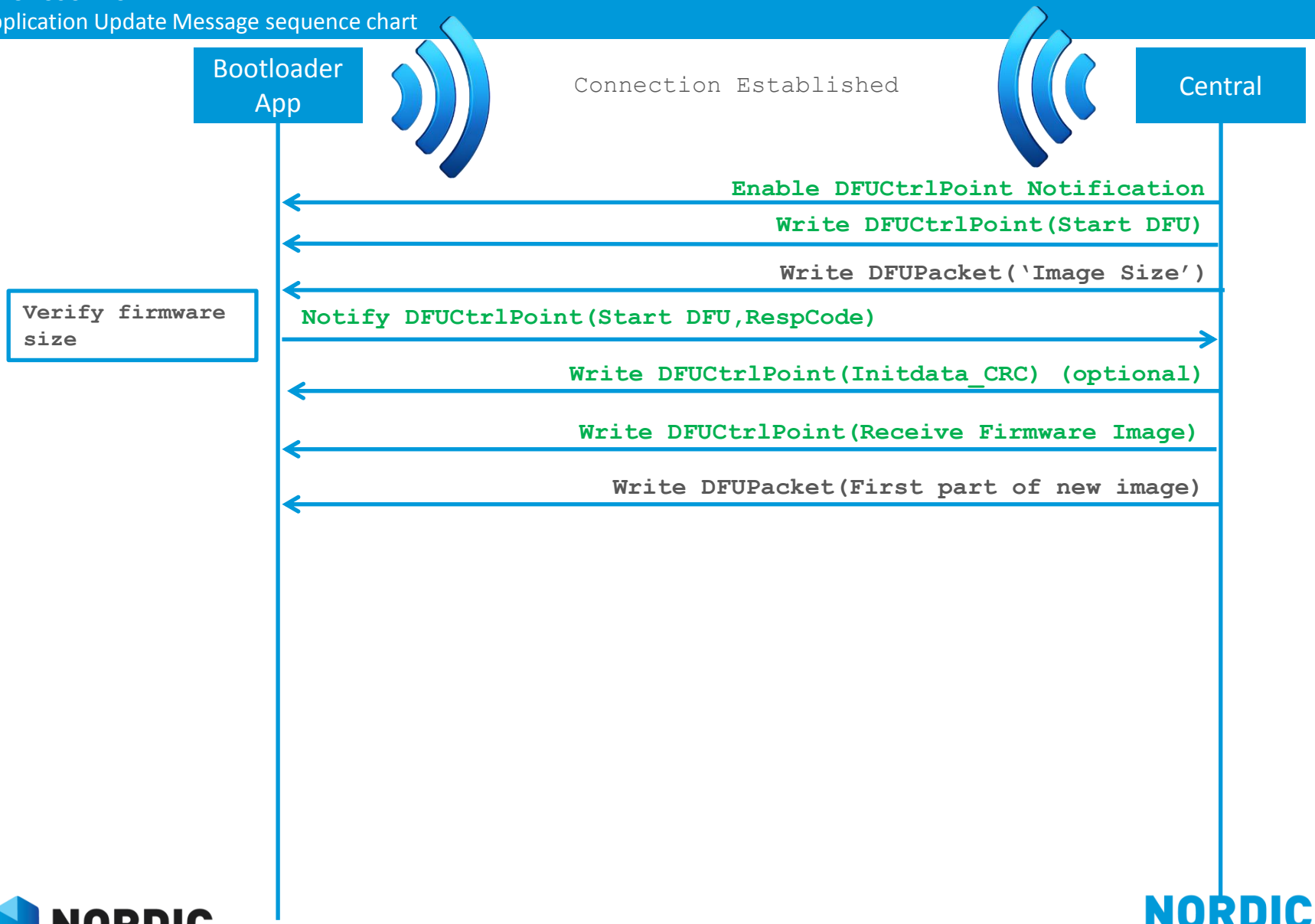

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')

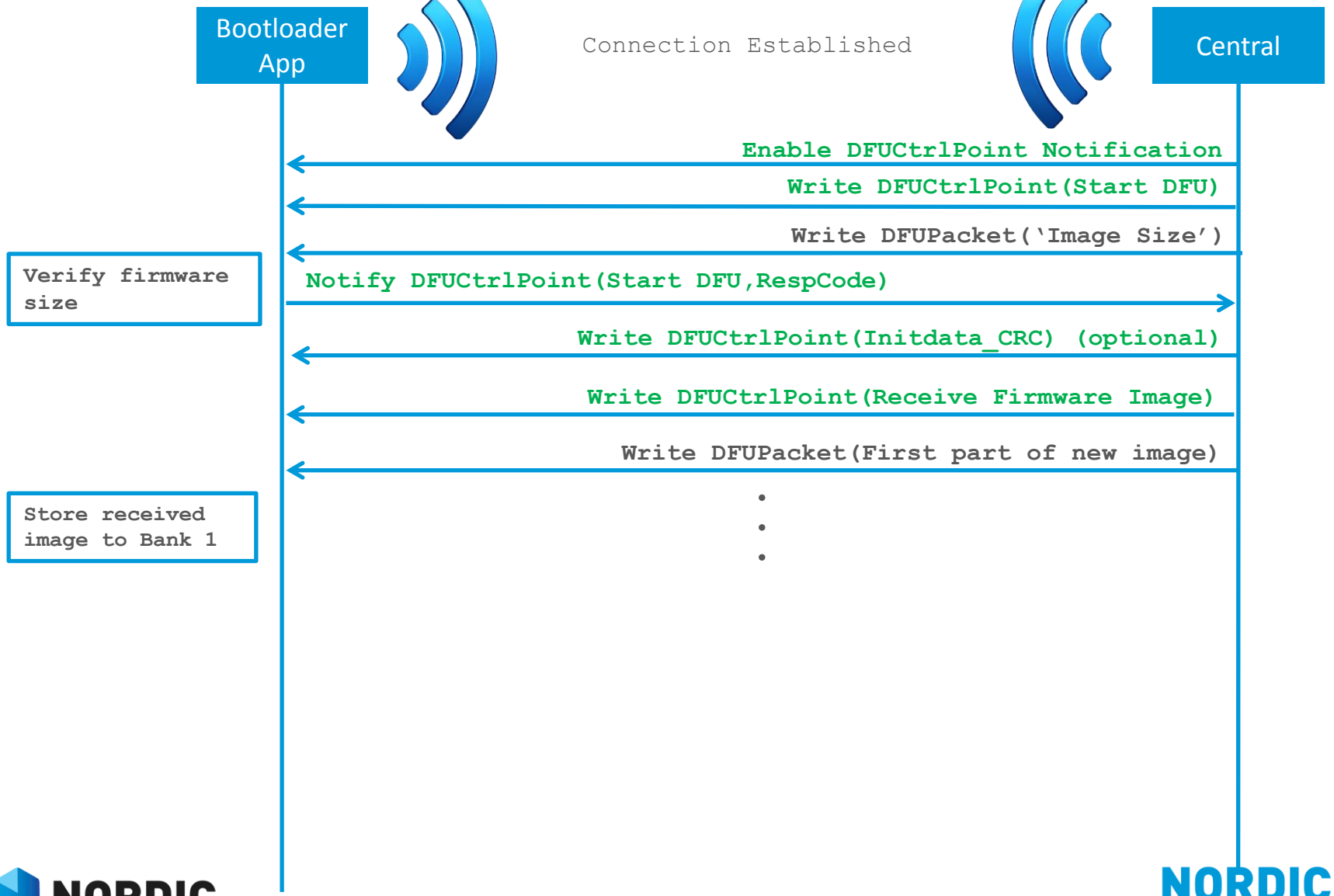

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size

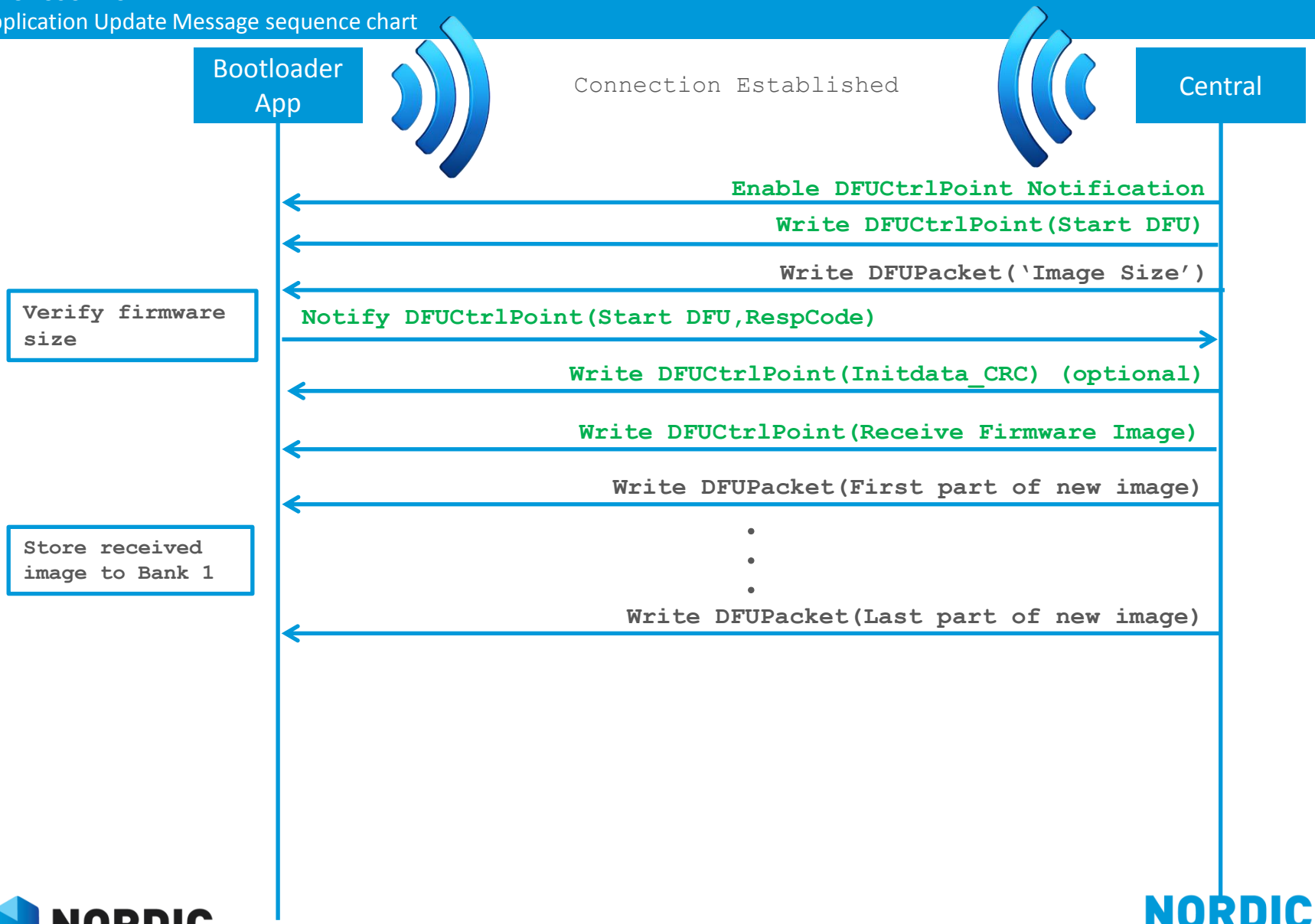

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)

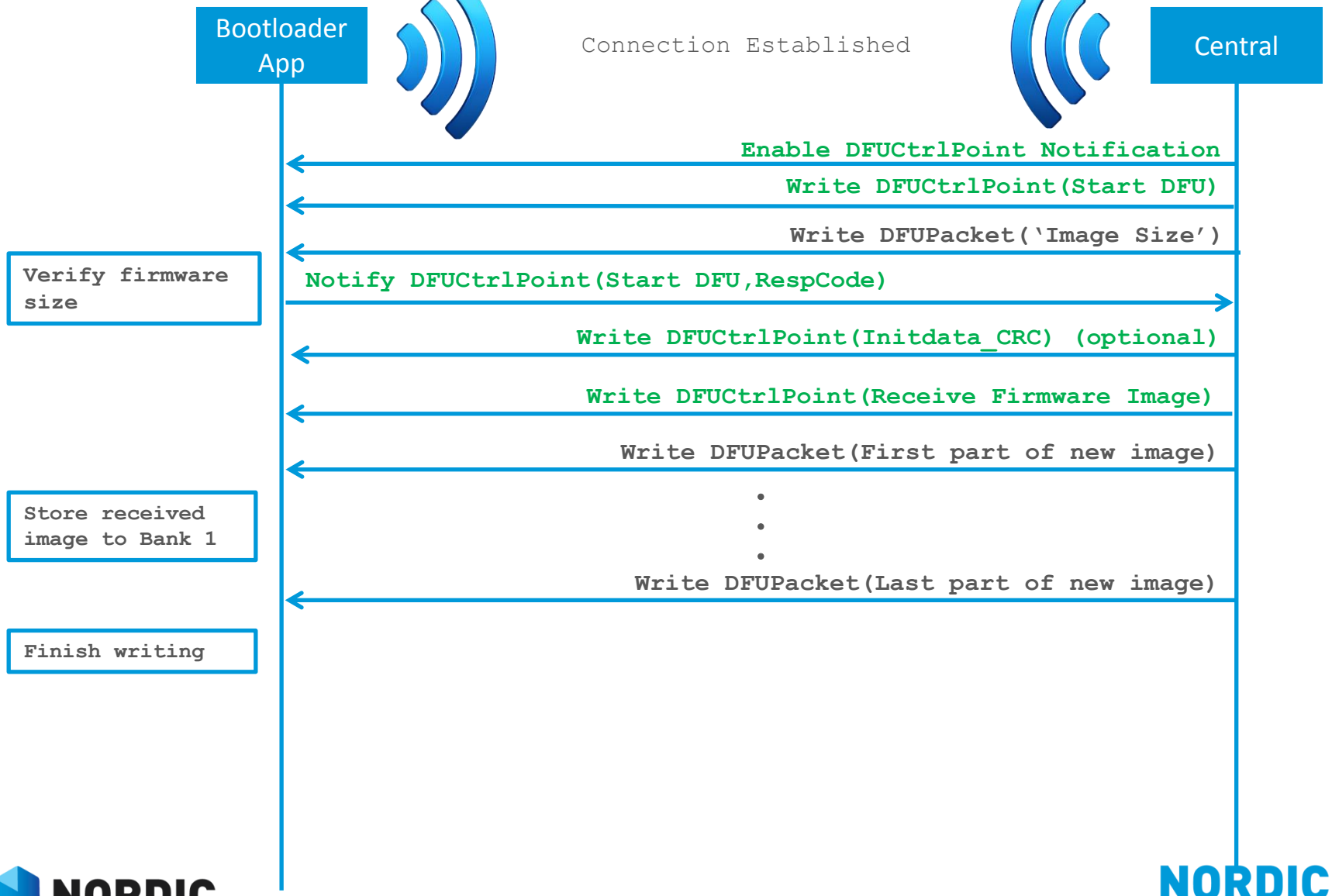

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)

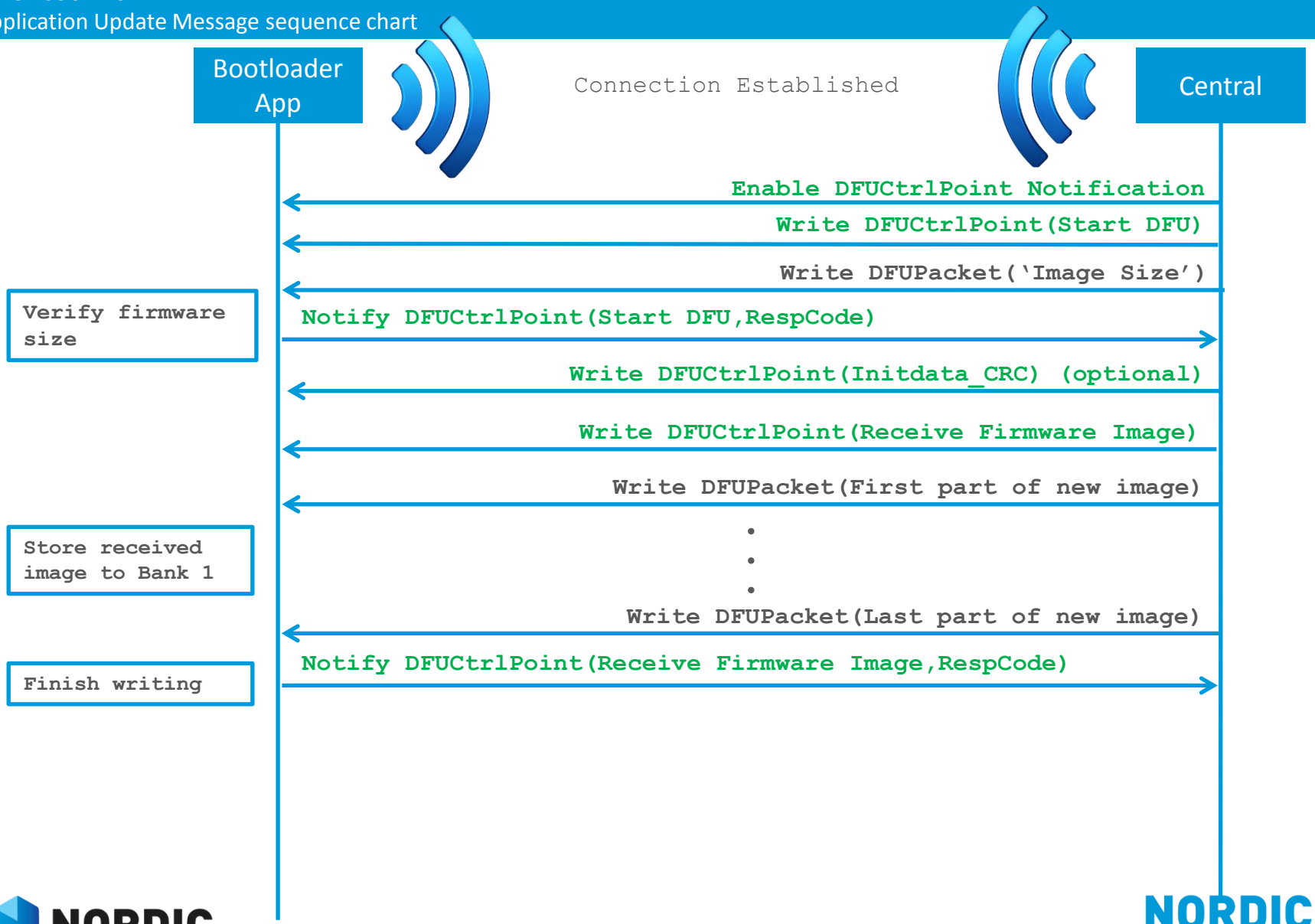

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)

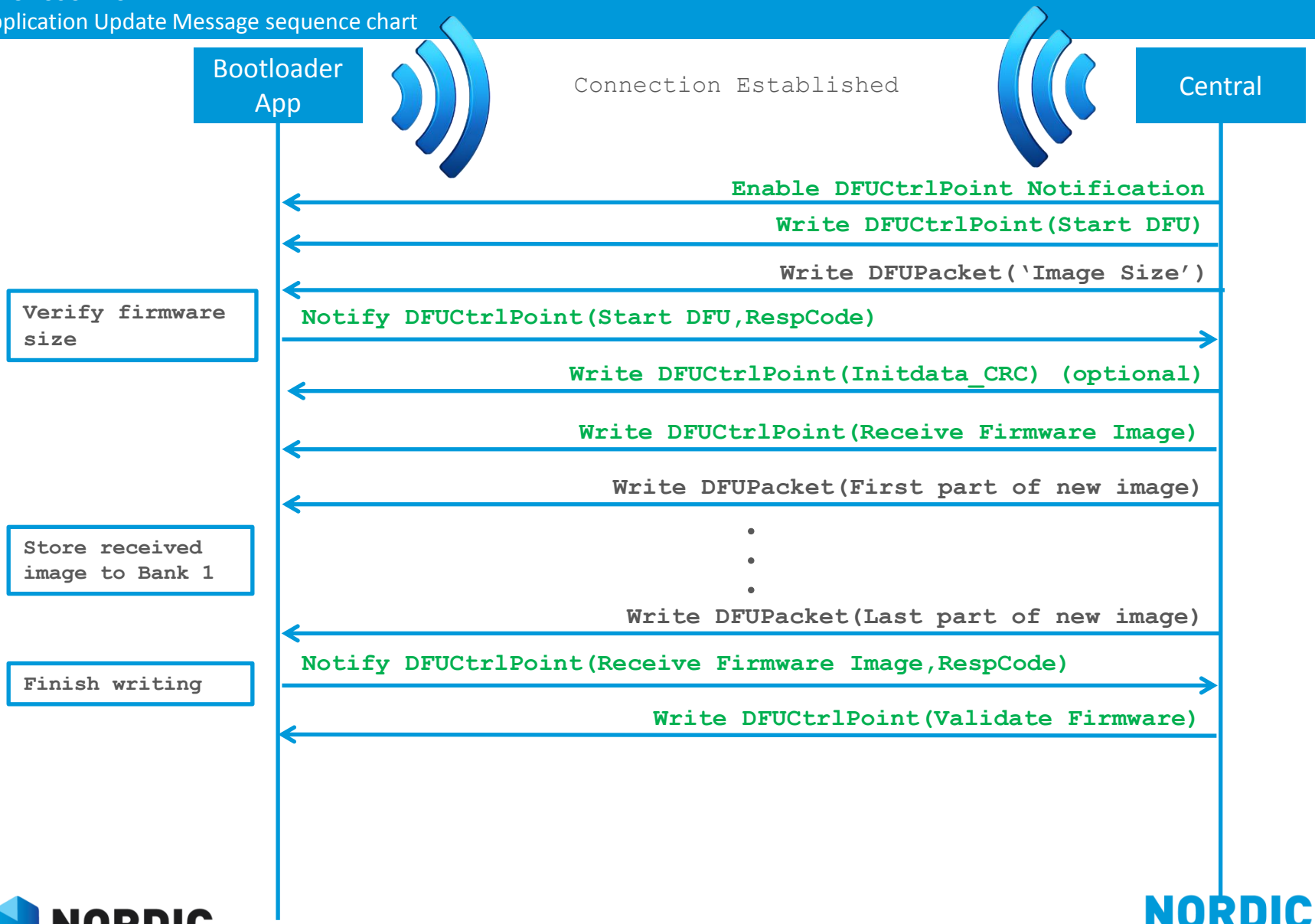

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)

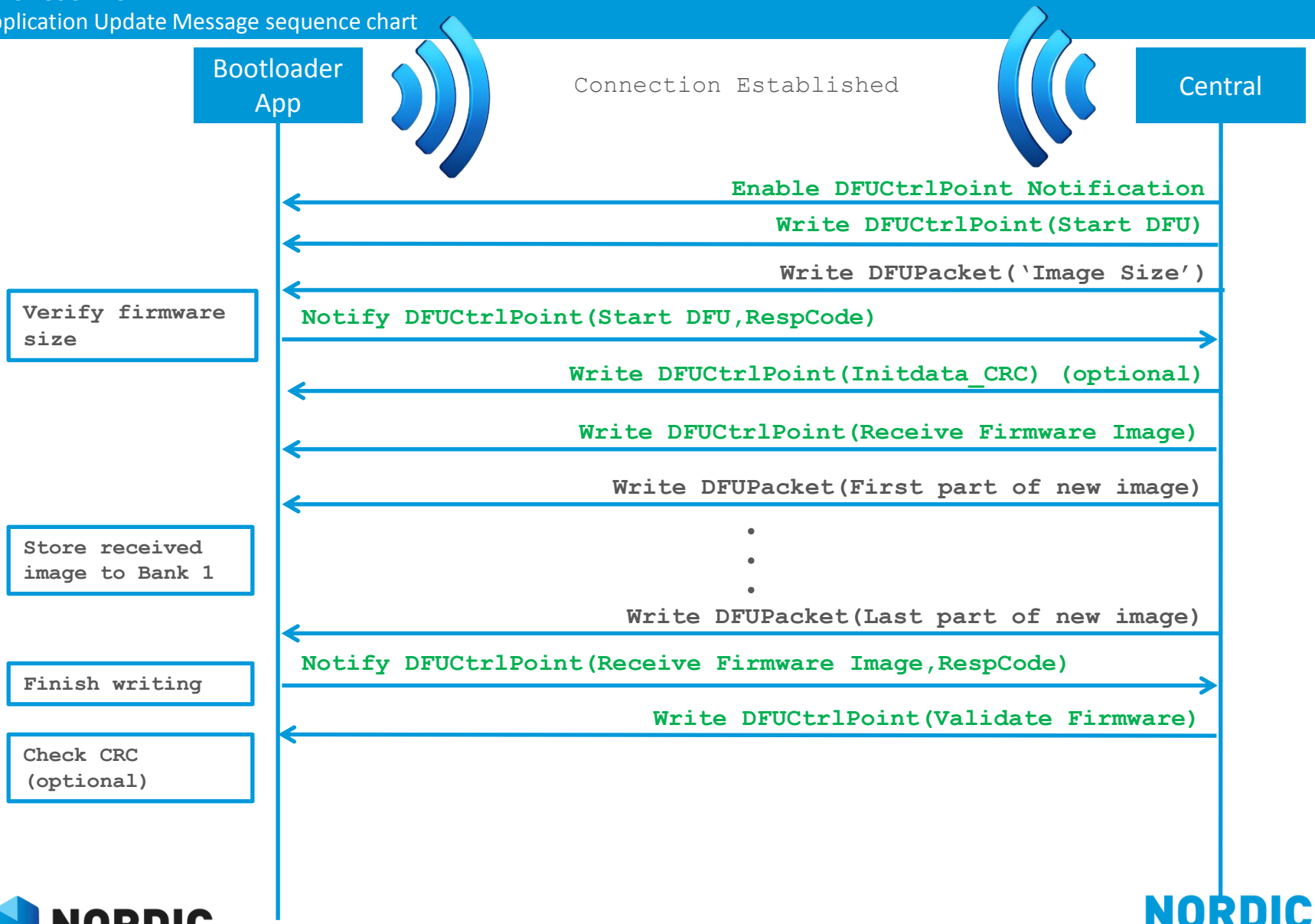

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1

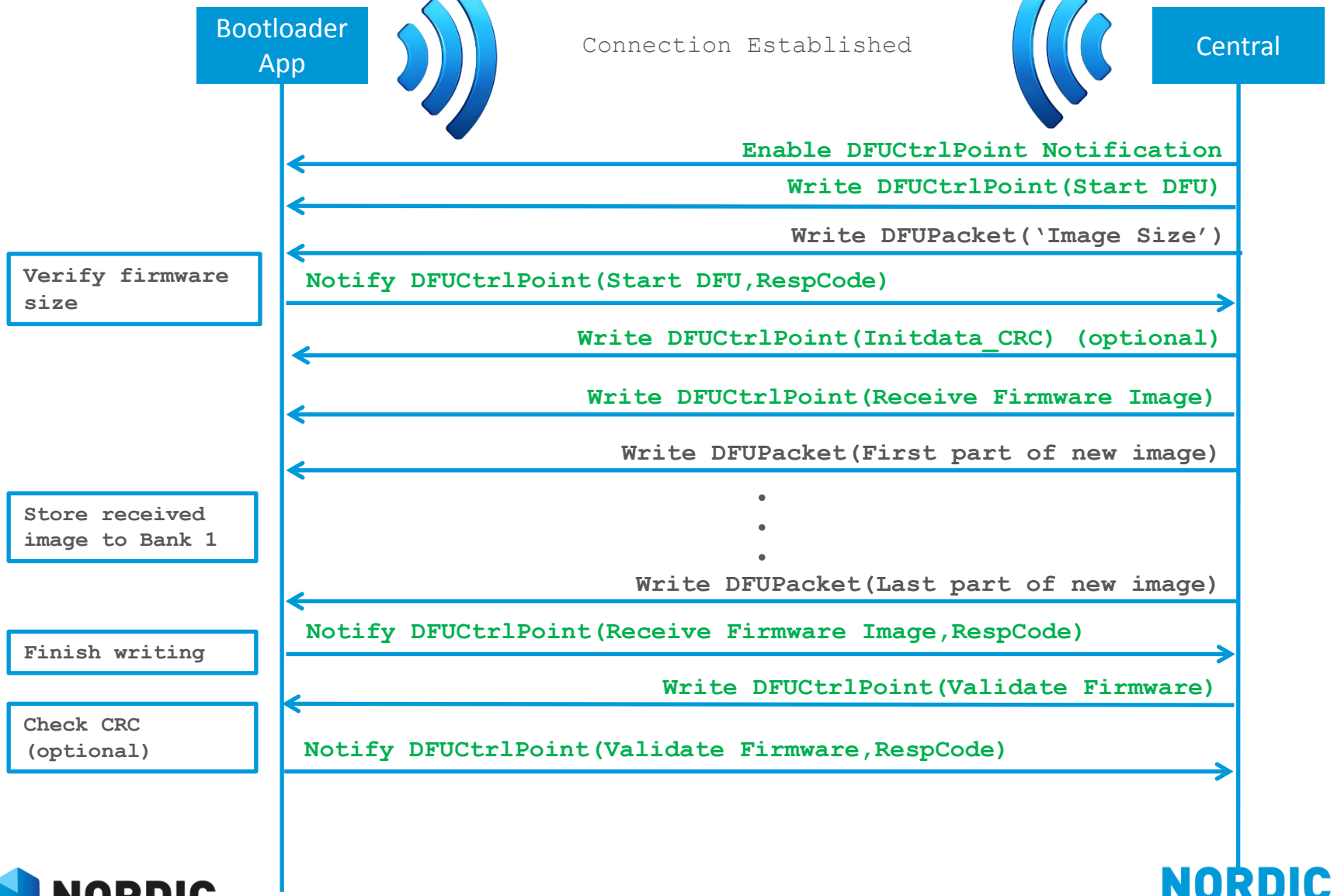

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)

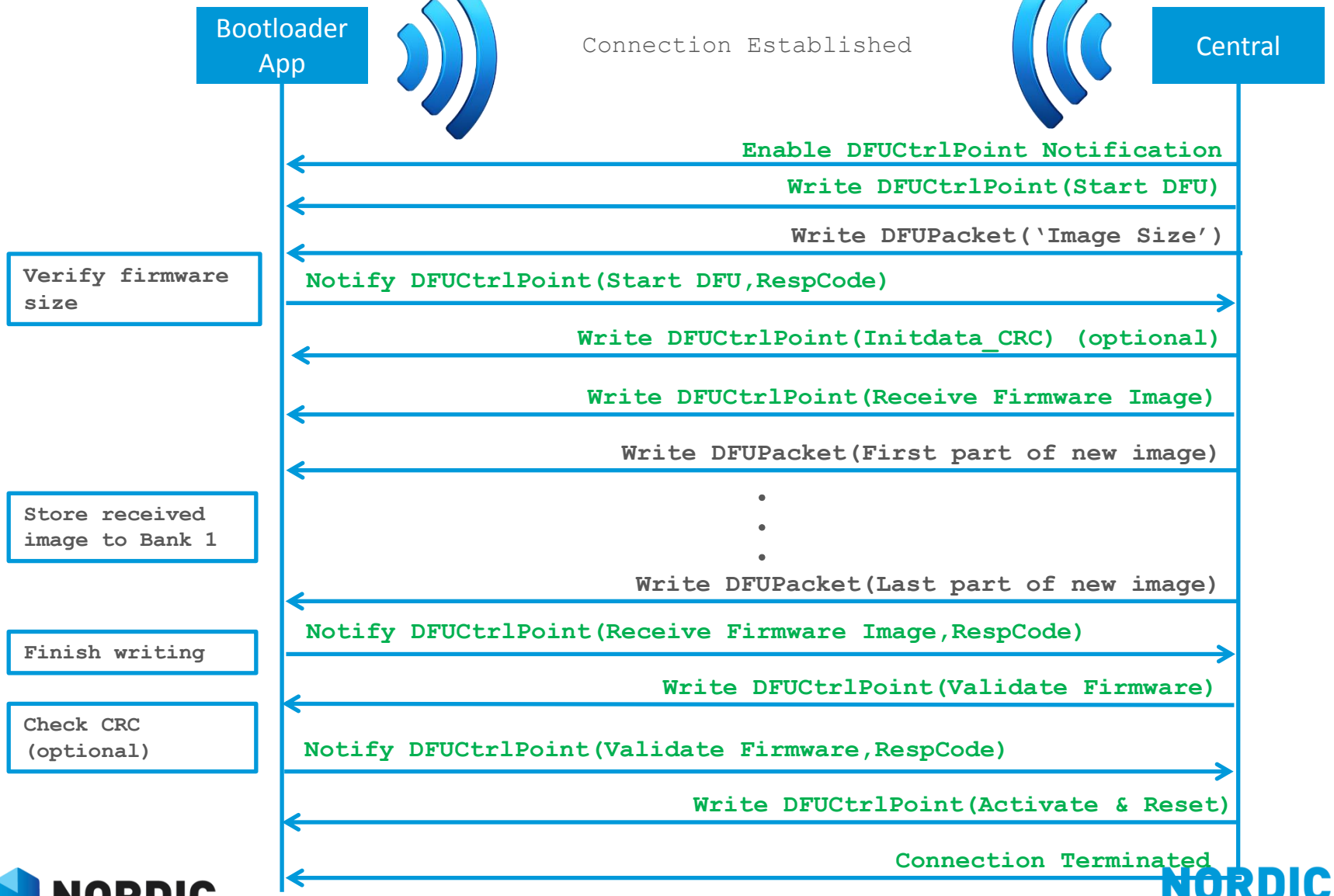

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Finish writing

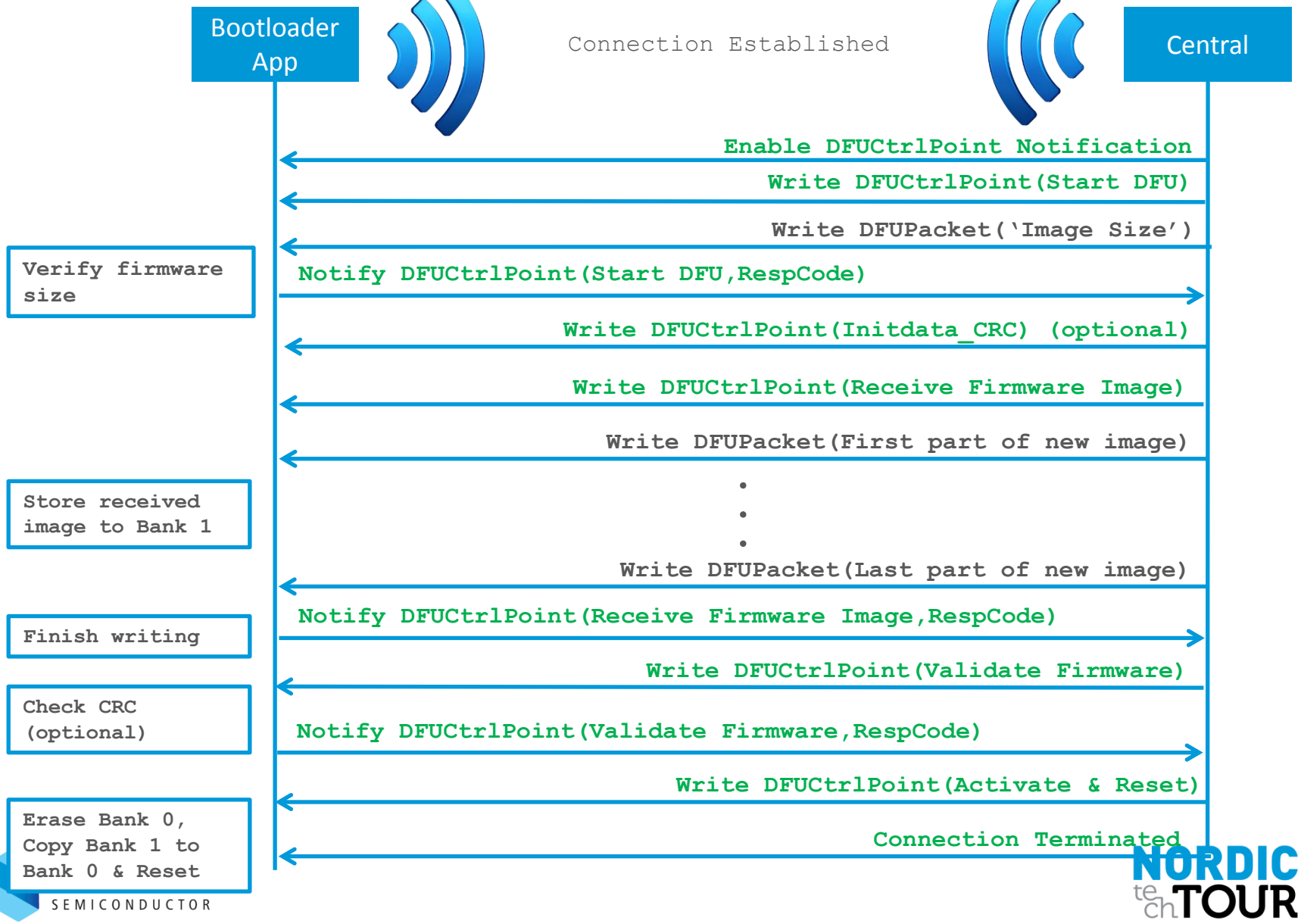

Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing


Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing
Write DFUCtrlPoint(Validate Firmware)


Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing
Write DFUCtrlPoint(Validate Firmware)
Check CRC (optional)


Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing
Write DFUCtrlPoint(Validate Firmware)
Check CRC (optional)
Notify DFUCtrlPoint(Validate Firmware,RespCode)


Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing
Write DFUCtrlPoint(Validate Firmware)
Check CRC (optional)
Notify DFUCtrlPoint(Validate Firmware,RespCode)
Write DFUCtrlPoint(Activate & Reset)
Connection Terminated


Bootloader App
Connection Established
Central
Enable DFUCtrlPoint Notification
Write DFUCtrlPoint(Start DFU)
Write DFUPacket('Image Size')
Verify firmware size
Notify DFUCtrlPoint(Start DFU,RespCode)
Write DFUCtrlPoint(Initdata_CRC) (optional)
Write DFUCtrlPoint(Receive Firmware Image)
Write DFUPacket(First part of new image)
Store received image to Bank 1
Write DFUPacket(Last part of new image)
Notify DFUCtrlPoint(Receive Firmware Image,RespCode)
Finish writing
Write DFUCtrlPoint(Validate Firmware)
Check CRC (optional)
Notify DFUCtrlPoint(Validate Firmware,RespCode)
Write DFUCtrlPoint(Activate & Reset)
Erase Bank 0, Copy Bank 1 to Bank 0 & Reset
Connection Terminated

Competitive Benchmark

nRF51 DFU:

- 15 ms connection interval
- 20 bytes pr packet
- 6 packets pr connection interval

- nRF51 SDK proximity example : 18kB
  - **Update: 5 sec**
- Full App bank update ~80 kB
  - **Update: 15 sec**
- Confirmed by measurements

CC254x OTA:

- 30 ms connection intervals
- 18 Byte pr. packet
- 1 packet pr conn interval

- TI SDK examples:
  - Keyfob: 116 kB
  - Blood glucose: 132 kB
  - OAD (upgrade image): 114 kB
- OAD image: ~114kB
  - **Update: ~3 min**

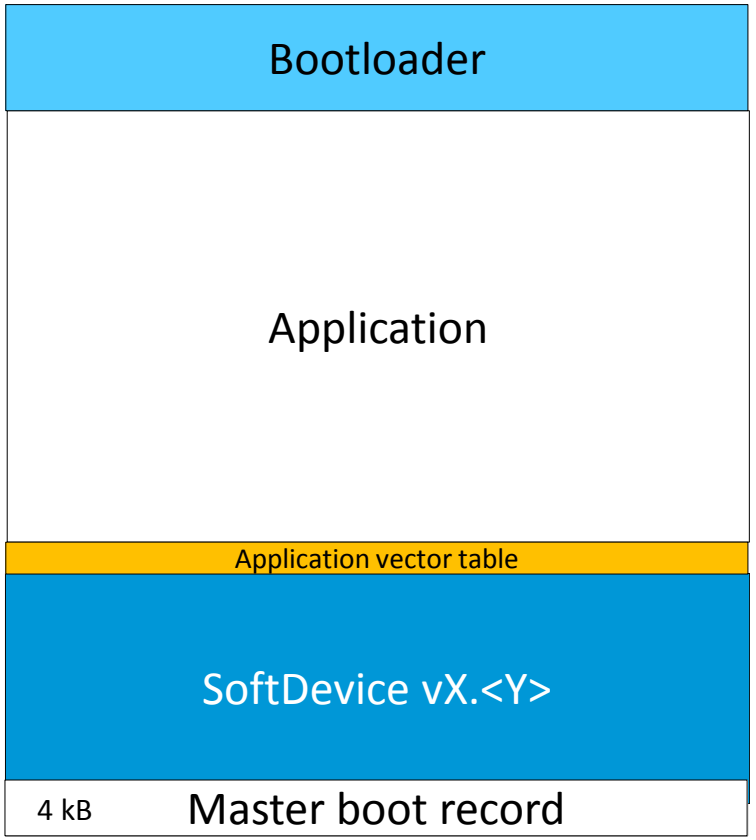
- Confirmed by measurement

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 ?

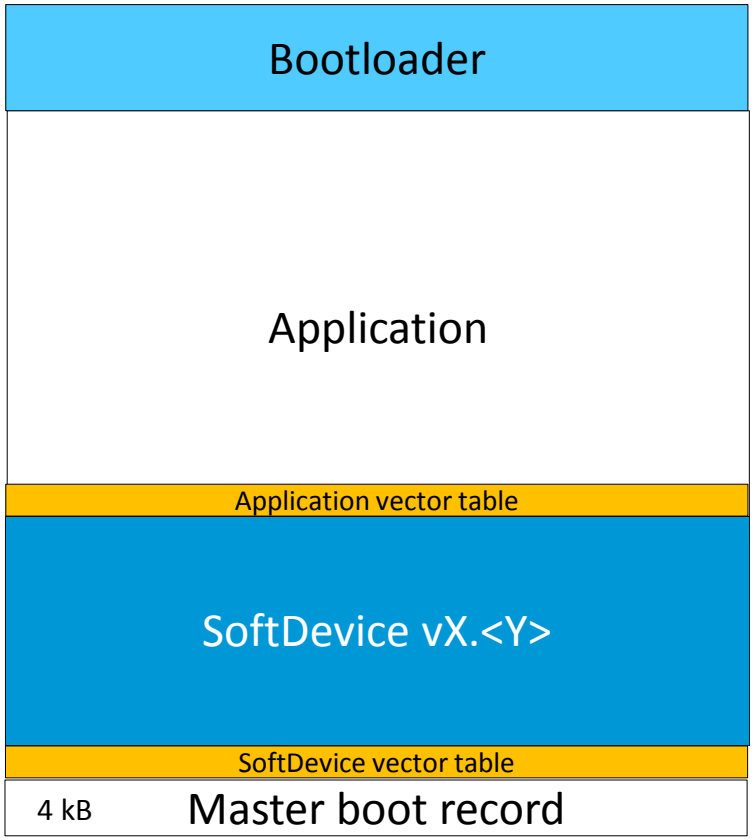
- Add a “Master boot record” under the Softdevice in code region 0 to handle flash updating for softdevice and bootloader.
- Three options:
  - Application update
  - Softdevice update
  - Softdevice + bootloader update



SoftDevice and Bootloader Update

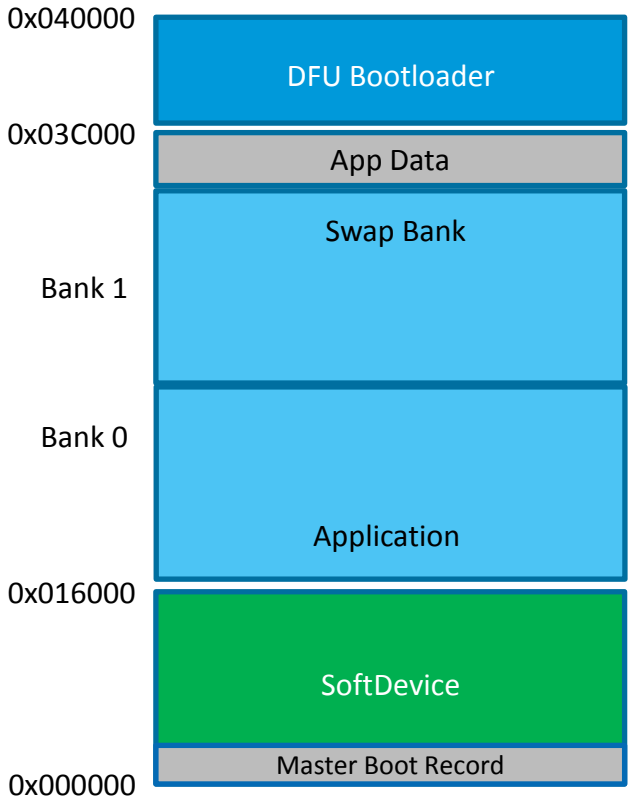
New feature from S110 v7.0 and later

- **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



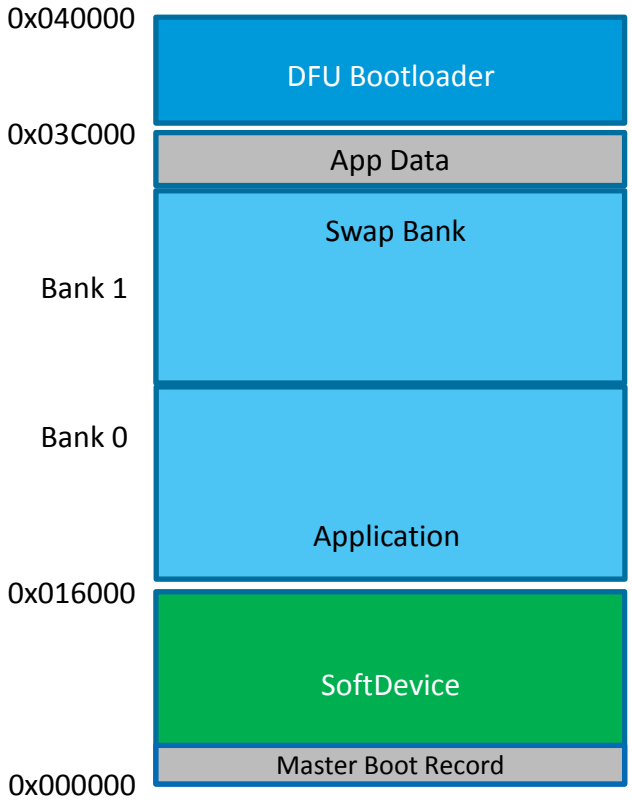
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



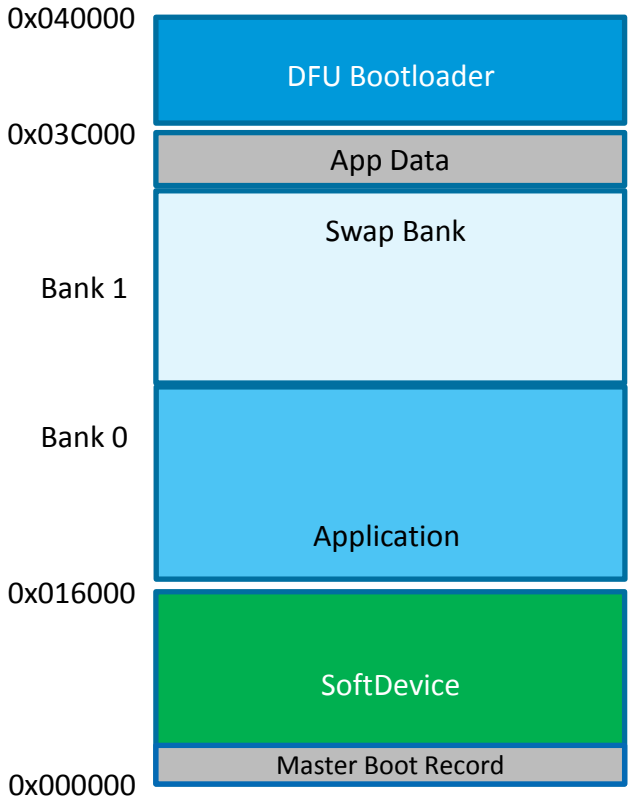
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



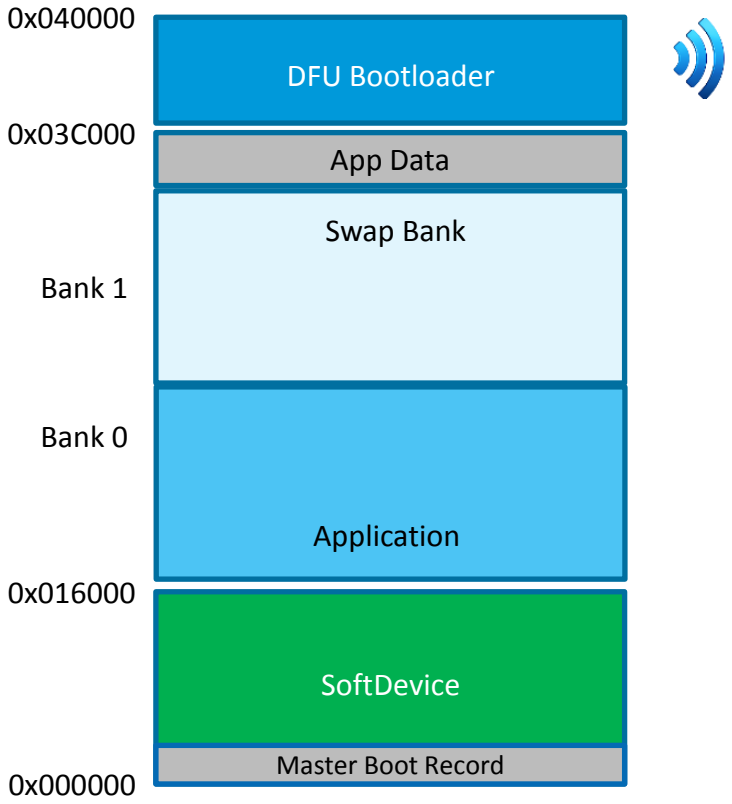
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



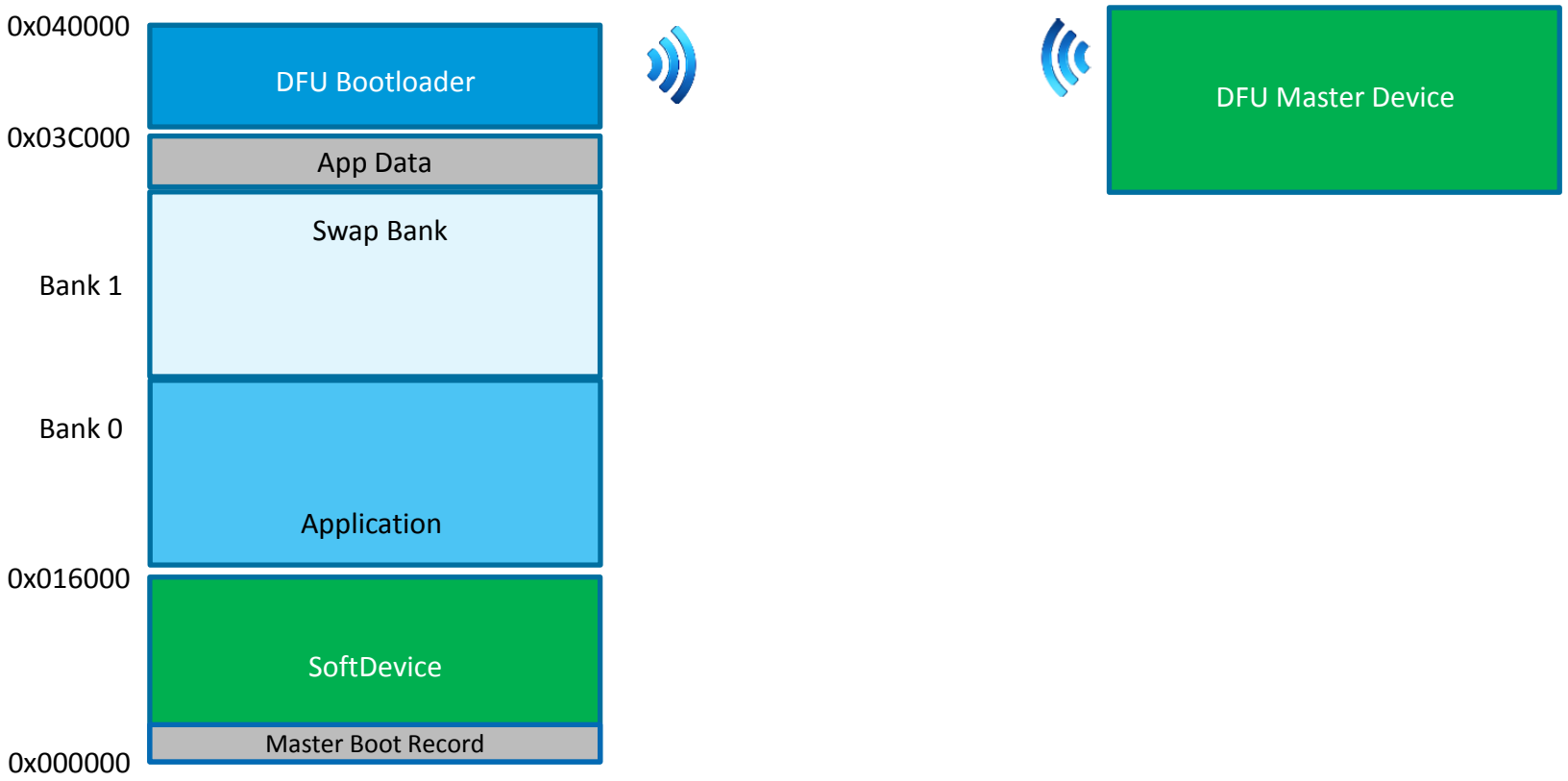
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



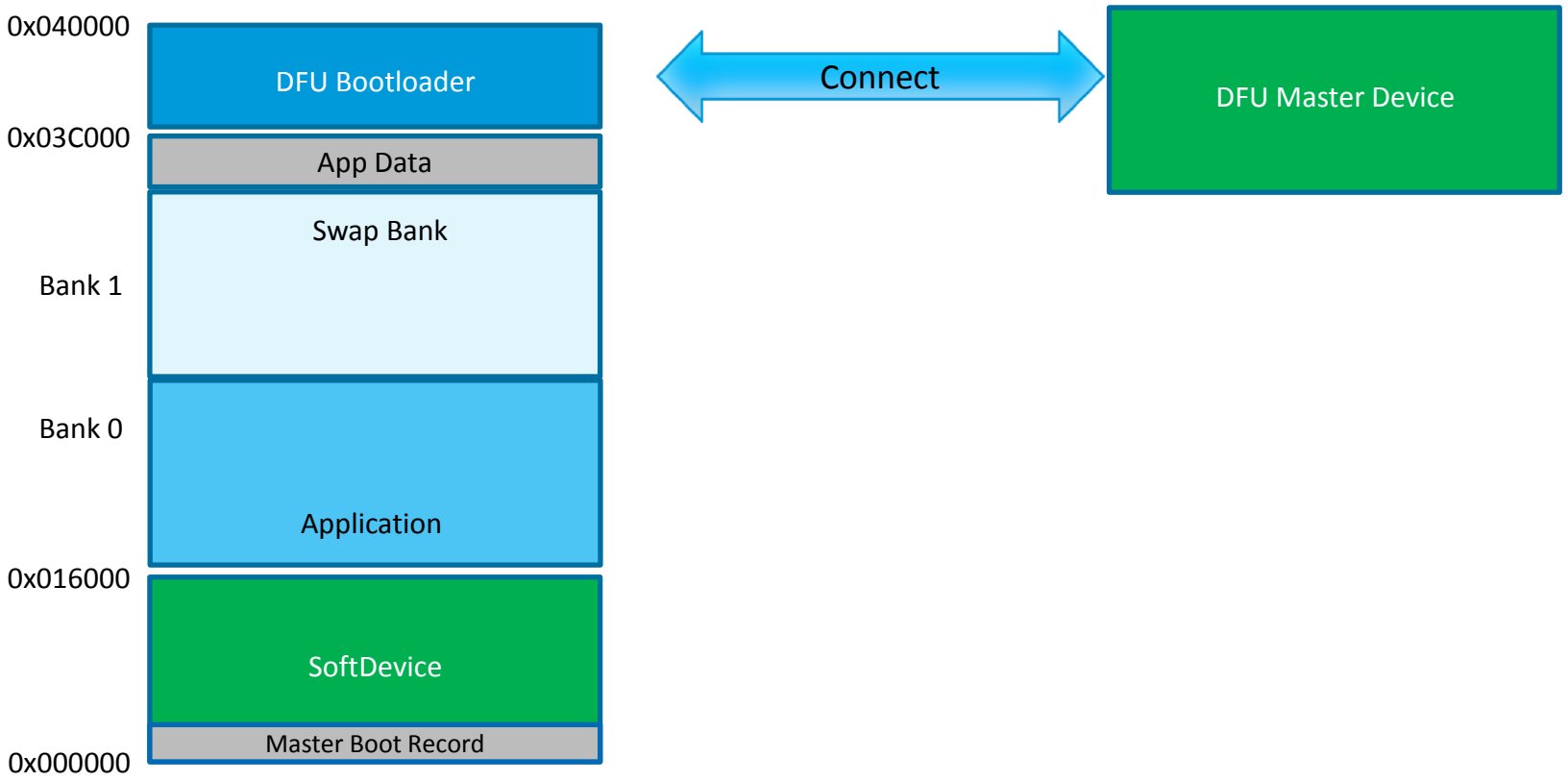
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



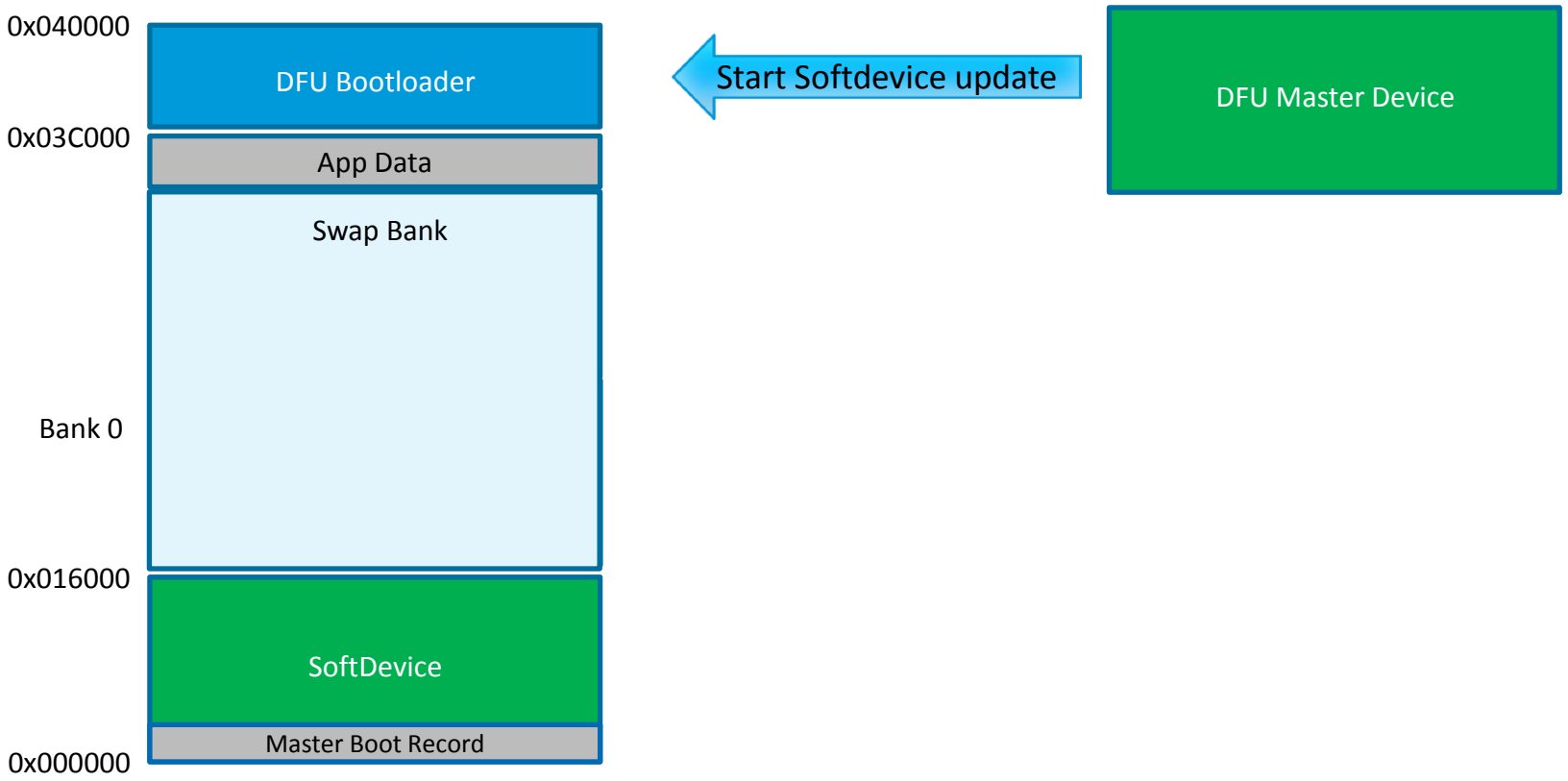
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



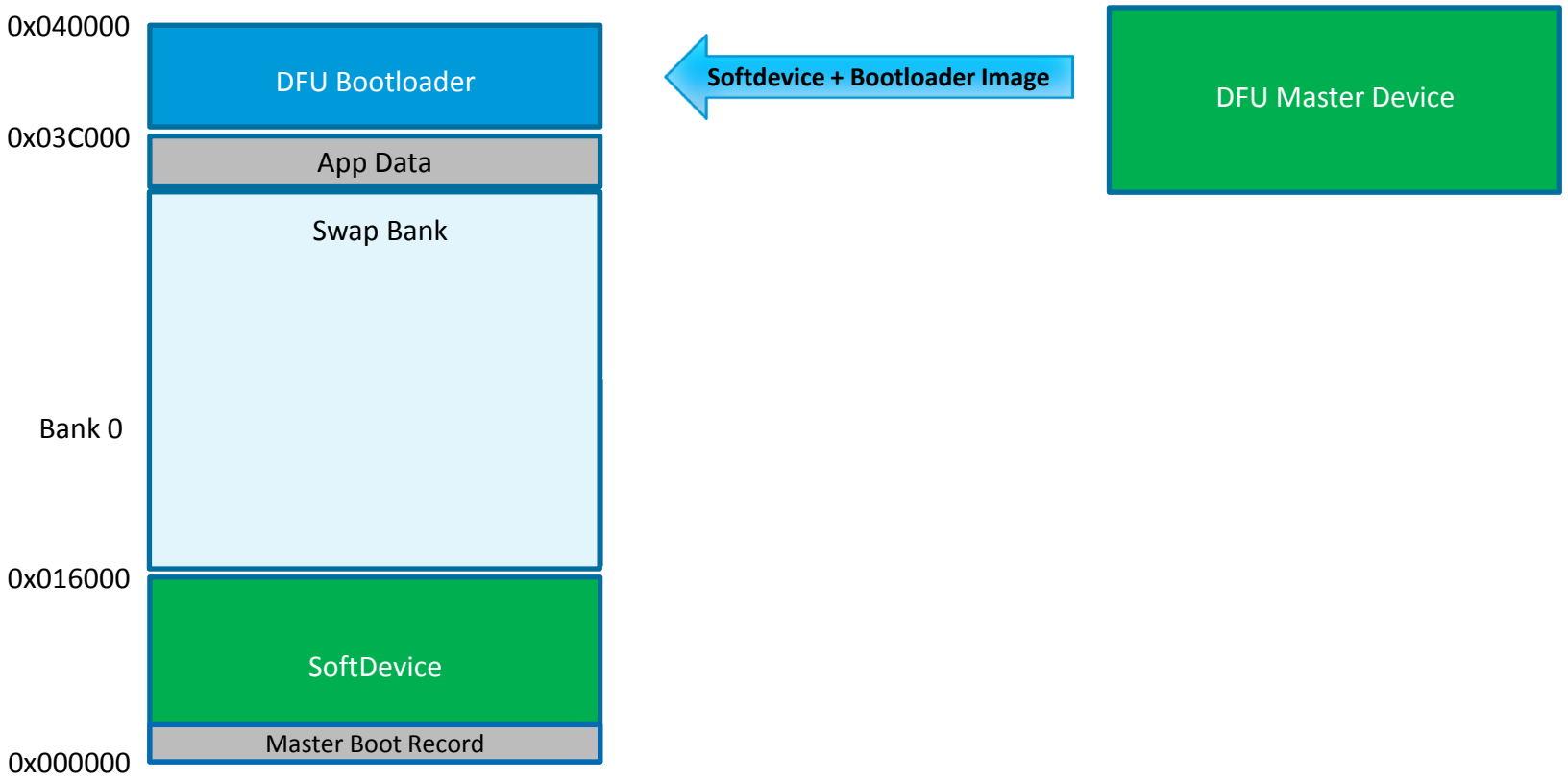
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



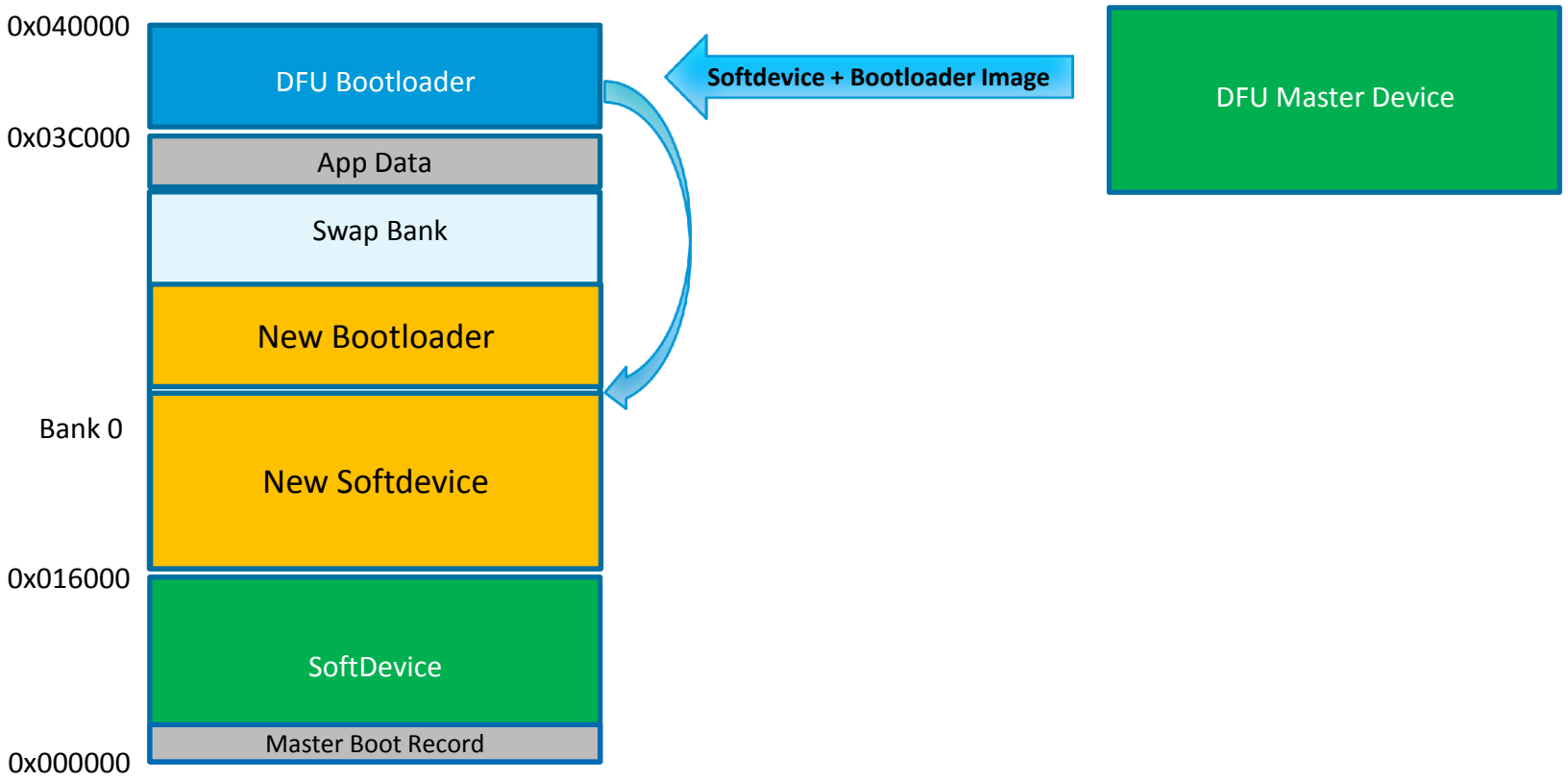
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



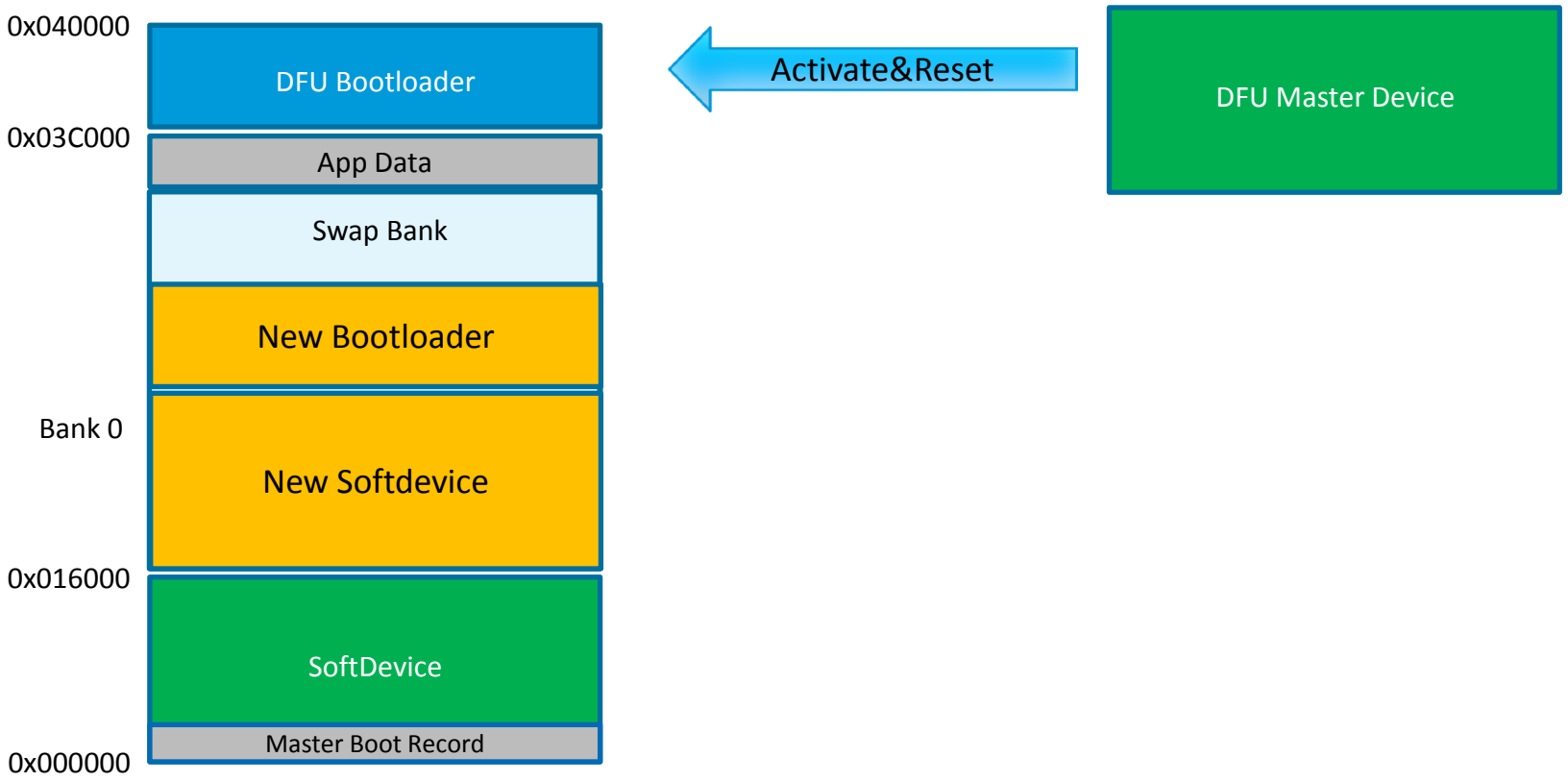
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



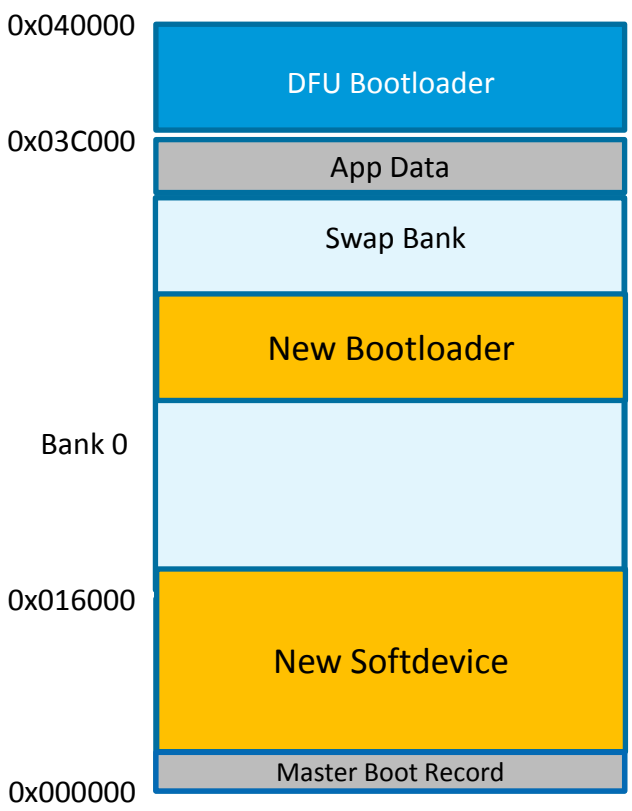
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



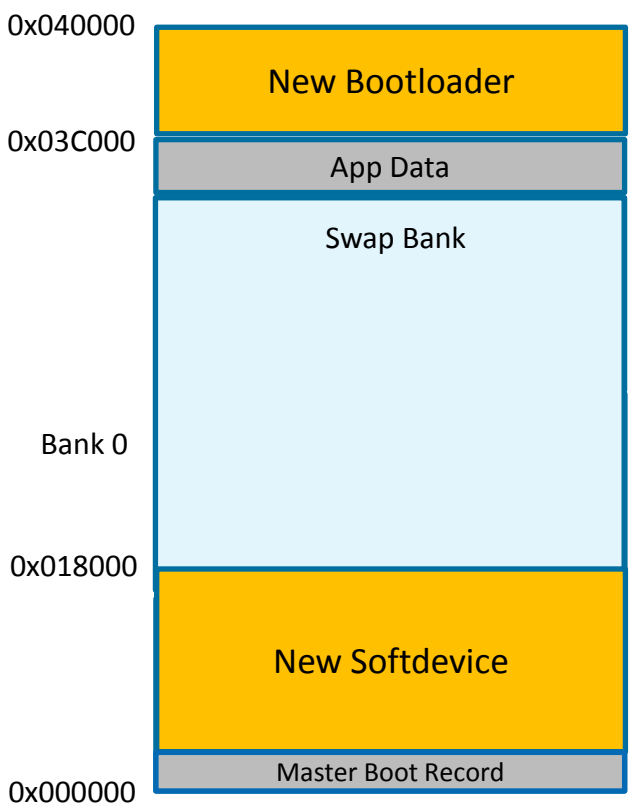
Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



Dual Bank update

Softdevice and bootloader update on a 256KB nRF51822



DFU Master Device

Topics

- Qualification when softdevice is updated ?
- What happens when something fails ?
- How to start bootloader from application (buttonless)
- Pairing issue when switching application and bootloader
- Extra DFU examples on Github

Topics

- Qualification when softdevice is updated ?

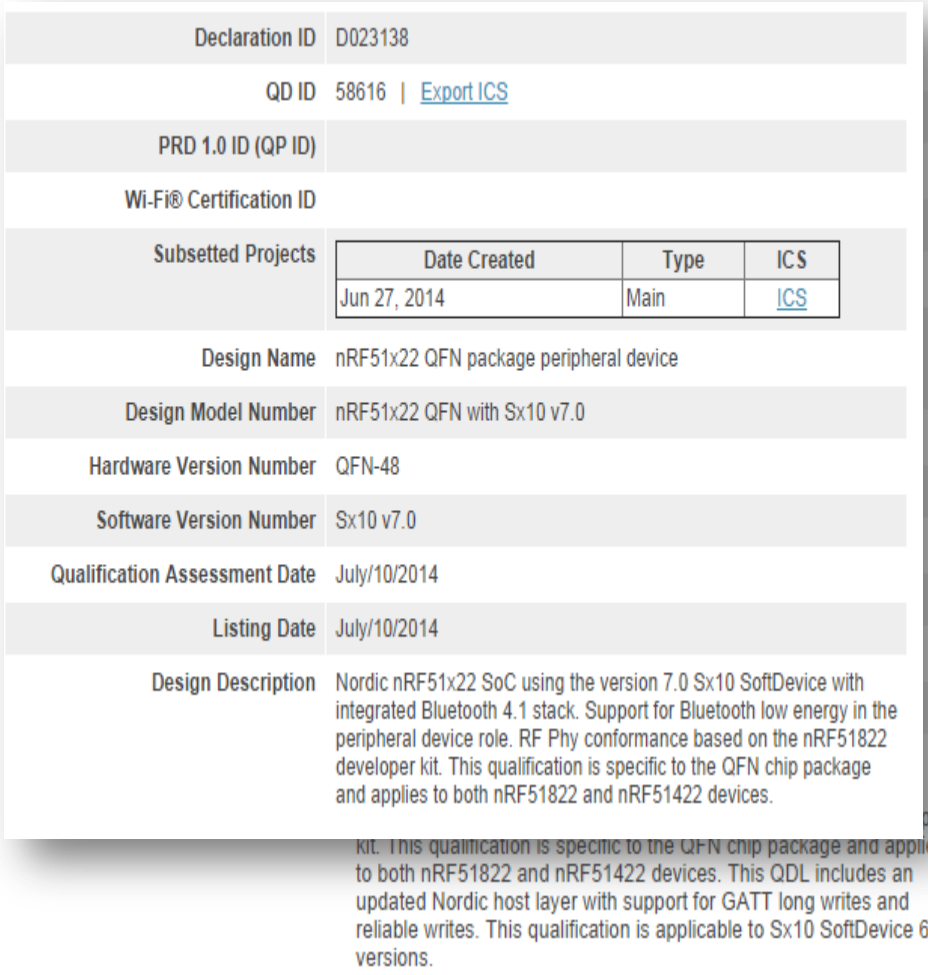
Topics

- Qualification when softdevice is updated ?
  - We're now qualifying softdevice with generic name so that it can be used for different minor softdevice variants (e.g v6.x) or different softdevices sharing the same core (e.g Sx10).

Declaration ID	B022009
QD ID	51744 | Export ICS
PRD 1.0 ID (QP ID)	
Wi-Fi® Certification ID	
Subsetted Projects	Date Created: Dec 5, 2013 / Type: Main / ICS: ICS
Design Name	nRF51x22 IC
Design Model Number	nRF51x22_QF with updated Sx10 stack
Hardware Version Number	QFN package
Software Version Number	6.xx
Qualification Assessment Date	January/30/2014
Listing Date	January/30/2014
Design Description	Nordic nRF51x22 SoC using the Sx10 SoftDevice with integrated Bluetooth 4.0 stack. Support for Bluetooth low energy peripheral device role. RF Phy conformance based on the nRF51822 developer kit. This qualification is specific to the QFN chip package and applies to both nRF51822 and nRF51422 devices. This QDL includes an updated Nordic host layer with support for GATT long writes and reliable writes. This qualification is applicable to Sx10 SoftDevice 6.x versions.

Topics

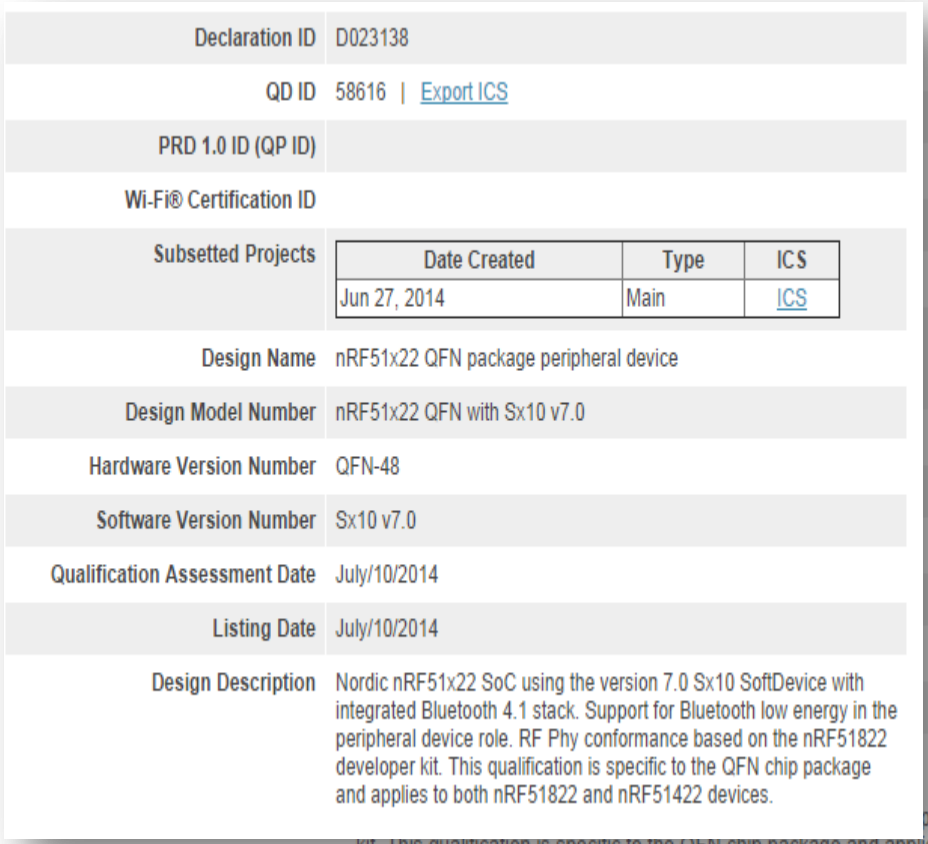
- Qualification when softdevice is updated ?
  - We’re now qualifying softdevice with generic name so that it can be used for different minor softdevice variants (e.g v6.x) or different softdevices sharing the same core (e.g Sx10).



Declaration ID	D023138
QD ID	58616 | Export ICS
PRD 1.0 ID (QP ID)	
Wi-Fi® Certification ID	
Subsetted Projects	Date Created: Jun 27, 2014 / Type: Main / ICS: ICS
Design Name	nRF51x22 QFN package peripheral device
Design Model Number	nRF51x22 QFN with Sx10 v7.0
Hardware Version Number	QFN-48
Software Version Number	Sx10 v7.0
Qualification Assessment Date	July/10/2014
Listing Date	July/10/2014
Design Description	Nordic nRF51x22 SoC using the version 7.0 Sx10 SoftDevice with integrated Bluetooth 4.1 stack. Support for Bluetooth low energy in the peripheral device role. RF Phy conformance based on the nRF51822 developer kit. This qualification is specific to the QFN chip package and applies to both nRF51822 and nRF51422 devices.

Topics

- Qualification when softdevice is updated ?
  - We're now qualifying softdevice with generic name so that it can be used for different minor softdevice variants (e.g v6.x) or different softdevices sharing the same core (e.g Sx10).
  - «If customers don't use new features, they don't need new QDL.»



Declaration ID	D023138
QD ID	58616 | Export ICS
PRD 1.0 ID (QP ID)	
Wi-Fi® Certification ID	
Subsetted Projects	Date Created: Jun 27, 2014 / Type: Main / ICS: ICS
Design Name	nRF51x22 QFN package peripheral device
Design Model Number	nRF51x22 QFN with Sx10 v7.0
Hardware Version Number	QFN-48
Software Version Number	Sx10 v7.0
Qualification Assessment Date	July/10/2014
Listing Date	July/10/2014
Design Description	Nordic nRF51x22 SoC using the version 7.0 Sx10 SoftDevice with integrated Bluetooth 4.1 stack. Support for Bluetooth low energy in the peripheral device role. RF Phy conformance based on the nRF51822 developer kit. This qualification is specific to the QFN chip package and applies to both nRF51822 and nRF51422 devices.

Topics

- What happens when something fails ?
  - Application Single bank update
    - If transfer failed, the old application is lost. The DFU will be restarted => unfunctional until a successful DFU. => not bricked
  - Application Dual bank update
    - If tranfer failed, the old application is untouched. The DFU update can be restarted or not. => not bricked.
    - If glitch happens, or power failure during flash swapping, bootloader will compare CRC and jump back to DFU mode => not bricked.
  - Non-functional application updated
    - If a buggy application is updated. DFU Bootloader won't be able to detect that with CRC, but end user can always reset the device in bootloader mode and reupdate a functional application => Not bricked.

Note: Here assumed that bootloader is not the buttonless bootloader. End user can start DFU mode with a button

Topics

- What happens when something fails ?
  - Softdevice update
    - Same as application dual bank update. Old softdevice will not be altered until correct image copied. The softdevice update can be restarted => not bricked
    - If glitch happens or or power failure during flash swaping, bootloader+MBR will try to swap again old softdevice with new softdevice image until CRC matched. => not bricked.
  - Bootloader update
    - Same as softdevice update. => not bricked
  - If a non-functional bootloader or softdevice is updated => **bricked**.

Topics

- How to start bootloader from application (buttonless)

  Two approaches, both uses GPREGRET to store a flag that tells bootloader to enter DFU or start application.

  - **Classic approach:**
    - Application set GPREGRET
    - **Reset** => Enter bootloader => bootloader checks for GPREGRET & decide
    - No example provided
  - **New approach from SDK v6.1**
    - Application sets GPREGRET
    - Jump to bootloader reset handler, **with softdevice kept initialized** => bootloader check for GPREGRET => decides
    - Example provided, DFU service can be easily integrated into customer application
    - Full DFU service is integrated, can be used to avoid the issue that ATT table is cached. (top part of the table, from GAP service to DFU service remain when changing from application to bootloader.

Topics

- Pairing issue when switching application and bootloader
  - Currently bootloader doesn't support bonding /pairing
  - When application bonded to central, switching to bootloader made it impossible to reconnect since the master will try to encrypt the link with the key that bootloader doesn't have.
  - Solution ?
    - Simplest: Change the device address to another (preferred random) address when switching to bootloader mode. The central will treat the bootloader as a new device. The old address can be added to the advertising packet if customer wants the central to detect the bootloader device automatically.
    - More complicated: Share the bond information between the bootloader and the application. Service change indication may needed to update the ATT table. Currently no example yet.
    - Even more complicated (but maybe not better): Integrate the bootloader into the application, start updating firmware right when in application mode.

Topics

- Extra DFU examples on Github

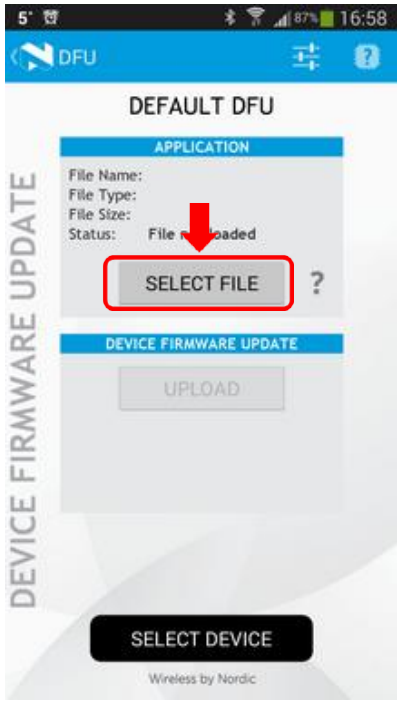
  Unofficial examples made by Nordic techsupport on customer needs
  - Single bank DFU via BLE
  - DFU bootloader for 128kB variant
  - DFU bootloader that compatible with gcc
  - DFU bootloader for the nRF51 Evaluation Kit
- Can be downloaded here: https://github.com/NordicSemiconductor

Demo

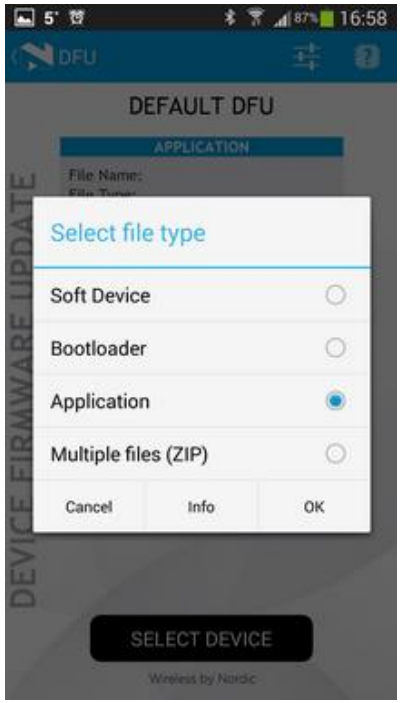
Demo


16:58
DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status: File not loaded
SELECT FILE
?
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic

Demo


16:58
DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic

Demo


16:58
DFU
DEFAULT DFU
APPLICATION
File Name:
DEVICE FIRMWARE UPDATE
Select file type
Soft Device
Bootloader
Application
Multiple files (ZIP)
Cancel
Info
OK
SELECT DEVICE
Wireless by Nordic

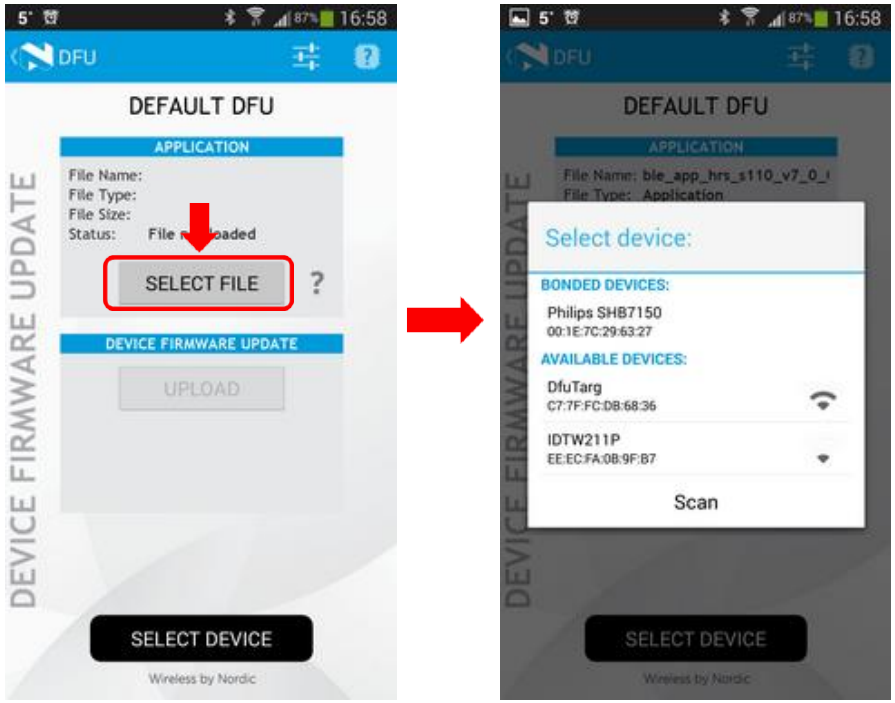
Demo


16:58
DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
?
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic
16:58
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
?
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic

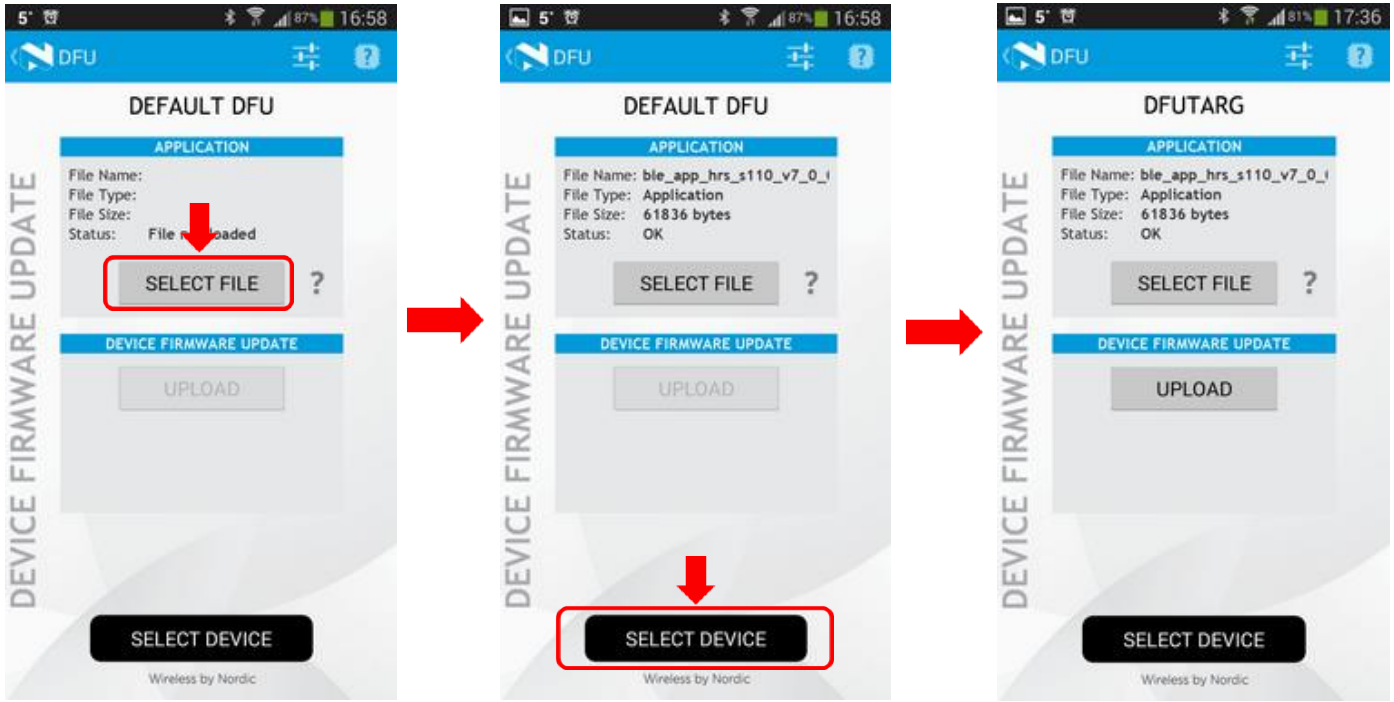
Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE

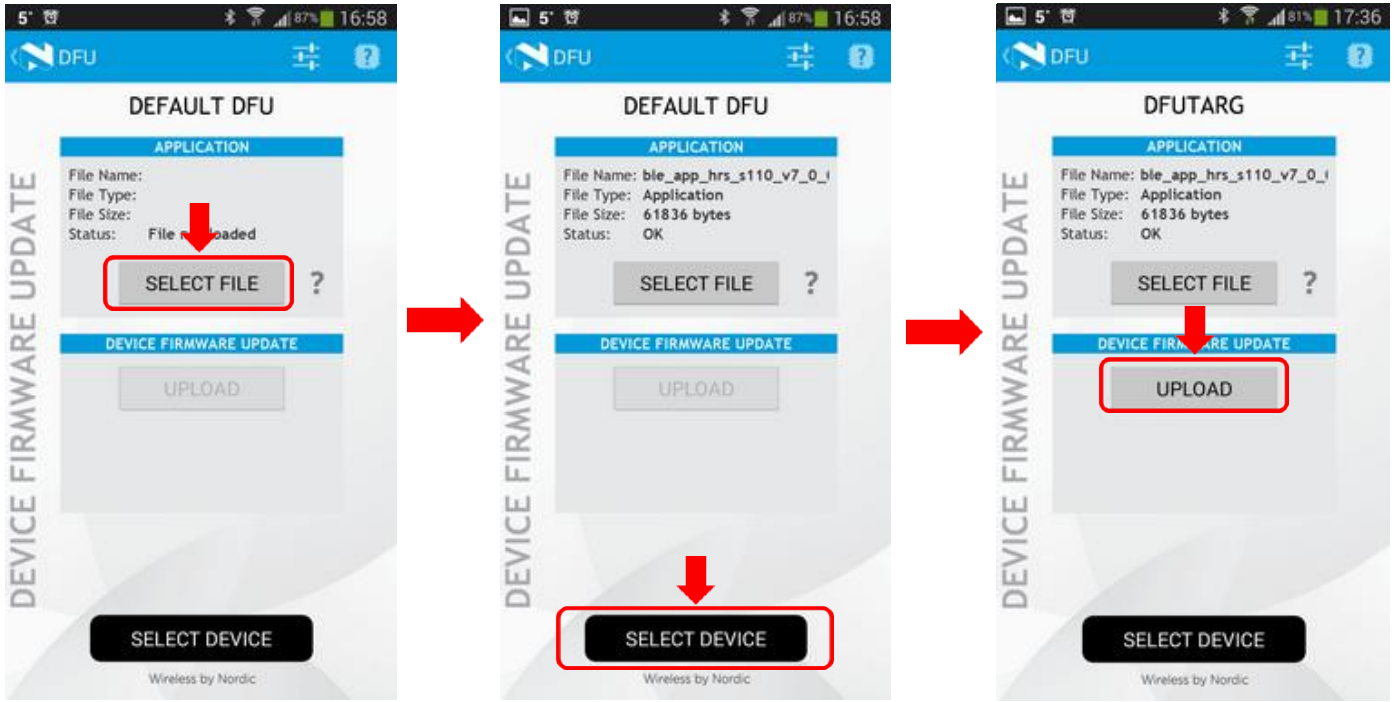
Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic
Select device:
BONDED DEVICES:
Philips SHB7150
00:1E:7C:29:63:27
AVAILABLE DEVICES:
DfuTarg
C7:7F:FC:DB:68:36
IDTW211P
EE:EC:FA:0B:9F:B7
Scan

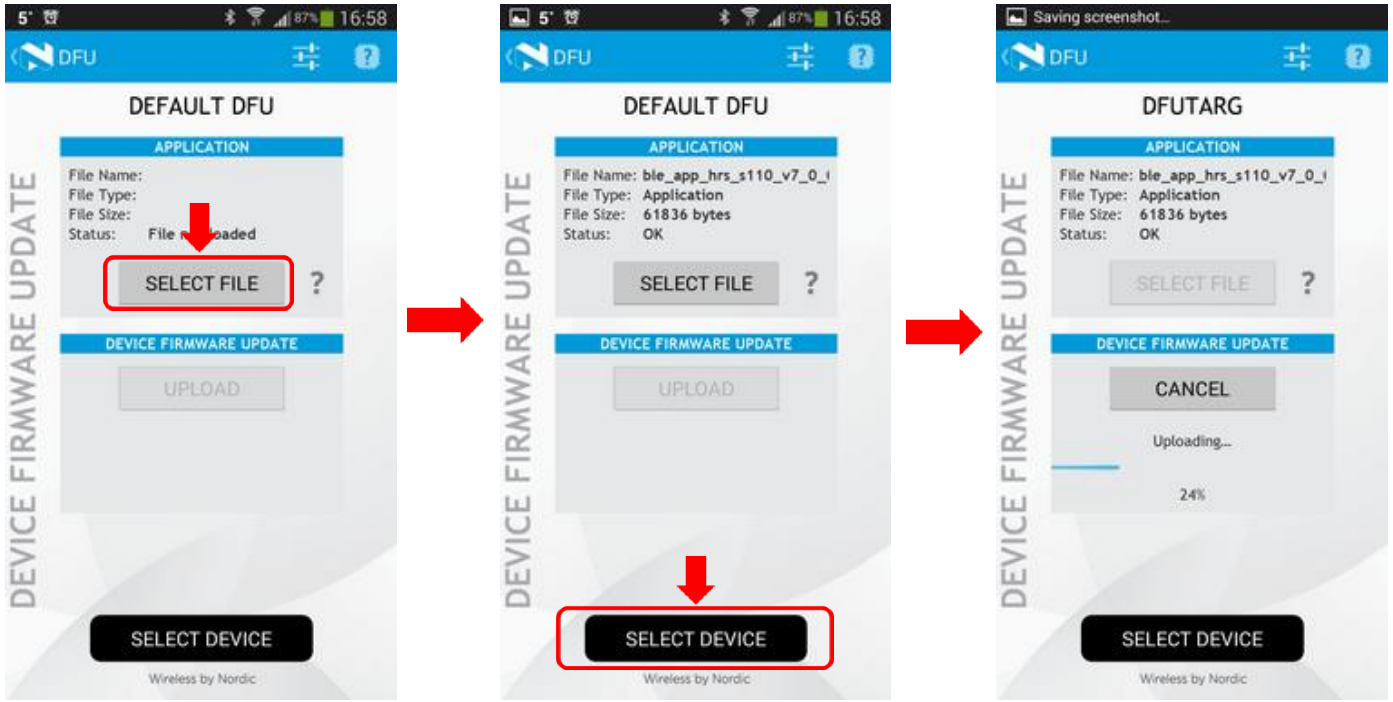
Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DFUTARG
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE

Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DFUTARG
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE

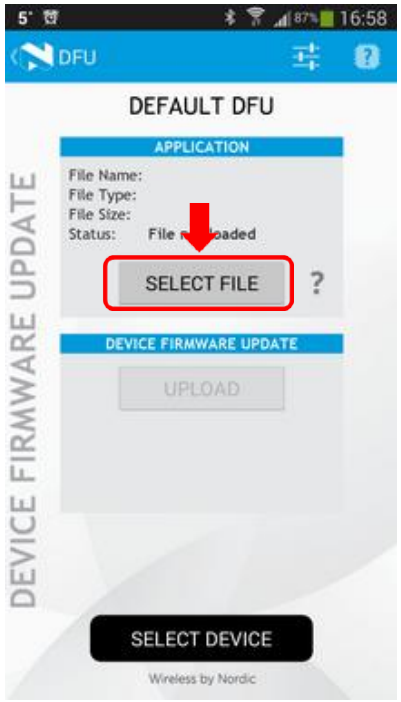
Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
DEVICE FIRMWARE UPDATE
SELECT DEVICE
Wireless by Nordic
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
Saving screenshot...
DFU
DFUTARG
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
CANCEL
Uploading...
24%
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE

Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE
DFU
DFUTARG
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE

Demo


DFU
DEFAULT DFU
APPLICATION
File Name:
File Type:
File Size:
Status:
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE


DFU
DEFAULT DFU
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
DEVICE FIRMWARE UPDATE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE


DFU
DFUTARG
APPLICATION
File Name: ble_app_hrs_s110_v7_0_
File Type: Application
File Size: 61836 bytes
Status: OK
SELECT FILE
UPLOAD
SELECT DEVICE
Wireless by Nordic
DEVICE FIRMWARE UPDATE


NORDIC_HRM
Finger
230
HEART RATE MONITOR
DISCONNECT

Q&A

Nordic Tech support

October, 2014

nRF51 SoC Device Firmware Update (DFU)

NORDIC SEMICONDUCTOR