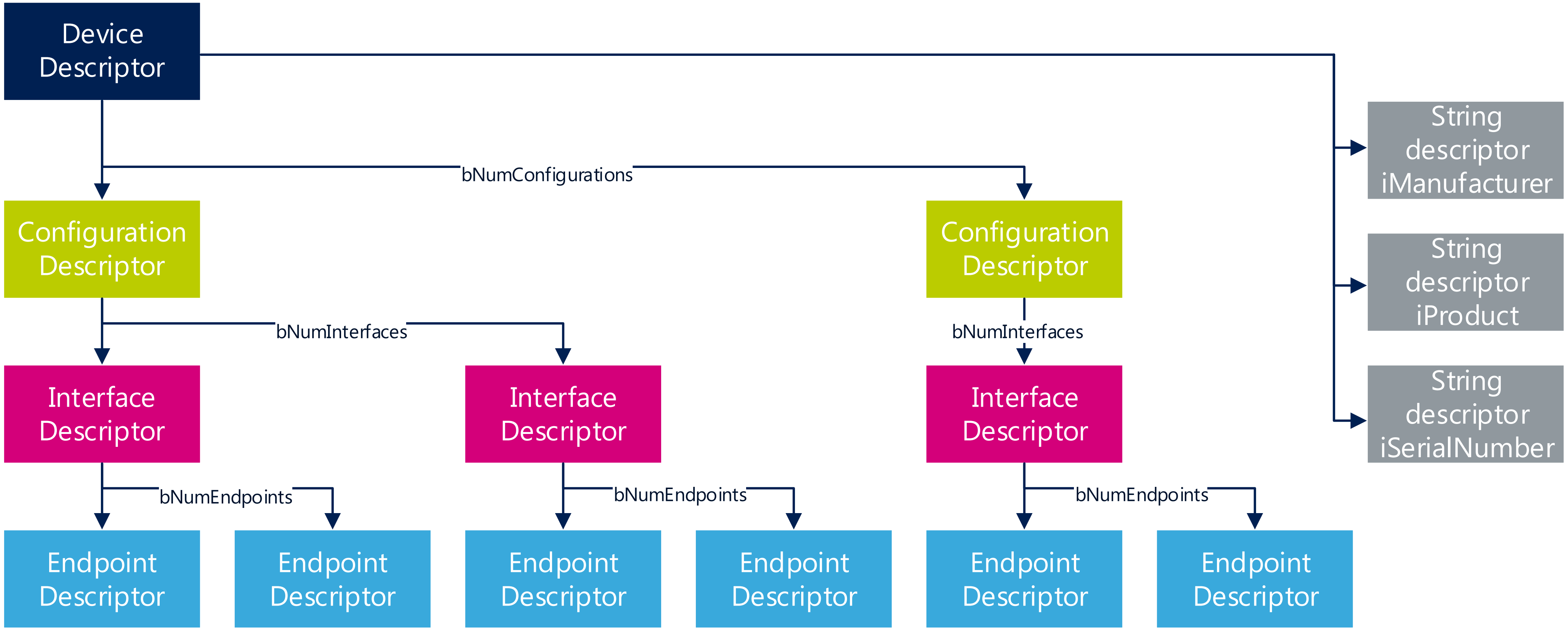
USB descriptors

USB descriptors

- The HOST using it to get information about device
  - If device don’t have the descriptor it is answering by STALL
- Each descriptor has 2-byte header specifying descriptor size and type
- String descriptors
  - Are referenced using index from other descriptors
  - Can be translated for different languages

Descriptor structure

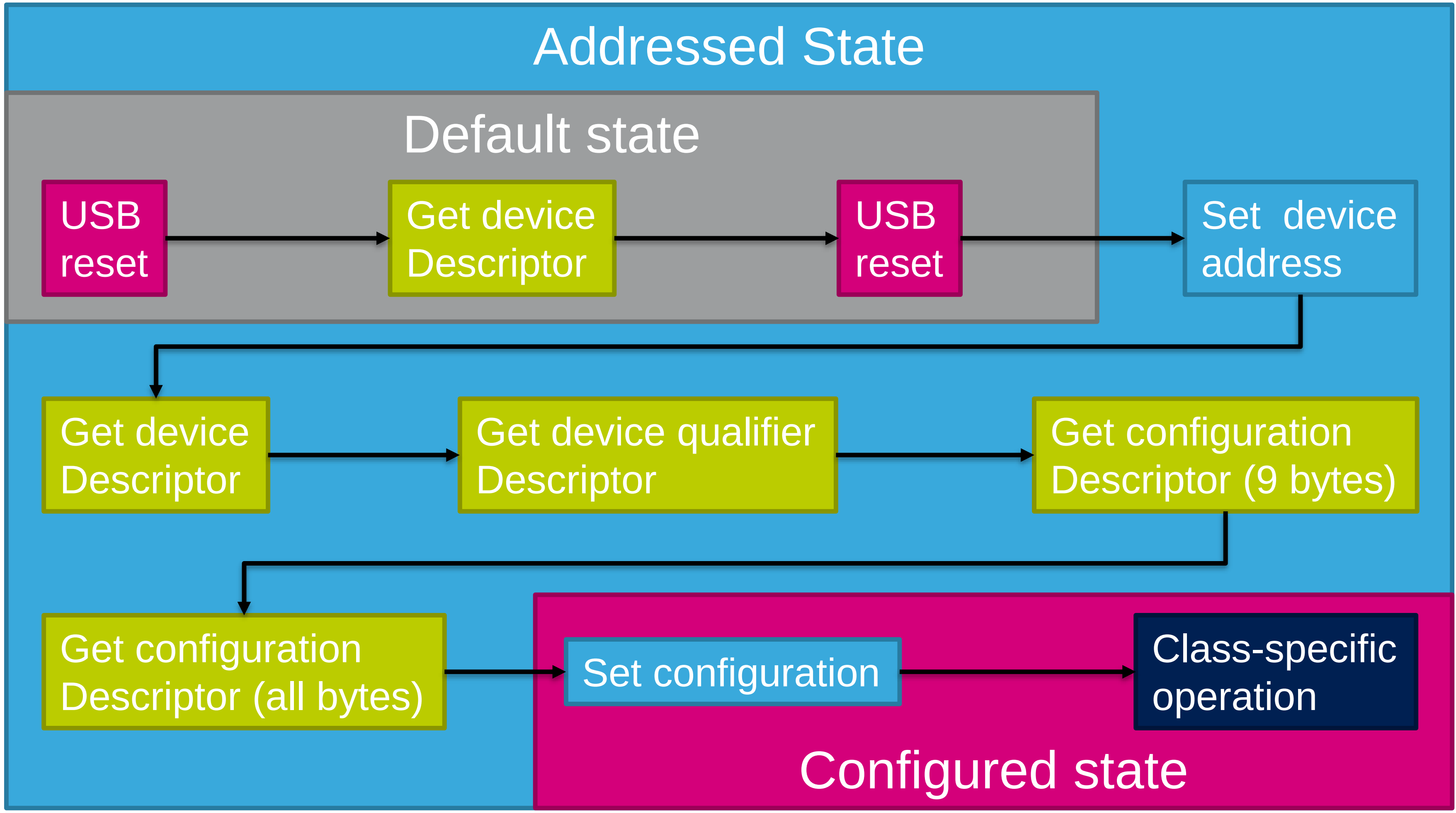


life.augmented

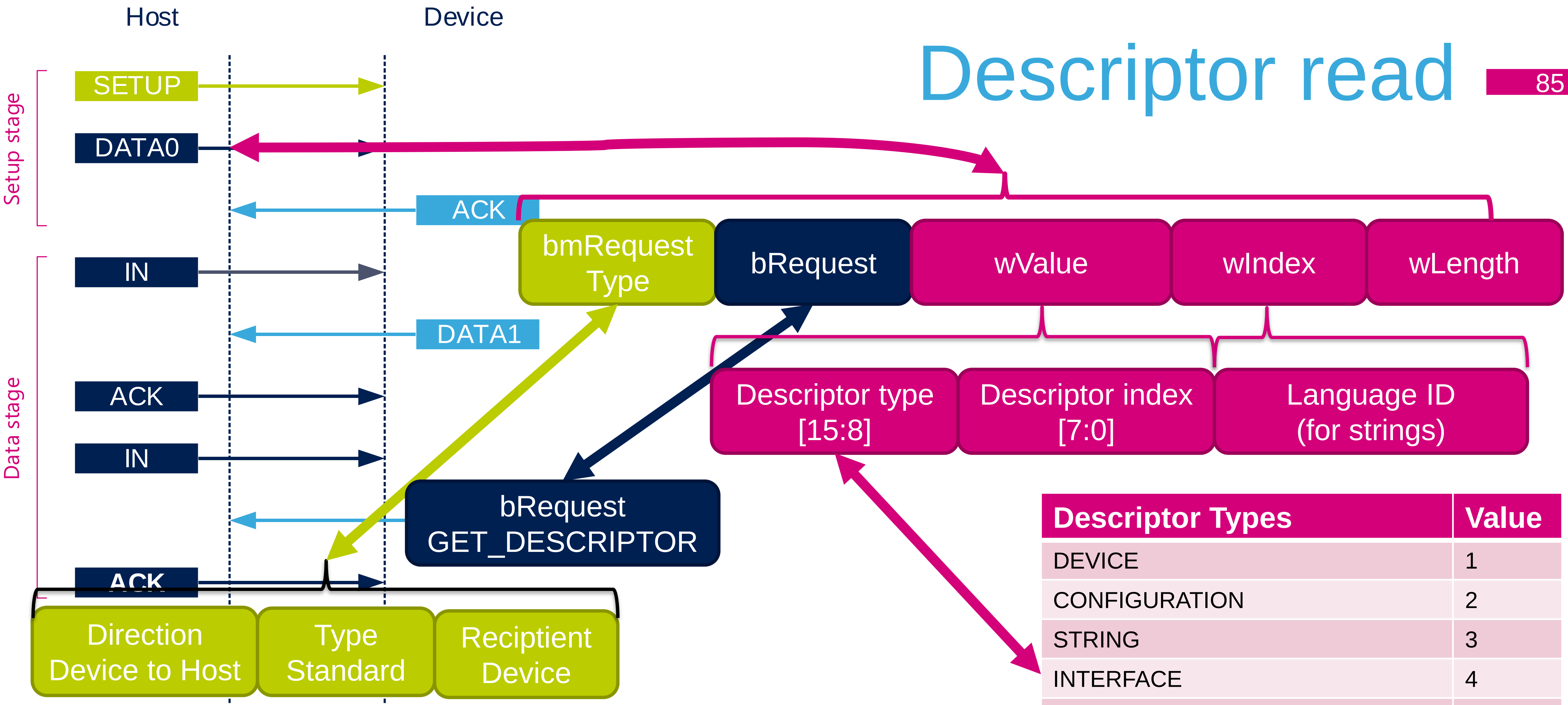
Example of enumeration process

- Order of some steps can depend on USB host implementation
  - USB specification doesn't specify the exact order – device should be prepared for everything

- Additional steps might be taken by the class-specific driver



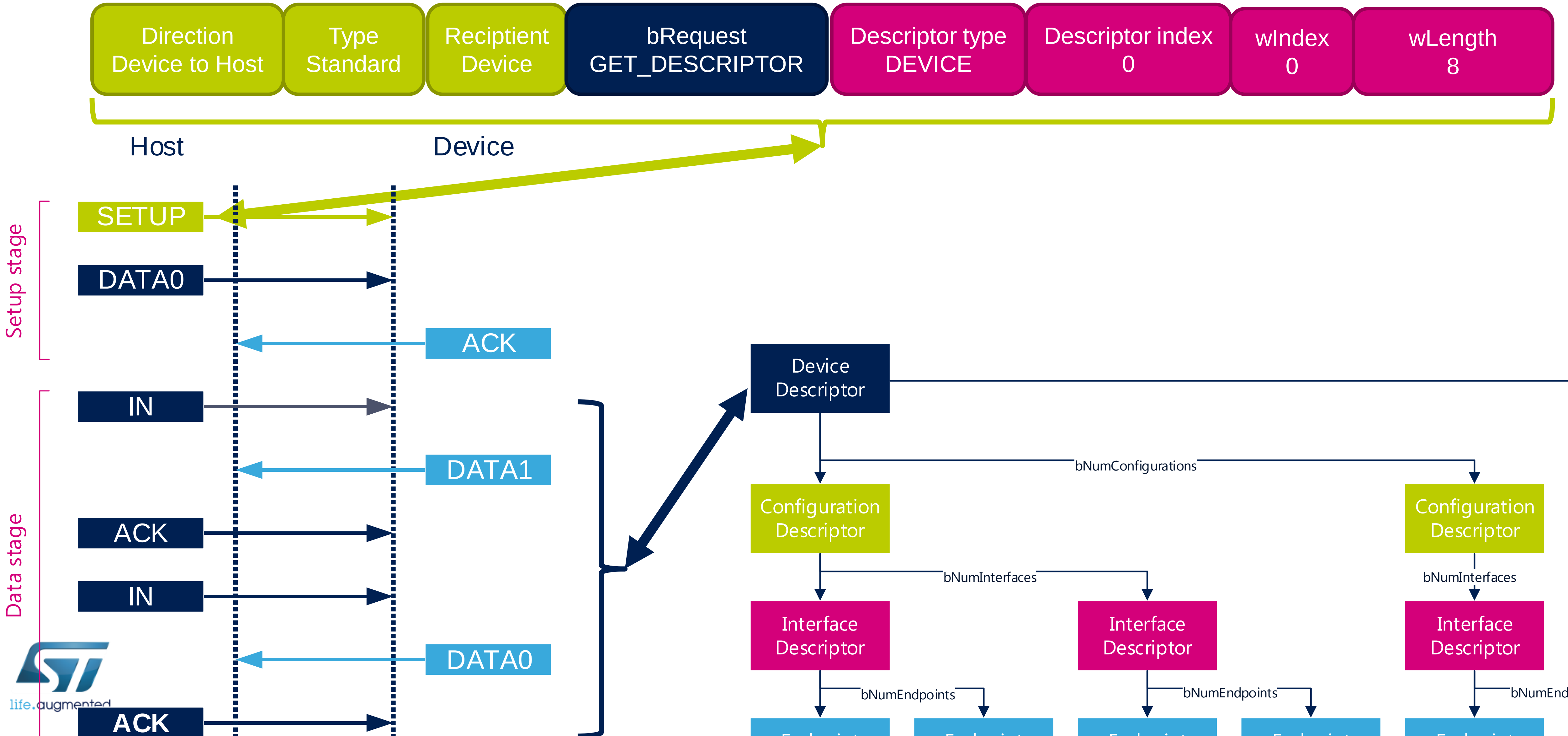
Descriptor read



Descriptor Types	Value
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7
INTERFACE_POWER	8

- If descriptor is not available or not supported send STALL

Reading Device Descriptor



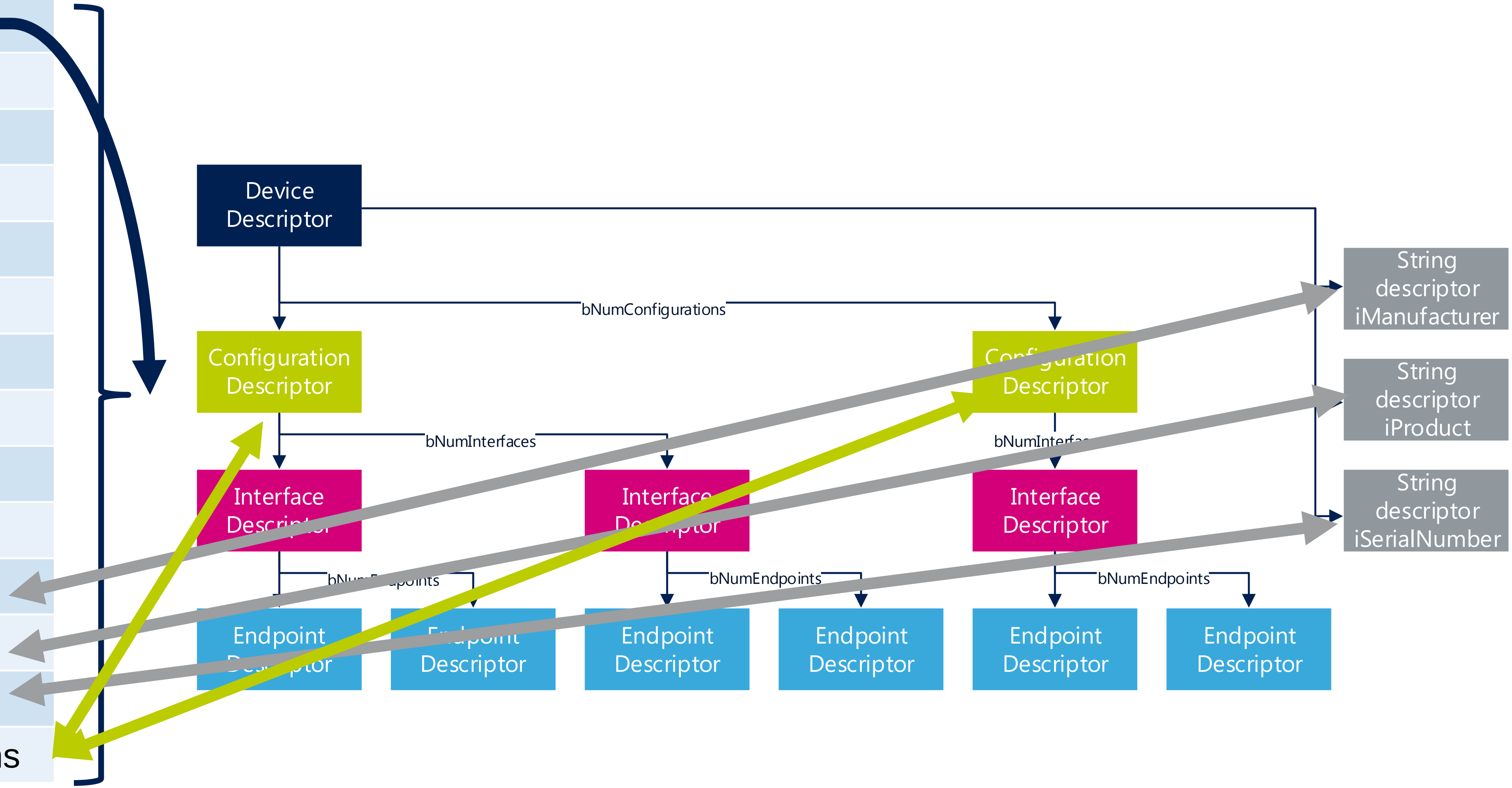
Direction	Type	Recipient	bRequest	Descriptor type	Descriptor index	wIndex	wLength
Device to Host	Standard	Device	GET_DESCRIPTOR	DEVICE	0	0	8

Device descriptor

Name	Description
bLength	Length of the descriptor (9 – bytes)
bDescriptorType	DEVICE (value = 1)
bcdUSB	USB version (e.g. 0x0200 => version 2.0)
bDeviceClass	Code for device standard class
bDeviceSubClass	Code for device standard subclass
bDeviceProtocol	Code for device protocol associated with the class
bMaxPacketSize0	Maximum packet size for endpoint 0
idVendor	Vendor ID assigned by USB-IF
idProduct	Product ID assigned by manufacturer
bcdDevice	Version of the device
iManufacturer	Description of the manufacturer
iProduct	Description of the product
iSerialNumber	Serial number (string)
bNumConfigurations	Number of configurations

Device descriptor

Name
bLength
bDescriptorType
bcdUSB
bDeviceClass
bDeviceSubClass
bDeviceProtocol
bMaxPacketSize0
idVendor
idProduct
bcdDevice
iManufacturer
iProduct
iSerialNumber
bNumConfigurations



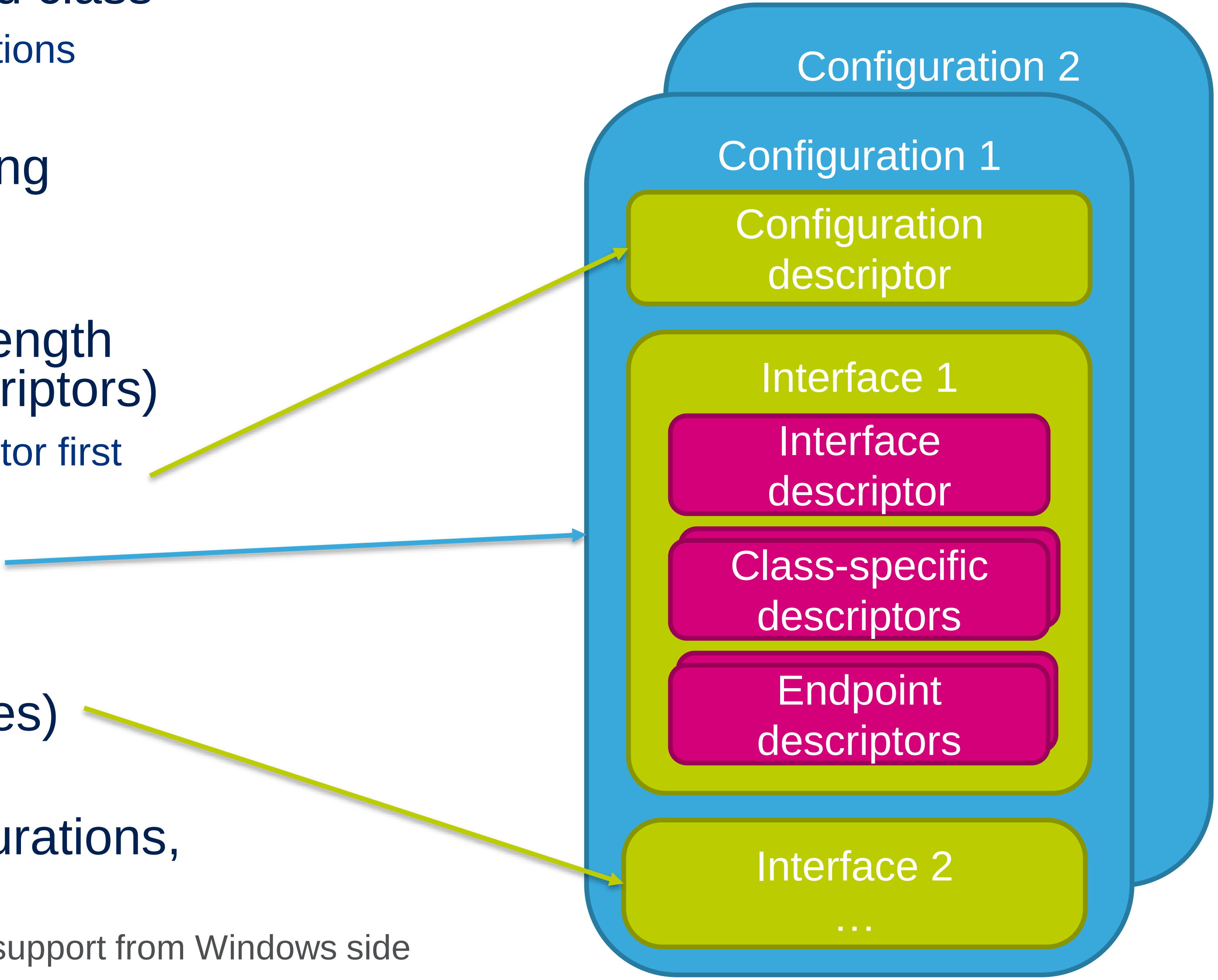
Device Qualifier descriptor

- Introduced in USB 2.0
  - Required for higher speeds of the USB (high-speed, super-speed)
  - Full-speed only device must respond with error (STALL)
- Contains device descriptor for other speed:
  - USB version
  - Class, subclass and protocol
  - Maximum size for endpoint 0
  - Number of configurations
- Value depends on selected speed
  - E.g. when device is in high-speed it returns values for full-speed

Name
bLength
bDescriptorType
bcdUSB
bDeviceClass
bDeviceSubClass
bDeviceProtocol
bMaxPacketSize0
bNumConfigurations
bReserved (must be zero)

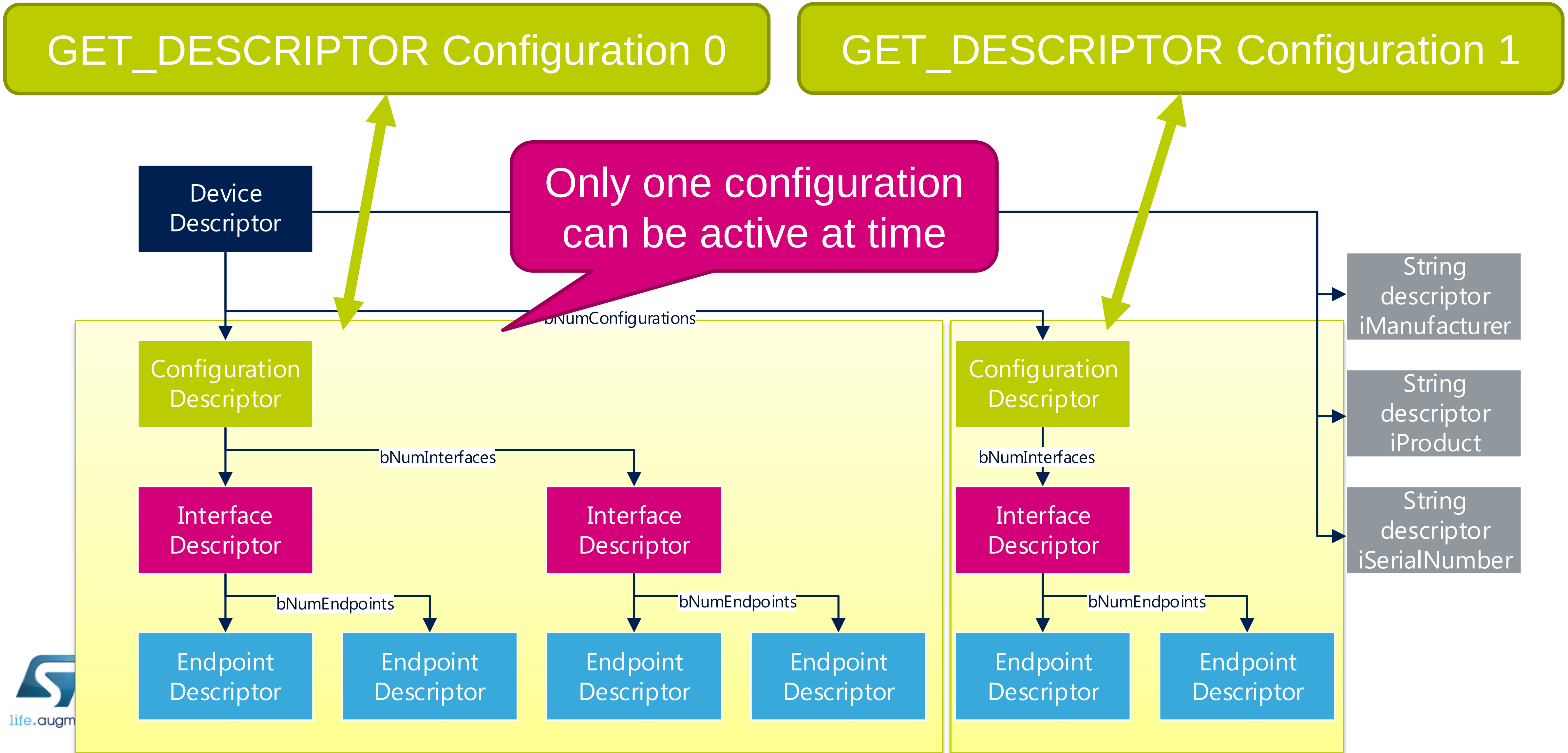
Configuration descriptor

- Contains description of the selected class
  - One device can have multiple configurations
- Contains other descriptors specifying the interfaces
- Configuration descriptor contains length of the whole configuration (all descriptors)
  - Host usually reads configuration descriptor first (to know the size)
  - And the reads the whole configuration
- Multiple interfaces can be used at the same time (composite devices)
- Device can contain multiple configurations, but only one can be active
  - Not recommended, since there is poor support from Windows side



life.augmented

Reading configuration descriptor



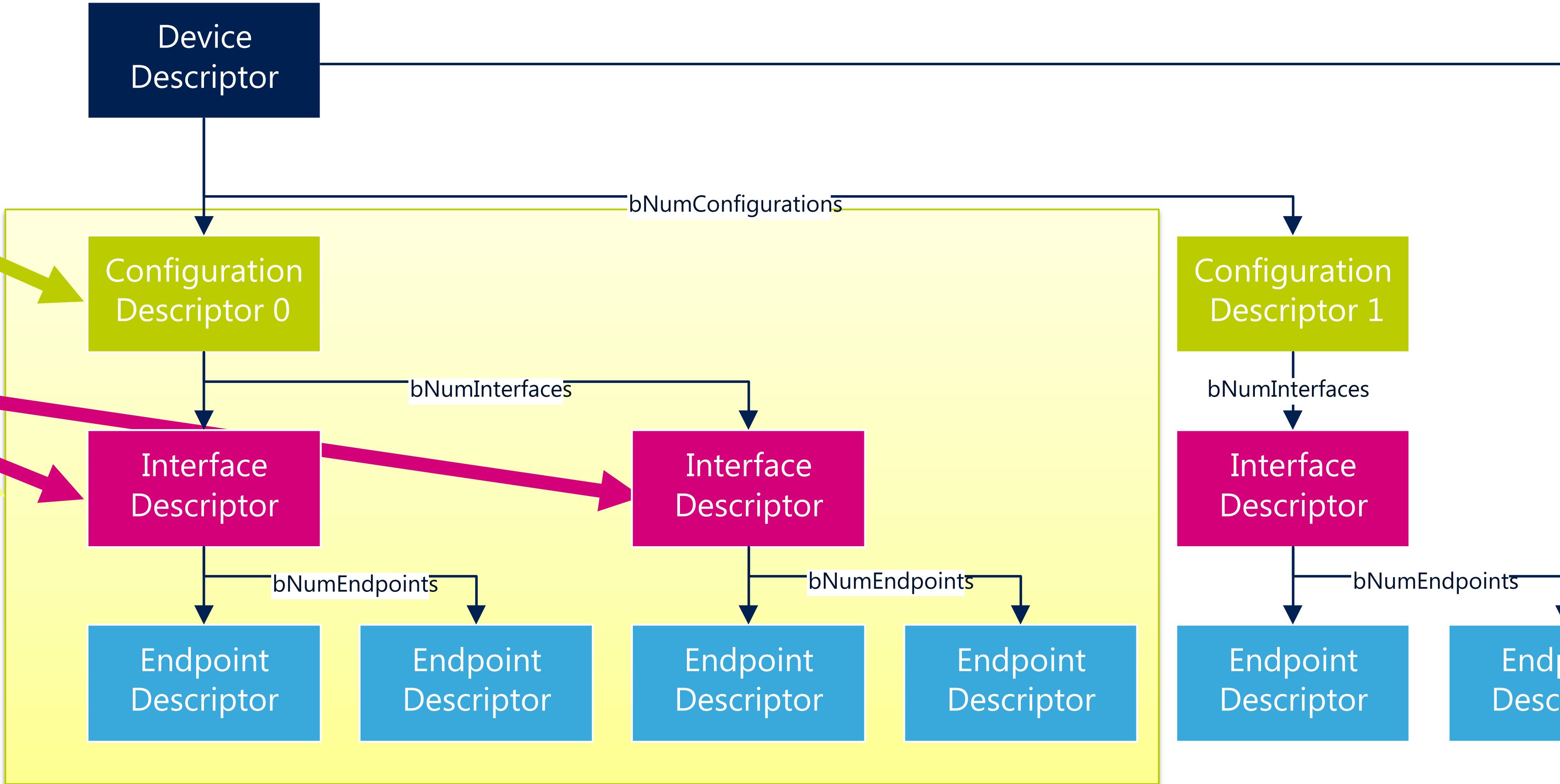
Configuration descriptor

- Device can have e.g. two configurations:
  - One for 500mA powered application
  - Reduced functionality when connected to 100mA port (e.g. USB-powered HUB)

Name	Description
bLength	Length of the descriptor (9 – bytes)
bDescriptorType	CONFIGURATION (value = 2)
wTotalLength	Total length of the configuration in bytes
bNumInterfaces	Number of interfaces in this configuration
bConfigurationValue	Value used by Set Configuration command (non-zero)
iConfiguration	String describing this configuration
bmAttributes	Remote wakeup, Self-powered
bMaxPower	Maximum allowed power consumption in mA

Configuration descriptor

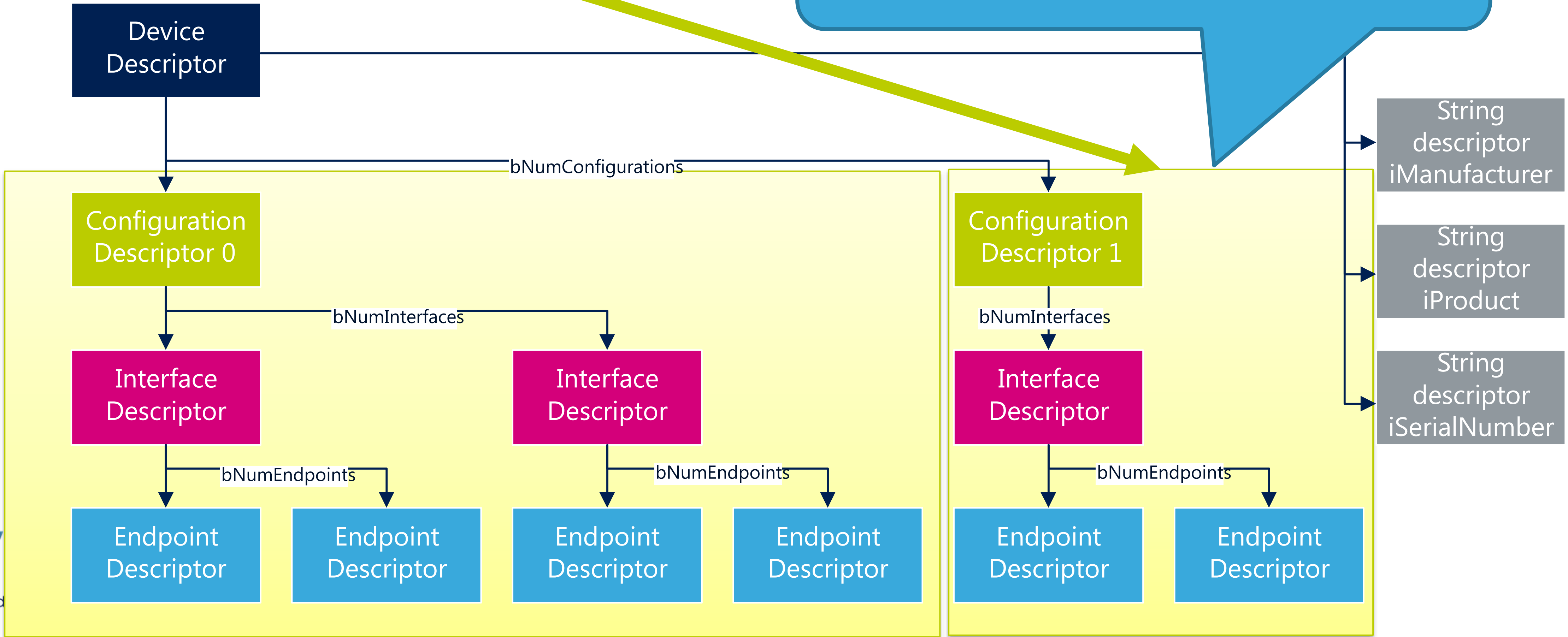
Name
bLength
bDescriptorType
wTotalLength
bNumInterfaces
bConfigurationValue
iConfiguration
bmAttributes
bMaxPower



Set Configuration

SET_CONFIGURATION
Configuration 1

Now can be used only endpoints from Configuration 1



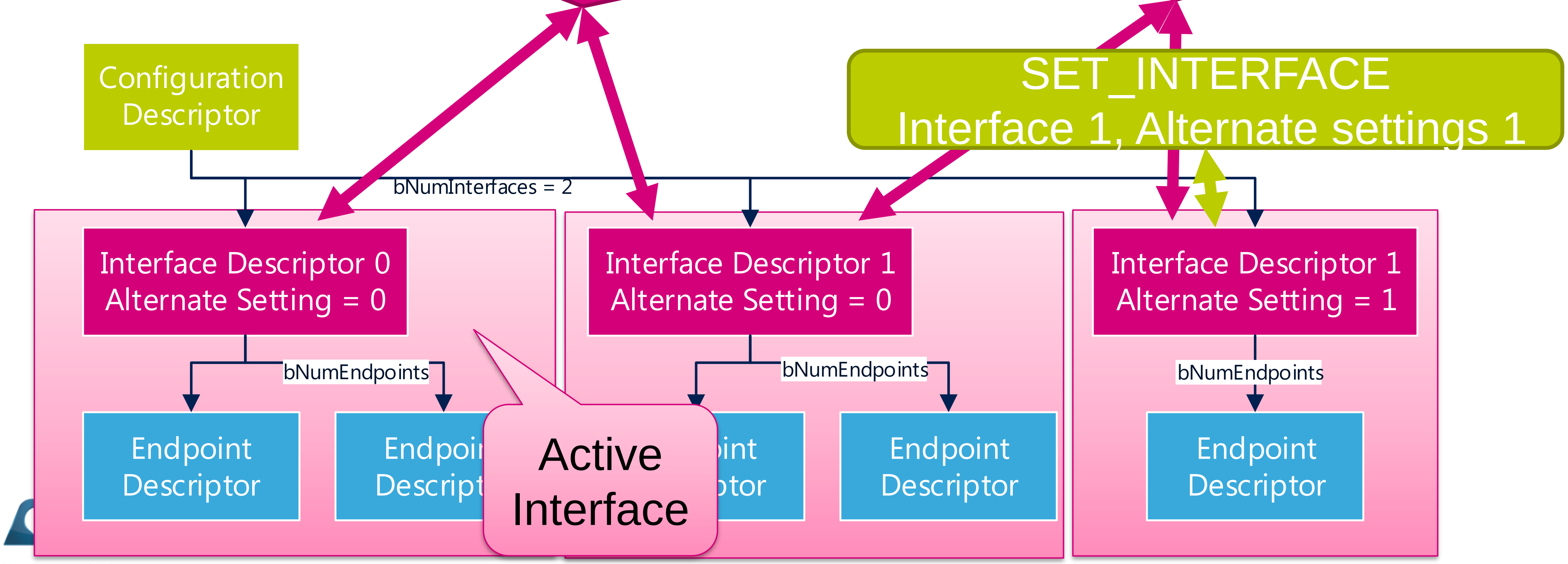
Interface descriptor

Name	Description
bLength	Length of the descriptor
bDescriptorType	INTERFACE (value = 4)
bInterfaceNumber	Interface identification number in configuration
bAlternateSetting	Alternate Interface settings
bNumEndpoints	Number of endpoints in this interface
bInterfaceClass	Interface class number
bInterfaceSubClass	Interface subclass number
bInterfaceProtocol	Interface class specific protocol
iInterface	Index of interface string descriptor

Interface

The Configuration can contatin multiple interfaces which can run in same time. Like COM port & mouse

Each interface can have also alternate settings which can be activated by SET_INTERFACE request

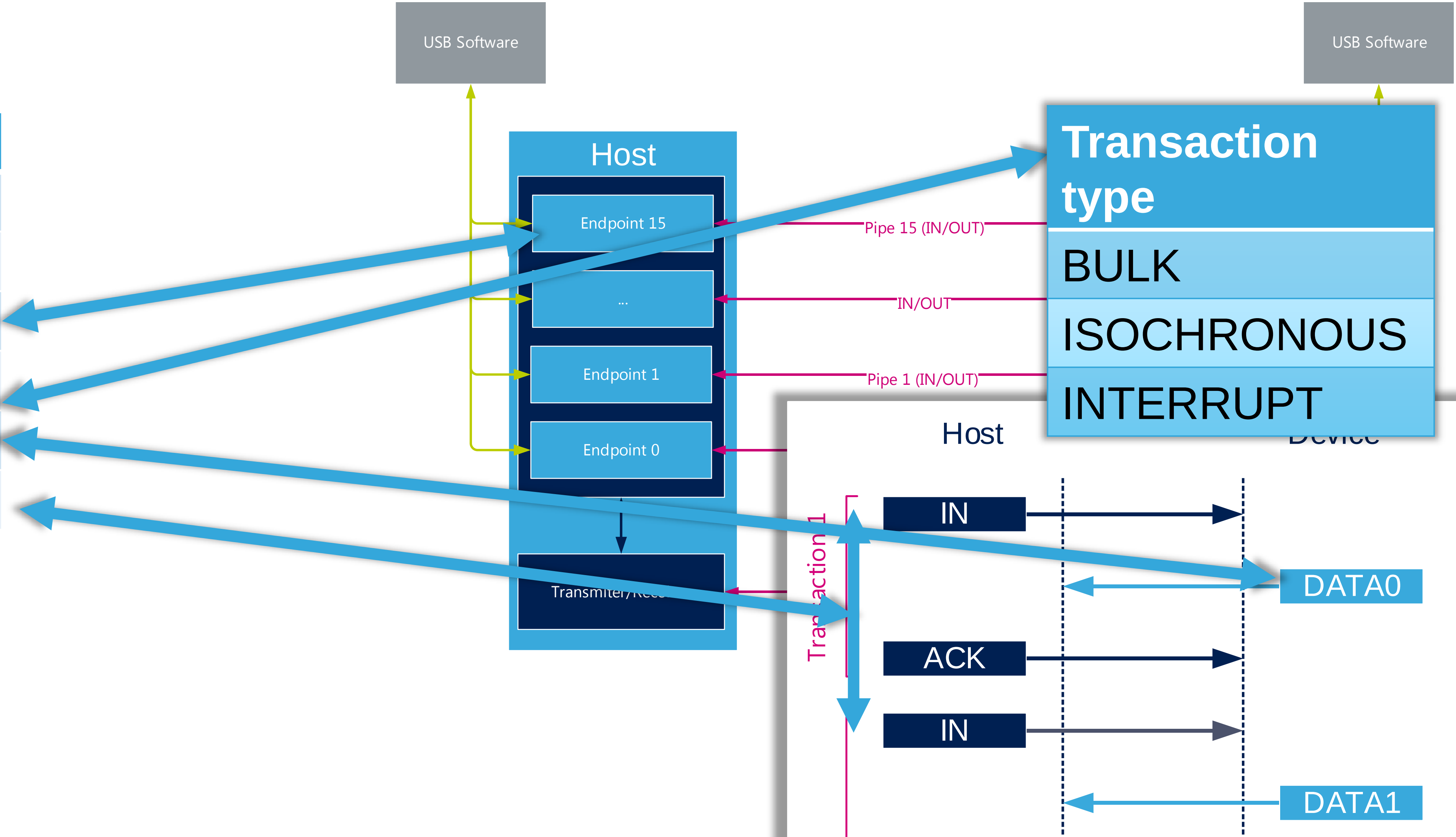


Endpoint descriptor

Name	Description
bLength	Length of the descriptor (7 – bytes)
bDescriptorType	Endpoint (value = 5)
bEndpointAdress	**[3:0]**: Endpoint number **[7]**: 0 = OUT, 1 = IN direction
bmAttributes	**[1:0]**: Transfer type **[3:2]**: Synchronization type (isochronous transfers only) **[5:4]**: Usage type (isochronous transfers only)
wMaxPacketSize	**[10:0]**: Maximum packet size **[12:11]**: Additional transactions per micro-frame (HS only)
bInterval	Polling interval per (micro-)frame

Endpoint descriptor

Name
bLength
bDescriptorType
bEndpointAdress
bmAttributes
wMaxPacketSize
bInterval



Transaction type
BULK
ISOCHRONOUS
INTERRUPT

String descriptors

- Accessed by indices from other descriptors
  - Index is specified in wValue

- Index 0 string contains list of supported languages

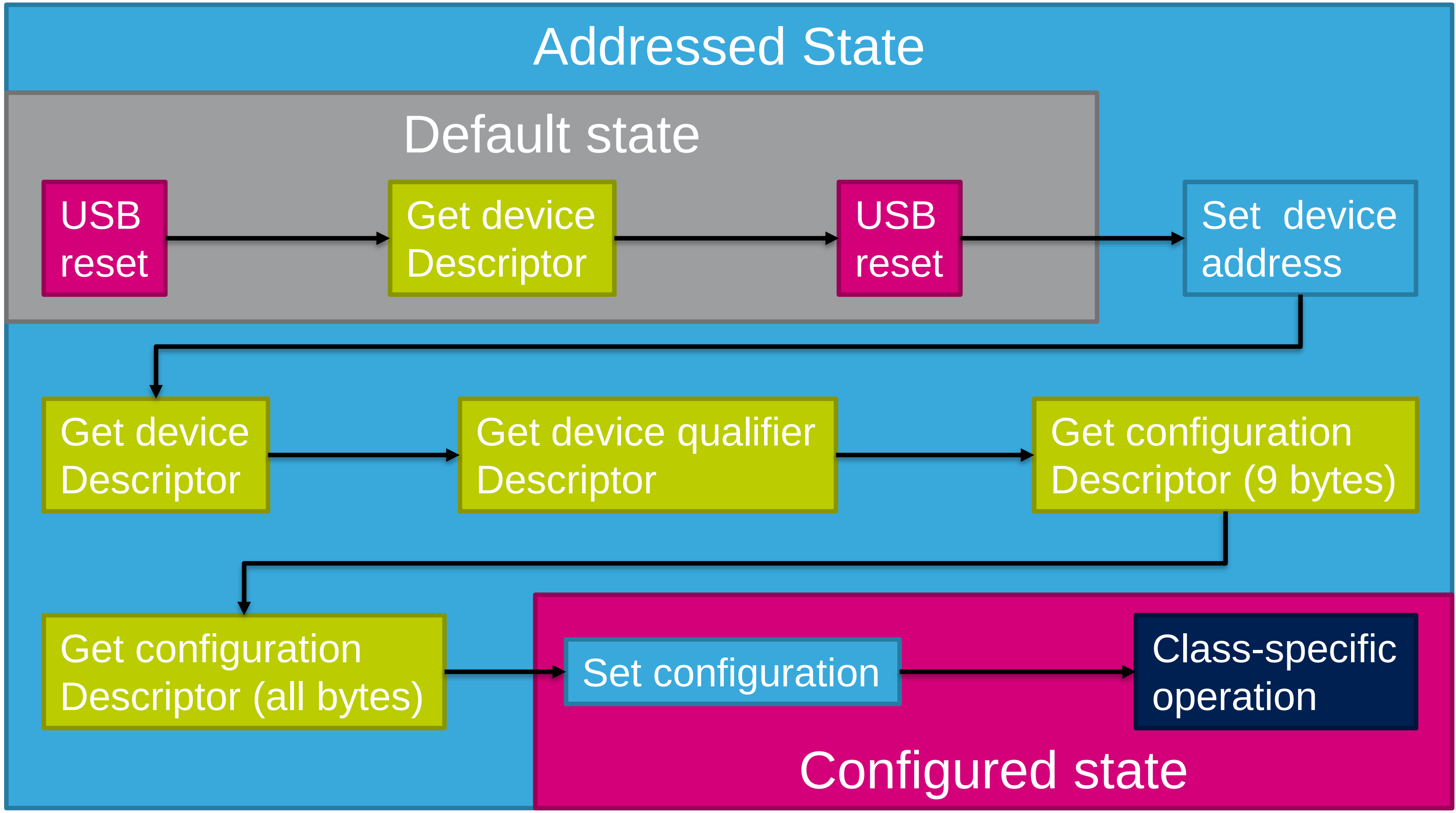
bLength	bDescriptorType	wLandID[0]	wLandID[1]	…

- Other indices contain individual strings
  - Strings are using UTF-16 encoding (16-bits per character)
  - Strings are not null terminated

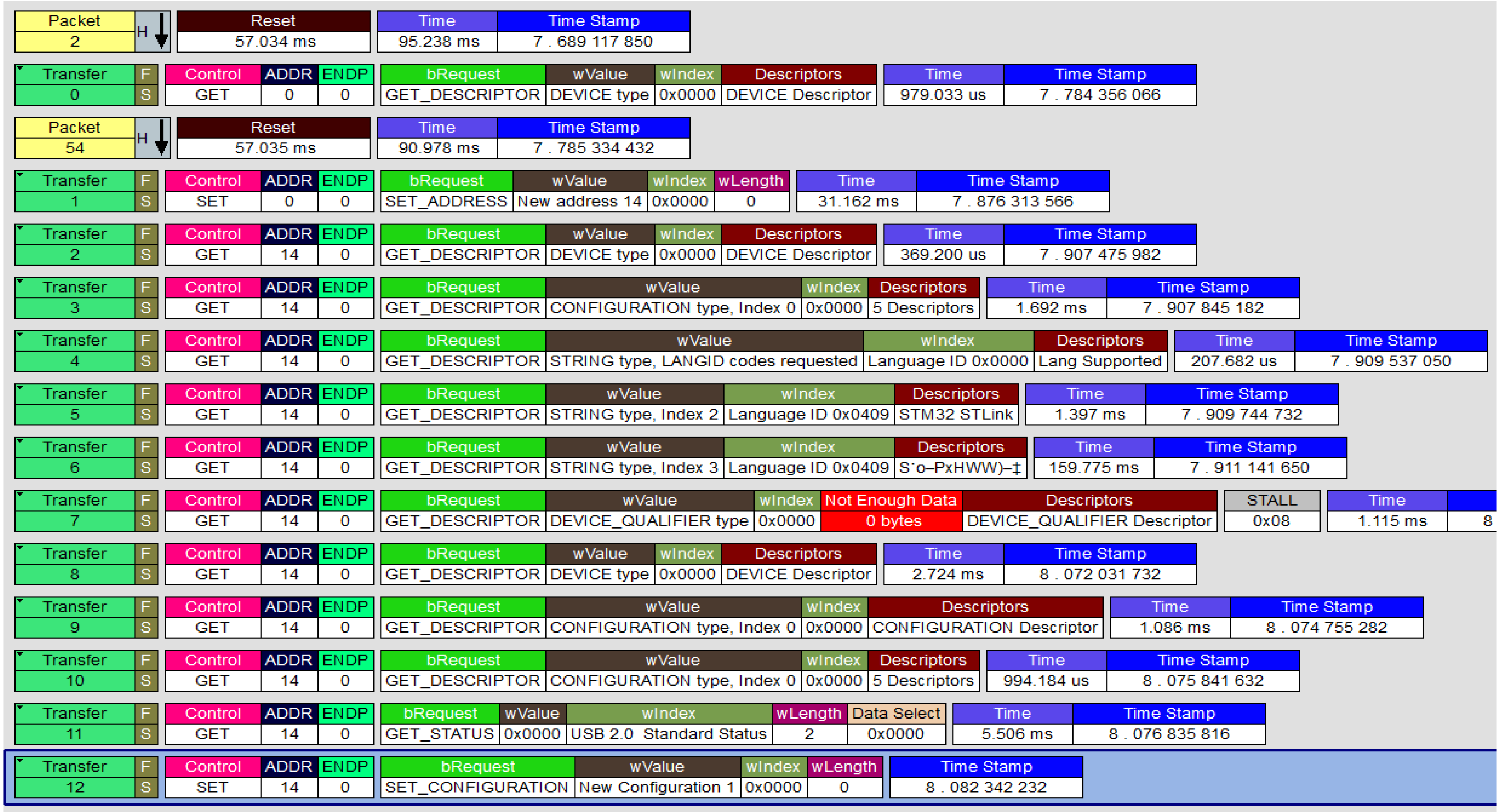
bLength	bDescriptorType	wChar[0]	wChar[1]	…

Example of enumeration process

- Order of some steps can depend on USB host implementation
  - USB specification doesn’t specify the exact order – device should be prepared for everything
- Additional steps might be taken by the class-specific driver



Example of enumeration process



Packet	H	Reset	Time	Time Stamp
2	↓	57.034 ms	95.238 ms	7 . 689 117 850

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
0	S	GET	0	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE Descriptor	979.033 us	7 . 784 356 066

Packet	H	Reset	Time	Time Stamp
54	↓	57.035 ms	90.978 ms	7 . 785 334 432

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time	Time Stamp
1	S	SET	0	0	SET_ADDRESS	New address 14	0x0000	0	31.162 ms	7 . 876 313 566

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
2	S	GET	14	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE Descriptor	369.200 us	7 . 907 475 982

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
3	S	GET	14	0	GET_DESCRIPTOR	CONFIGURATION type, Index 0	0x0000	5 Descriptors	1.692 ms	7 . 907 845 182

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
4	S	GET	14	0	GET_DESCRIPTOR	STRING type, LANGID codes requested	Language ID 0x0000	Lang Supported	207.682 us	7 . 909 537 050

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
5	S	GET	14	0	GET_DESCRIPTOR	STRING type, Index 2	Language ID 0x0409	STM32 STLink	1.397 ms	7 . 909 744 732

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
6	S	GET	14	0	GET_DESCRIPTOR	STRING type, Index 3	Language ID 0x0409	S˙o–PxHWW)–‡	159.775 ms	7 . 911 141 650

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Not Enough Data	Descriptors	STALL	Time	
7	S	GET	14	0	GET_DESCRIPTOR	DEVICE_QUALIFIER type	0x0000	0 bytes	DEVICE_QUALIFIER Descriptor	0x08	1.115 ms	8

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
8	S	GET	14	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE Descriptor	2.724 ms	8 . 072 031 732

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
9	S	GET	14	0	GET_DESCRIPTOR	CONFIGURATION type, Index 0	0x0000	CONFIGURATION Descriptor	1.086 ms	8 . 074 755 282

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time	Time Stamp
10	S	GET	14	0	GET_DESCRIPTOR	CONFIGURATION type, Index 0	0x0000	5 Descriptors	994.184 us	8 . 075 841 632

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Data Select	Time	Time Stamp
11	S	GET	14	0	GET_STATUS	0x0000	USB 2.0 Standard Status	2	0x0000	5.506 ms	8 . 076 835 816

Transfer	F	Control	ADDR	ENDP	bRequest	wValue	wIndex	wLength	Time Stamp
12	S	SET	14	0	SET_CONFIGURATION	New Configuration 1	0x0000	0	8 . 082 342 232

life.augmented