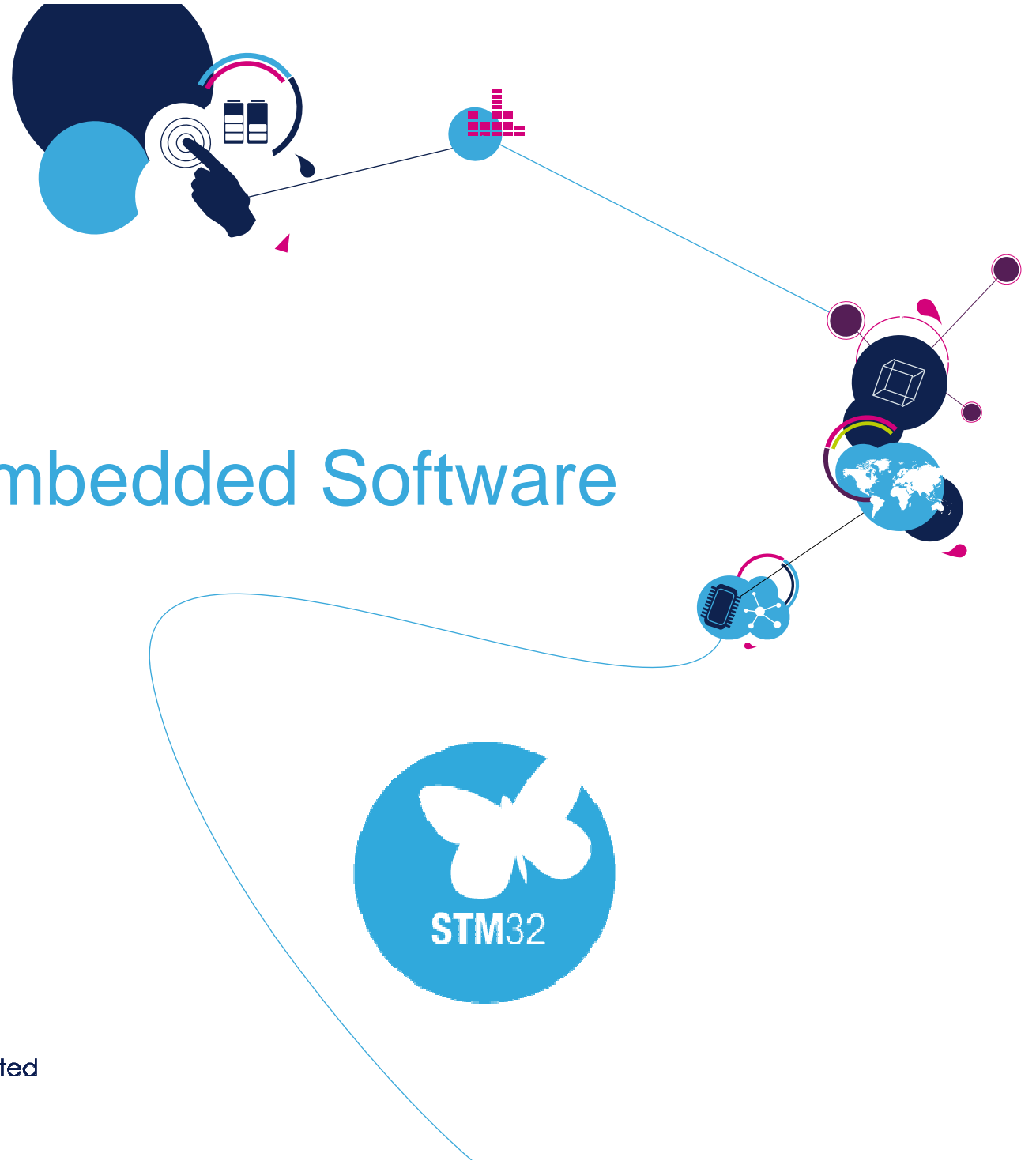
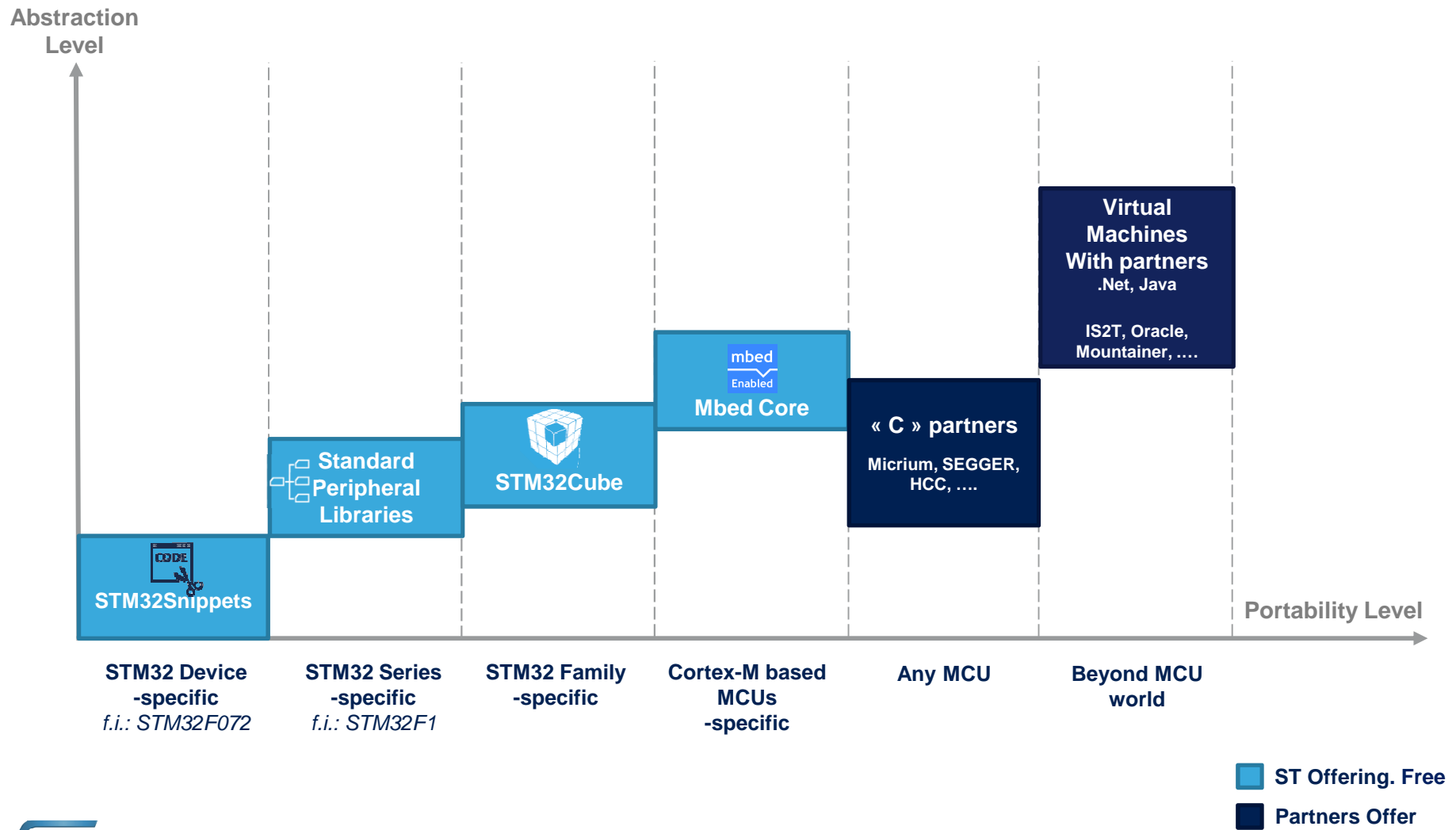


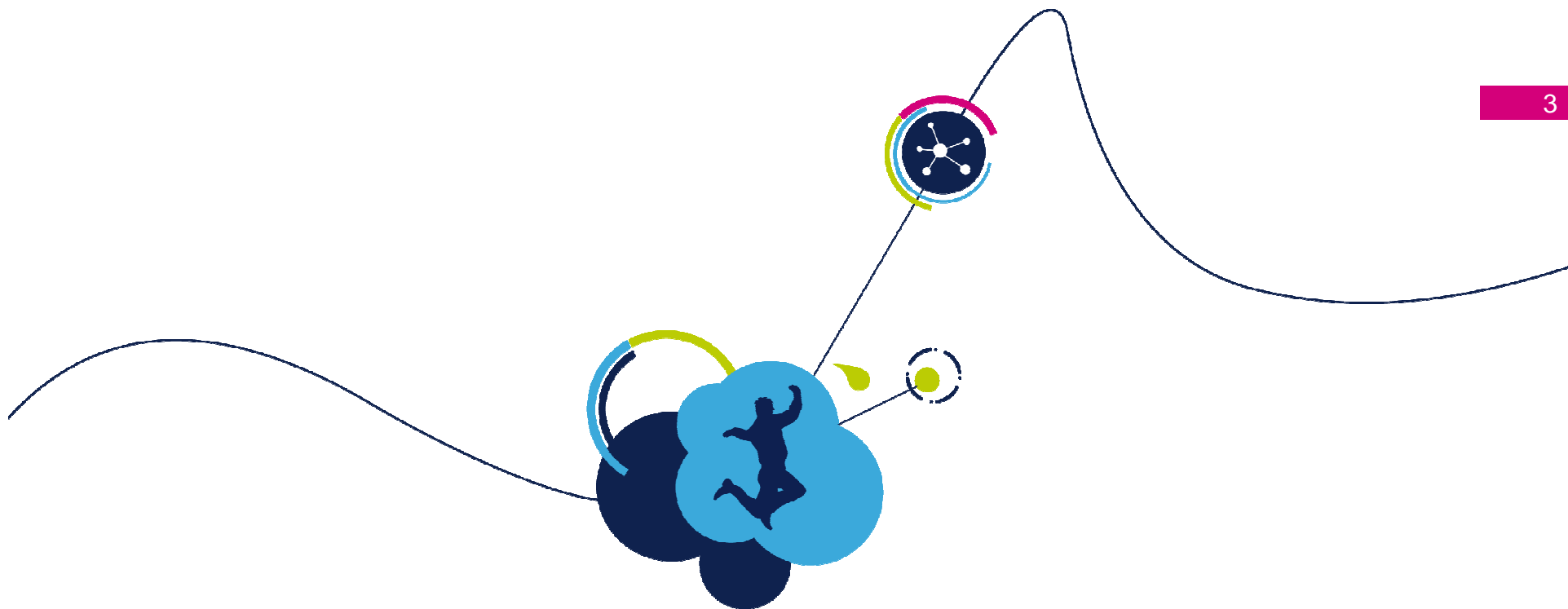
STM32 Embedded Software

ST's Offer

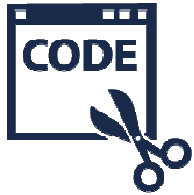


STM32 Embedded Software Offer - Overview





Focus on ST's Offer (Free)



STM32Snippets

4

- What is it ?

- A collection of code examples, directly based on STM32 peripheral registers, available in documentation and as software bundles

- Target Audience

- low level embedded system developers, typically coming from an 8 bit background, used to assembly or C with little abstraction

- Features:

- Highly Optimized
- Register Level Access
- Small code expressions
- Closely follows the reference manual
- Debugging close to register level

- Limitations:

- Specific to STM32 devices, not portable directly between series
- Not matching complex peripherals such as USB
- Lack of abstraction means developers must understand peripheral operation at register level
- Available (today) on STM32 L0 and F0 series



Portability	Optimization (Memory & Mips)	Easy	Readiness	Hardware coverage
	+++			+



Standard Peripheral Libraries (SPL)

5

- What is it ?

- Collection of C Libraries covering STM32 peripherals

- Target Audience

- Embedded systems developers with procedural C background. All existing STM32 customer base prior to the STM32Cube launch, willing to keep same supporting technology for future projects, and same STM32 series

- Features:

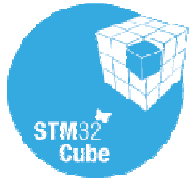
- Average optimization, fitting lots of situations
- No need for direct register manipulation
- 100% coverage of all peripherals
- Easier debugging of procedural code
- Extensions for complex middleware such as USB/TCP-IP/Graphics/Touch Sense

- Limitations:

- Specific to certain STM32 series.
- No common HAL API prevents application portability between series
- Middleware libraries may not be unified for each series
- Doesn't support forward STM32 series starting with STM32 L0 and F7



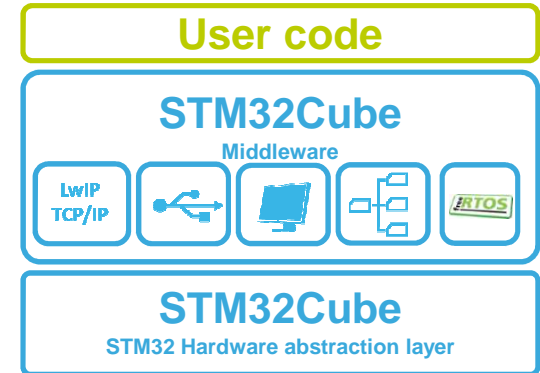
Portability	Optimization (Memory & Mips)	Easy	Readiness	Hardware coverage
++	++	+	++	+++



STM32Cube - Embedded software

- What is it ?

- Full featured packages with drivers, USB, TCP/IP, Graphics, File system and RTOS
- Set of common application programming interfaces, ensuring high portability inside whole STM32 family



- Target Audience

- Embedded system developers with a strong structured C background. New customers looking for a fast way to evaluate STM32 and easy portability

- Features:

- High level and functional abstraction
- Easy port from one series to another
- 100% coverage of all peripherals
- Integrates complex middleware such as USB/TCP-IP/Graphics/Touch Sense/RTOS
- Can work with STM32CubeMX tool on the PC to generate initialization code




- Limitations:

- May be challenging to low level C programmers in the embedded space.
- Higher portability creates bigger software footprints or more time spent executing adaptation code
- Not Available (today) on STM32 F1 series

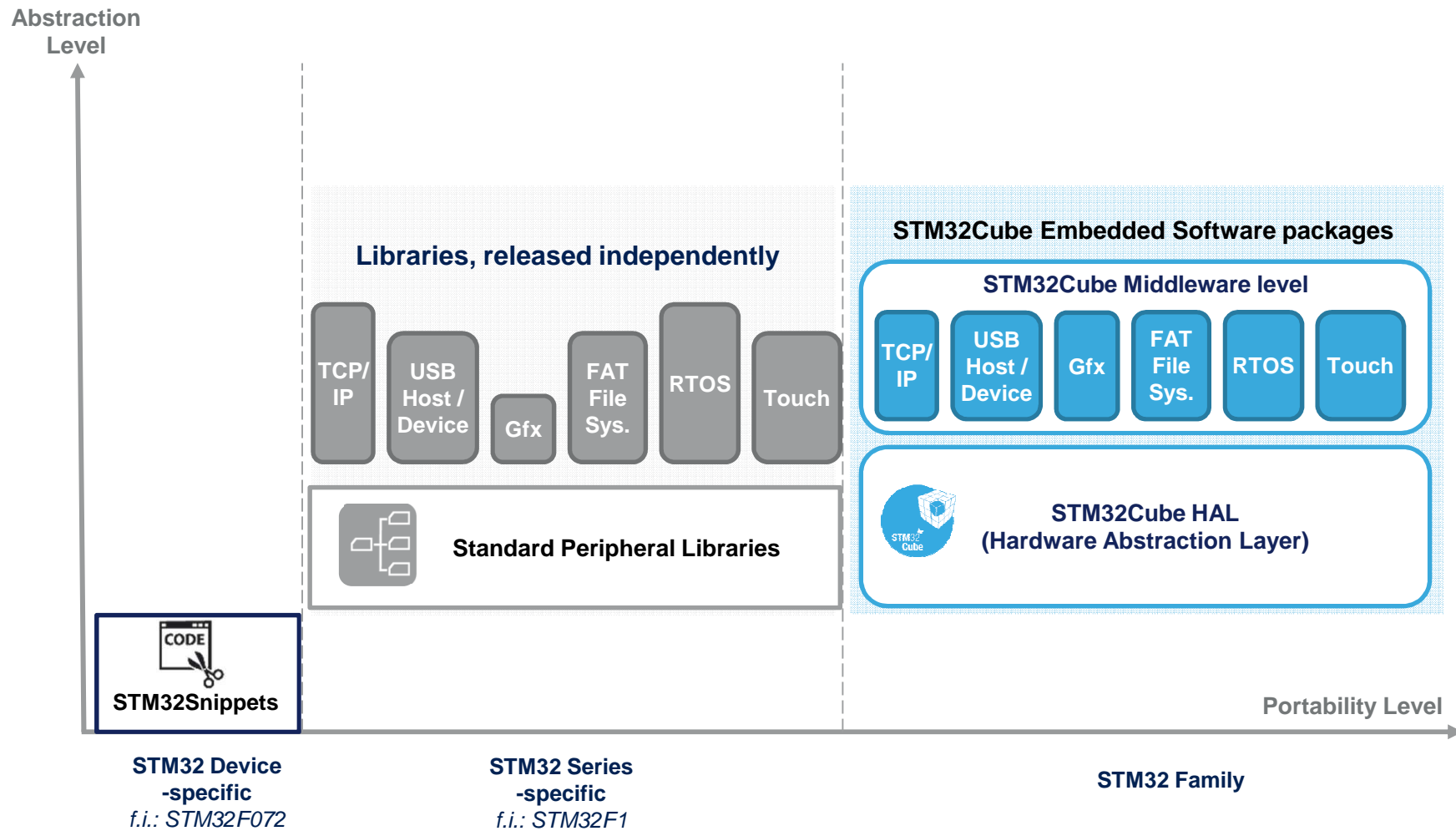


Portability	Optimization (Memory & Mips)	Easy	Readiness	Hardware coverage
+++	+	++	+++	++

ST Embedded software offer – Comparison

Offer	Portability	Optimization (Memory & Mips)	Easy	Readiness	Hardware coverage
 STM32Snippets		++++			+
 Standard Peripheral Library	++	++	+	++	++++
 STM32Cube	++++	+	++	++++	++

ST Embedded software offer - Positioning





Availability

Offer	Available for STM32							
	 STM32 F0	 STM32 F1	 STM32 F3	 STM32 F2	 STM32 F4	 STM32 F7	 STM32 L0	 STM32 L1
STM32Snippets	Now	N.A.	N.A.	N.A.	N.A.	N.A.	Now	N.A.
Standard Peripheral Library	Now	Now	Now	Now	Now	N.A.	N.A.	Now
STM32Cube	Now	Jan. 15	Now	Now	Now	May 15	Now	Now

What solution to choose ? An FAQ

10

1. I want to use a small footprint MCU, what should I use?

Abstraction has a cost. Therefore, if you need to take benefit from every single bit of memory, STM32Snippets will be the best choice

2. I come from 8-bit MCU world, what should I use?

If you prefer direct register manipulation then the STM32Snippets would be a good starting point. However, if you prefer structure 'C' level programming, then we recommend using the STM32Cube or SPL

3. I today use SPL on STM32F103. Should I switch to STM32Cube?

If you intend to use only MCUs that are part of the same series in the future (in this case STM32 F1 series), then you should remain using SPL


If you plan to use different STM32 series in the future then we recommend considering STM32Cube as this will make it much easier to move between series

4. I need a mix of portability and optimization. What can I do?

You can use the STM32Cube and replace some of the calls with your specific optimized implementations, thus keeping maximum portability and isolating areas that are not portable, but optimized



Migrating between offers



From	To		
	STM32Snippets	SPL	STM32Cube
STM32Snippets	Easy within same STM32 serie. Ex: Between STM32F072 and STM32F030	Not possible	Not possible
	Almost not possible between different series Ex: Between STM32F103 and STM32F334		
Standard Peripheral Library (SPL)	Some (but not all) SPL functions can be replaced with Snippets	Easy within same STM32 serie. Ex: Between STM32F401 and STM32F429	Not possible today <i>Note: ST studies addition of APIs in STM32Cube to ease that migration</i>
		Difficult between different STM32 series Ex: Between STM32F100 and STM32F407	
STM32Cube embedded software package	Some (but not all) STM32Cube functions can be replaced with Snippets	Not possible	Yes. Across all STM32 family



Thank you 12



 /STM32

 @ST_World

 st.com/e2e



<http://www.st.com/stm32embeddedsoftware>