

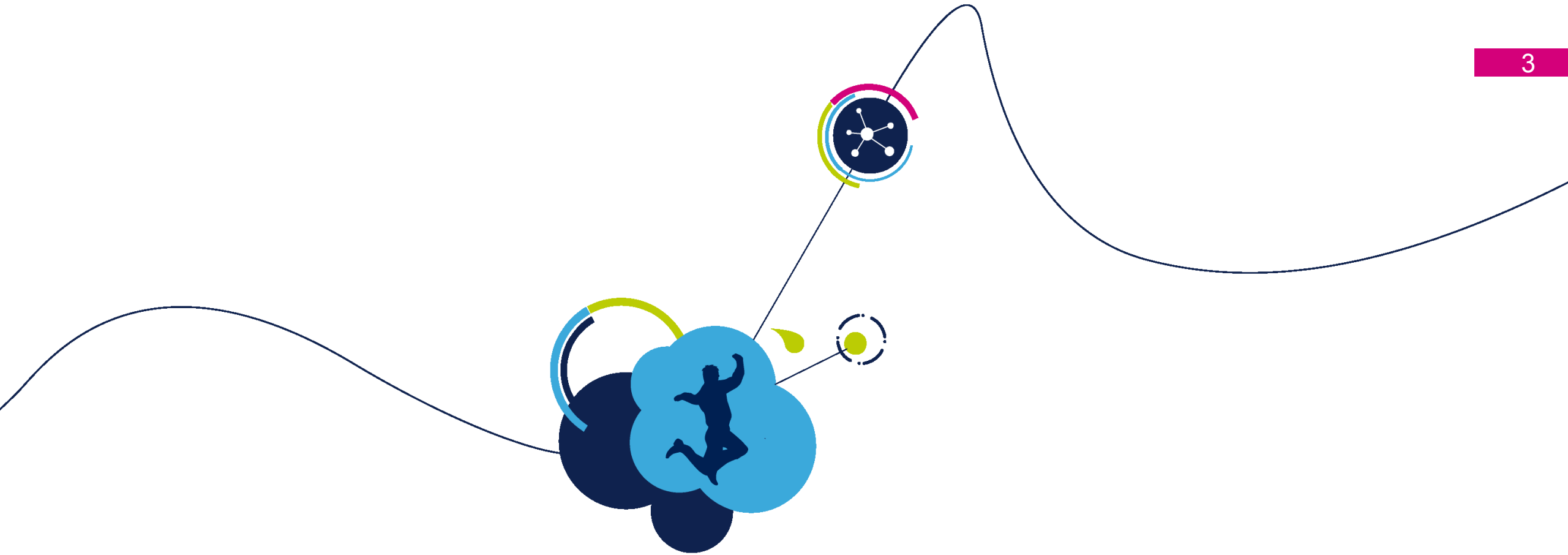
STM32MP1 Embedded Software



Presentation

1h

- *STM32MP1 Embedded Software Distribution Components*
- Demo launcher
- ST Mainlining – Support
- Third Party Ecosystem for STM32MP1
- Embedded Software Distribution Delivery
 - Starter Package
 - Development package
 - Distribution package
- Licensing Term
- Wiki User Guide

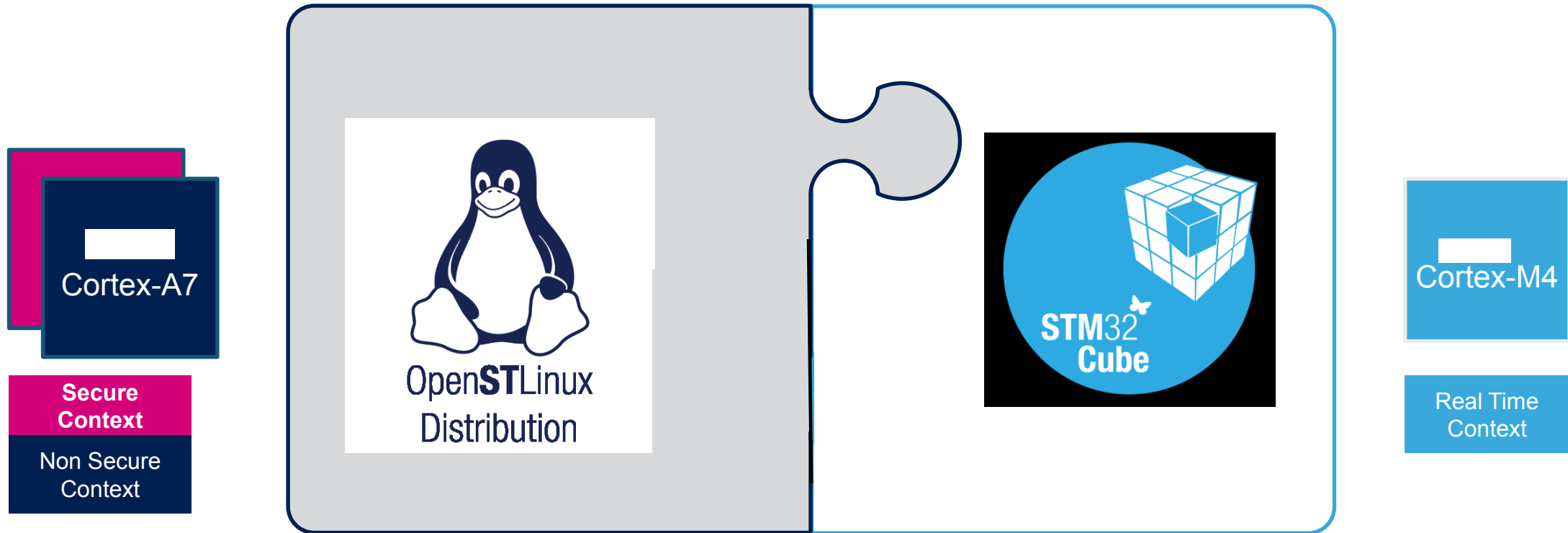


STM32MP1 Embedded Software Software Components

A Fully Integrated Design Suite

Leveraging the STM32Cube Environment

4

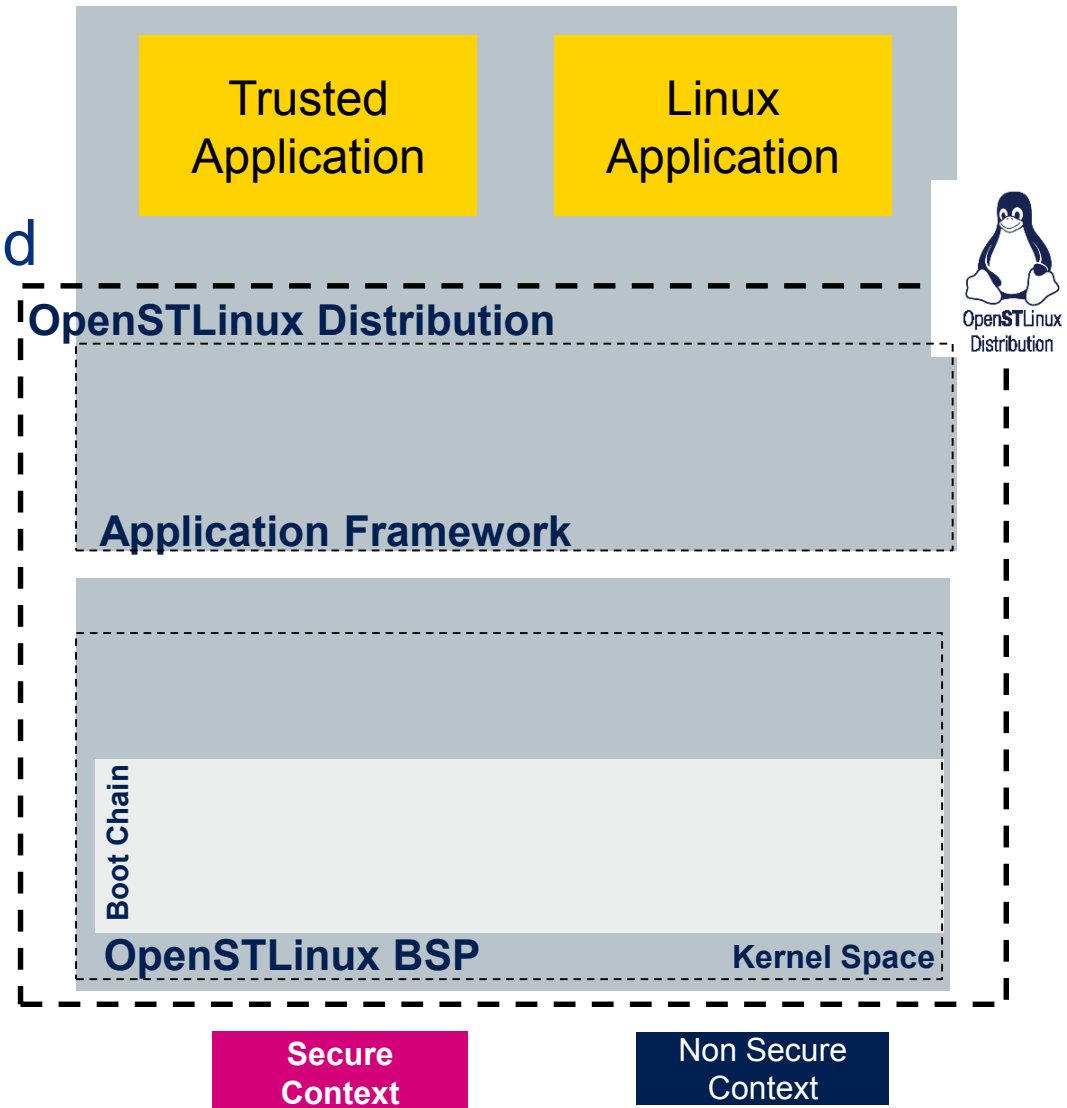


STM32MP1 Embedded Software Distribution

STM32MP1 Embedded Software Architecture

5

- OpenSTLinux Distribution
 - Mainlined environment
95% of the ST code upstreamed
- Using Yocto community tools
- Open-Source Solution adopted by Linaro



STM32MP1 Embedded Software Architecture

6

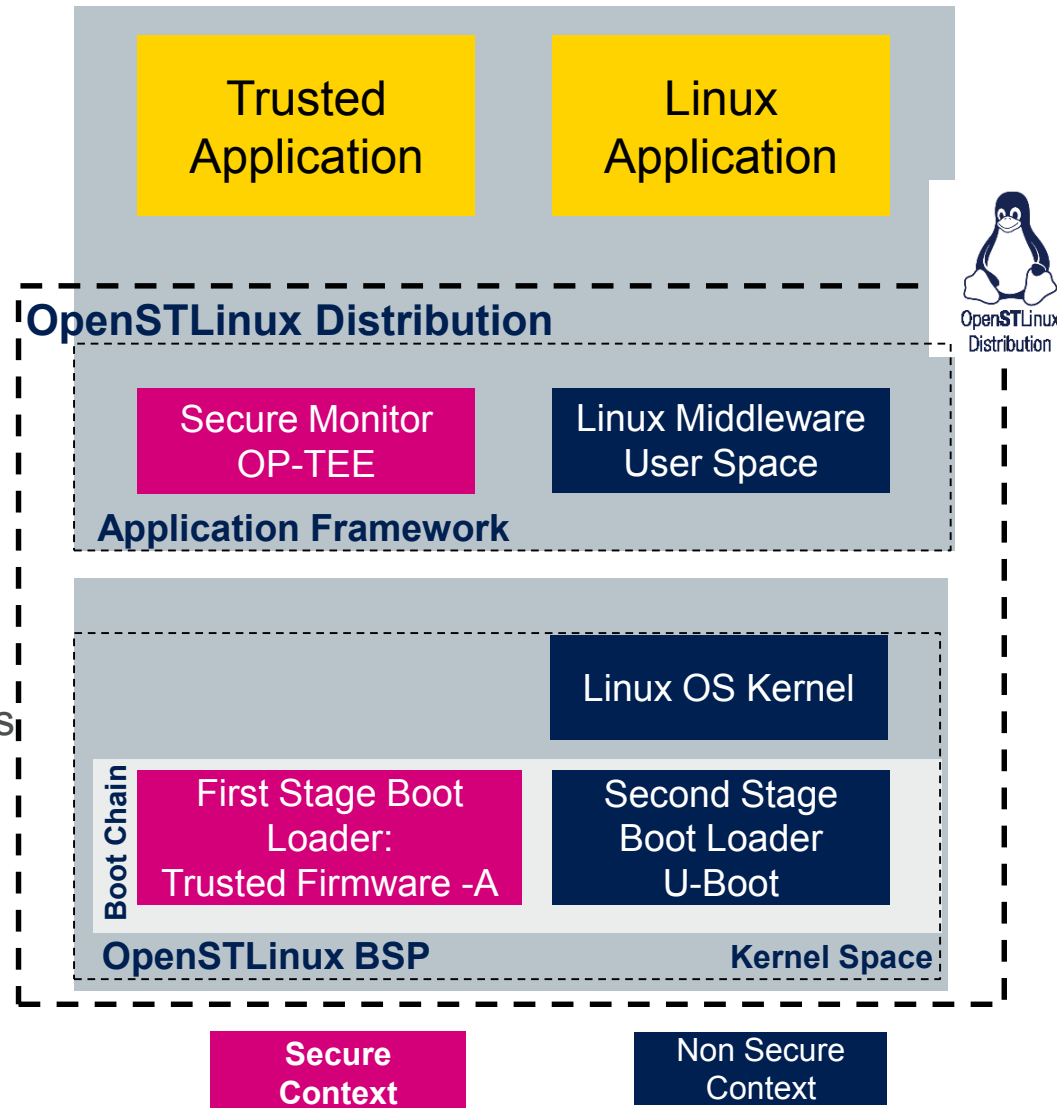
- OpenSTLinux Distribution

- OpenSTLinux BSP

- Secure boot
- Standard boot chain
- Linux Kernel

- Application framework

- **Linux Middleware:** To ease the Linux applications (ASLA, Gstreamer, Wayland-Weston ...)
- OP-TEE : to run Trusted Applications (Maintained by Linaro Security WG)



STM32MP1 Embedded Software Architecture

7

- OpenSTLinux Distribution

- Mainlined environment using Yocto tools
- Open-Source Solution adopted by Linaro

- OpenSTLinux BSP

- Secure boot
- Standard boot chain
- Linux Kernel

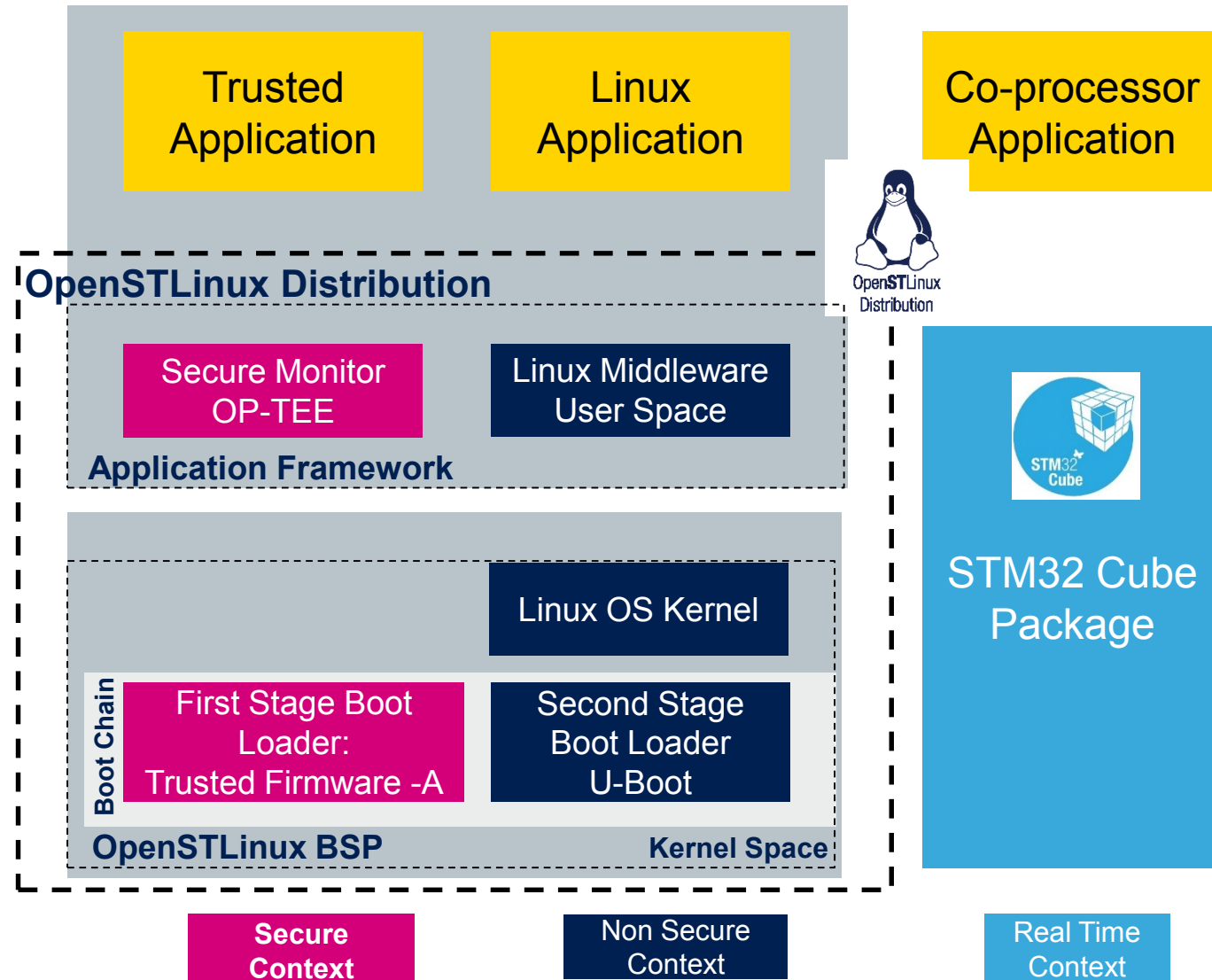
- Application framework

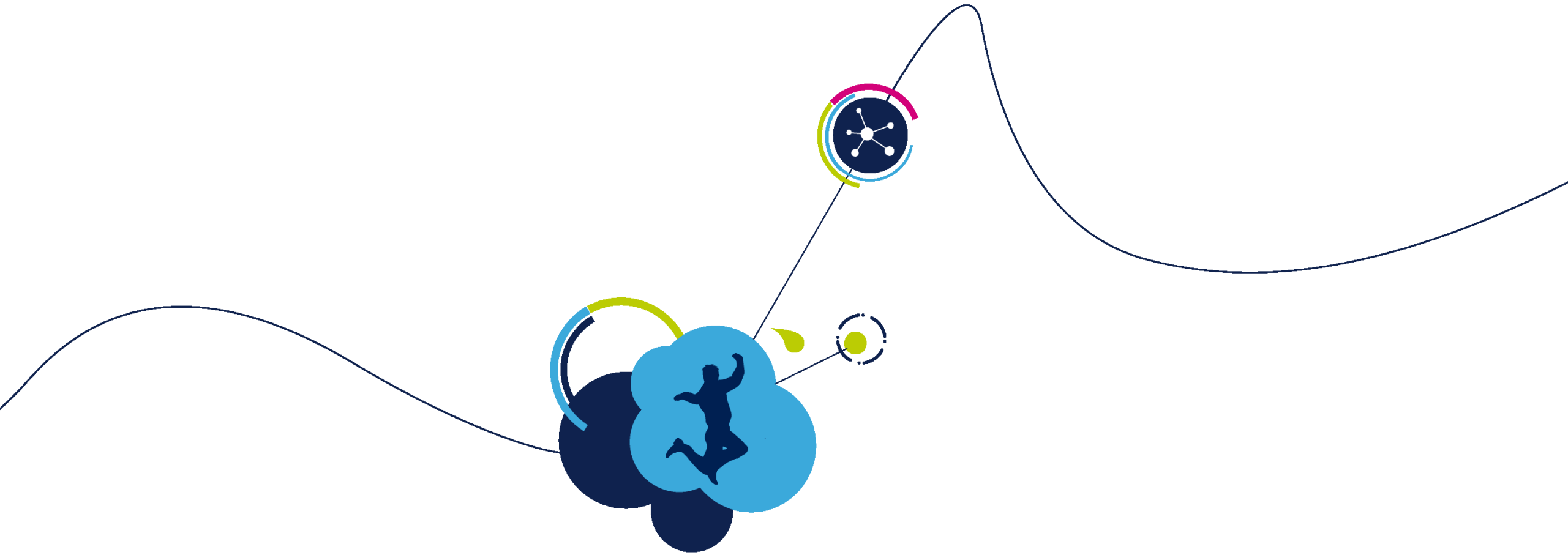
- **Linux Middleware:** To ease the Linux applications (ASLA, Gstreamer, Wayland-Weston ...)
- OP-TEE : to run Trusted Applications (Maintained by Linaro Security WG)

- STM32Cube Package



Legacy MCU Software Package





OpenSTLinux Distribution Components

3rd Party

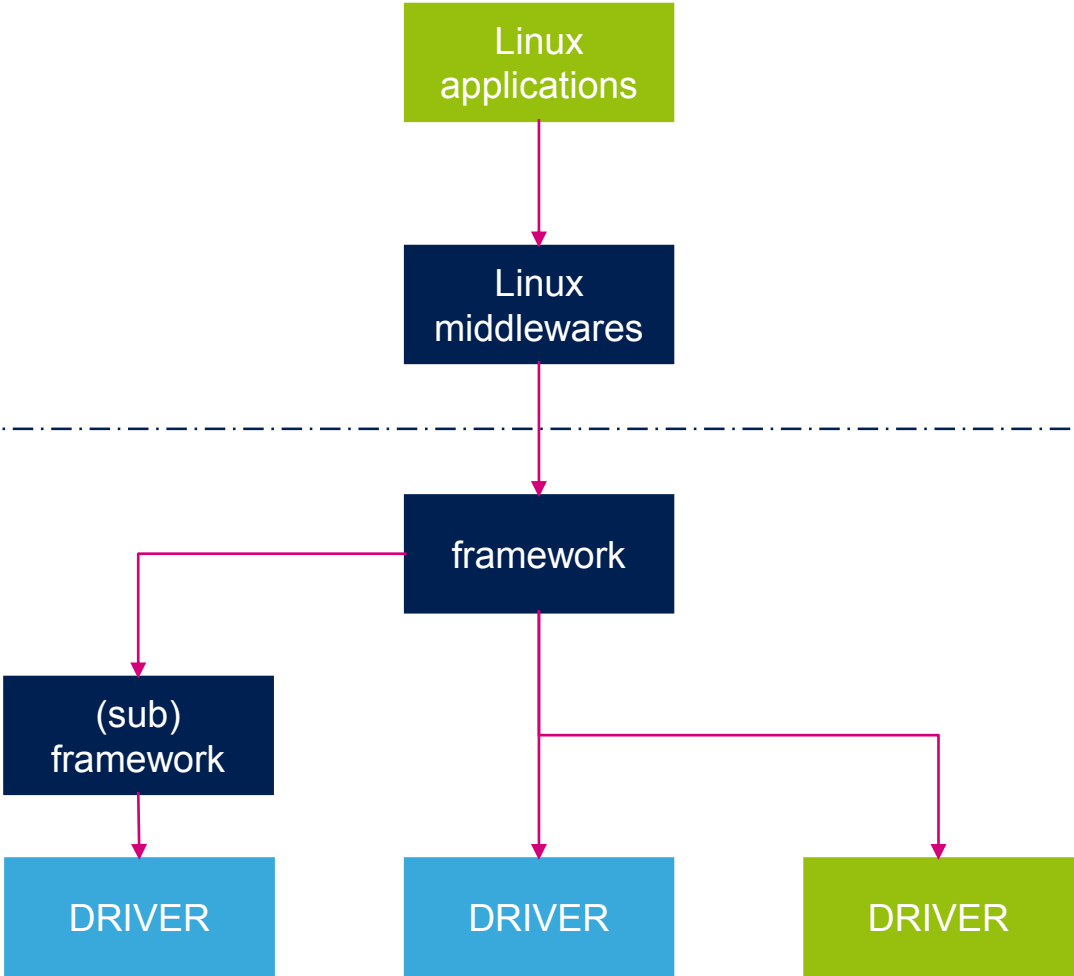
STCommunity

- lowercase = community framework
- UPPERCASE = peripheral driver

Legend

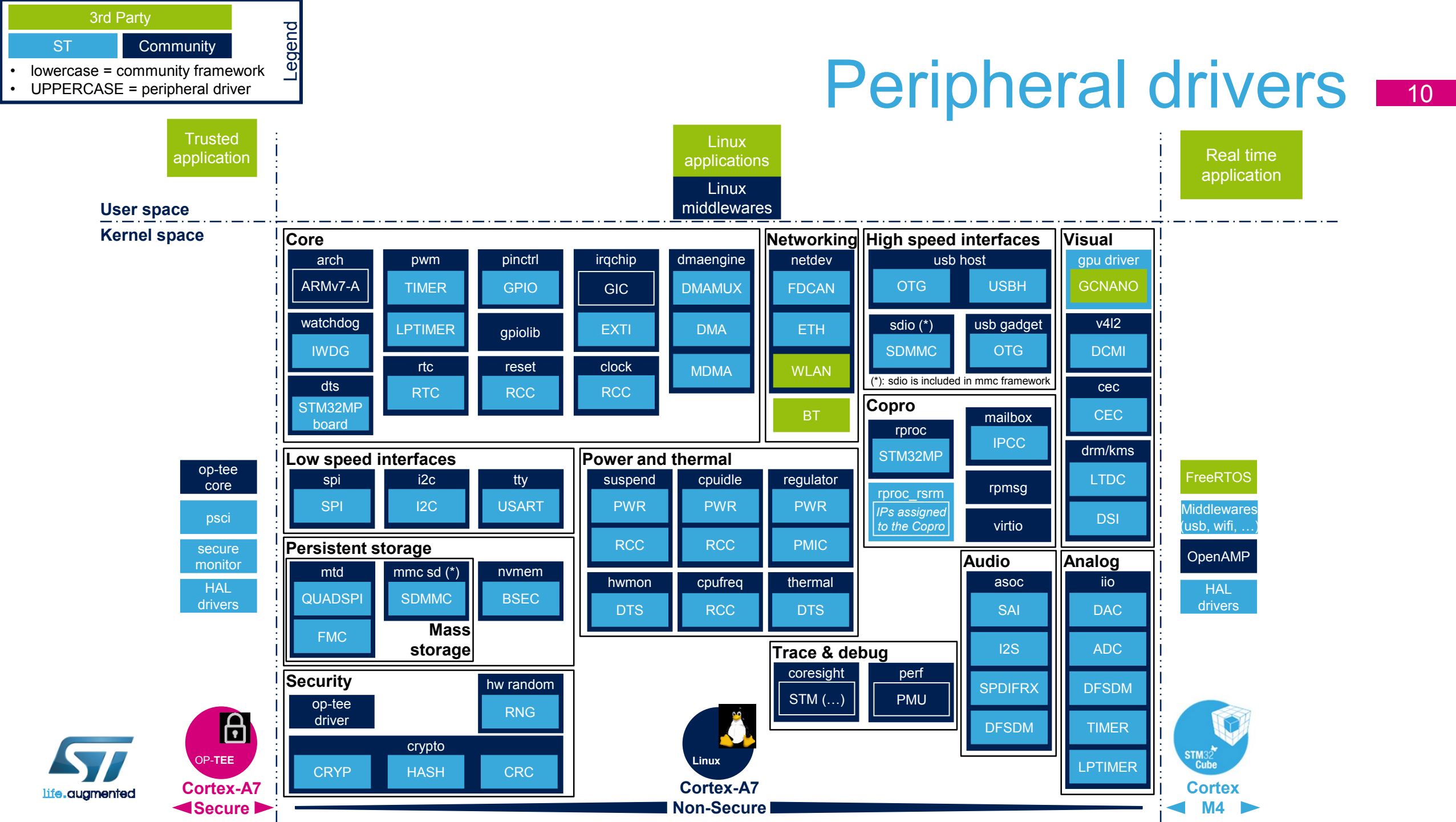
Linux framework & driver

User space
Kernel space



Peripheral drivers

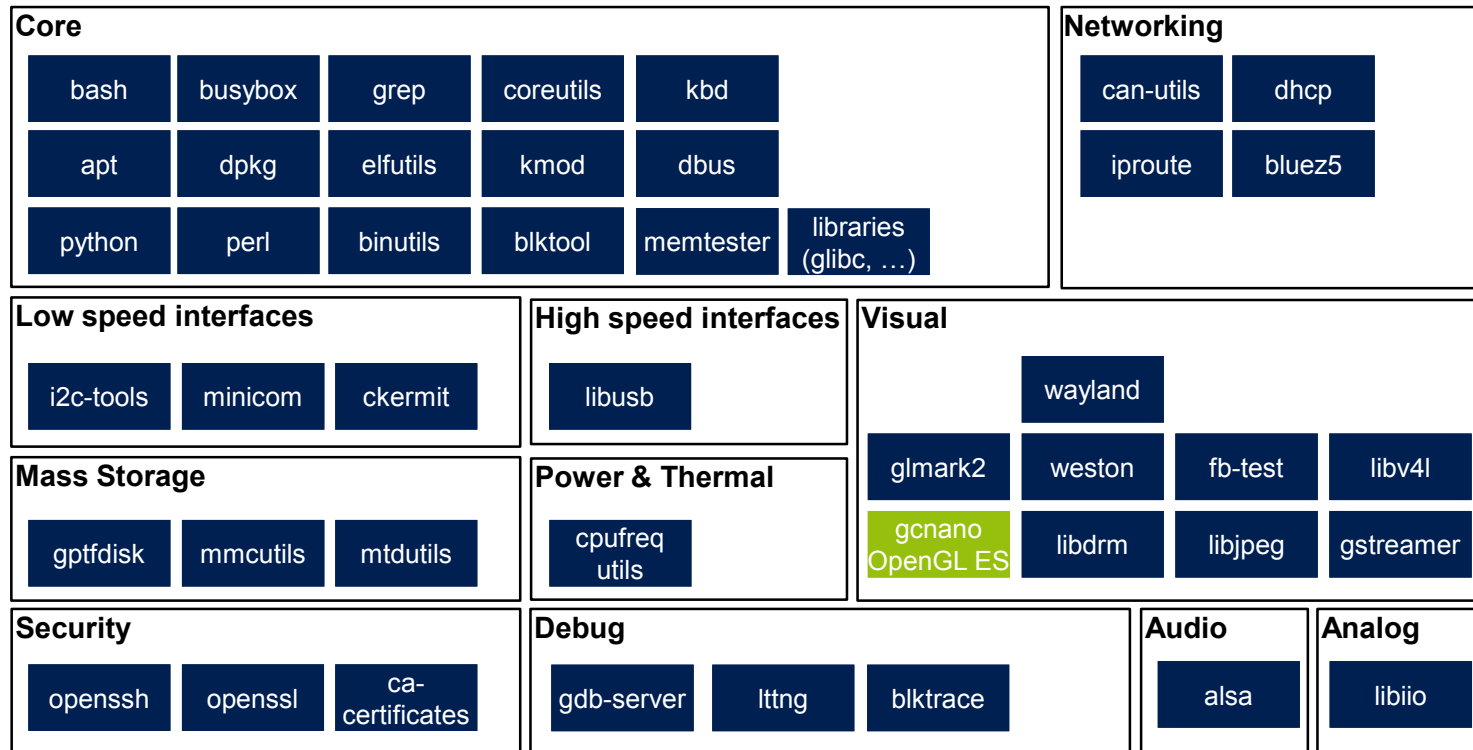
10



Open-Embedded User space

- The middleware components list shown here is not exhaustive and can be tuned by the customer to fit with applications needs.

Linux applications

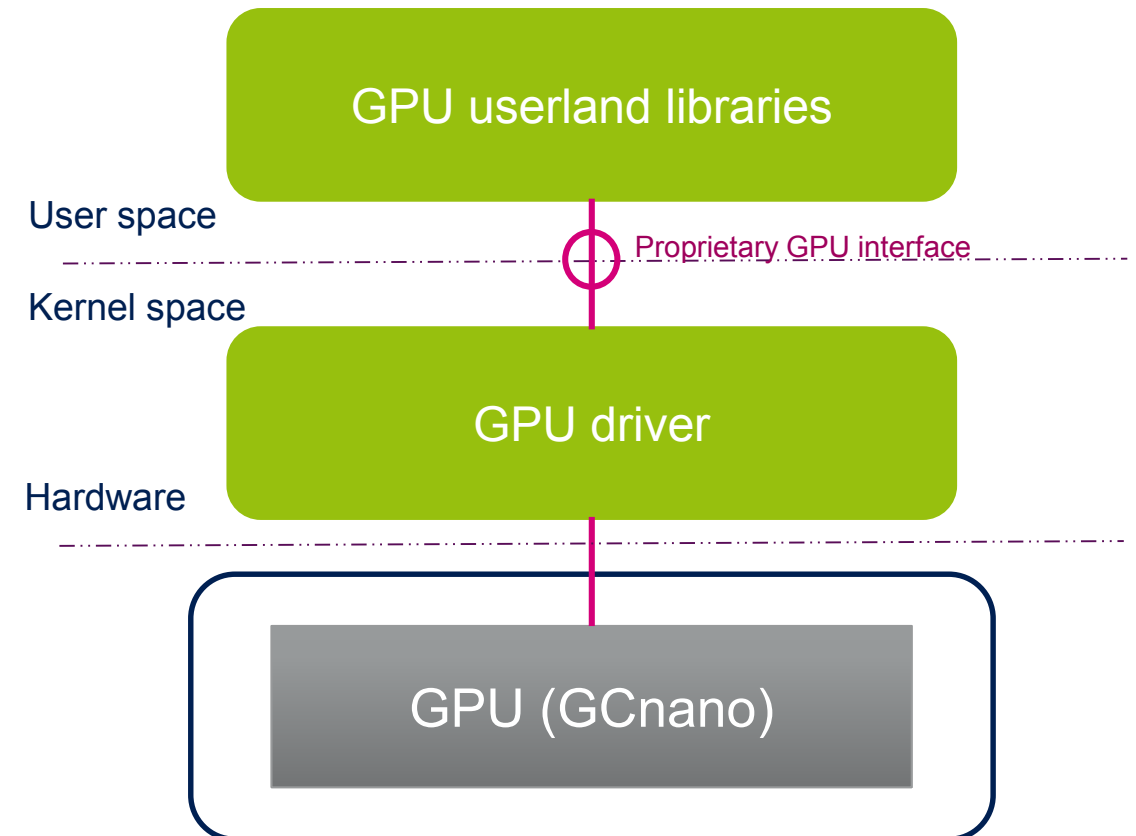


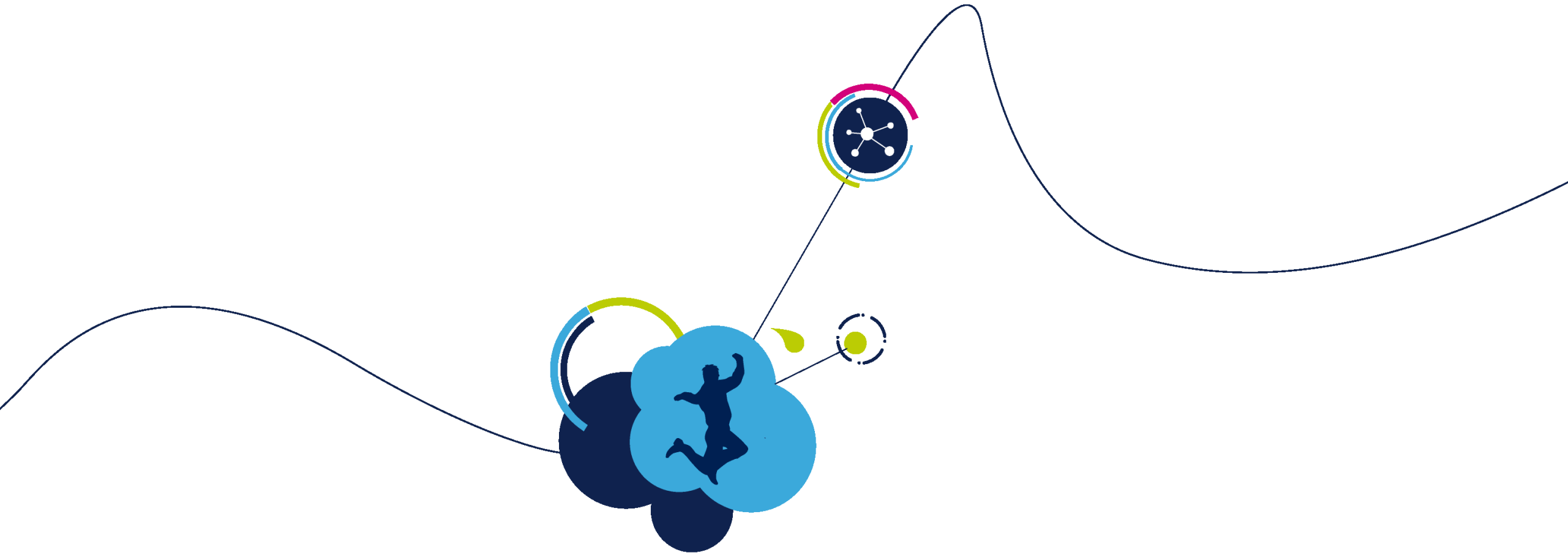
User space

Kernel space



- GCnano IP from VeriSilicon (Vivante)
- User space:
 - GPU libraries provided in OpenSTLinux distribution (source code access under NDA with VeriSilicon)
 - GPU libraries supported API :
Open GL ES 1.1 and 2.0,
OpenVG 1.1 and EGL
- Kernel space
 - GPU driver source code provided
- Verisilicon ToolKit provided by ST for debug/analyse customer applications





STM32MP1 Embedded Software

STM32Cube Package

Benefit from Field-Proven RTOS Tools

14

Full re-use of STM32 MCU Cube firmware on Arm Cortex-M



Hardware abstraction layer & Low-level layer peripheral drivers



Collection of Middleware components (RTOS, USB...) for Cortex-M



Hundreds of examples



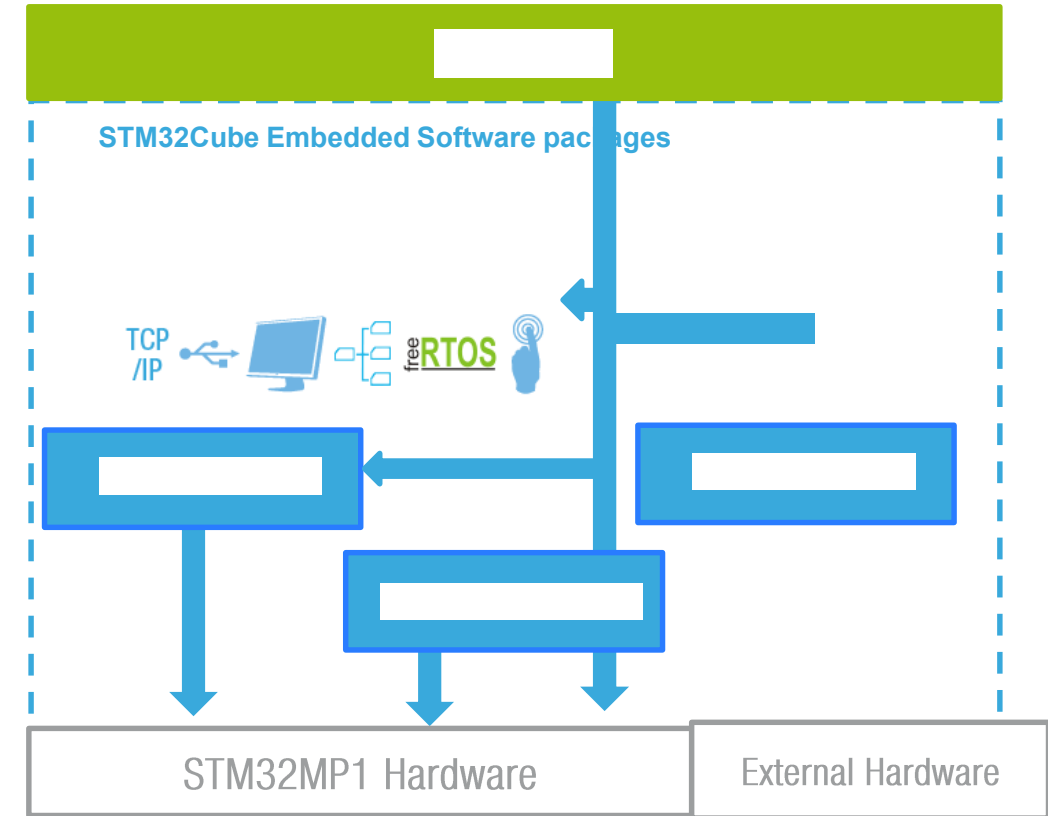
Production-ready quality

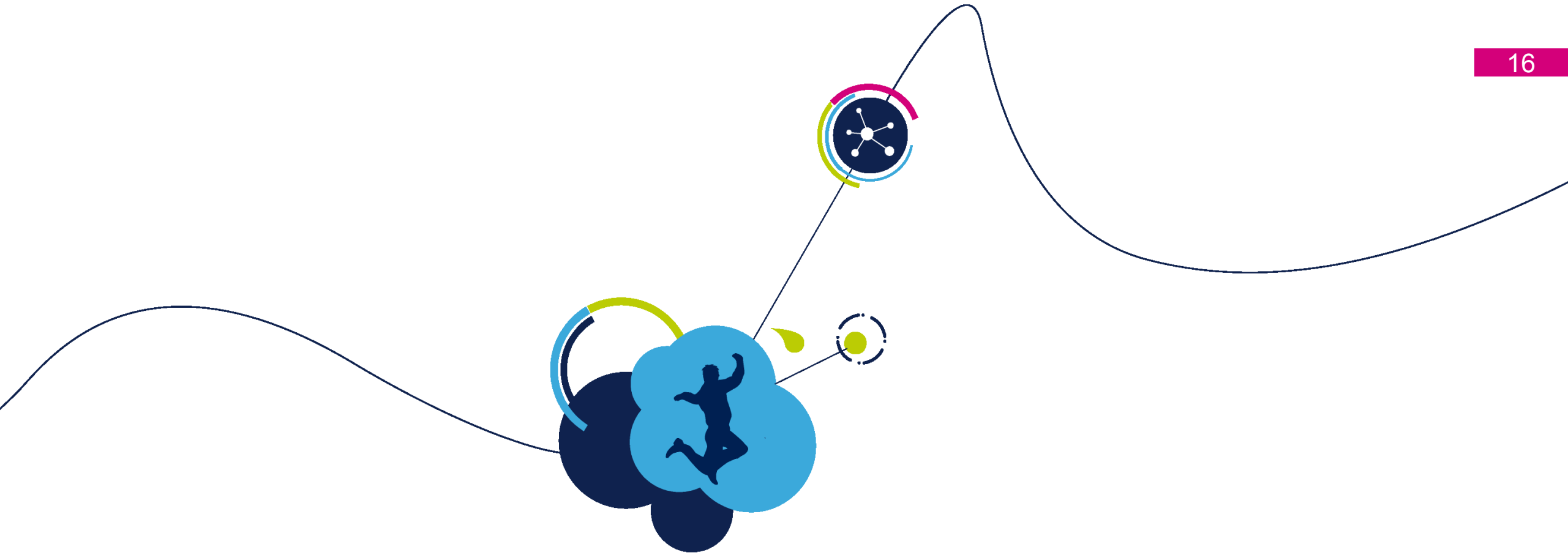


Business-friendly license terms

STM32CubeMP1 Package: Drivers

- Legacy framework from STM32 MCU
- Hardware Abstraction Layer (HAL)
 - High-level and feature-oriented APIs, hardware complexity hidden to users
- Low Level Drivers (LL)
 - Low-level APIs at registers level with better optimization
 - Available for most of the peripherals
- Board Support Package (BSP) drivers
 - Based on HAL peripheral drivers
 - Provide APIs for external components on ST development boards



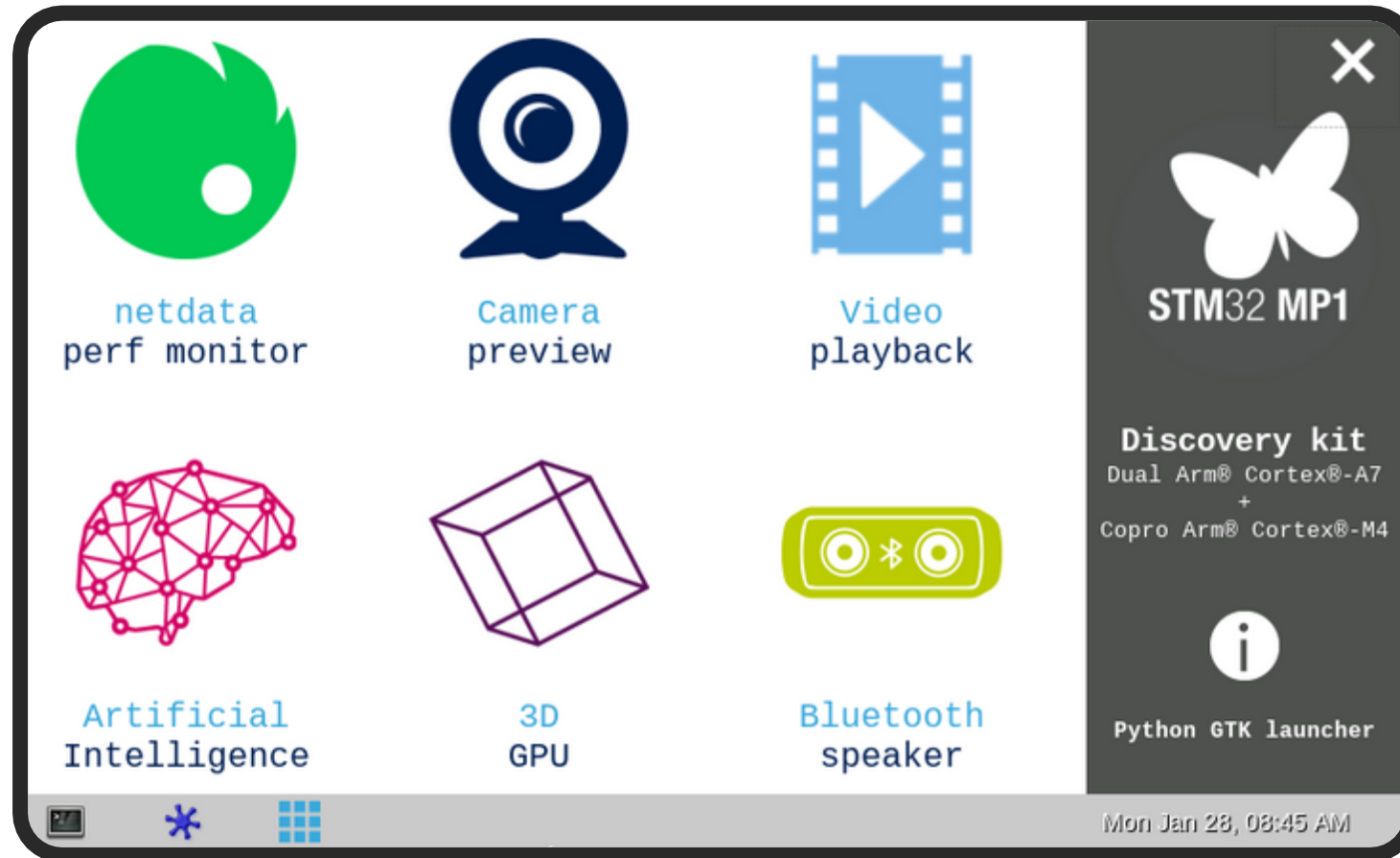


Starter Package demo launcher

Examples

One demo launcher

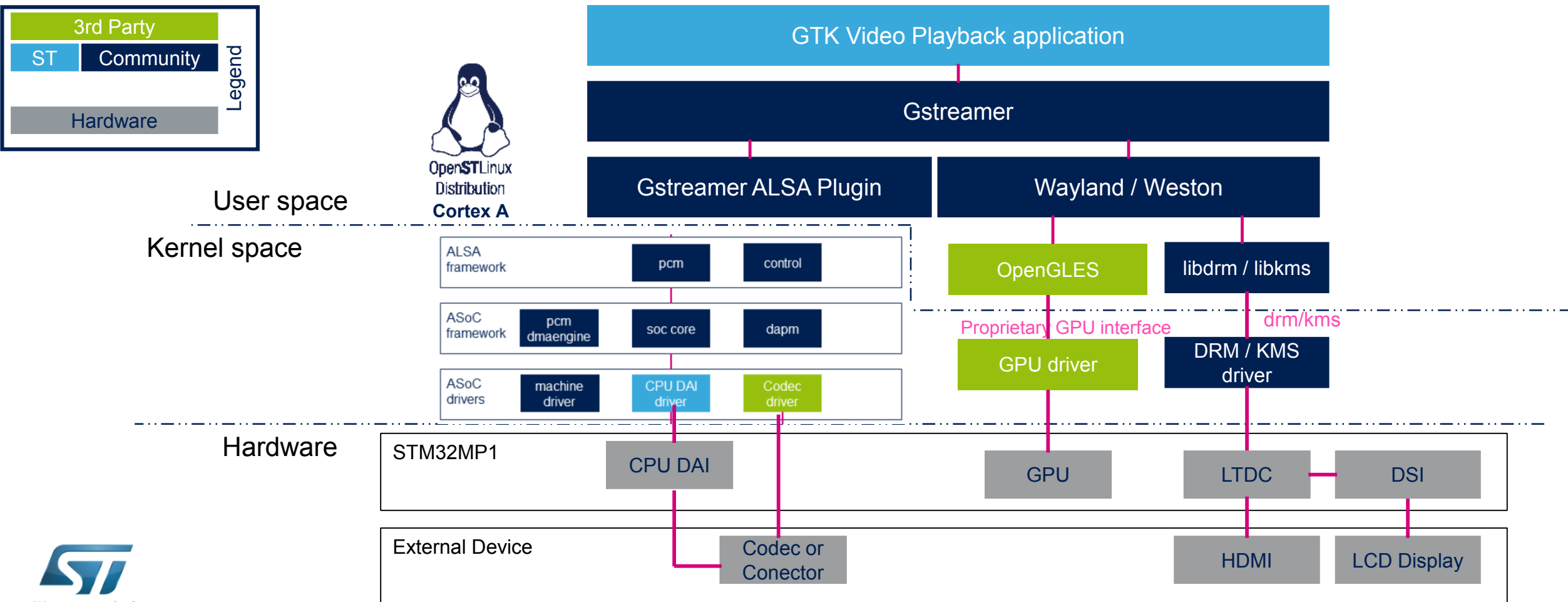
17



Discovery Kit : GTK Demo Launcher

18

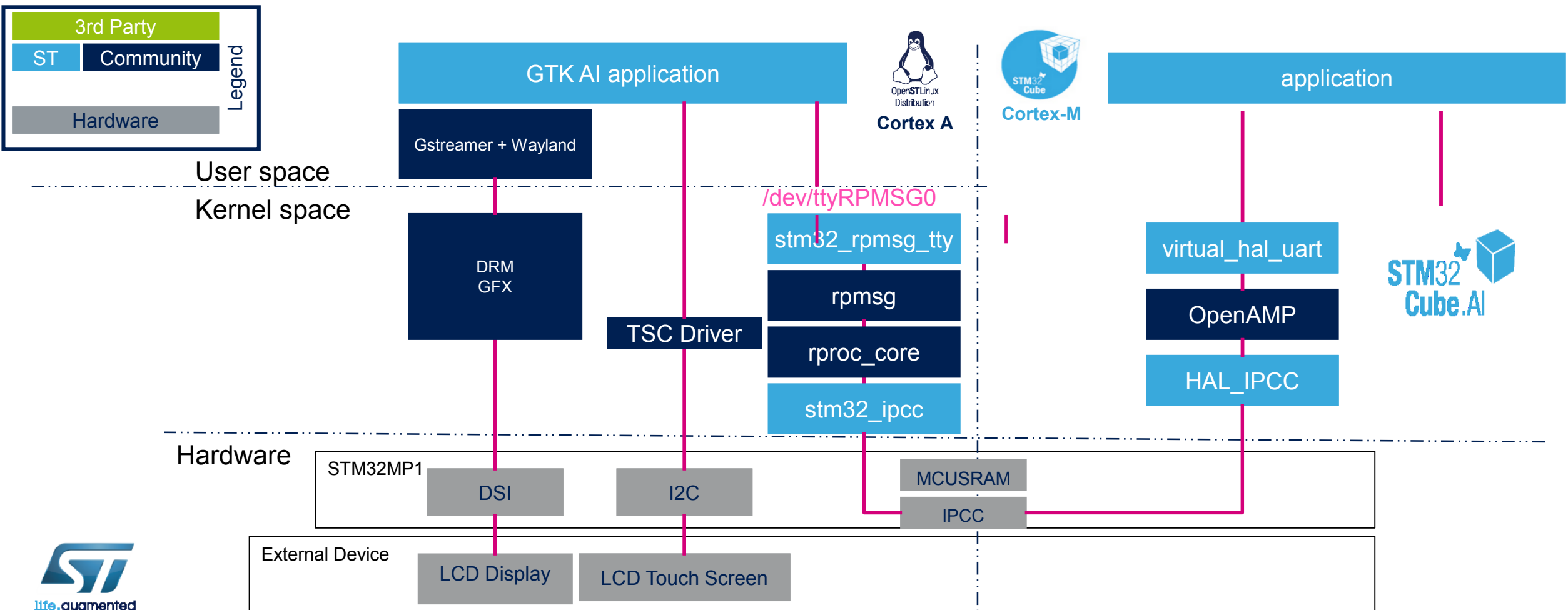
Video Playback

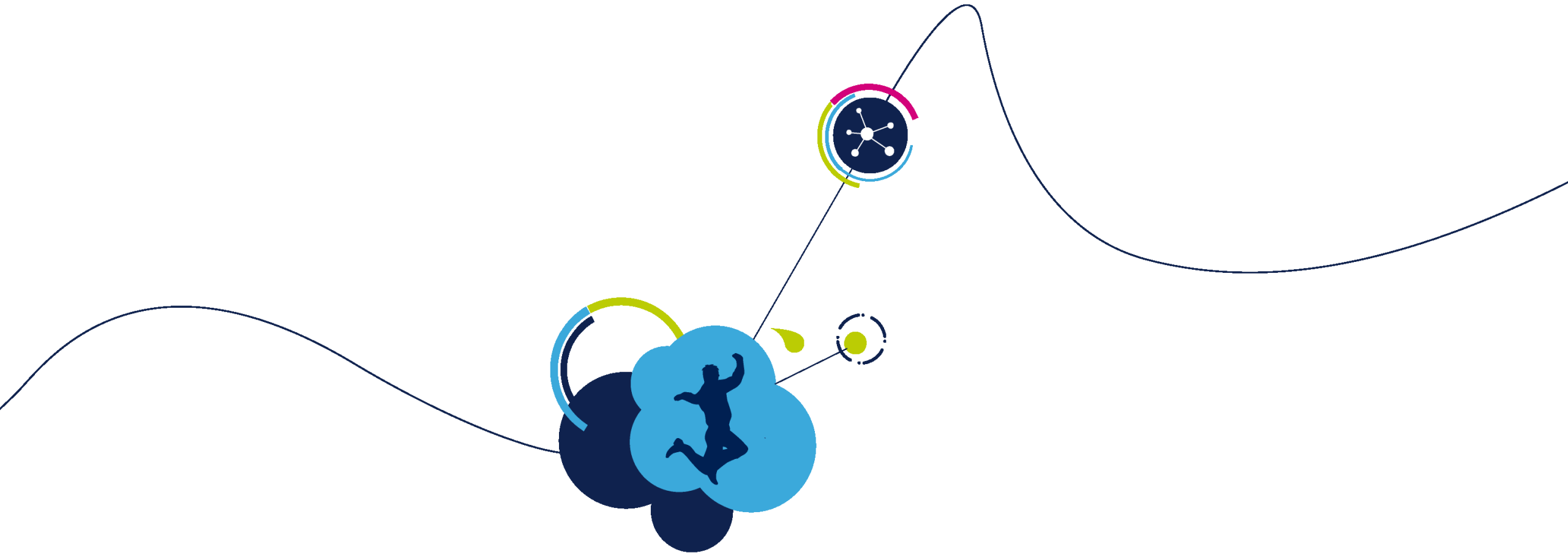


Discovery Kit : GTK Demo Launcher

19

Artificial Intelligence

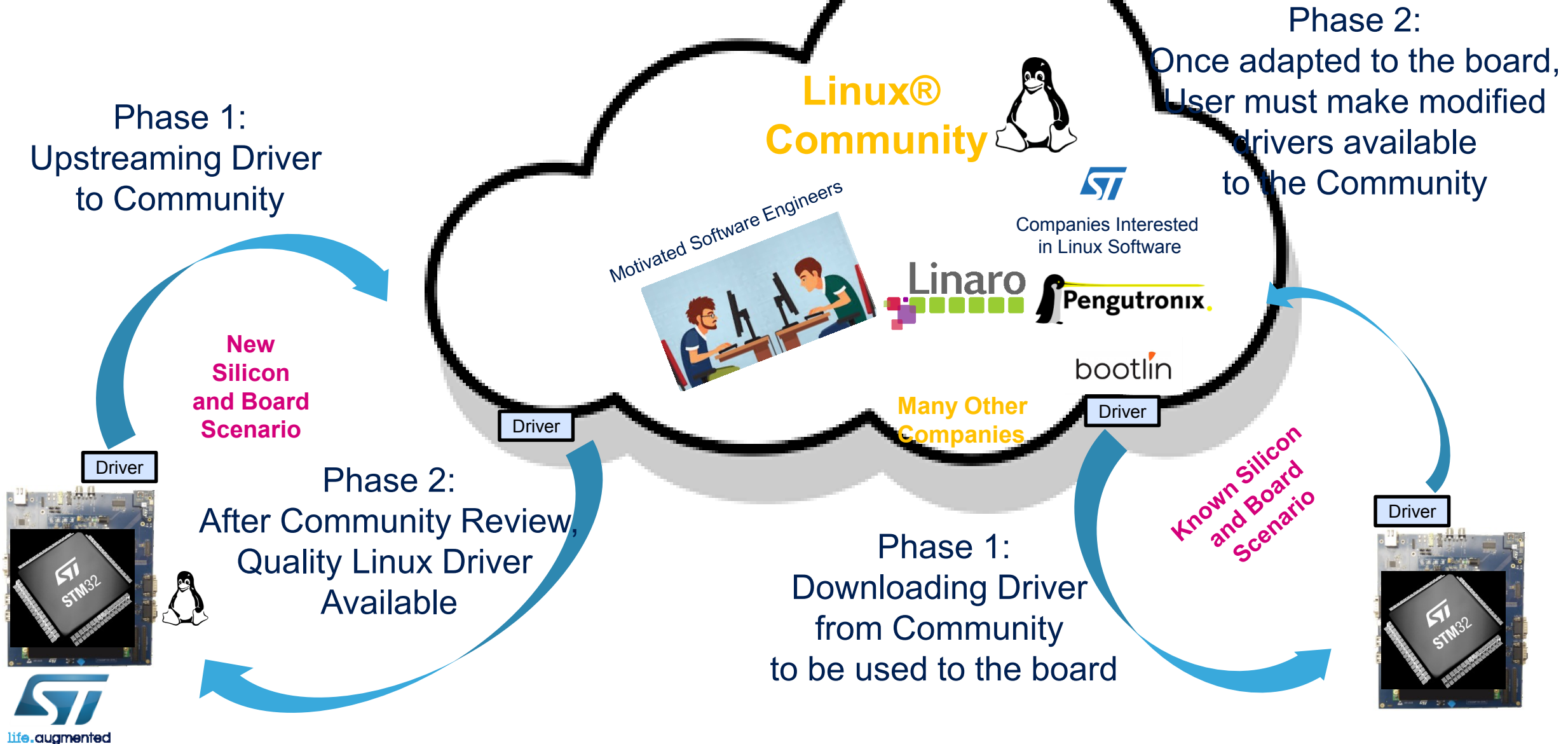




Mainline Explanation & Linux Support

What Does Mainlining Mean?

22



Upstream Process: Customers Top Advantages

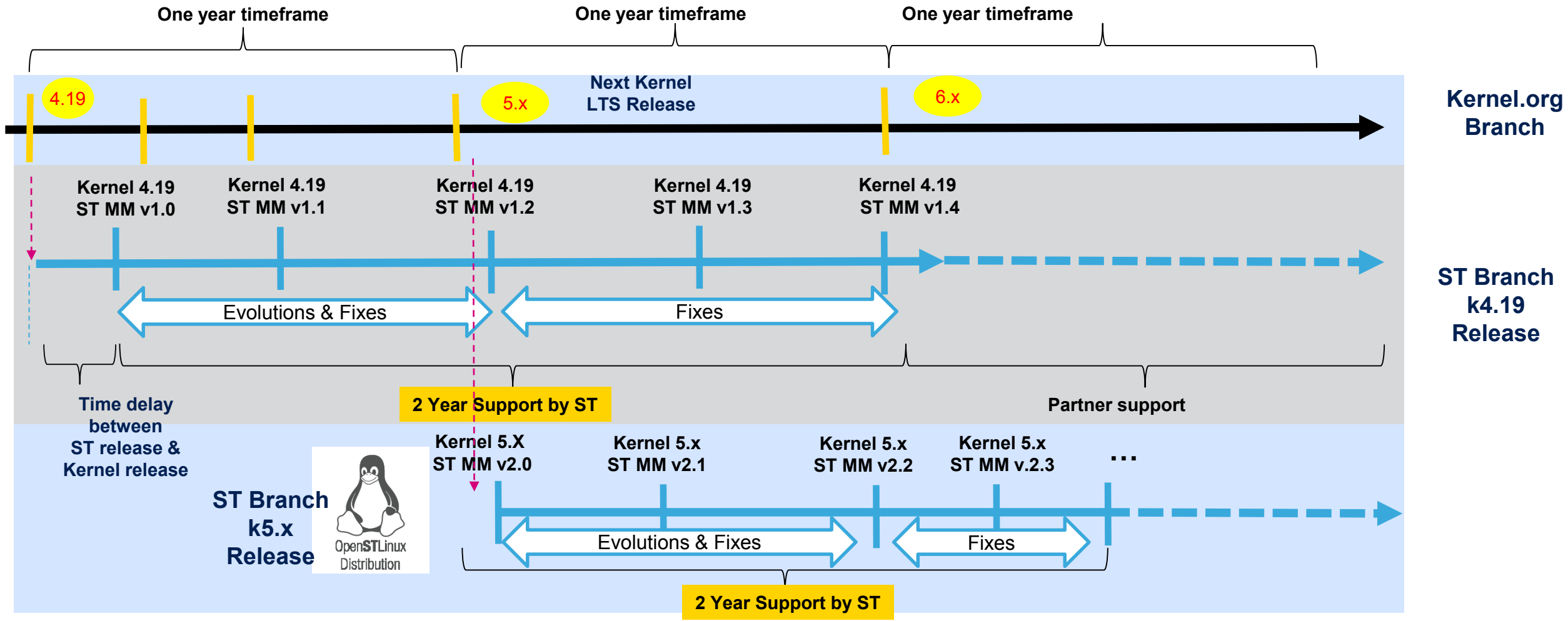
24

- Software Scalability towards platforms
 - Seamless Transition during kernel upgrade for already up-streamed drivers
 - Seamless Transition when porting towards derivative Hardware and Product Series
- Ensuring Software Quality
 - Self-enhanced security breaches from the Community
 - Enrich Distribution with newest and well tested features set
 - Reducing Software Risks
- Reducing Time to Market & Cost of the Solution
 - Strengthening the core business differentiators to drive innovation and profit



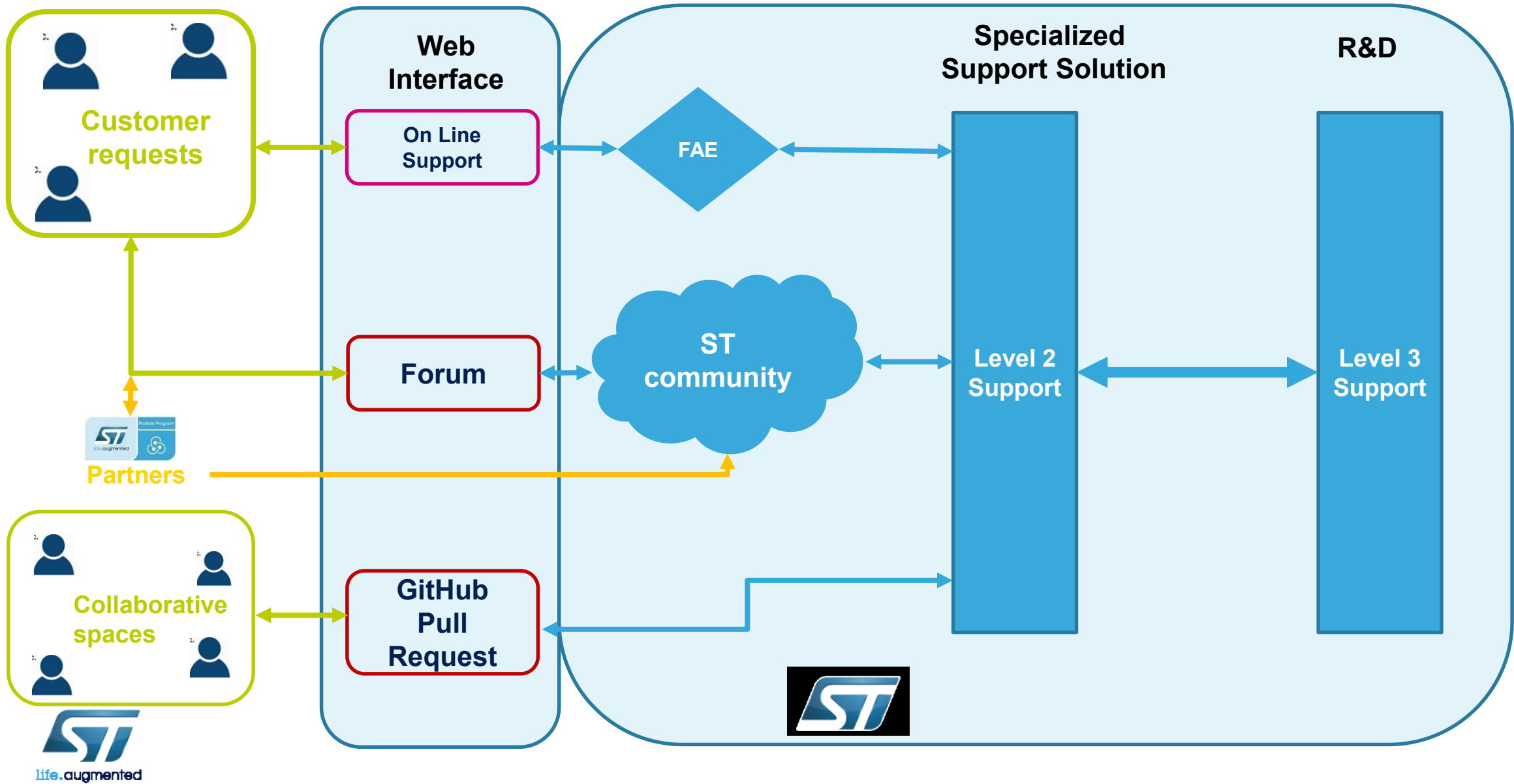
Linux Delivery & ST Support Process

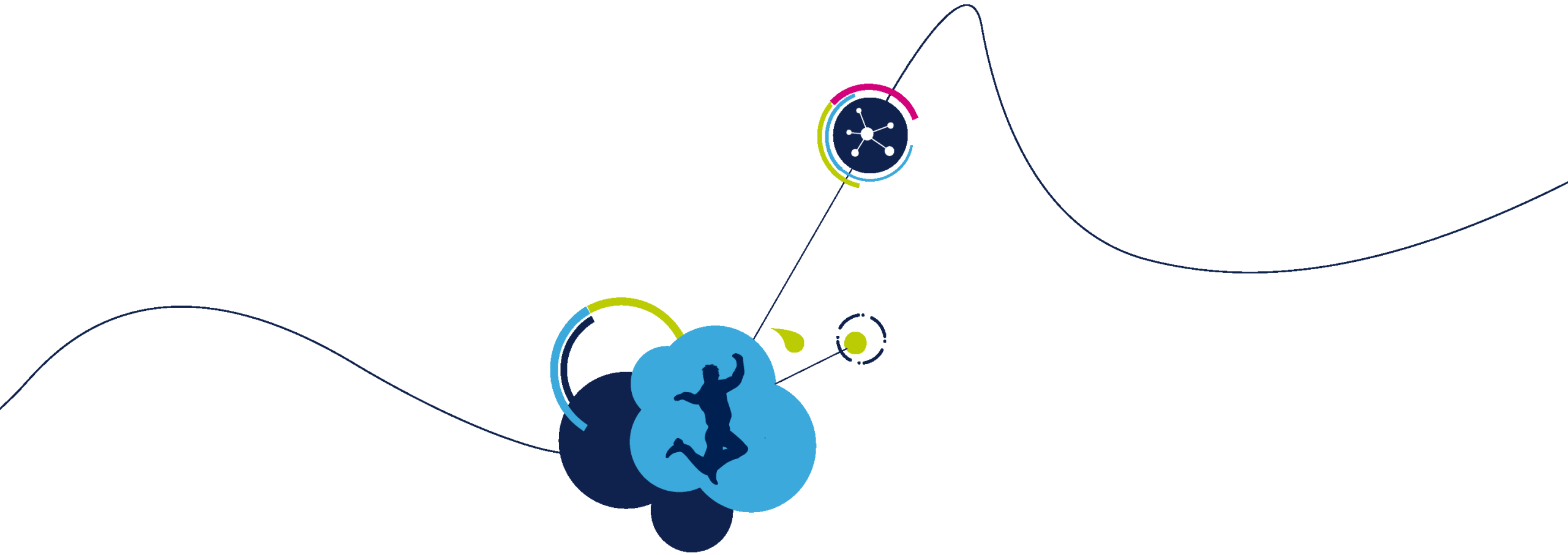
25



Support Channels, Communities

26





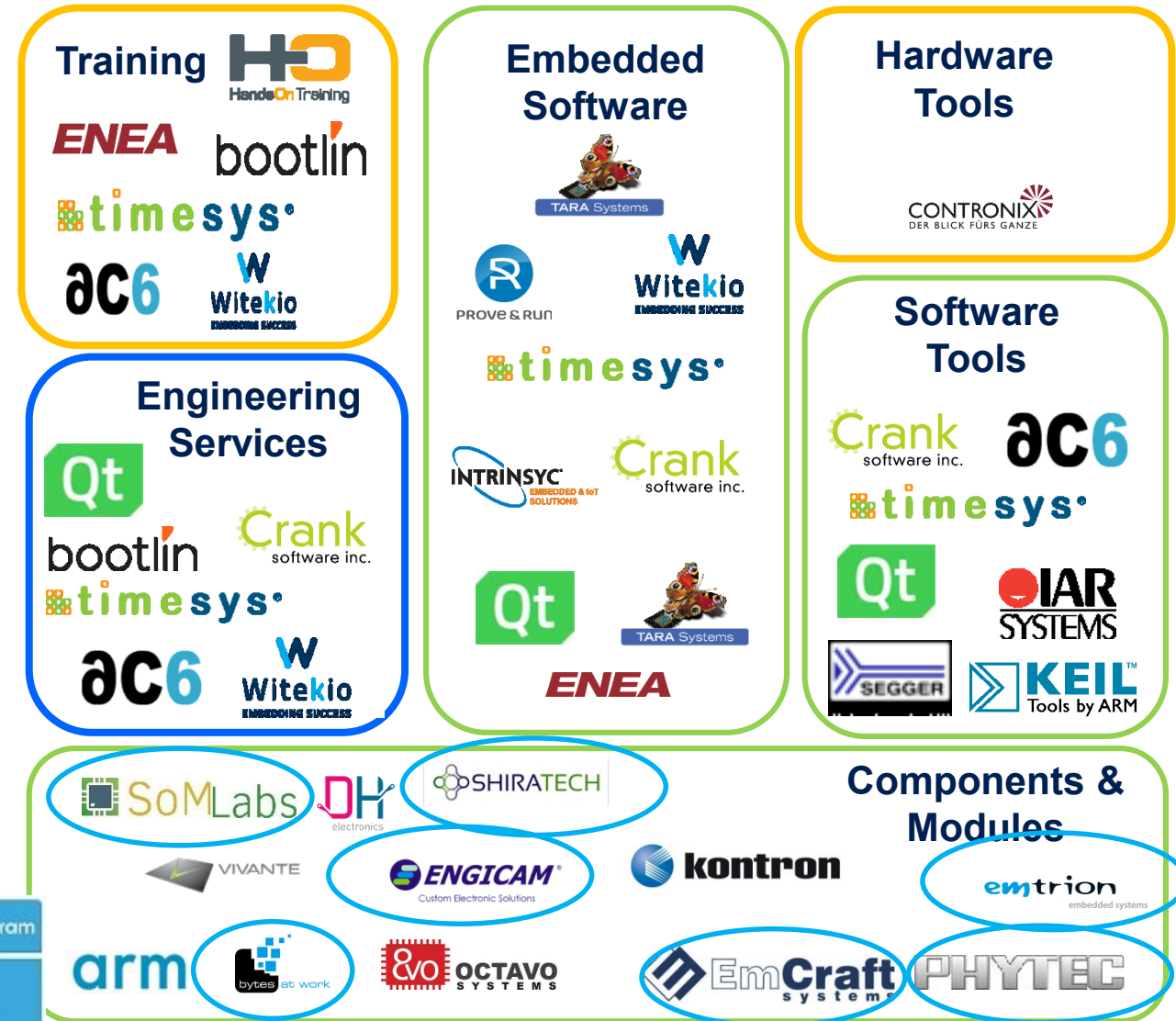
Third Party Ecosystem

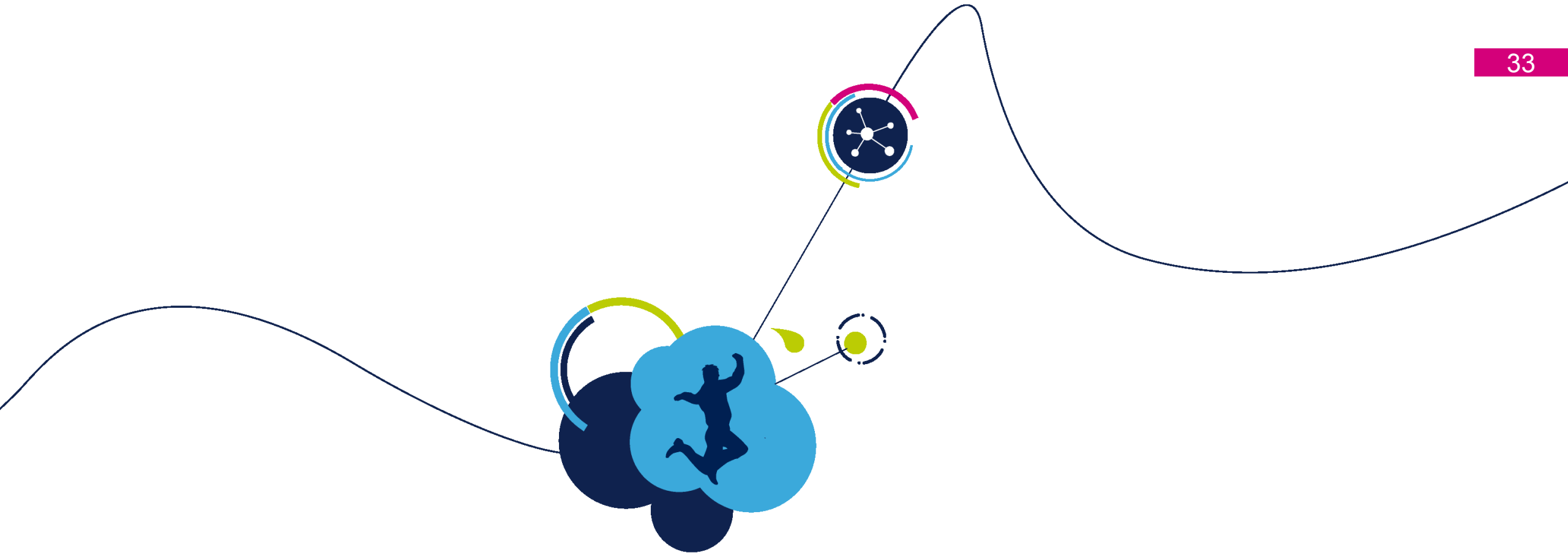


STM32MP1 Series Partners List

28

- STM32MP1 Partners List evangelizing the STM32MP1 Series Solution
- Highly skilled recognized leaders in their specific domain to help customers
- Early access of the technology to benefit Customers program in order to make success
- On-going ST Partner Program to enrich STM32MP1 Series Ecosystem
- 3rd party service description on St.com : https://www.st.com/content/st_com/en/partner/partner-program/partnerpage.html?key=MPU&country=country





Embedded Software Distribution Delivery

One distribution, three packages

34



Starter Package

- Evaluation (a-la-Raspberry)



Developer Package

- For customers adaptation of the OpenSTLinuxBSP, the kernel
- Create an application



Distribution Package

- For customers who want to deliver their distribution in standard way
- Customize application framework

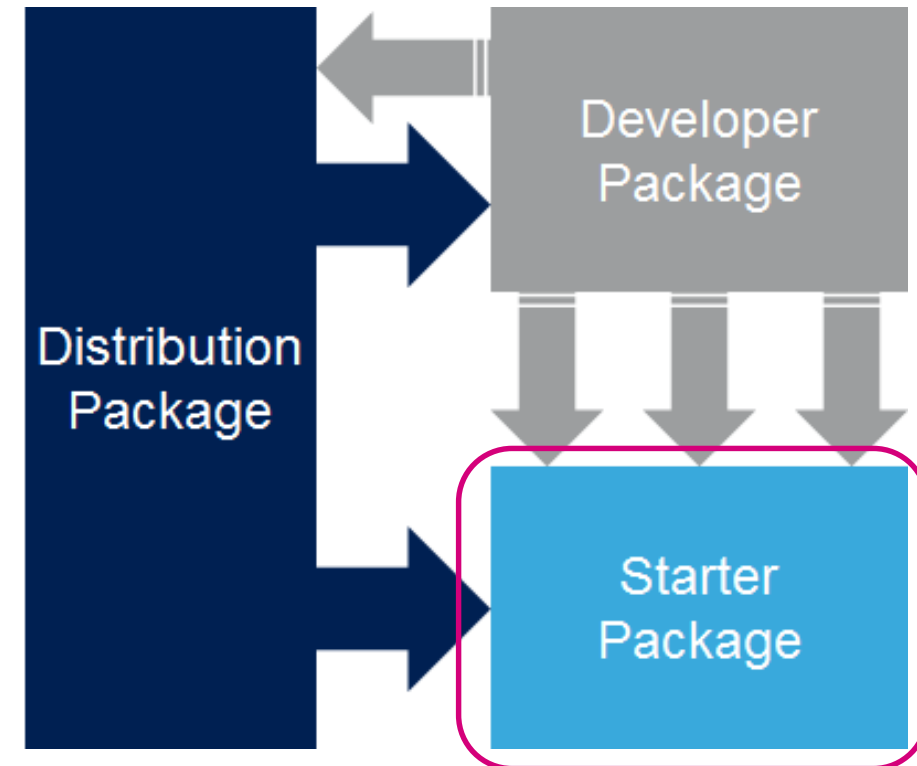
Purpose is MP1 evaluation

What you can do

- Linux script programming (bash, python)
- Download on board (through scp or usb key) an application binary

Contents : “st-image-Weston” distribution image

- OpenSTLinuxBSP (bootChain, Kernel)
- OpenSTlinux Application framework (Weston/Wayland GUI)
- The binary image generated from the distribution package



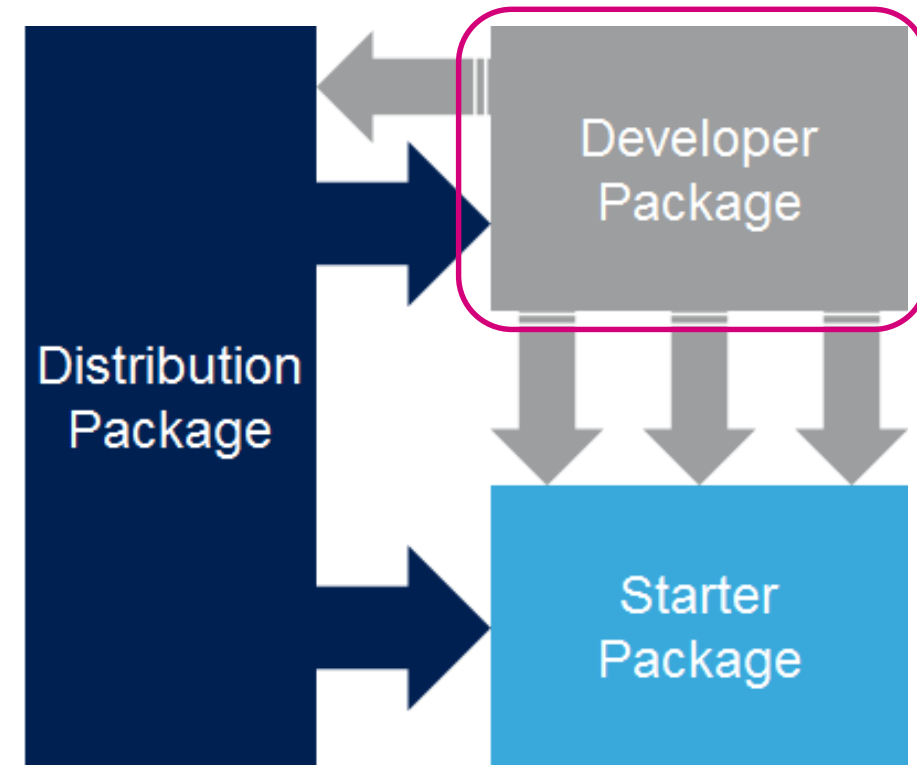
Purpose for developers

What you can do

- Adapt OpenSTLinuxBSP to customer board
modify Kernel, device tree, boot chain, Op-tee
- Develop application
- Quick build and debug time

Contents:

- Starter kit st-image-weston binaries
- SDK -makefiles
- Tarball of *OpenSTLinuxBSP* source code
Kernel, Uboot, TF-A, Op-tee, M4Cube-FW



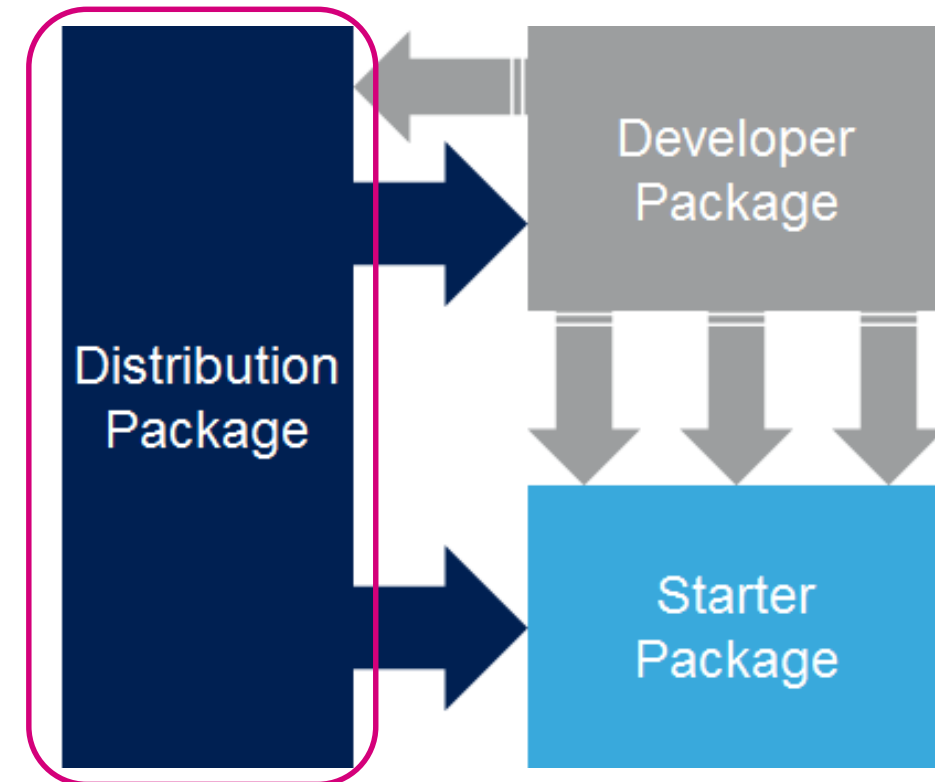
Purpose create a customer distribution

What you can do

- A “distribution Yocto interoperable”
- Add/remove application framework adjusted to customer application
- BSP adaptations

Contents

- Full OpenSTLinux distribution source code
- STM32CubeMPU Package (tool, M4 drivers)
- ST Metadata layers on Yocto servers
- YoctoOpenEmbedded tool (bitbake & metadata)



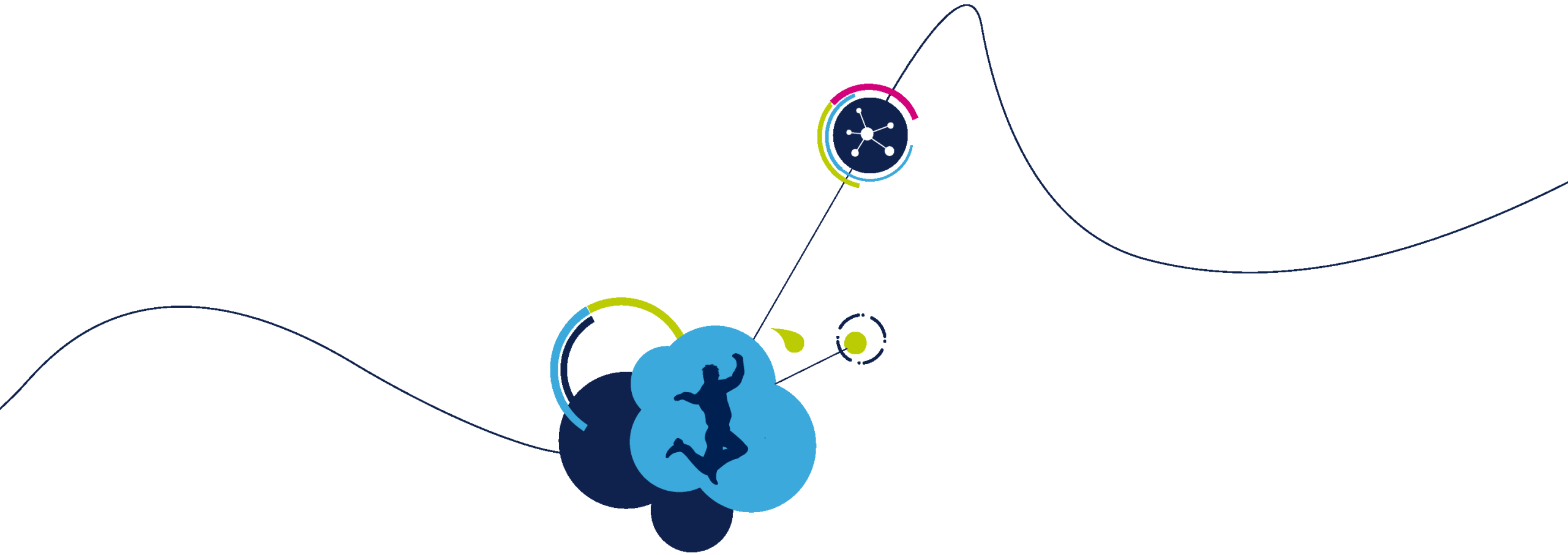
ST provides 2 yocto layers

OpenSTLinux BSP: **meta-st-stm32mp**

For Machines (board setup) :

- stm32mp1 <- common to all boards support
- stm32mp1-eval, stm32mp1-dk <- variant specific to one board

OpenSTLinuxApplication: **meta-st-openstlinux**



Licensing Terms

Generic Licensing Model

42

- Code delivered under different terms and conditions
 - While free, using the code must follow some rules
 - Development code can be used without restriction
 - Production code must follow the Licensing rules
 - More Information on : <https://www.gnu.org/licenses/licenses.en.html>

| Licensing Type | Rules | Remark |
|----------------|---|---|
| GPLV2, LGPLv2 | <ul style="list-style-type: none">• Freely Using the Community Code• Modified Code must be made available upon request | GPLv2 LGPLv2 |
| GPLv3, LGPLv3 | <ul style="list-style-type: none">• Entire Source code must be made available to the community | GPLv3 LGPLv3 |
| BSD, MIT, PSF | <ul style="list-style-type: none">• Uses as is without constraint | |
| Apache | <ul style="list-style-type: none">• Free use, modification, distribution of the code | here |

ST Linux® Licensing Delivery (1/2)

43

- **STM32 OpenSTLinux Distribution Delivery:**
 - Underneath the SLA048 clickthru license.
 - Includes:
 - OpenSTLinux as FOSS Compliant Distribution
 - CubeMP1 as FOSS
 - 3 Proprietaries licences
 - **GCNano** from Vivante, **BT** & **Wifi** from Cypress
- >95% of the drivers Delivered under GPLv2, MIT, BSD Licensing
- Some code is provided under GPLv3
- To avoid code contamination of the source code by GPLv3:
 - Vendors partition has been created to isolate the non GPLv3 Compatible Code (GCnano from Vivante IP) to follow the ST Policy
 - GPLv3 code mainly used for development purposes (devTools)

- **More Information on:**

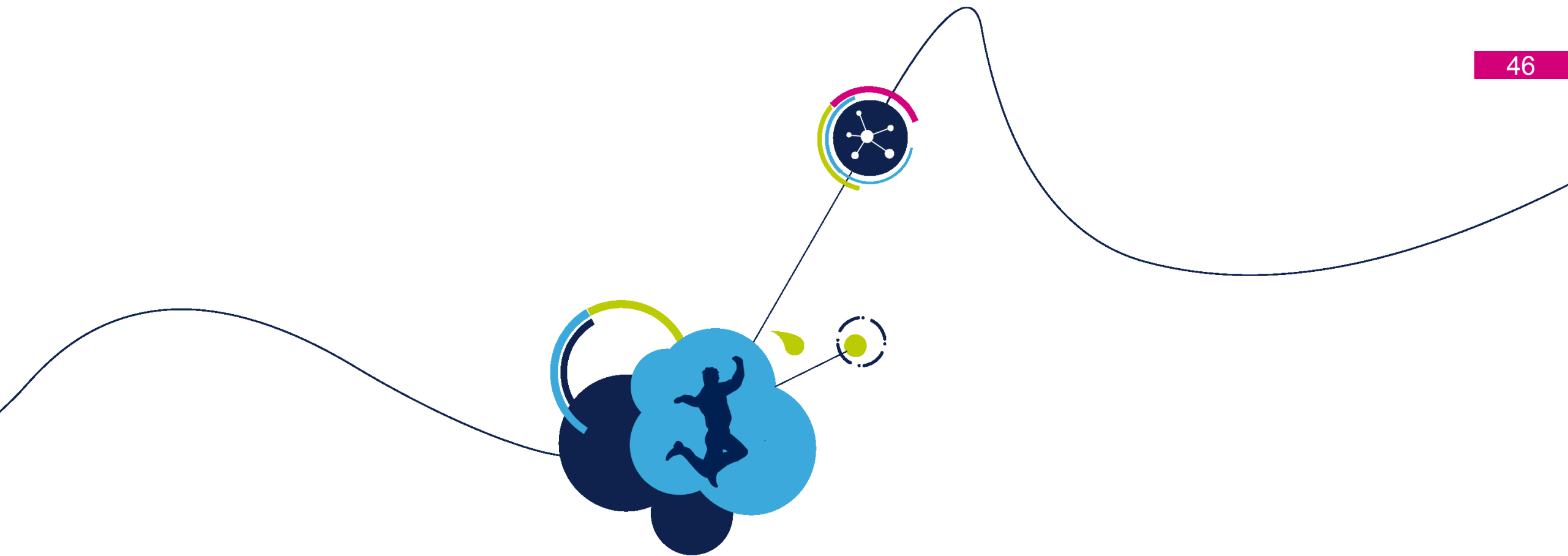
- https://wiki.st.com/stm32mpu/wiki/OpenSTLinux_licenses

- **Audio / Video Codecs**

- All codec provided thru OpenSTLinux is Open Source
- In case of proprietary Codec, Customers / Partners are responsible to be legally bounded to the Audio / Video Codec IP Providers

- **Wifi & Bluetooth**

- Wireless Solution is Murata Module based on BCDM/Cypress BCM83xx Device
- Mainly using the Community Drivers from both Kernel and user spaces.
- Firmware included into rootfs partition, fully open source code



Software wiki guide

One Wiki, <https://wiki.st.com/>

47

The screenshot shows the STM32 MPU wiki homepage. At the top left is the STM logo and the text "STM32 MPU wiki by". To the right is a search bar with a magnifying glass icon and a "Search" button. Further right are icons for a document and a plus sign. Below the header is a dark blue navigation bar with the following items: "Welcome", "Reading tips", "Getting started", "Development zone" (circled in red), and "Legal notice". On the left side of the page is a sidebar menu with the following items: "Main page", "Popular Pages", "Help & Syntax", "Glossary", "Contributors", "Release notes", "Getting started with ST boards", "STM32 MPU microprocessor devices", "Embedded software", "Tools", "How to", "Other resources", "Development zone" (highlighted in blue), "PC prerequisites", and "Which Package better suits your needs". The main content area features five cards with icons and titles: "Getting started with ST boards" (yellow board icon), "STM32 MPU microprocessor devices" (blue chip icon), "Embedded software" (pink circuit icon), "Tools" (blue laptop icon), and "How to" (pink lifebuoy icon). Each card has a brief description of its content.

STM32 MPU wiki by

Search

Welcome Reading tips Getting started **Development zone** Legal notice

Main page
Popular Pages
Help & Syntax
Glossary
Contributors
Release notes
Getting started with ST boards
STM32 MPU microprocessor devices
Embedded software
Tools
How to
Other resources
Development zone
PC prerequisites
Which Package better suits your needs

Getting started with ST boards
Discover the description of the boards for the STM32 MPUs microprocessor devices, and how to start with them

STM32 MPU microprocessor devices
Browse the peripherals and the multiple-core architecture of the STM32 MPUs microprocessor devices

Embedded software
Find the supported distributions, the architecture overviews, the embedded software components (frameworks and drivers)...

Tools
Discover our ecosystem software tools, and our suggested standard tools and methods

How to
Browse the advices and tips to tune the software

One Wiki, Getting Started

48



Main page

Glossary

STM32MP1 boards

- Getting started

General Books

Getting started

- 1. Getting started
 - 1.1. STM32MP1 boards

Welcome

Reading tips

Getting started

Development zone

Legal notice

Approved version. Approved on: 13:26, 15 February 2019

Page76 days, 17 hours and 56 minutes -

Getting started★★★★★ (2)

Select your series

One Wiki, Development Zone

49



[Main page](#)
[Glossary](#)
[Release notes](#)

- Release notes archives
 - STM32MP15 ecosystem release note
 - STM32MP15 OpenSTLinux release note
 - STM32CubeMP1 Package release note
 - STM32CubeProgrammer release note
 - KeyGen release note
 - Signing tool release note
 - STM32-CoPro-MPU plugin release note
- Getting started with ST boards
- STM32 MPU microprocessor devices
- Embedded software
- Tools
- How to
- Other resources

[Welcome](#)[Reading tips](#)[Getting started](#)[Development zone](#)[Legal notice](#)

Approved version. Approved on: 13:19, 25 February 2019

Page 66 days, 18 hours and 6 minutes -

Development zone (0)

PC prerequisites





Check the recommended configuration of the host environment

Release notes

Find out the release notes related to the [STM32 MPU](#) microprocessor devices

Which Package better suits your needs

Discover the software Packages delivered for the [STM32 MPU](#) microprocessor devices



WelcomeReading tipsGetting startedDevelopment zoneLegal notice

↶ Main page

📖 Glossary

✓ Release notes

➤ Release notes archives

- STM32MP15 ecosystem release note
- STM32MP15 OpenSTLinux release note
- STM32CubeMP1 Package release note
- STM32CubeProgrammer release note
- KeyGen release note
- Signing tool release note
- STM32-CoPro-MPU plugin release note

➤ Getting started with ST boards

➤ STM32 MPU microprocessor devices

➤ Embedded software

Pages in category "Release notes"↑

The following 7 pages are in this category, out of 7 total.

0↑

- STM32MP15 ecosystem release note

1↑

- STM32MP15 OpenSTLinux release note

3↑

- STM32CubeMP1 Package release note
- STM32CubeProgrammer release note

K↑


- KeyGen release note

S↑



- Signing tool release note

5↑

- STM32-CoPro-MPU plugin release note


life.augmented

https://wiki.st.com/stm32mpu/wiki/Category:Release_notes

 Main page
  Glossary

 > Release notes

 > Getting started with ST boards

 > STM32 MPU microprocessor devices

 > Embedded software

 > Tools

 > How to

 > Other resources

 Development zone

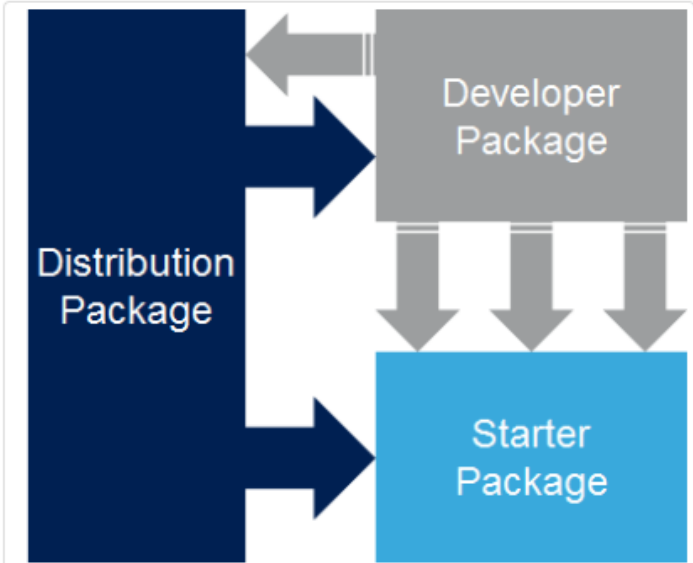
 PC prerequisites

Which Package better suits your needs



WelcomeReading tipsGetting startedDevelopment zoneLe

The STM32MPU Embedded Software distribution for STM32 microprocessor platforms supports three software Packa



- The **Starter Package** to quickly and easily start with any STM32MP microprocessor device. The Starter Package is generated from the Distribution Package.
- The **Developer Package** to add your own developments on top of the STM32MPU Embedded Software distributio replace the Starter Package pre-built binaries. The Developer Package is generated from the Distribution Package
- The **Distribution Package** to create your own Linux® distribution, your own Starter Package and your own Develo Package.








- > Release notes
- > Getting started with ST boards
- > STM32 MPU microprocessor devices
- Embedded software**
- ✓ Distributions
 - ✓ STM32MPU Embedded Software distri...
 - ✓ OpenSTLinux distribution
 - > OpenSTLinux filesystem
 - OpenSTLinux distribution
 - Dpkg
 - GTK demo launcher
 - OpenEmbedded
 - STM32MP15 OpenSTLinux relea...
 - ✓ STM32Cube MPU Package
 - STM32CubeMP1 Package
 - STM32CubeMP1 architecture
 - STM32CubeMP1 Package relea...
 - Example of directory structure for P...

Category  

Embedded software Help













Embedded software components




Architecture overviews, software components for platform boot (TF-A, U-Boot...) and for platform configuration, operating system frameworks and drivers

Supported distributions for STM32 MPU devices

Subcategories 





WelcomeReading tipsGetting startedDevelopment zoneLegal notice>>



[Main page](#)
[Glossary](#)
[Release notes](#)
[Getting started with ST boards](#)
[STM32 MPU microprocessor devices](#)
[Embedded software](#)
 [Distributions](#)
 [STM32MPU Embedded Software distri...](#)
 [OpenSTLinux distribution](#)
 [OpenSTLinux filesystem](#)
 [OpenSTLinux distribution](#)
 [Dpkg](#)
 [GTK demo launcher](#)
 [OpenEmbedded](#)
 [STM32MP15 OpenSTLinux relea...](#)
 [STM32Cube MPU Package](#)
 [Example of directory structure for P...](#)

Approved version. Approved on: 12:17, 14 March 2019

Page52 days, 18 hours and 33 minutes -  

OpenSTLinux distribution



★★★★★ (0)

Contents

[↑](#) [\[hide\]](#)

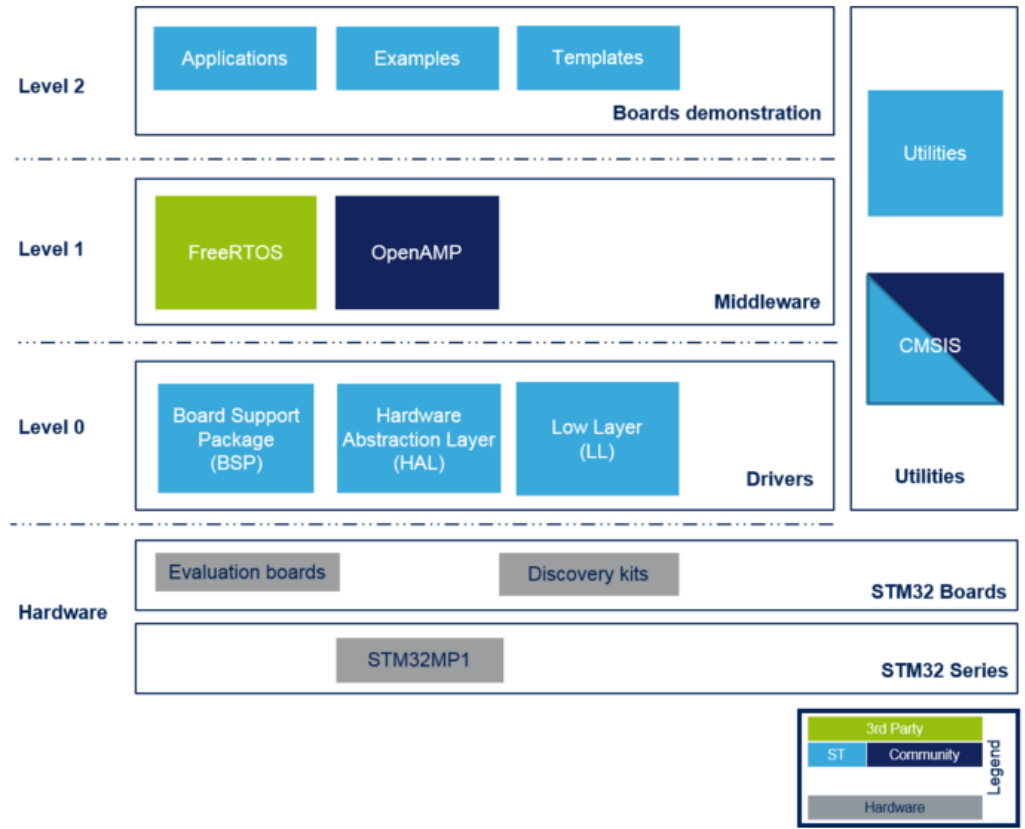
- 1 What is the OpenSTLinux distribution?
- 2 Software architecture overview
- 3 OpenSTLinux concept
 - 3.1 Layers
 - 3.2 Machines
 - 3.3 Images
 - 3.4 Distro
 - 3.5 Reference source code
- 4 How to get the software for this distribution
- 5 References

[1 What is the OpenSTLinux distribution?](#)
[2 Software architecture overview](#)
[3 OpenSTLinux concept](#)
[4 How to get the software for this distribution](#)
[5 References](#)

 Main page
 Glossary
 > Release notes
 > Getting started with ST boards
 > STM32 MPU microprocessor devices
 > Embedded software
 > Distributions
 > STM32MPU Embedded Software distri...
 > OpenSTLinux distribution
 > STM32Cube MPU Package
 [STM32CubeMP1 Package](#)
 STM32CubeMP1 architecture
 STM32CubeMP1 Package relea...
 Example of directory structure for P...
 > STM32MPU Android Embedded Softw...
 > Embedded software components
 > Tools
 > How to

Welcome Reading tips Getting started Development zone Legal notice

The figure below gives an overview of the [STM32CubeMP1_architecture](#).



The diagram illustrates the STM32CubeMP1 architecture across four levels:





- Level 2:** Applications, Examples, Templates (Boards demonstration)
- Level 1:** FreeRTOS, OpenAMP (Middleware)
- Level 0:** Board Support Package (BSP), Hardware Abstraction Layer (HAL), Low Layer (LL) (Drivers)
- Hardware:** Evaluation boards, Discovery kits (STM32 Boards)

Additional components shown on the right include Utilities and CMSIS.



Legend:

- 3rd Party (Green)
- ST (Blue)
- Community (Dark Blue)
- Hardware (Grey)

1 Introduction to STM32Cube
 2 STM32CubeMP1 package overview
 3 [Software architecture overview](#)
 4 STM32CubeMP1 Package Content
 5 Getting started with STM32CubeMP1 Package
 6 References

[Welcome](#)
[Reading tips](#)
[Getting started](#)
[Development zone](#)
[Legal notice](#)



[Category](#)



Embedded software components Help

This category groups together all articles and subcategories related to the software components designed for the STM32 MPU microprocessor devices and their associated boards.

**What is the architecture of the software running on my STM32MP1 board?
How can I configure the software for my needs?**

Click on the links in the frame below and let you drive!








STM32MPU Embedded Software architecture overview

Subcategories

This category has the following 7 subcategories, out of 7 total.

- Embedded software components
 - Architecture overview
 - OpenSTLinux architecture overview
 - OpenSTLinux BSP architecture overview
 - Security overview
 - STM32MPU Embedded Software architecture overview
 - Platform boot
 - ROM code
 - Trusted Firmware-A (TF-A)
 - U-Boot
 - Boot chains overview
 - Platform configuration
 - Device tree configuration
 - STM32MP15 platform configuration
 - Platform configuration overview
 - Device tree
 - Linux Operating System
 - OP-TEE secure OS

- Linux Operating System
- > Analog
- > Audio
- > Coprocessor management Linux
- > Core
- > Embedded trace and debug
- > High speed interface
- > Low speed interface
- > Networking
- > Persistent storage
- > Power and Thermal
- > Security
- > Visual
- STM32MP15 Linux kernel overview
- > OP-TEE secure OS
- > STM32Cube MPU
- > Applications and UI frameworks
- STM32MPU Embedded Software archi...

Welcome
Reading tips
Getting started
Development zone
Legal notice

Category
Linux Operating System
Help

This category groups together all articles and subcategories related to the Linux® software frameworks for the STM32 MPU microprocessor devices and their associated boards.

Subcategories↑

This category has the following 12 subcategories, out of 12 total.

A↑

- > Analog (1 C)
- > Audio (1 C, 2 P)

C↑

- > Coprocessor management Linux (8 P)
- > Core (11 C)

N↑

- > Networking (2 C, 1 P)

P↑

- > Persistent storage (1 C, 1 P)

Low speed interface (3 C)

Linux Operating System

- > Analog
- > Audio
- > Coprocessor management Linux
- > Core
- > Embedded trace and debug
- > High speed interface

Low speed interface

I2C

I2C i2c-tools

I2C device tree configuration

I2C driver

I2C i2c-dev

I2C overview

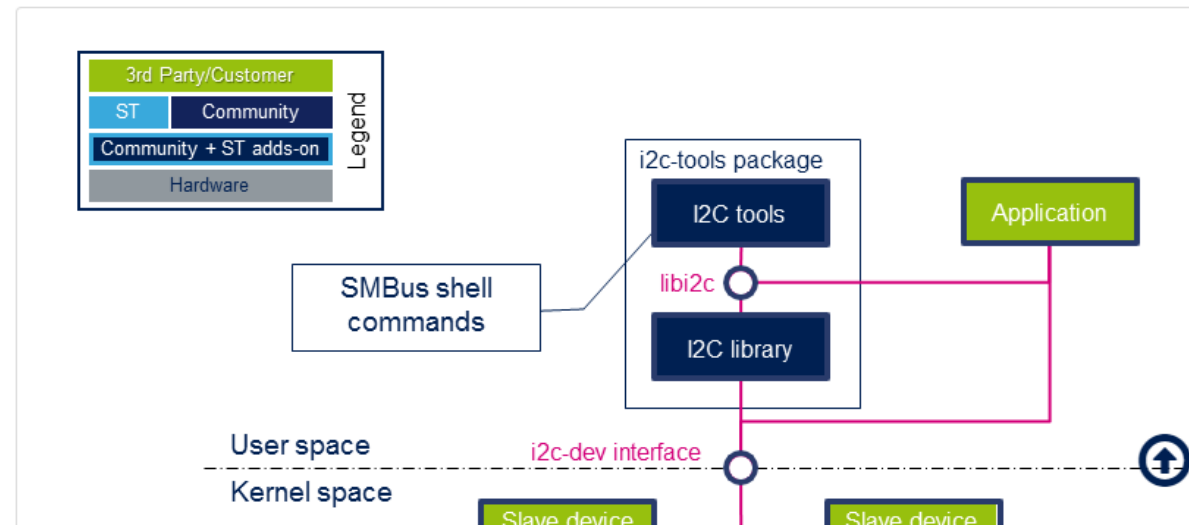
- > Serial TTY
- > SPI
- > Networking
- > Persistent storage

2 System overview

I²C is an acronym for the "Inter-IC" bus, a simple bus protocol which is widely used where low data rate communications suffice.

I2C is the acronym for the microprocessor I²C peripheral interface.

Around the microprocessor device, the user can add many I²C external devices to create a custom board. Each external device can be accessed through the I2C from the user space or the kernel space.



2.1 Component description

2.1.1 Board external I²C devices

2.1.2 STM32 I2C internal peripheral controller

2.1.3 i2c-stm32

2.1.4 i2c-core

2.1.5 Board peripheral drivers

2.1.6 i2c-dev

2.1.7 i2c-tools

2.1.8 Application

2.2 API description

2.2.1 libi2c

2.2.2 User space application

2.2.3 Kernel space peripheral driver

3 Configuration

4 How to use the framework

5 How to access I2C peripherals

- Main page
 Glossary
- > Release notes
 > Getting started with ST boards
 > STM32 MPU microprocessor devices
 > Embedded software
 > Distributions
 > Embedded software components
 > Architecture overview
 > Platform boot
 > Platform configuration
 > Linux Operating System
 > Analog
 > Audio
 > **ALSA**

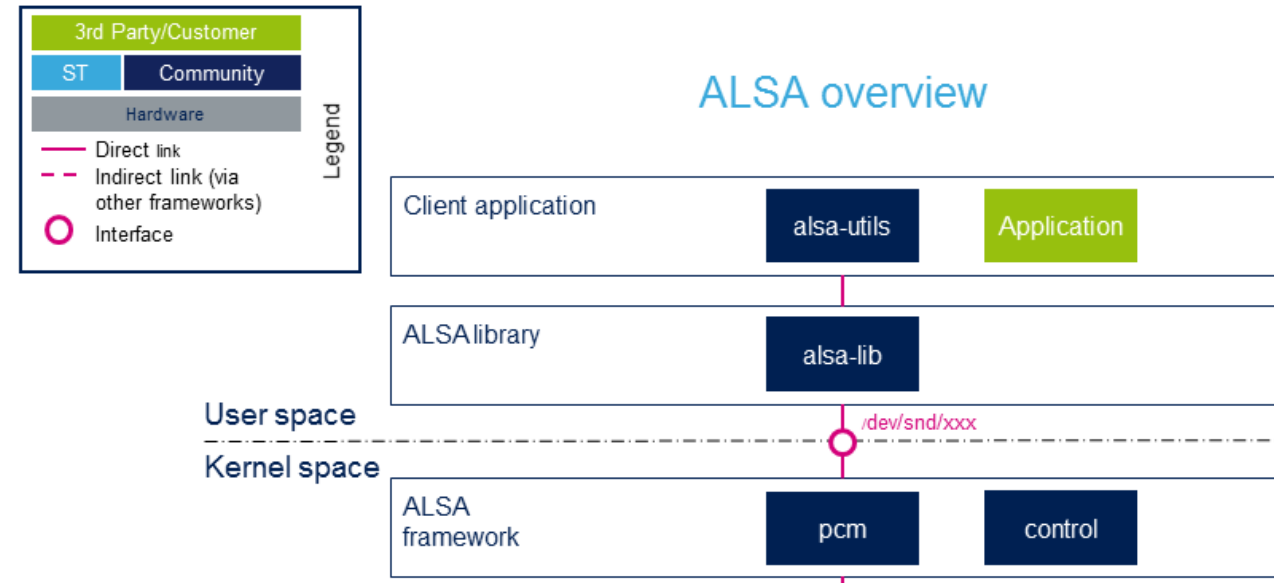
[ALSA overview](#)
[ALSA troubleshooting grid](#)
[DFSDM device tree configurat...](#)





1 Purpose

The purpose of this article is to introduce the [ALSA](#) framework.

The [ALSA](#) framework provides comprehensive audio functionality for Linux which includes recording and playing of audio streams, in either analog or digital formats together with routing and mixing capabilities. [ALSA](#) also supports audio middleware such as [PulseAudio](#), [Gstreamer](#) or [Android](#).

2 System overview



Welcome
Reading tips
Getting started
Development zone
Legal notice

Security
Visual
CEC
DRM KMS

DRM KMS overview
DRM KMS troubleshooting grid
DSI device tree configuration
How to get DRM KMS logs
How to get name and current ...
How to modify the panel backl...
How to monitor the display fra...
LTDC device tree configuration

GPU
GStreamer
V4L2
Wayland Weston
Visual troubleshooting grid
STM32MP15 Linux kernel overview

2 System overview

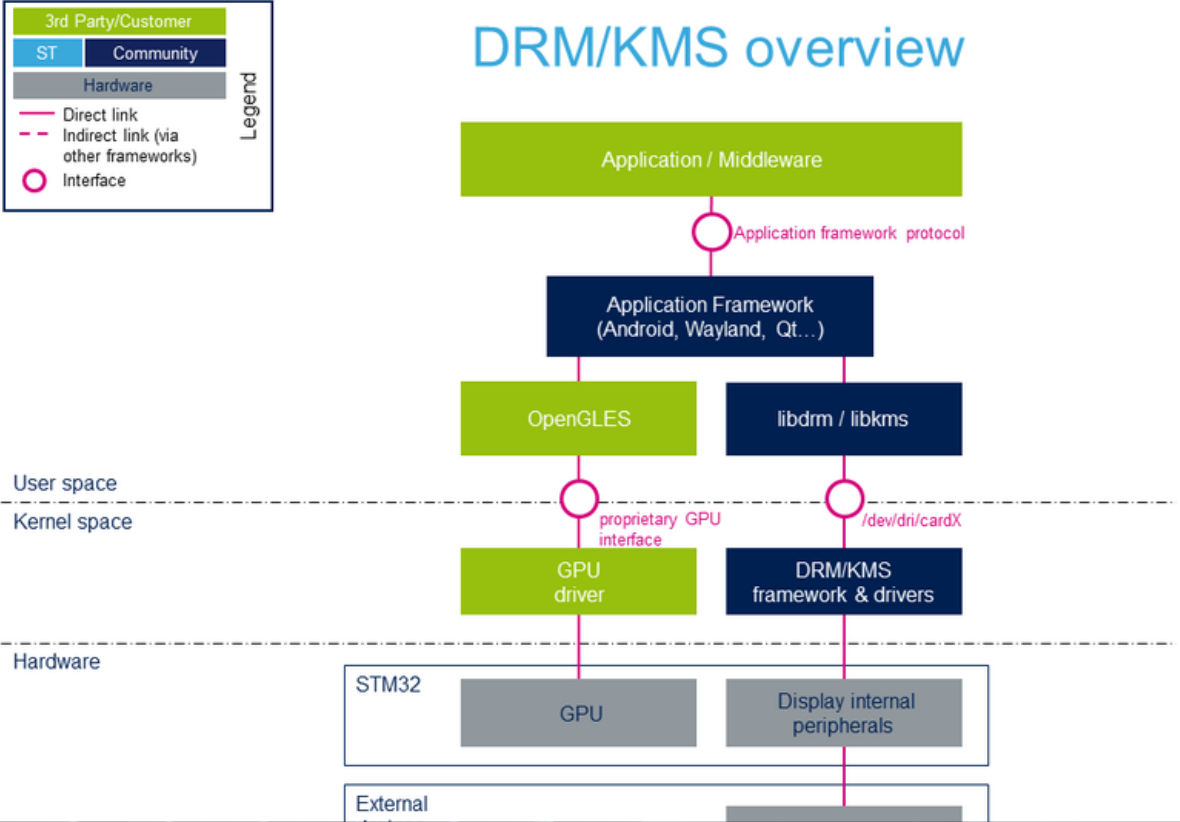
3rd Party/Customer

ST Community

Hardware

Direct link
Indirect link (via other frameworks)
Interface

DRM/KMS overview



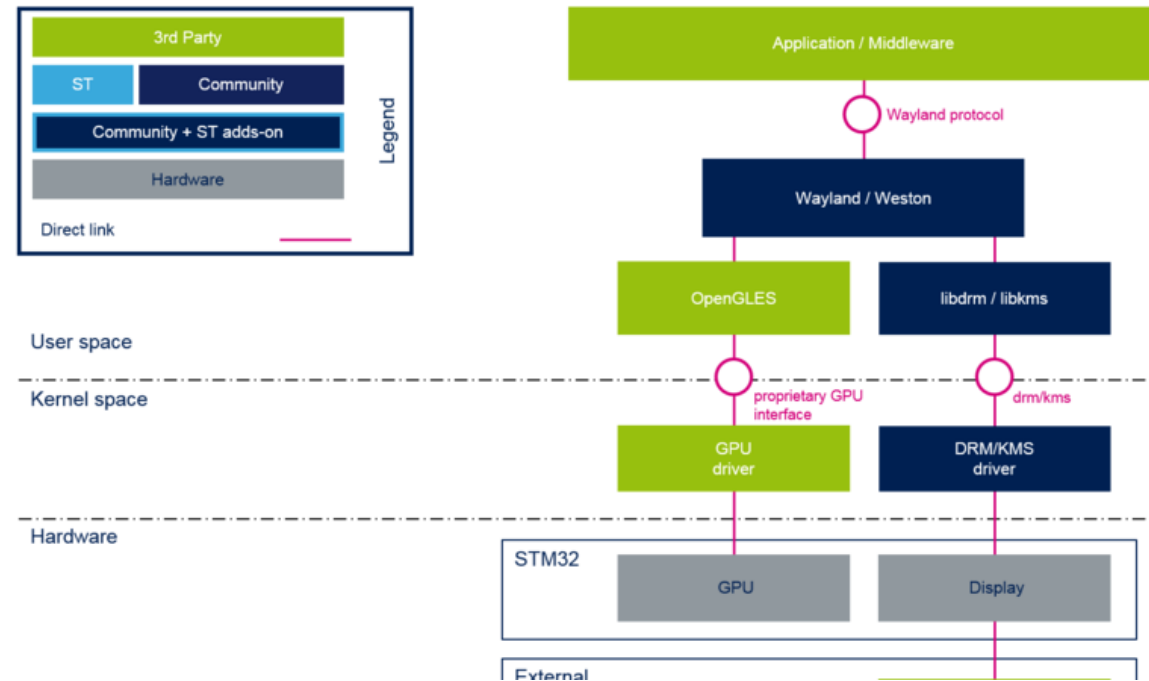
```





graph TD
    subgraph User_Space [User space]
        A[Application / Middleware] -- "Application framework protocol" --> B[Application Framework  
(Android, Wayland, Qt...)]
        B --> C[OpenGL]
        B --> D[libdrm / libkms]
    end
    subgraph Kernel_Space [Kernel space]
        C -- "proprietary GPU interface" --> E[GPU driver]
        D -- "/dev/dri/cardX" --> F[DRM/KMS framework & drivers]
    end
    subgraph Hardware [Hardware]
        E --> G[GPU]
        F --> H[Display internal peripherals]
    end
    subgraph External [External]
        H --> I[External]
    end
    
```

- Visual
 - > CEC
 - > DRM KMS
 - > GPU
 - > GStreamer
 - > V4L2
- Wayland Weston
 - How to calibrate the touchscreen
 - How to configure the Weston ...
 - How to configure Weston scre...
 - How to debug Weston
 - How to display an image usin...
 - How to make a screen captur...
 - How to make a video capture ...
 - How to stop and start Weston
 - Wayland Weston overview**
 - Wayland Weston troubleshooti...
 - Weston keyboard shortcuts

2 System overview

Wayland / Weston overview











[Welcome](#)
[Reading tips](#)
[Getting started](#)
[Development zone](#)
[Legal notice](#)

- Tools
 - Development tools
 - Trace and debug tools**
 - Tracing tools
 - Linux tracing tools
 - OP-TEE - How to debug
 - TF-A - How to debug
 - U-Boot - How to debug
 - Monitoring tools
 - Debugging tools
 - HW probes
 - Linux debugging tools
 - GDB
 - GDB commands
 - Gdbgui
 - OP-TEE - How to debug
 - STM32-CoPro-MPU plugin for S...
 - TF-A - How to debug


Subcategories

This category has the following 3 subcategories, out of 3 total.

- 
 - Tracing tools (1 C, 3 P)
- 
 - Monitoring tools (1 C)
- 
 - Debugging tools (2 C, 7 P)

Pages in category "Trace and debug tools"

The following 4 pages are in this category, out of 4 total.

- 
 - STM32MP1 Platform trace and debug environment overview



Welcome

Reading tips

Getting started

Development zone

Legal notice

🏠 Main page

📖 Glossary

> Release notes

> Getting started with ST boards

> STM32 MPU microprocessor devices

> Embedded software

✓ Tools

> Development tools

✓ Trace and debug tools

> Tracing tools

> Monitoring tools

> Debugging tools

STM32MP1 Platform trace and debug ...

STM32MP1 Platform trace and debug ...

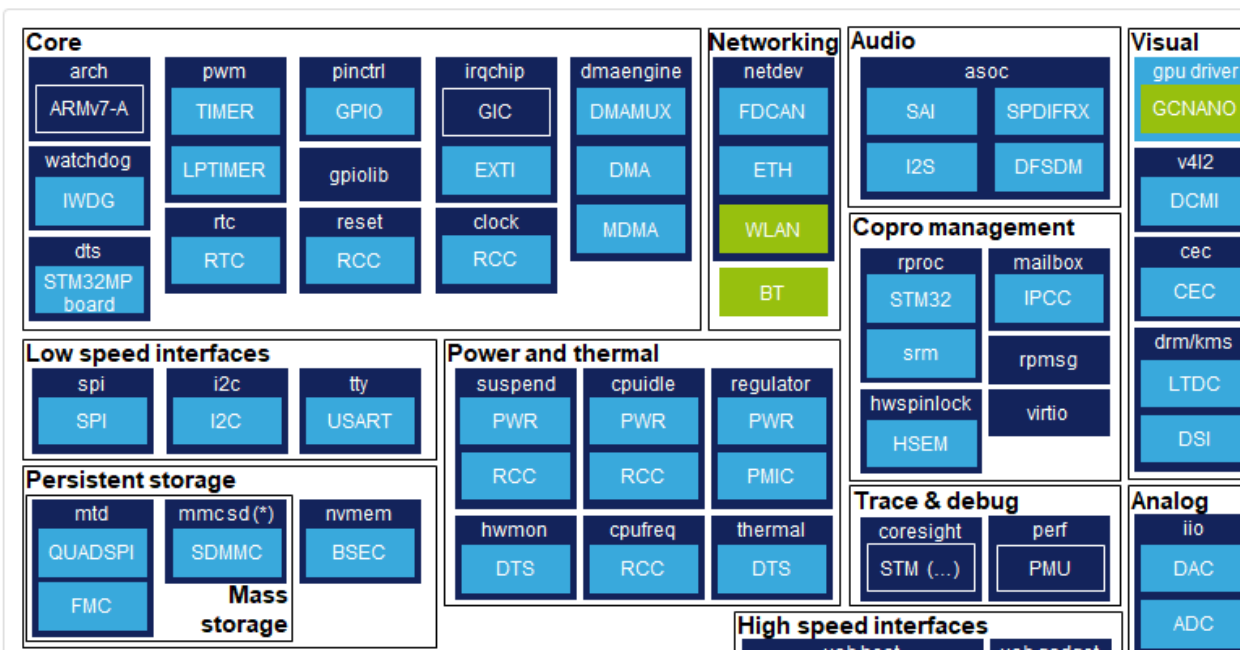
Linux tracing, monitoring and debugging

Trace and debug scenario - UART issue

> How to

4 Trace and debug overview per Linux software frameworks [\[edit\]](#)

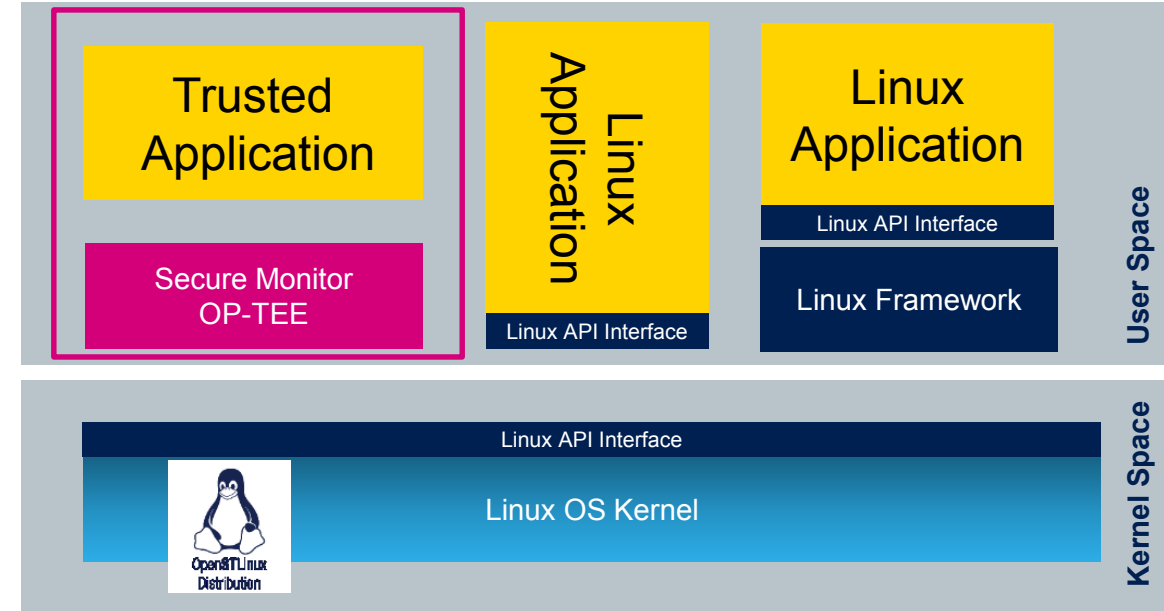
The picture below allows accessing to different Linux software frameworks which provide specific trace and debug information in their *"How to trace and debug the framework"* dedicated chapter.

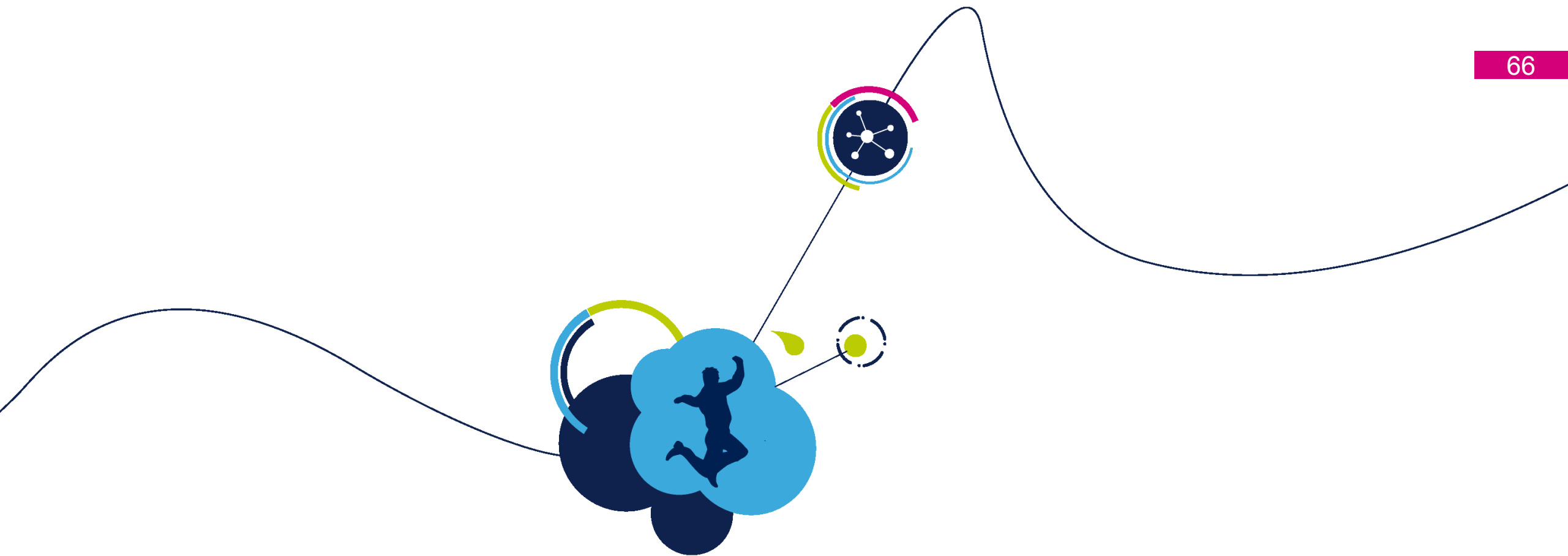


Wrap-up ST Software Offering: Easy&Fast

65

- Making Easy the Code Generation
 - MP1 Device Tree Generation
 - Generate HAL peripheral initialisation for Cortex-M4
 - Distribution package to generate final image
- OpenSTLinux distribution fully mainlined
 - Ensuring Software Quality
 - Scalable implementation
 - Removing Security Breaches
- Linux Community API compliancy
 - Easy changes from Linux Framework
 - Yocto bringing access to different environments to ease the build
- OP-TEE fully adapted to v7 ARM instruction set





Thanks