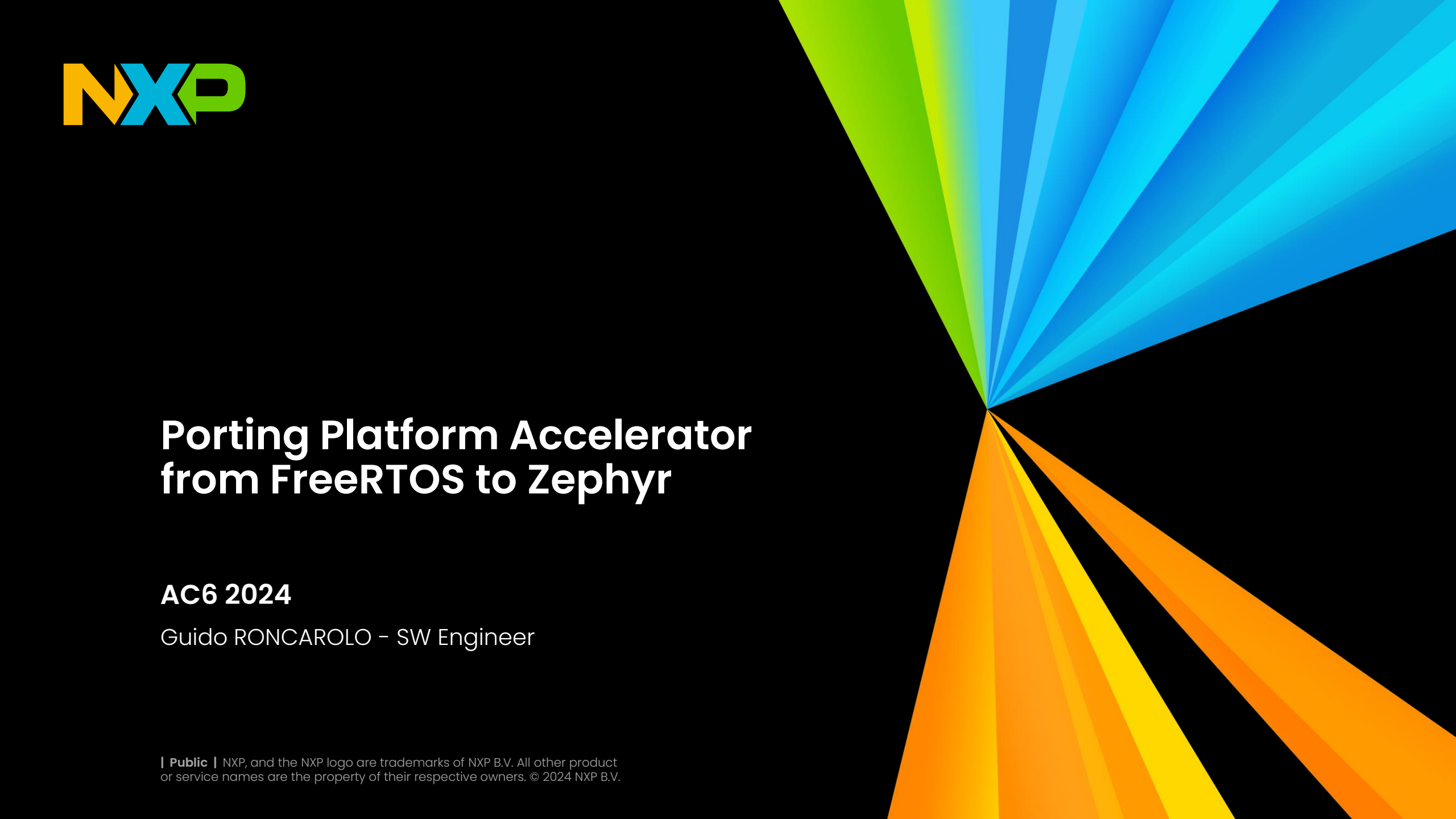


Porting Platform Accelerator from FreeRTOS to Zephyr

AC6 2024

Guido RONCAROLO - SW Engineer

About Me

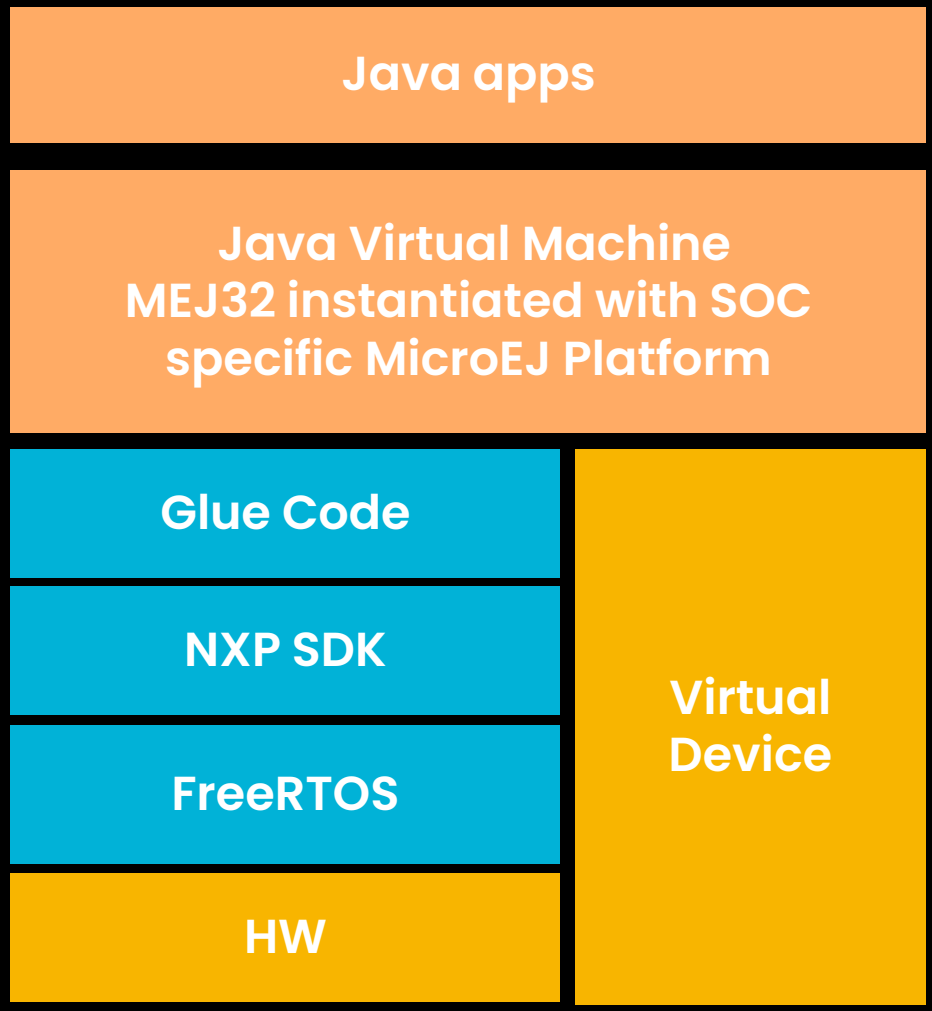
- Guido RONCAROLO
- Software engineer at NXP for 10 years
- 20 years of experience in embedded development
  - Telecommunications
  - Kernel
  - Real time audio solutions
  - Virtualization

What is Platform Accelerator

- Virtualization solution developed with NXP partner MicroEJ
- [SCALABLE] Platform Accelerator is a standard, safe and secure container for embedded systems platform capable of running on many different processors including microcontrollers (MCU) and microprocessors (MPU).
- [PROMISCUOUS] Platform Accelerator acts as a software container that runs on any OS/RTOS commonly used in embedded systems (**Zephyr ,** FreeRTOS, Linux or even bare metal).
- [EASY TO PROGRAM] Platform Accelerator includes the small virtual processor MEJ32 (a 32-bit virtual core), along with a wide range of free libraries and specially developed NXP APIs.

The Virtual Execution Environment For Embedded Systems: The Architecture

- Virtualization of Hardware and Middleware
- Multi-app trusted execution model (sandboxes)
- Standard services such as networking, cryptography, file systems, UI, ...
- Java technology mastered by 20+ million developers
- Digital twin simulator
- Platform Accelerator is available
  - as a simulated environment called Virtual Devices
  - as an embedded runtime for physical processors



Platform Accelerator Virtual Device

- **Digital twin of future product.**
- It allows to design embedded software applications on comfortable PC/desktop, using a true simulator while the actual electronics is being designed.
- The Virtual Device simulates the code execution, along with all peripherals such as displays, connectivity, sensors and any specific hardware feature of your system.

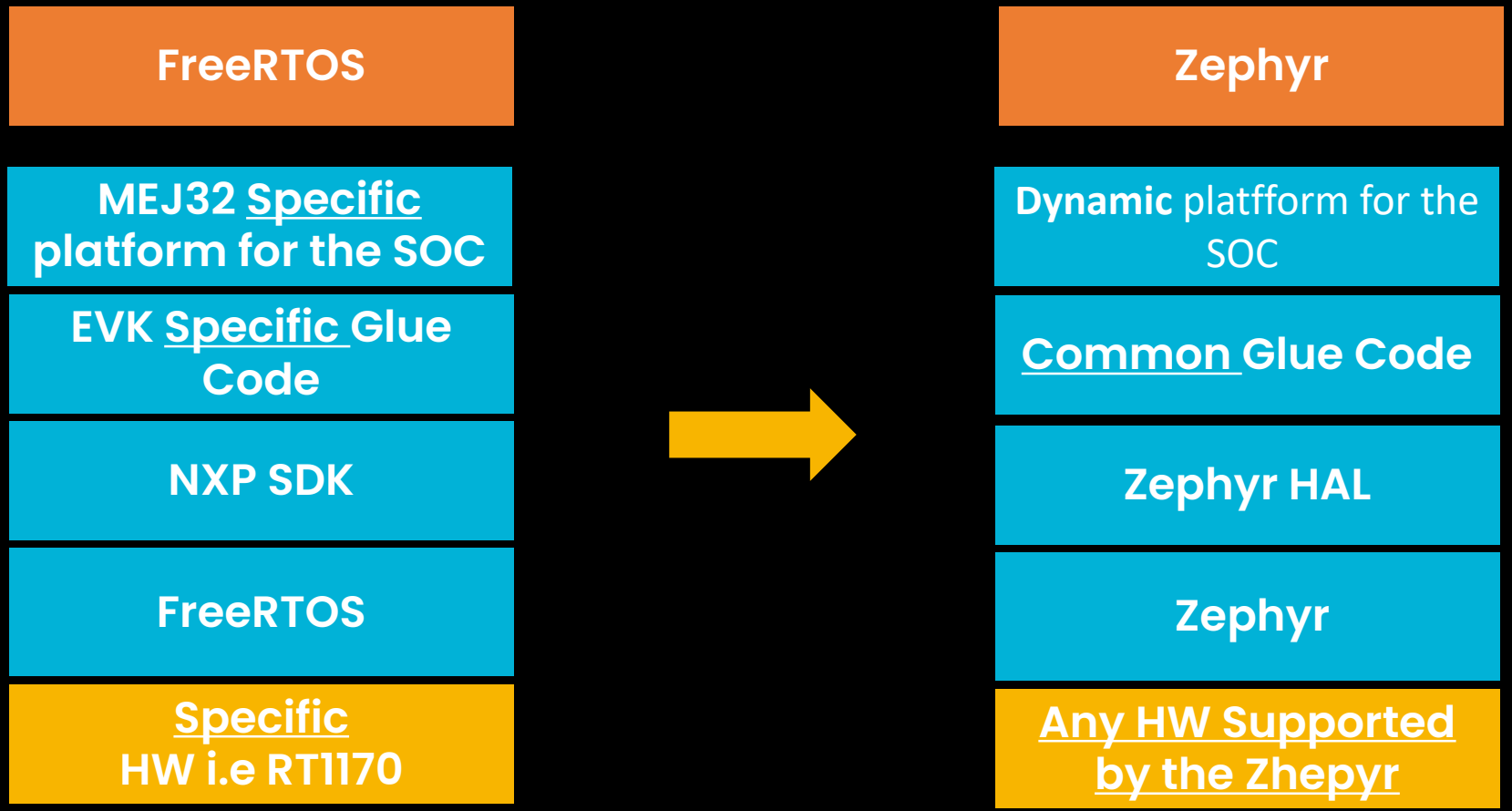
Platform Accelerator Embedded Runtime

- Embedded runtime
- Applications developed on the Virtual Device execute the same without any change on the real device: write once, run and deploy anywhere.
- MEJ32 comes in various flavors, named **architectures**, virtualizing the SoC cores.

FreeRTOS + NXP SDK: port public releases on github

- A port of Platform Accelerator usually has these features:
  - Core: tasks, timers, serial port
  - File System
  - UI
  - Networking
  - SSL
  - SECURITY
- We ported Platform Accelerator to these HW EVKs
  - i.MXRT595 EVK with watch display panel
  - i.MXRT1170 EVK with 5.5” display panel
  - MCXN947 Freedom board with 3.5” display panel
- Each port contains a **substantial amount of duplicated code**
- Lack of common driver APIs makes it difficult to factorize code
- Porting the core part on a new platform takes, usually, a week

Zephyr advantages over FreeRTOS



Zephyr multi-architecture port

- Porting Platform Accelerator on Zephyr we realize we could greatly benefit from its **modularity and driver model**
- We strived to make Platform Accelerator port on Zephyr **agnostic of the platform it runs ON**
- Our port can now run on all (**NXP**) Hardware supported by Zephyr for which we have a MicroEJ Platform available
  - M4
  - M33
  - M7
  - Fusion1
- Supporting platform is extremely straightforward
- CMake uses Kconfig defined symbols to
  - Compile the correct platform for the SOC chosen
  - Compile the best fitting Java application

Supporting a new SOC

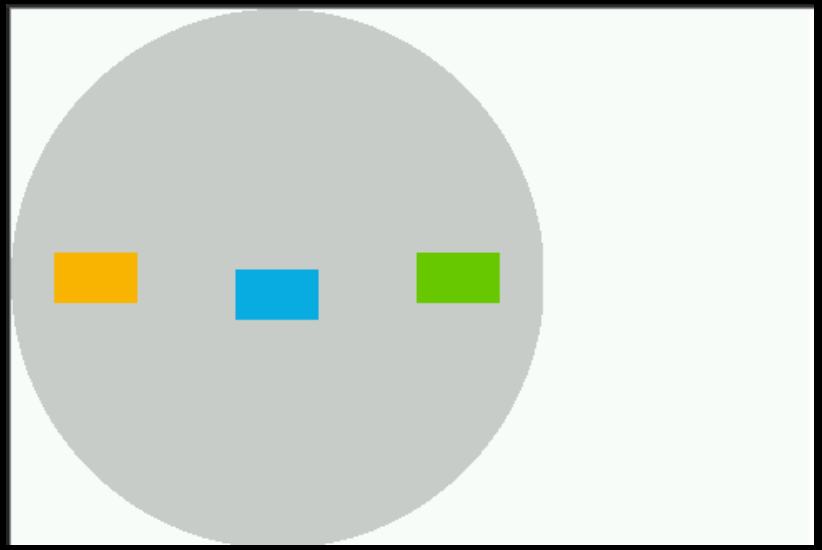
- Getting the demo running on a new EVKs can be as simple as just **compiling for the right target**
- At most we need to add a board.conf to configure specific features
  - If display is supported
  - How much heap memory
  - As an example:
    - zephyr/samples/nxpvee$ ls boards/ -1
      - frdm_mcxn947_mcxn947_cpu0.conf
      - imx8mp_evk_mimx8ml8_m7.conf
      - mimxrt1060_evkb.conf
      - mimxrt1064_evk_ai.conf
      - mimxrt1064_evk.conf
      - mimxrt1170_evk_mimxrt1176_cm7_ai.conf
      - mimxrt1170_evk_mimxrt1176_cm7.conf
      - mimxrt595_evk_mimxrt595s_cm33_ai.conf
      - mimxrt595_evk_mimxrt595s_cm33.conf
- We ported Platform Accelerator on SOCs/EVKs that we never had on FreeRTOS in a matter of **minutes not weeks**
- We gain **enormously** in term of time and ease of maintenance
- On FreeRTOS the glue code is dependent and customized on the EVK used
- On Zephyr the glue code leverages Zephyr HAL so it is common for all

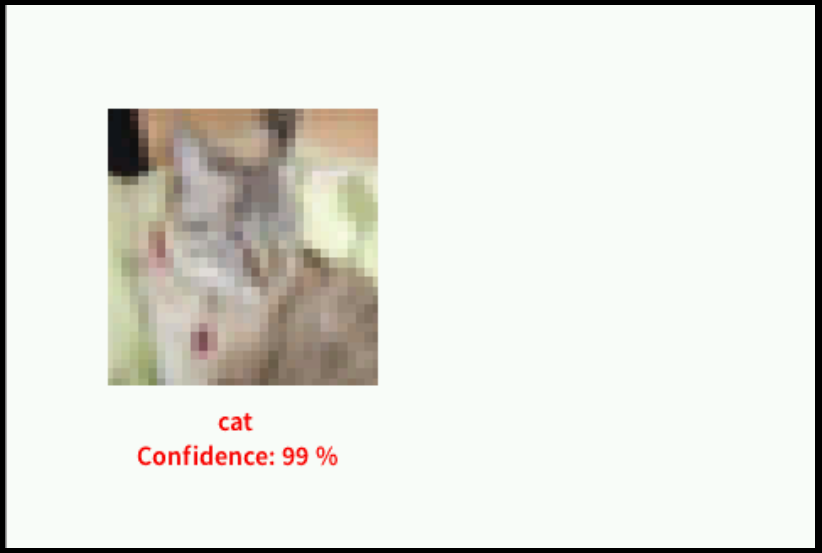
Zephyr Port Feature as of today

- These features are available
  - Basic enablement (core)
  - Display enablement
  - **AI demo using TensorFlow Lite Micro**
  - Simulation (digital twin)
  - Validation suite
  - Flashing (jlink, cmsis)
  - Debugging (jlink, cmsis)
  - MicroEJ IDE works too

Java apps available on Zephyr

- HelloWorld: most basic app just prints hello world on serial port for EVKs where no display is available
- SimpleGFX: display demo
- AiSample: AI demo using TensorFlow with Cifar based model and display to show image slide with best matching label



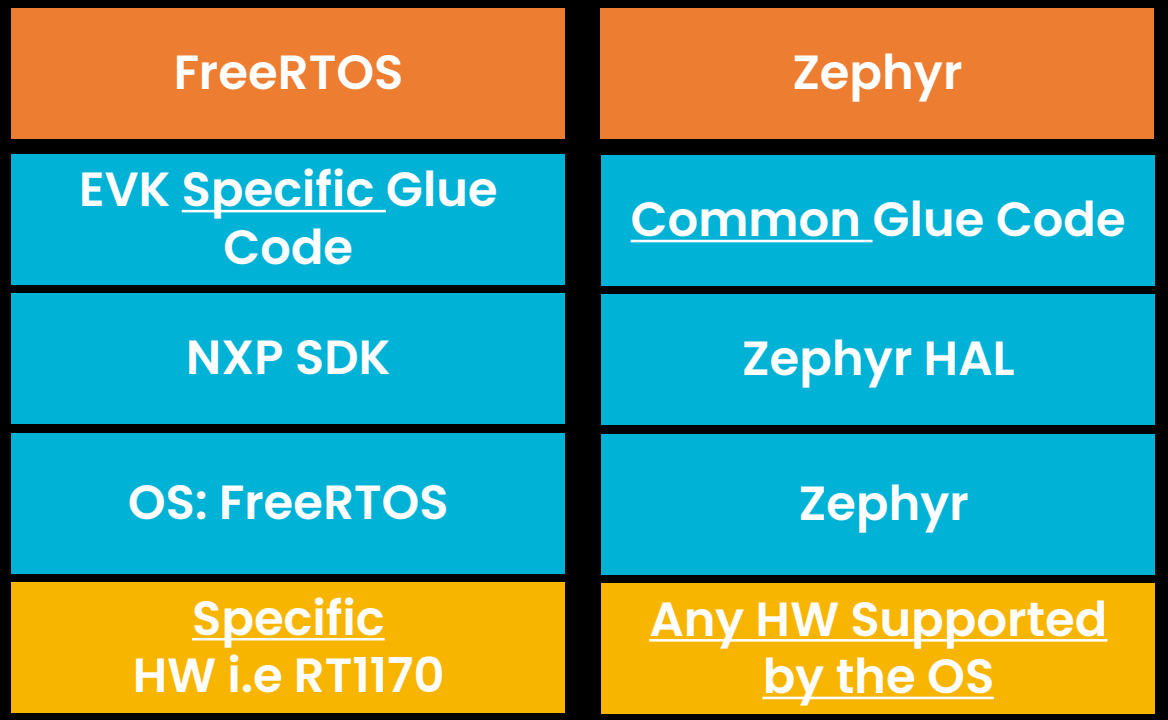


Zephyr and shields

- Thanks to the concept of shield, we can test different displays on boards where the connectivity allows it
- For example:
  - west -v build -b mimxrt1170_evk/mimxrt1176/cm7 zephyr/samples/nxpvee/ -DSHIELD=**rk055hdmipi4ma0**
  - west -v build -b mimxrt1170_evk/mimxrt1176/cm7 zephyr/samples/nxpvee/ -DSHIELD=**g1120b0mipi**
- **Without changing any code, just recompiling**

Wrap up

- Zephyr is Modular and Scalable
- Zephyr supports:
  - KConfig as in Linux
  - DTS for HW description
  - A native networking stack
- Zephyr has a driver model
  - Writing an application that uses a certain HW (i.e. a display) will work on all EVKs that support that HW
  - One can run an application on new SOCs/EVKs as soon as the support for it is integrated
- Fast porting on different platforms



List of NXP supported HW as of today

- imx8mm_evk
- imx8mn_evk
- imx8mp_evk
- imx8mq_evk
- imx8qm_mek
- imx8qxp_mek
- imx8ulp_evk
- mimxrt1010_evk
- mimxrt1015_evk
- mimxrt1020_evk
- mimxrt1024_evk
- mimxrt1040_evk
- mimxrt1050_evk
- mimxrt1060_evk
- mimxrt1060_evkb
- mimxrt1062_fmurt6
- mimxrt1064_evk
- mimxrt1160_evk
- mimxrt1170_evk
- mimxrt595_evk
- mimxrt685_evk
- frdm_mcxn947

Demo: Compile Sample Apps for MCXN947

- cd ~/fsl/microej/zephyr
- rm build -rf && west -v build -b frdm_mcxn947/mcxn947/cpu0 zephyr/samples/nxpvee -- -DSHIELD=lcd_par_s035_8080 -DJUSAGE=prod
- west flash
- rm build/ -rf && west -v build -b frdm_mcxn947/mcxn947/cpu0 zephyr/samples/nxpvee -- -DSHIELD=lcd_par_s035_8080 -DJUSAGE=prod -DCONF_FILE=prj_ai.conf
- west flash

Github Links

- https://github.com/nxp-mcuxpresso/nxp-vee-imxrt1170-evk
- https://github.com/nxp-mcuxpresso/nxp-vee-imxrt595-evk
- https://github.com/nxp-mcuxpresso/nxp-vee-mcxn947-frdm