

SOUND OPEN FIRMWARE MEETS ZEPHYR

Daniel Baluta
Iuliana Prodan
SEPTEMBER 2021



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2020 NXP B.V.



CONTENT

What is Sound Open Firmware (SOF) ?

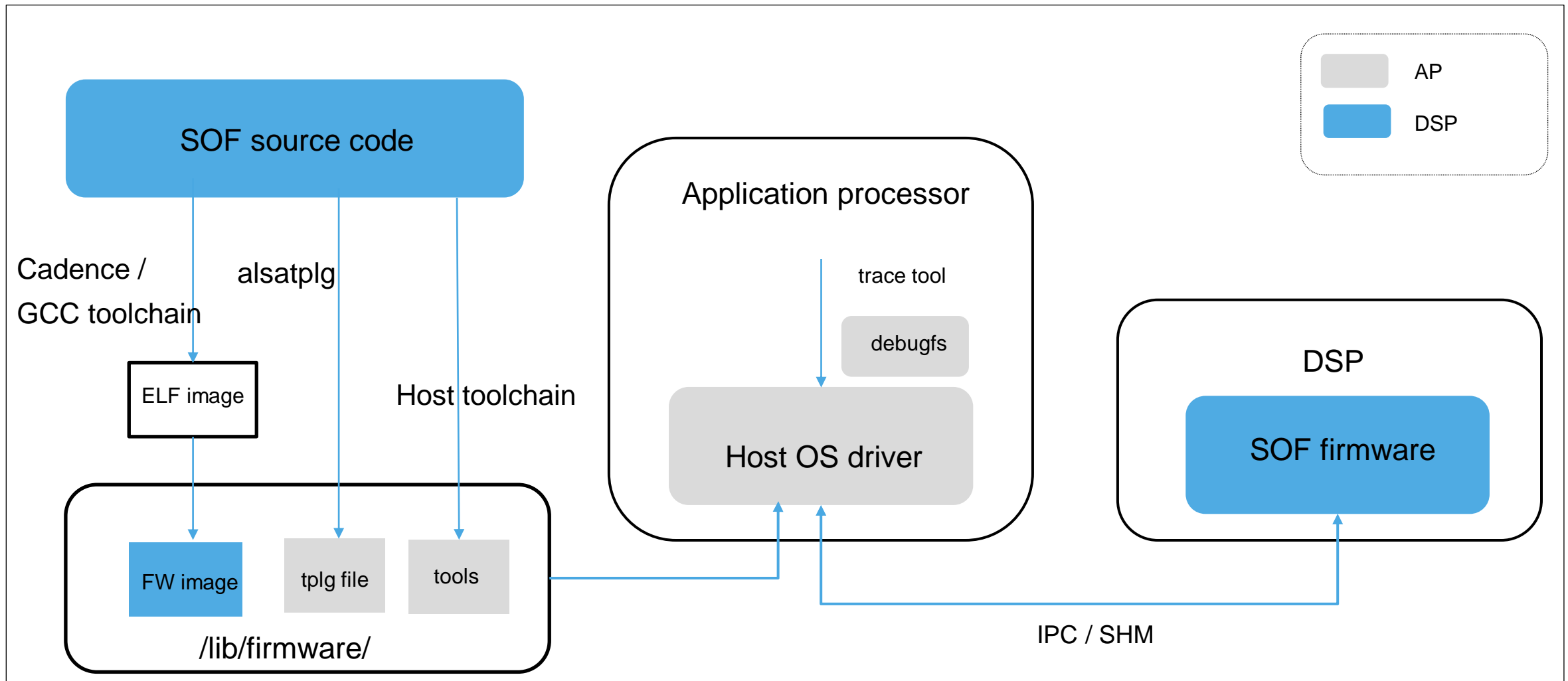
SOF with Zephyr

Changes needed for i.MX8

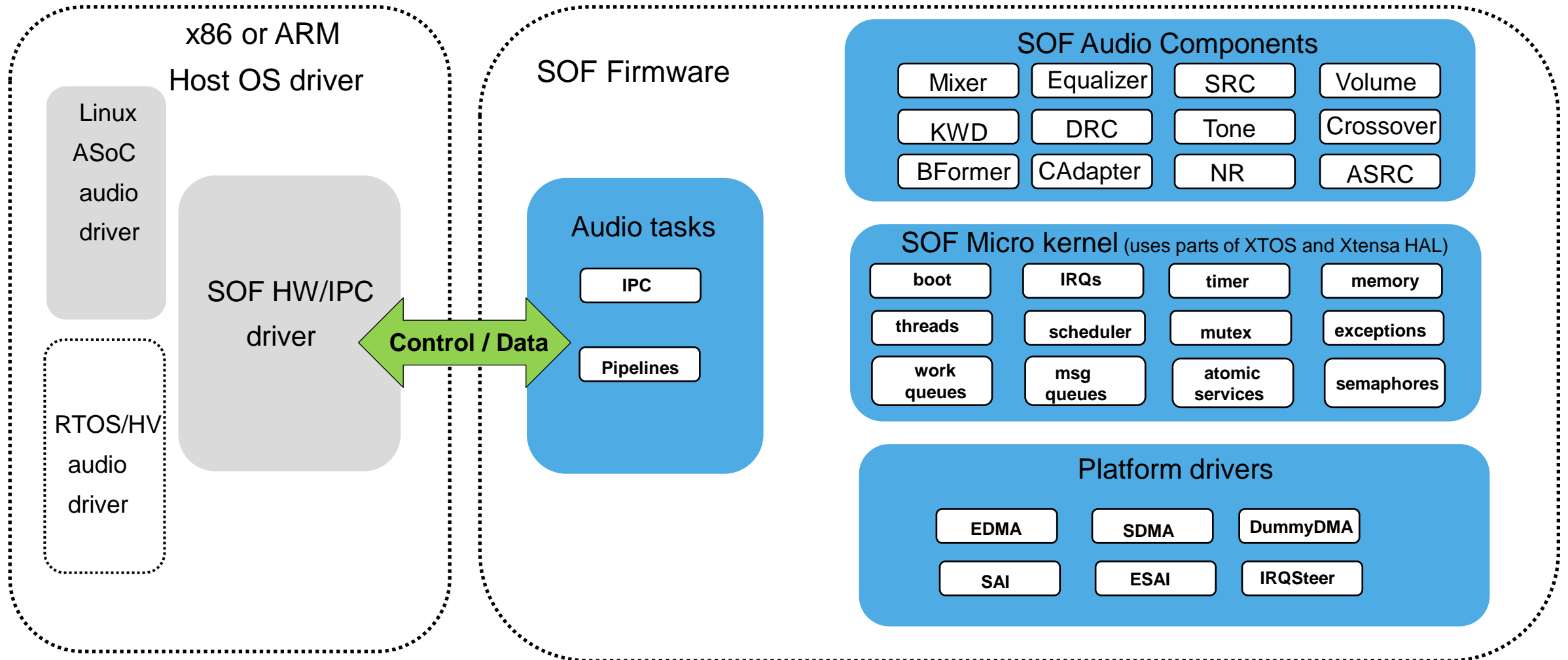
Challenges

Future work

WHAT IS SOUND OPEN FIRMWARE?



WHAT IS SOUND OPEN FIRMWARE?

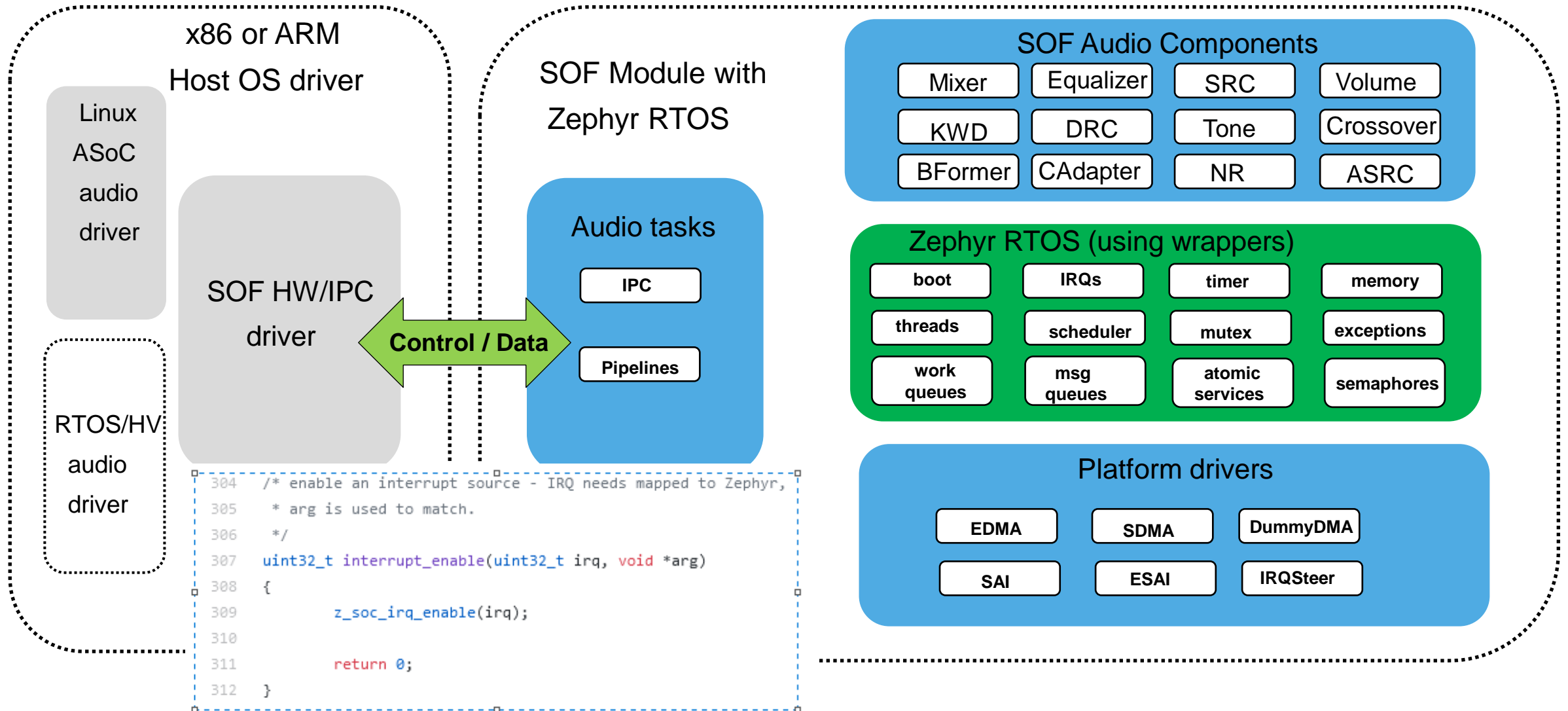


WHY ZEPHYR?

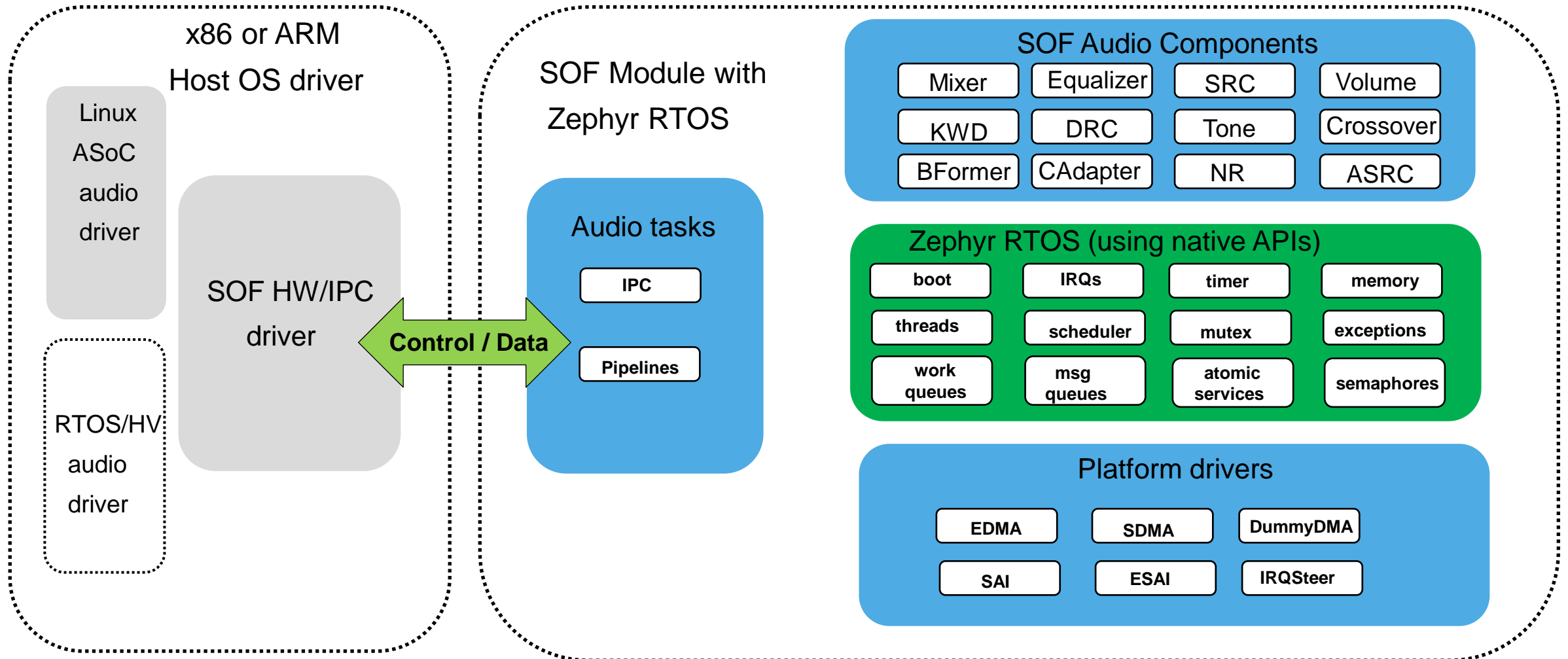


- ✓ open source
- ✓ vibrant community
- ✓ richer feature set
- ✓ multi-platform support
- ✓ multi-threaded execution
- ✓ built with safety and security in mind

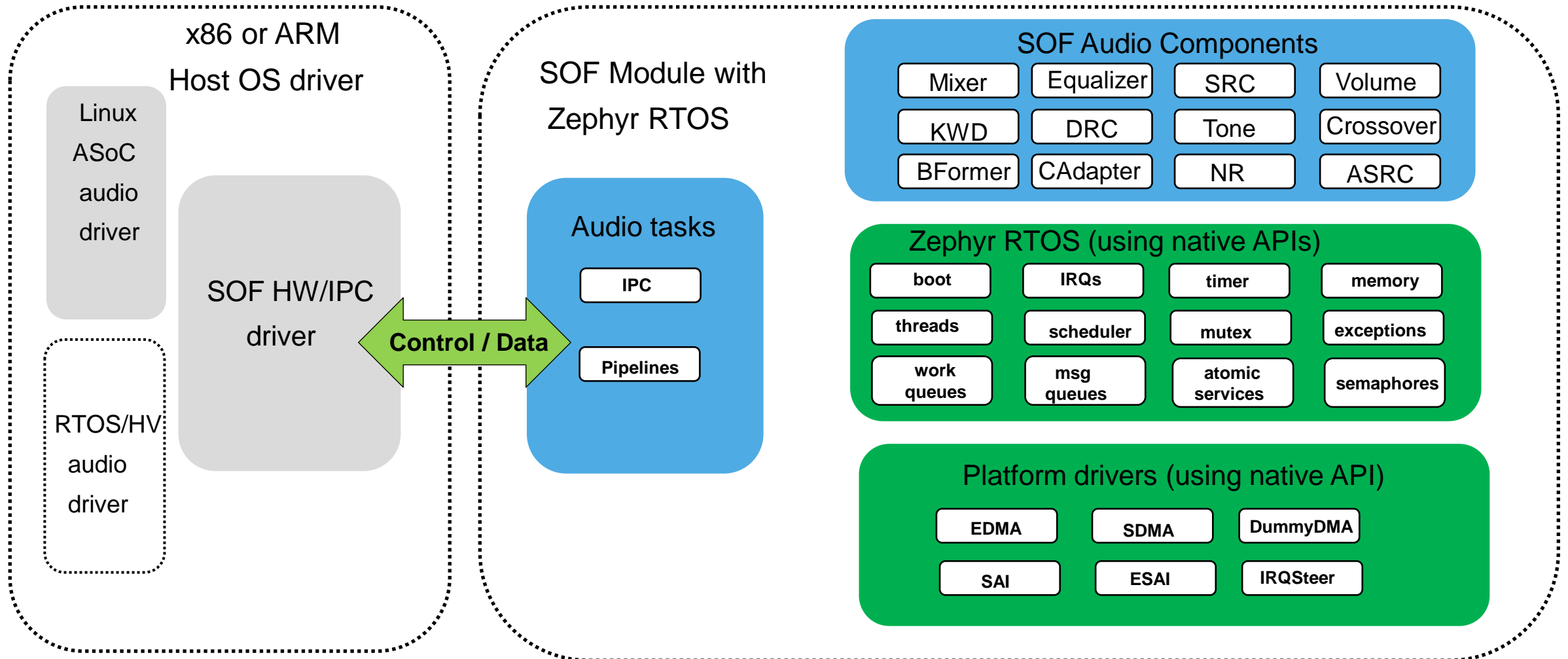
SOF WITH ZEPHYR



SOF WITH ZEPHYR

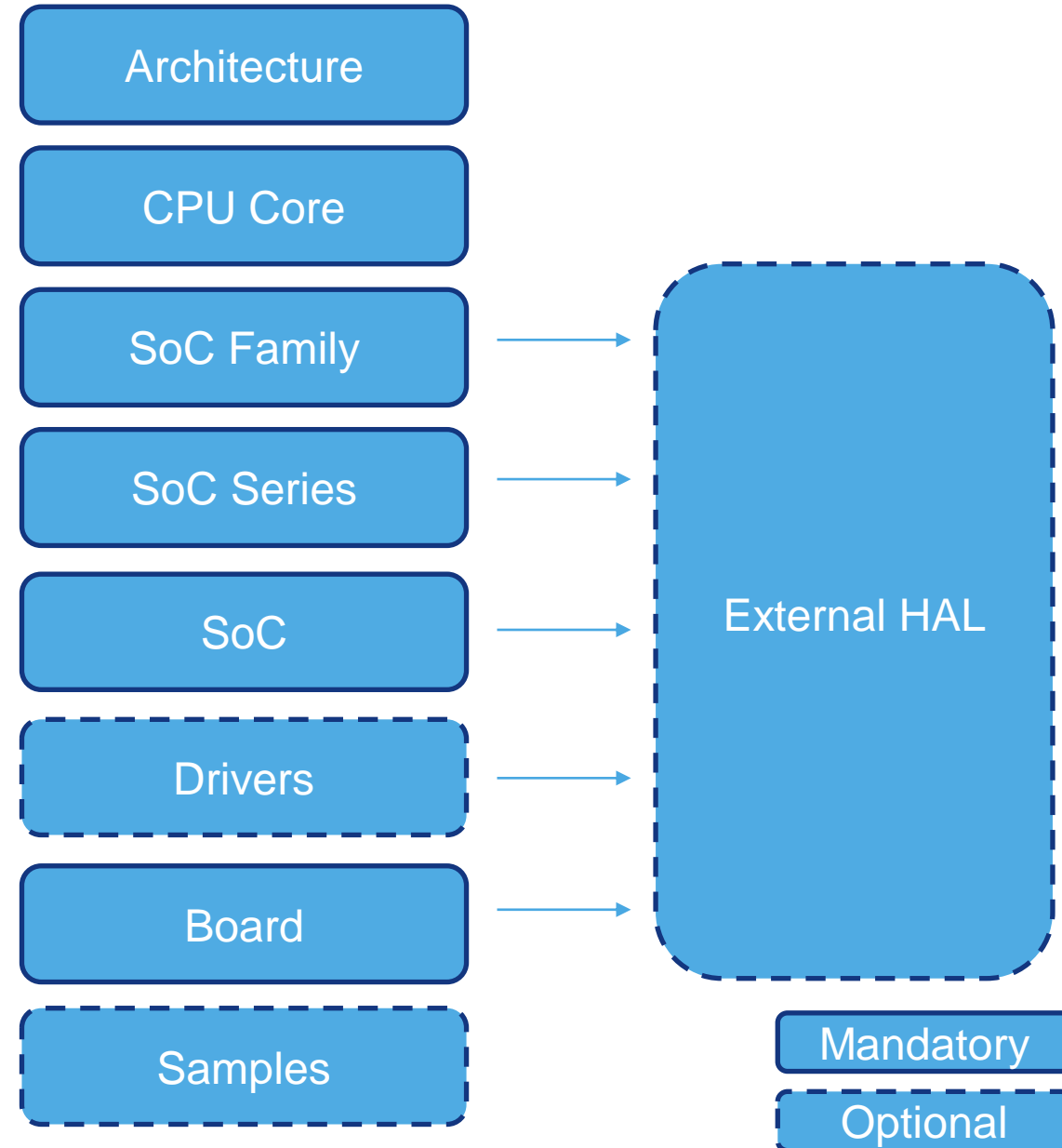


SOF WITH ZEPHYR

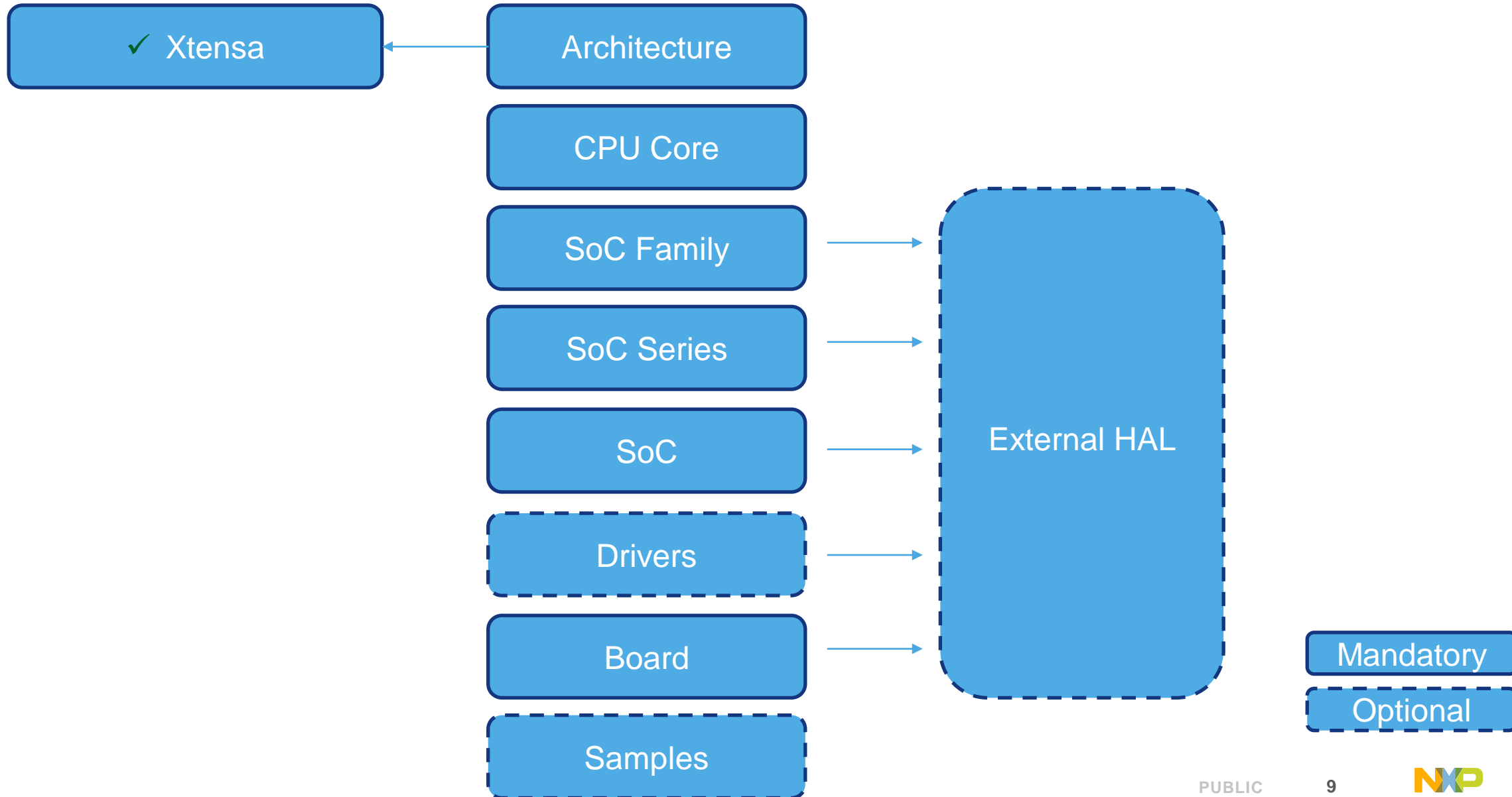


CHANGES NEEDED FOR I.MX8

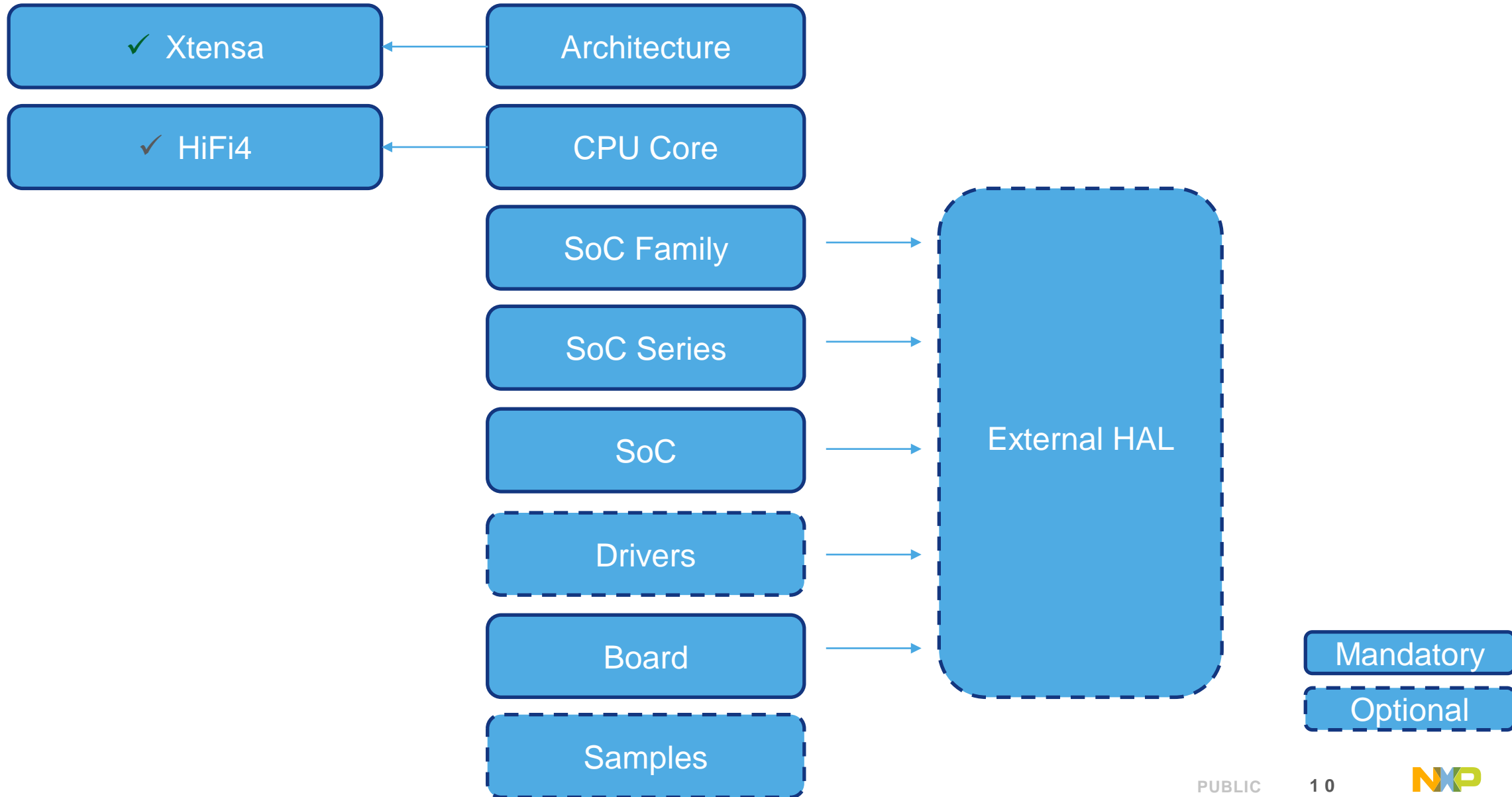
- Start from Zephyr documentation:
 - [Board Porting Guide](#)
- SOF is a module in Zephyr Project:
 - `zephyrproject/modules/audio/sof`
- DSP core
 - Tensilica HIFI4
- Build, test and fix



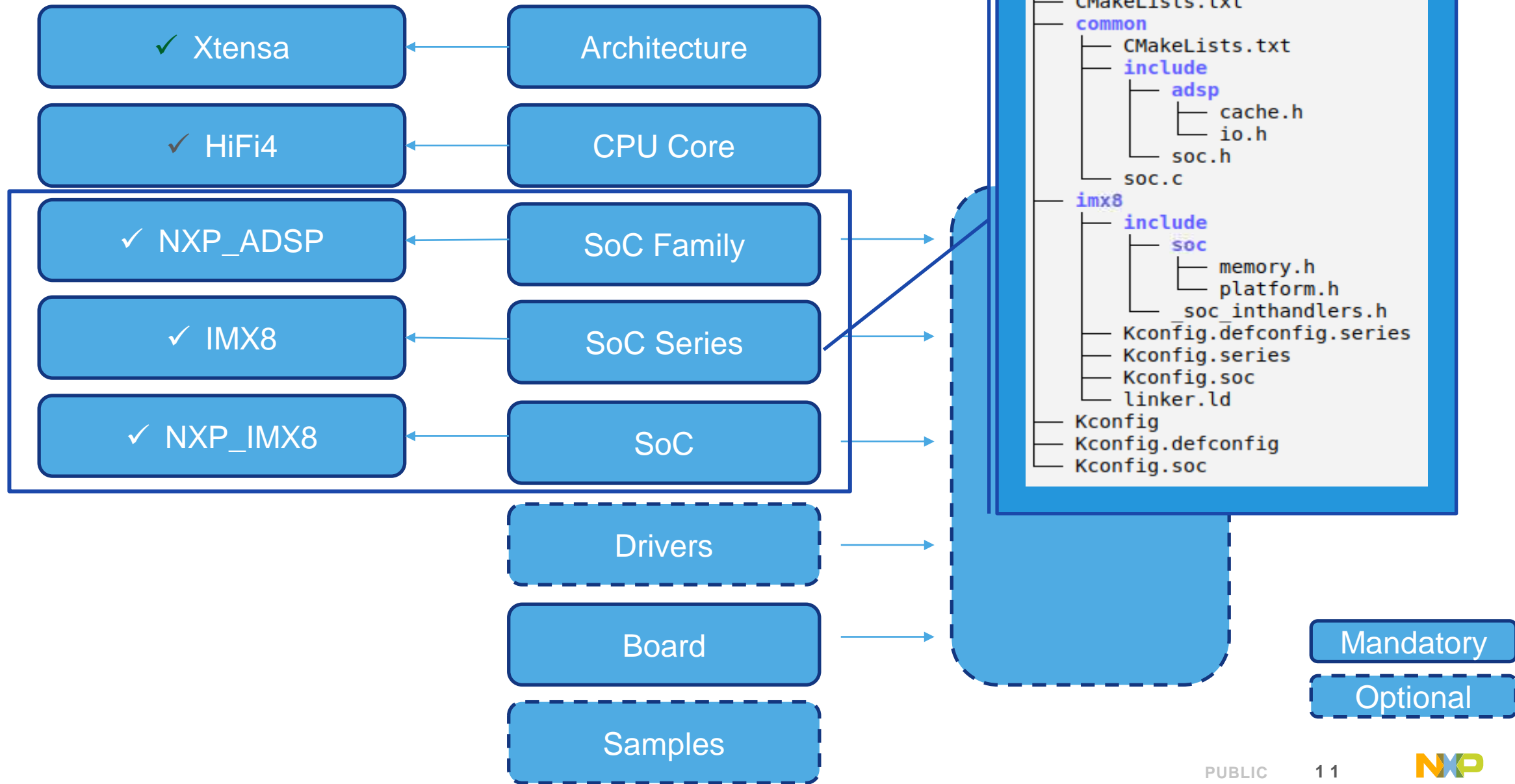
CHANGES NEEDED FOR I.MX8



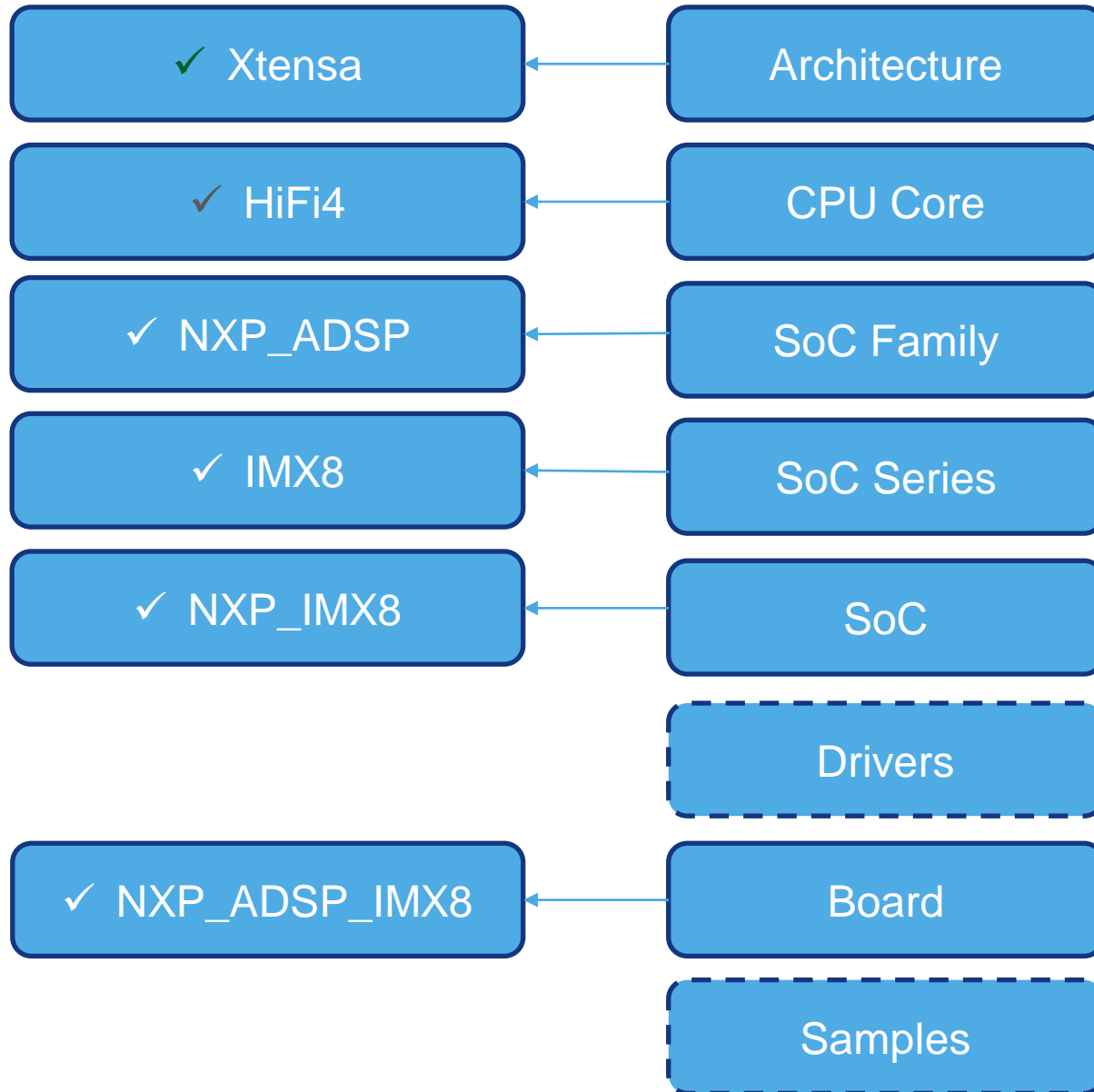
CHANGES NEEDED FOR I.MX8



CHANGES NEEDED FOR I.MX8



CHANGES NEEDED FOR I.MX8



```

/ {
    cpus {
        #address-cells = <1>;
        #size-cells = <0>;

        cpu0: cpu@0 {
            device_type = "cpu";
            compatible = "cdns,tensilica-xtensa-lx6";
            reg = <0>;
        };
    };

    sram0: memory@92400000 {
        device_type = "memory";
        compatible = "mmio-sram";
        reg = <0x92400000 DT_SIZE_K(512)>;
    };

    sram1: memory@92c00000 {
        device_type = "memory";
        compatible = "mmio-sram";
        reg = <0x92c00000 DT_SIZE_K(512)>;
    };
};

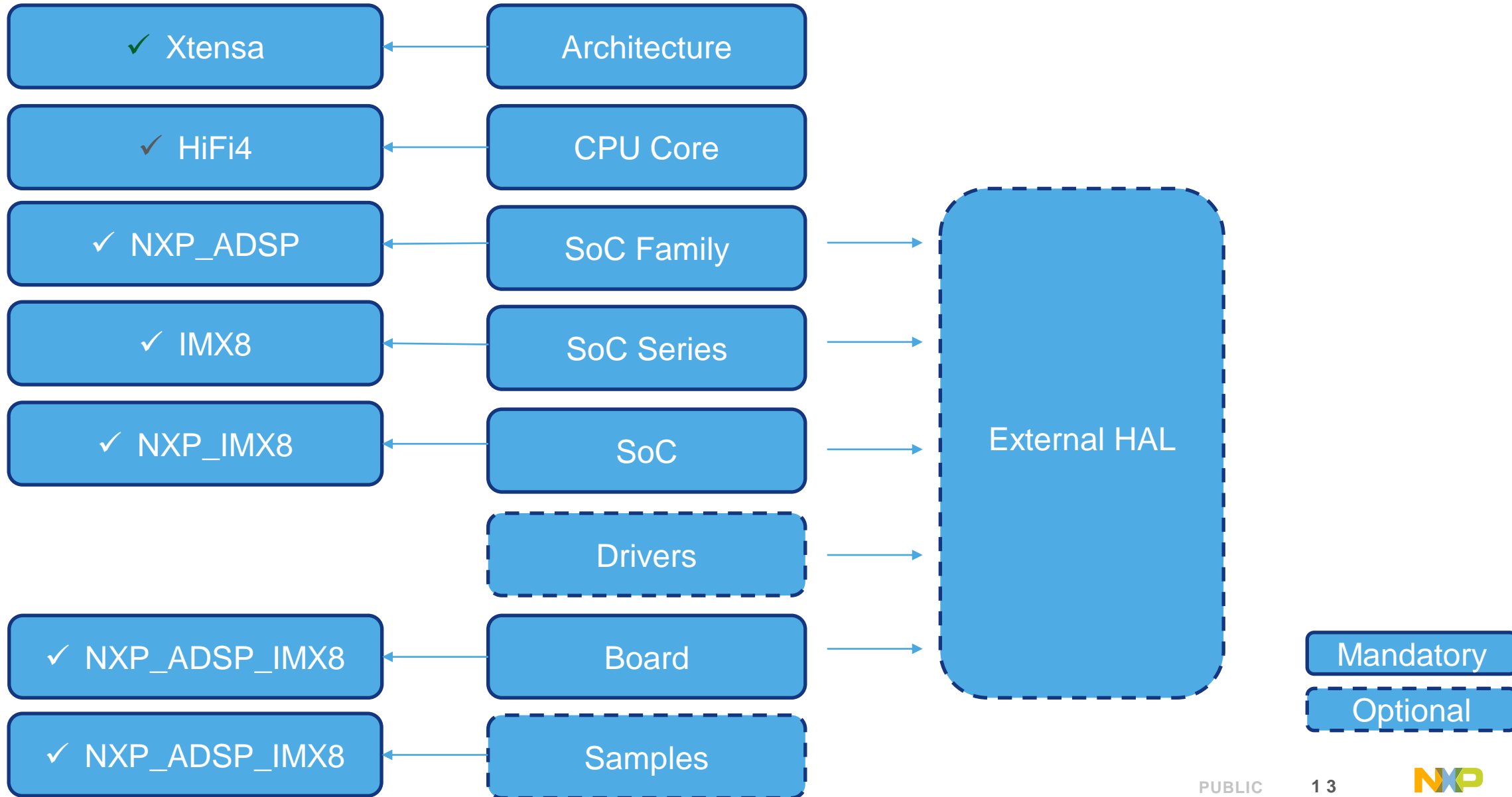
```

- boards/xtensa/nxp_adsp_imx8/
 - board.cmake
 - Kconfig.board
 - Kconfig.defconfig
 - nxp_adsp_imx8_defconfig
 - nxp_adsp_imx8.dts
 - nxp_adsp_imx8.yaml

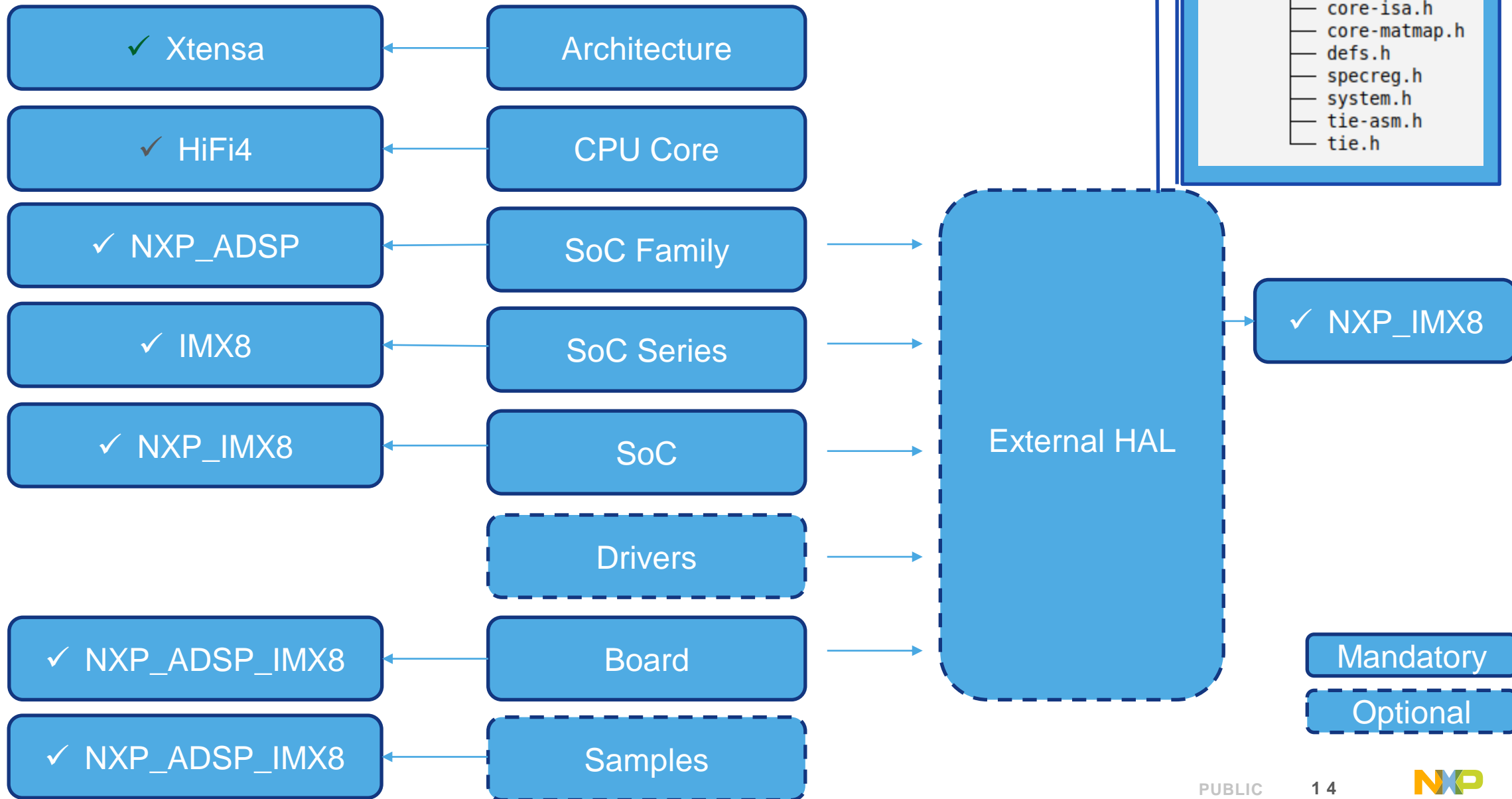
Mandatory

Optional

CHANGES NEEDED FOR I.MX8



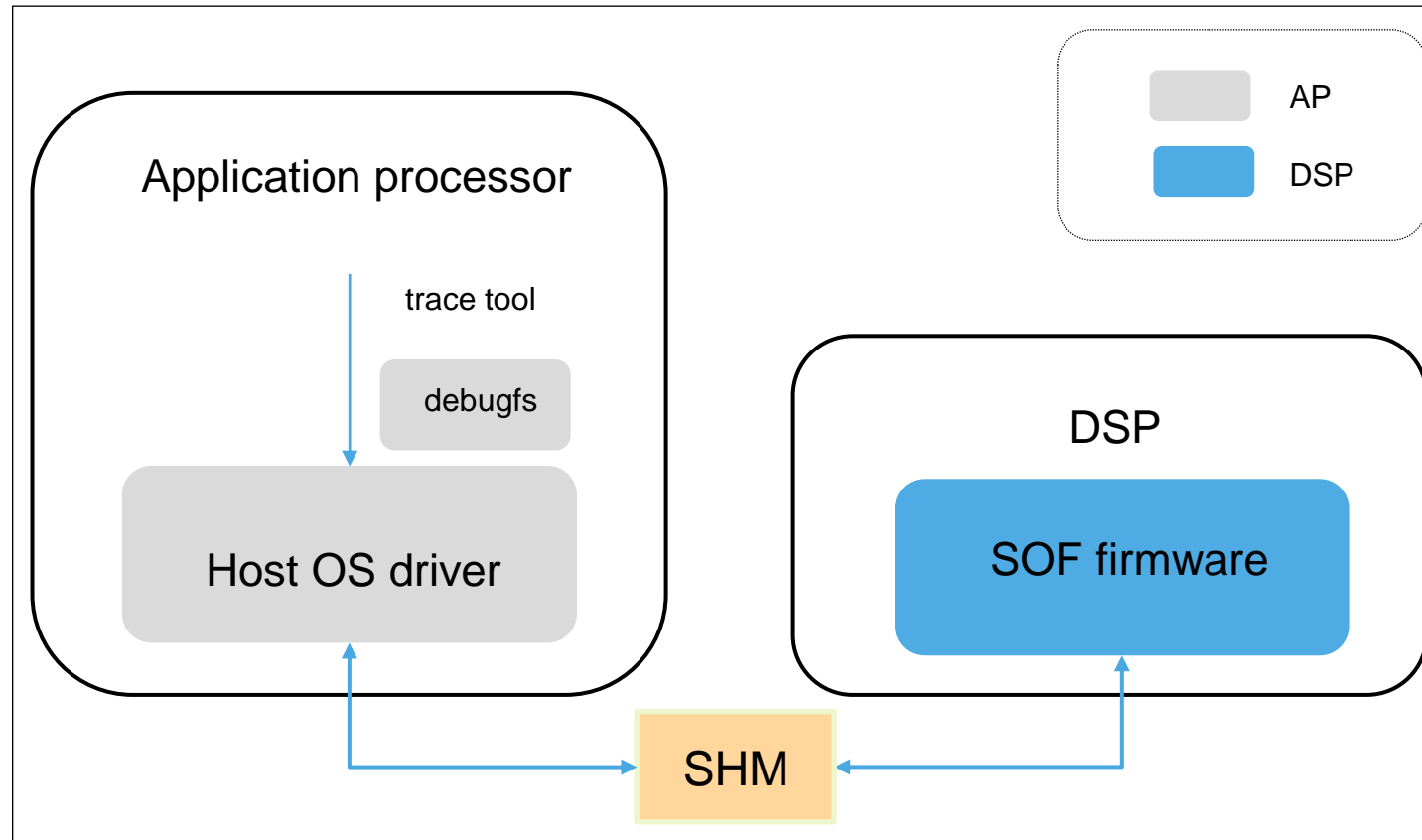
CHANGES NEEDED FOR I.MX8



CHALLENGES



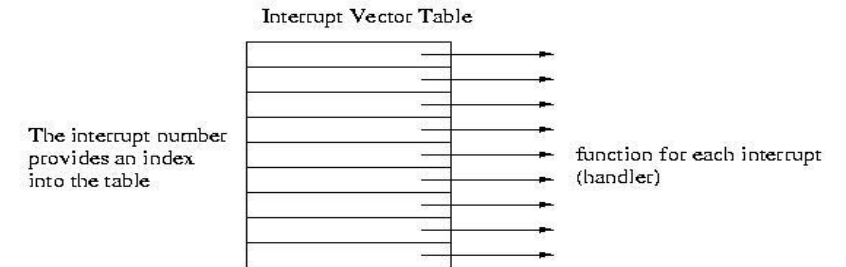
- Very hard to debug – no trace, no logger, no available debugger
 - ✓ Used global variables to set program state and read them later from Linux using `/dev/mem`



CHALLENGES

- Entry point was set incorrectly
 - ✓ Set it to `.ResetVector` as in SOF, default is `__start`
- The size constraint of the interrupt vector table entry, for i.MX, is less than usual
 - ✓ Added a small indirection to bypass this size constraint and moved the default handlers to the end of vector table
- Timer interrupt (from level 3) was generated endlessly
 - ✓ Update Exception mode of Program State Register to a higher value

```
ENTRY(CONFIG_KERNEL_ENTRY)
...
.text : ALIGN(4)
{
    _stext = .;
    _text_start = ABSOLUTE(.);
    KEEP (*(ResetVector.text))
    *(ResetVector.literal)
    *(entry.text)
    *(init.literal)
    KEEP(*(init))
}
```



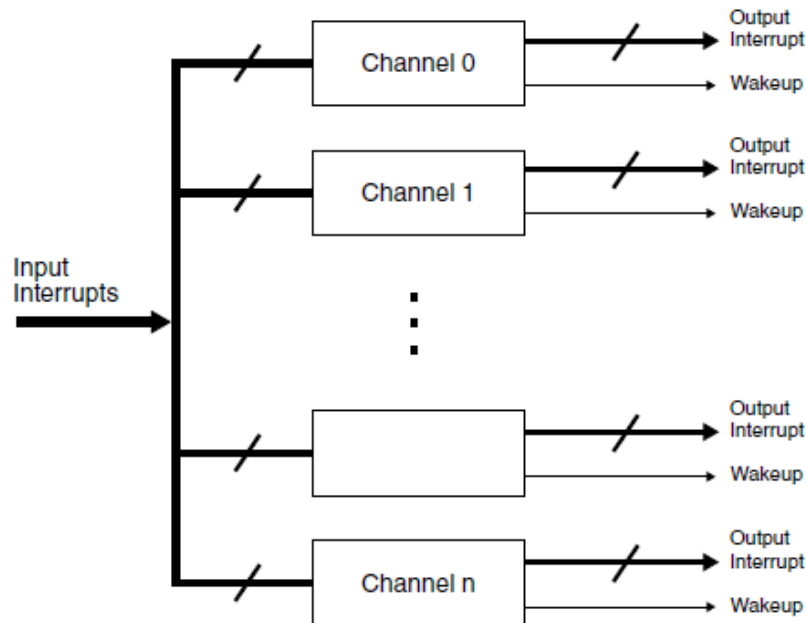
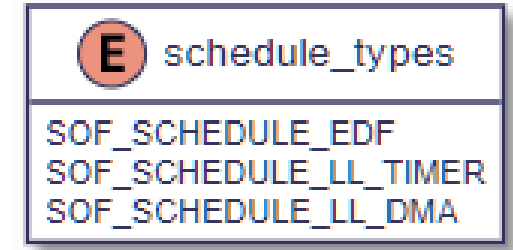
```
timer_interrupt_handler()
{
    get_spinlock();

    rearm_timer();

    release_spinlock();
}
```

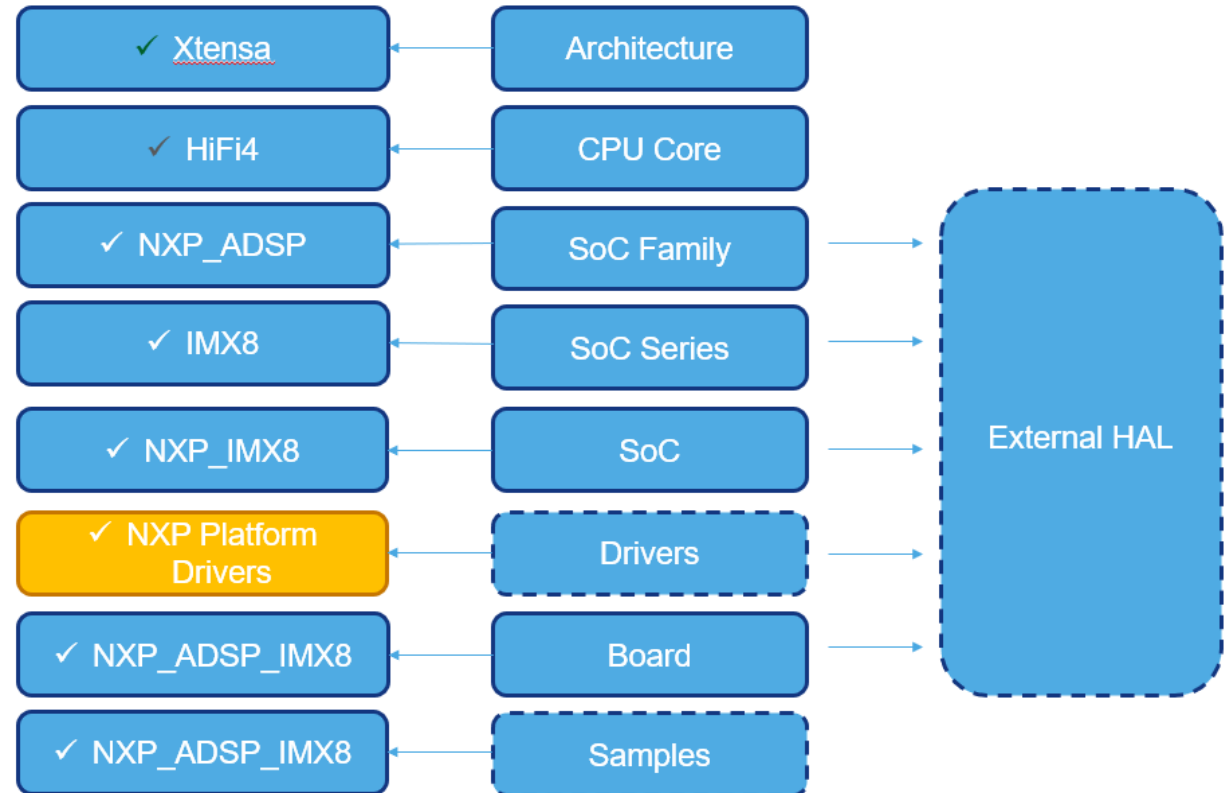
CHALLENGES

- Zephyr Low Latency scheduler (`ll_scheduler`) is limited to timer interrupt
 - ✓ Use `ll_scheduler` from SOF which supports also DMA interrupts
- The second level interrupt handling was not enabled in SOF with Zephyr
 - ✓ Activate cascade interrupt in SOF and Interrupt Request Steering (`IRQ_steer`) module for i.MX



FUTURE WORK

- Extend Zephyr Low Latency scheduler (`ll_scheduler`) to support also DMA interrupts
- Implement a driver for each NXP Platform Driver (EDMA, SAI, IRQ_Steer, etc.)



FUTURE WORK

- Common interface / wrapper to easily support multiple RTOSes
 - FreeRTOS, Zephyr, XTOS
- Support SOF on all i.MX board even if they don't have a DSP – support ARM core
 - run Zephyr on an ARM core and use Jailhouse

THANK YOU

Questions?



SECURE CONNECTIONS
FOR A SMARTER WORLD