

ENABLING SOUND OPEN FIRMWARE ON ARM

CORTEX®-A BASED DEVICES USING ZEPHYR RTOS

Daniel BALUTA

JUNE 2023



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. © 2023 NXP B.V.

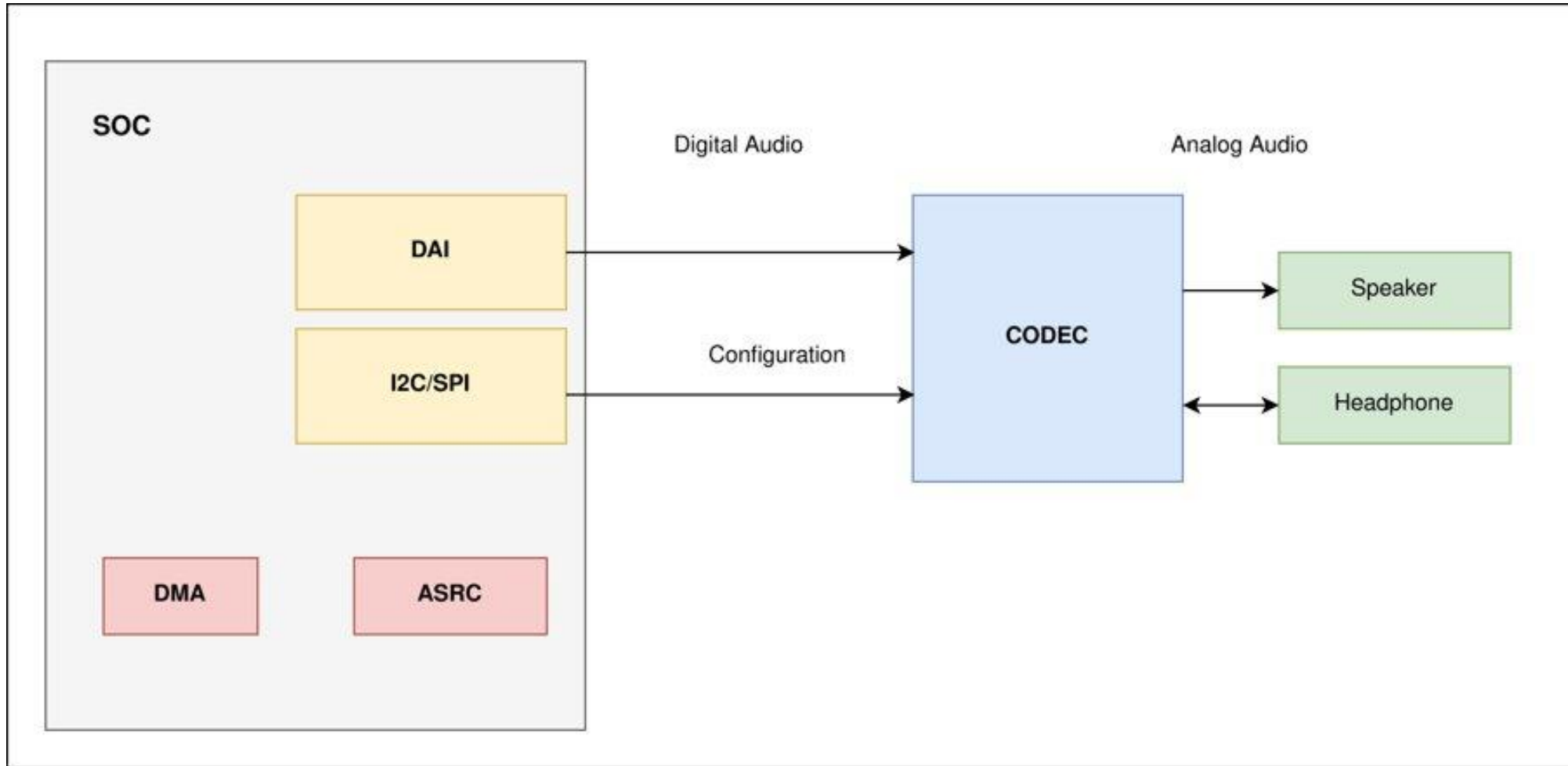




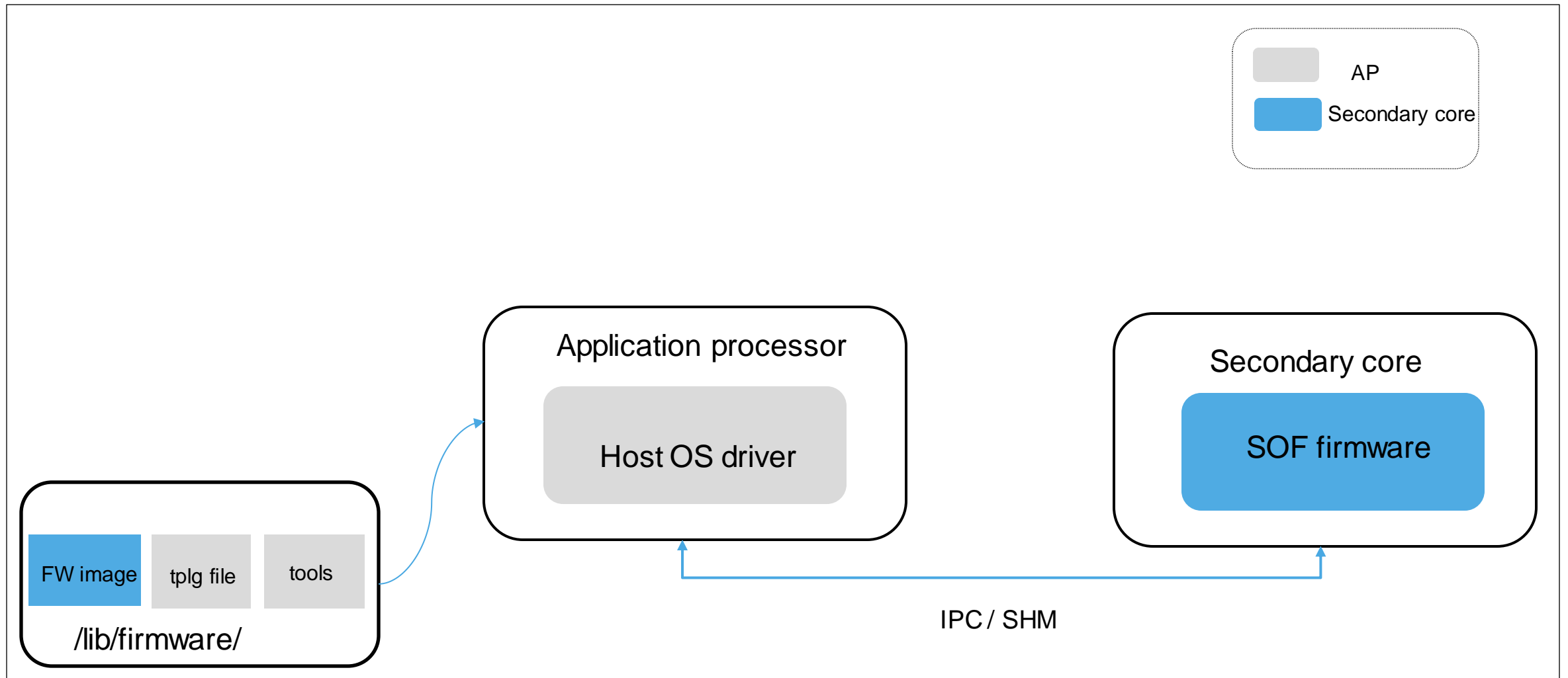
OVERVIEW

- Audio basics & solution overview
- Jailhouse support
- Linux driver support
- Zephyr support

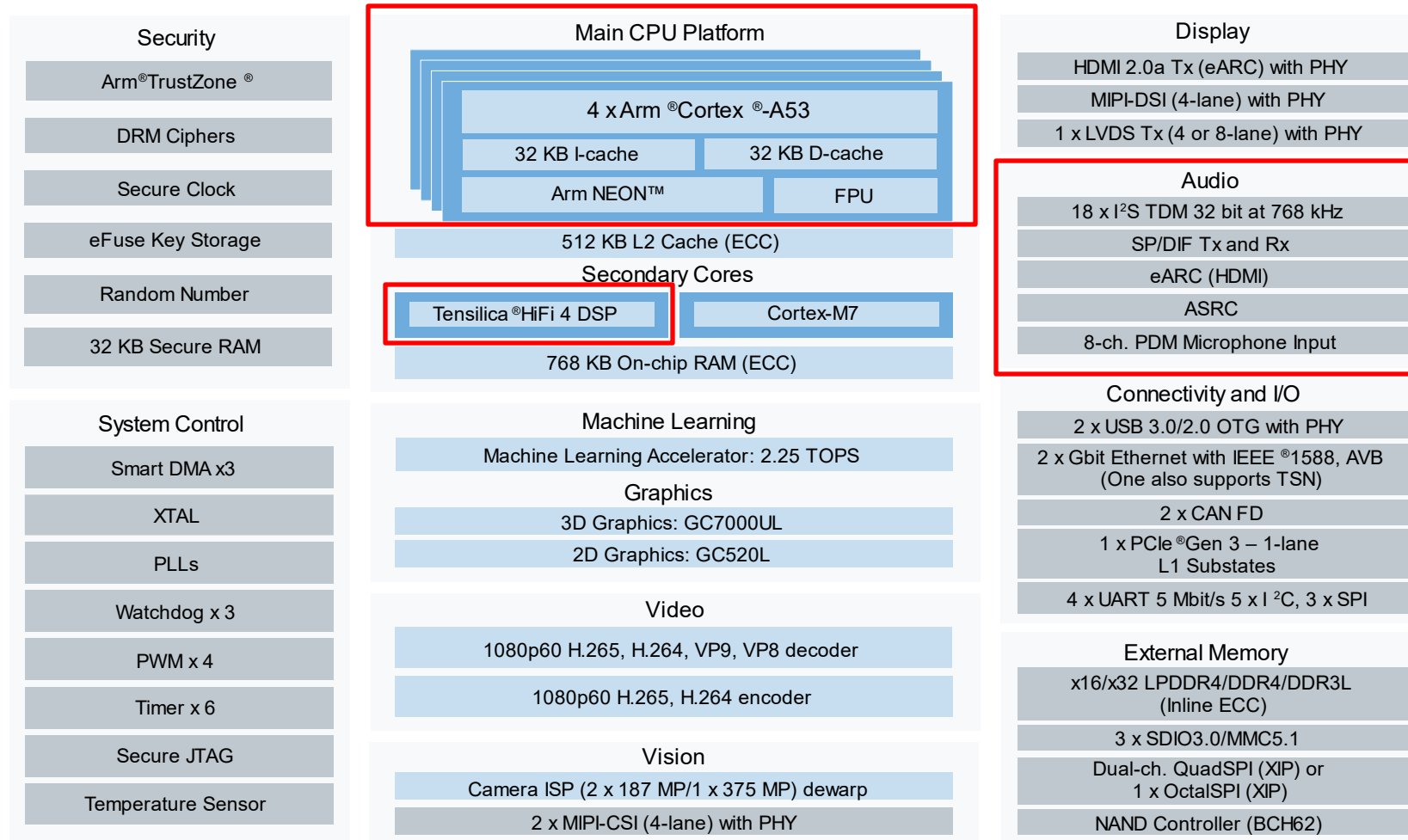
ANATOMY OF AN EMBEDDED AUDIO SYSTEM



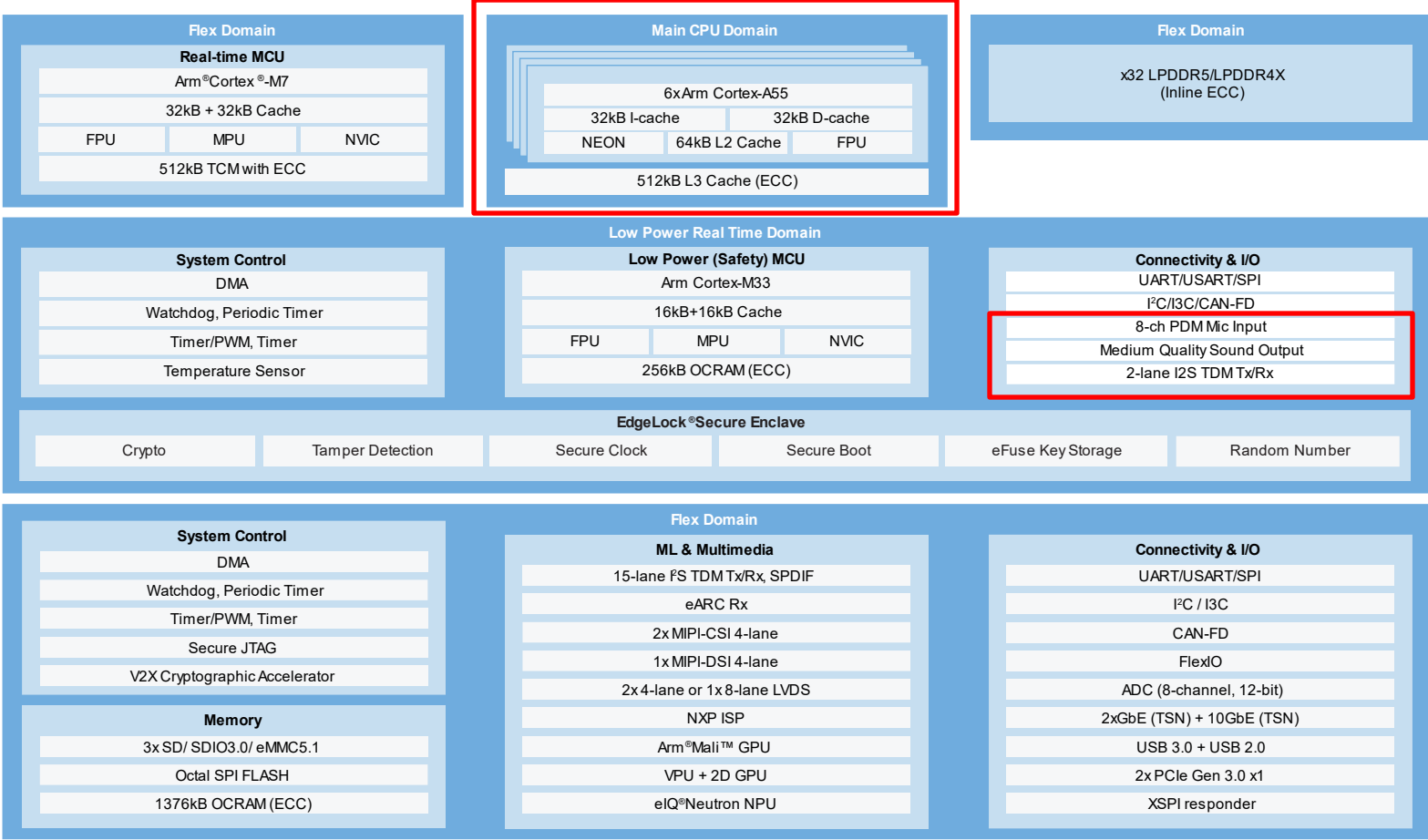
SOUND OPEN FIRMWARE SOLUTION OVERVIEW



NXP I.MX8MP APPLICATION PROCESSOR



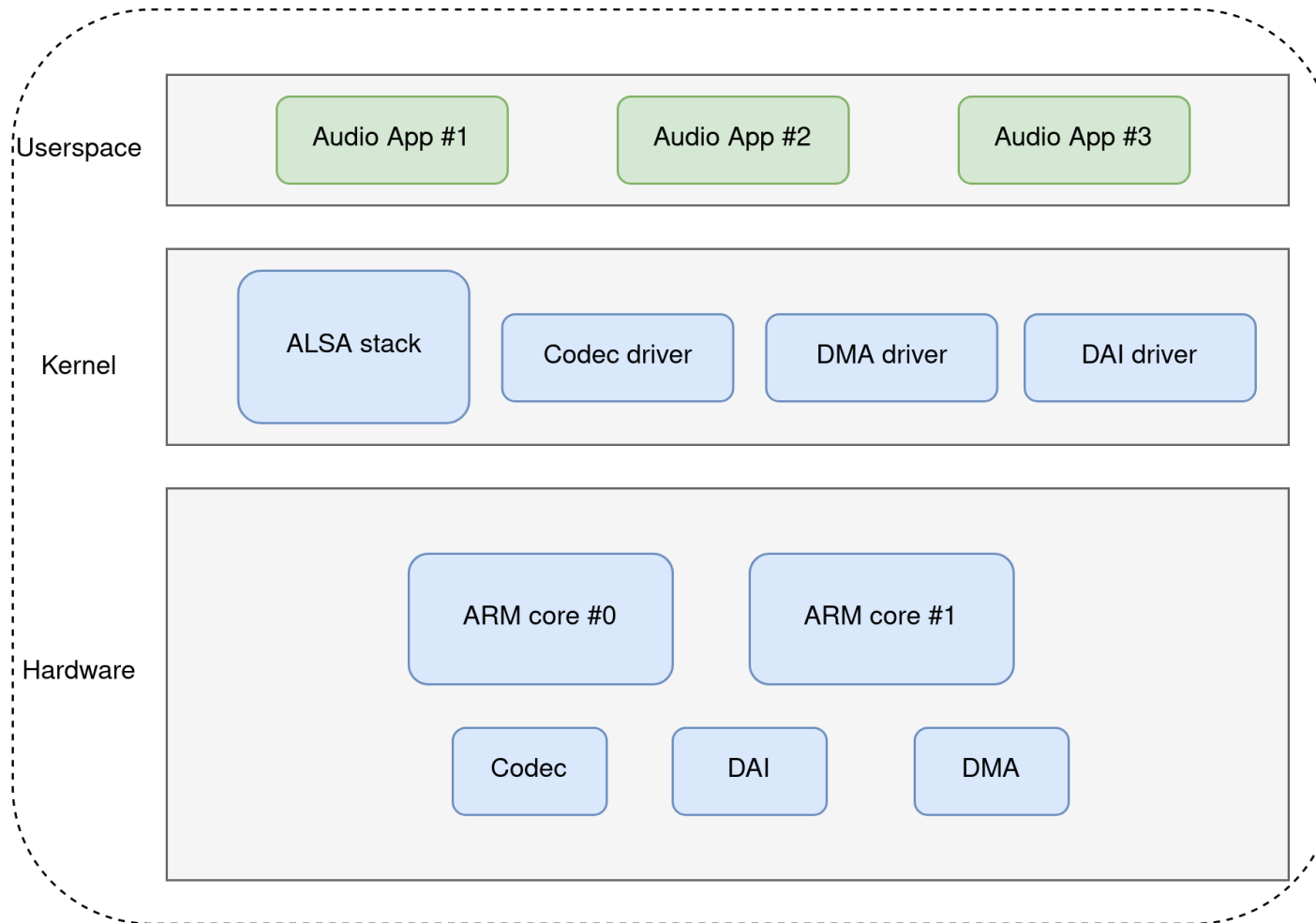
NXP I.MX95 APPLICATION PROCESSOR



WHAT IS SOUND OPEN FIRMWARE?

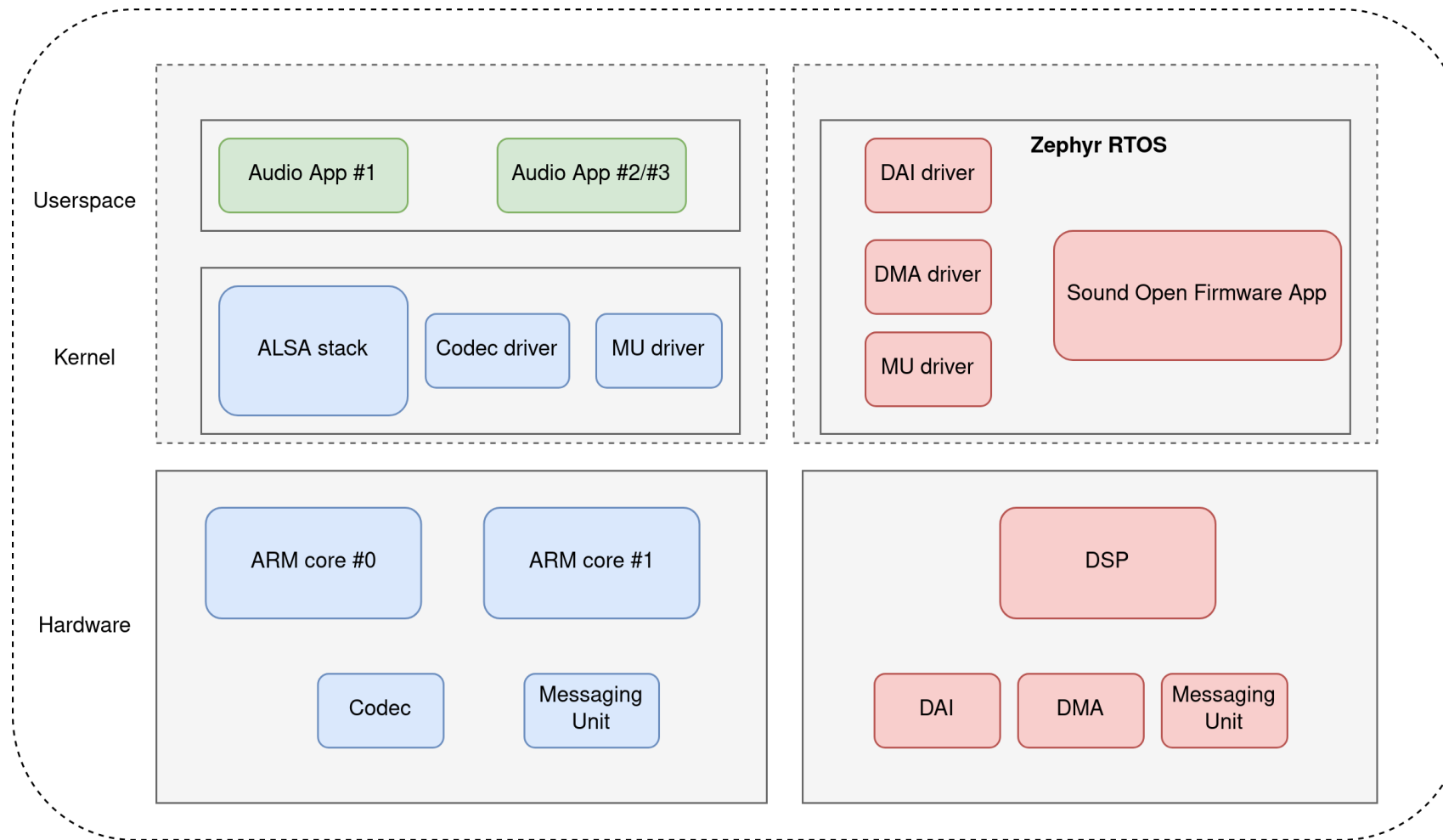
- Open Source Audio firmware and SDK
 - BSD/MIT licensed firmware, BSD/GPL licensed Linux drivers
 - Developed initially for Cadence HIFI DSP family
- Platform agnostic
 - Supports multiple hosts platforms (x86 with Intel, AMD and arm64 with NXP, Mediatek)
- DSP agnostic
 - Generic OS interface
- Tools
 - Includes proprietary / open toolchain and libs
 - Includes a logging system and runtime debugging capabilities
 - Uses ALSA interface

AUDIO STACK ON LINUX



- All devices are managed by Linux
- Audio apps are using ALSA interface
- Typical HW IPs involved
 - Digital Audio Interface
 - Codec
 - DMA

AUDIO STACK ON LINUX WITH DSP



- Offload audio processing
- Real time capabilities
- DSP takes over some IPs management
- Dedicate one core just for Audio processing
- Introduce Messaging Unit for IPC

Jailhouse Hypervisor



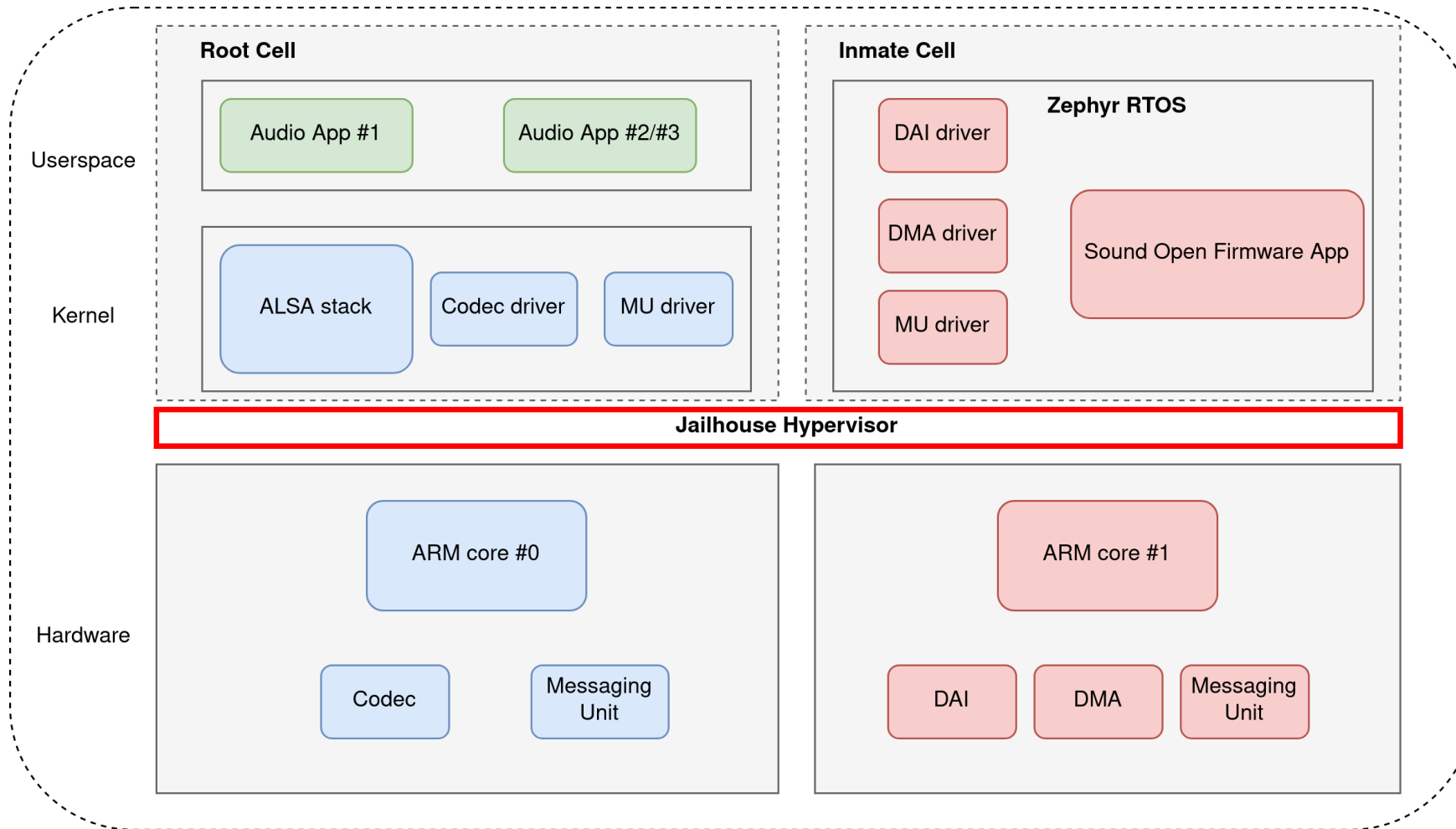
SECURE CONNECTIONS
FOR A SMARTER WORLD

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. © 2023 NXP B.V.



OFFLOADING AUDIO ON A SEPARATE CORE (ARM)



- Use jailhouse for isolation
- Dedicate one ARM core to run the Firmware
- Port firmware on arm64

JAILHOUSE HYPERVISOR

- static partitioning hypervisor
- splits existing hardware resources into isolated "cells"
- partition a **booted** system
- Linux starts first
 - loads jailhouse kernel module
 - enables **root cell**
 - creates the **inmate cell**
 - loads **zephyr.bin**
 - starts inmate cell



```
1 root@imx93mek:~# modprobe jailhouse.ko
2 root@imx93mek:~# jailhouse enable ./configs/arm64/imx93.cell
3 root@imx93mek:~# jailhouse cell create ./configs/arm64/imx93-zephyr-inmate.cell
4 root@imx93mek:~# jailhouse cell load zephyr ./zephyr.bin --address 0xce000000
5 root@imx93mek:~# jailhouse cell start zephyr
```

Sound Open Firmware Linux driver



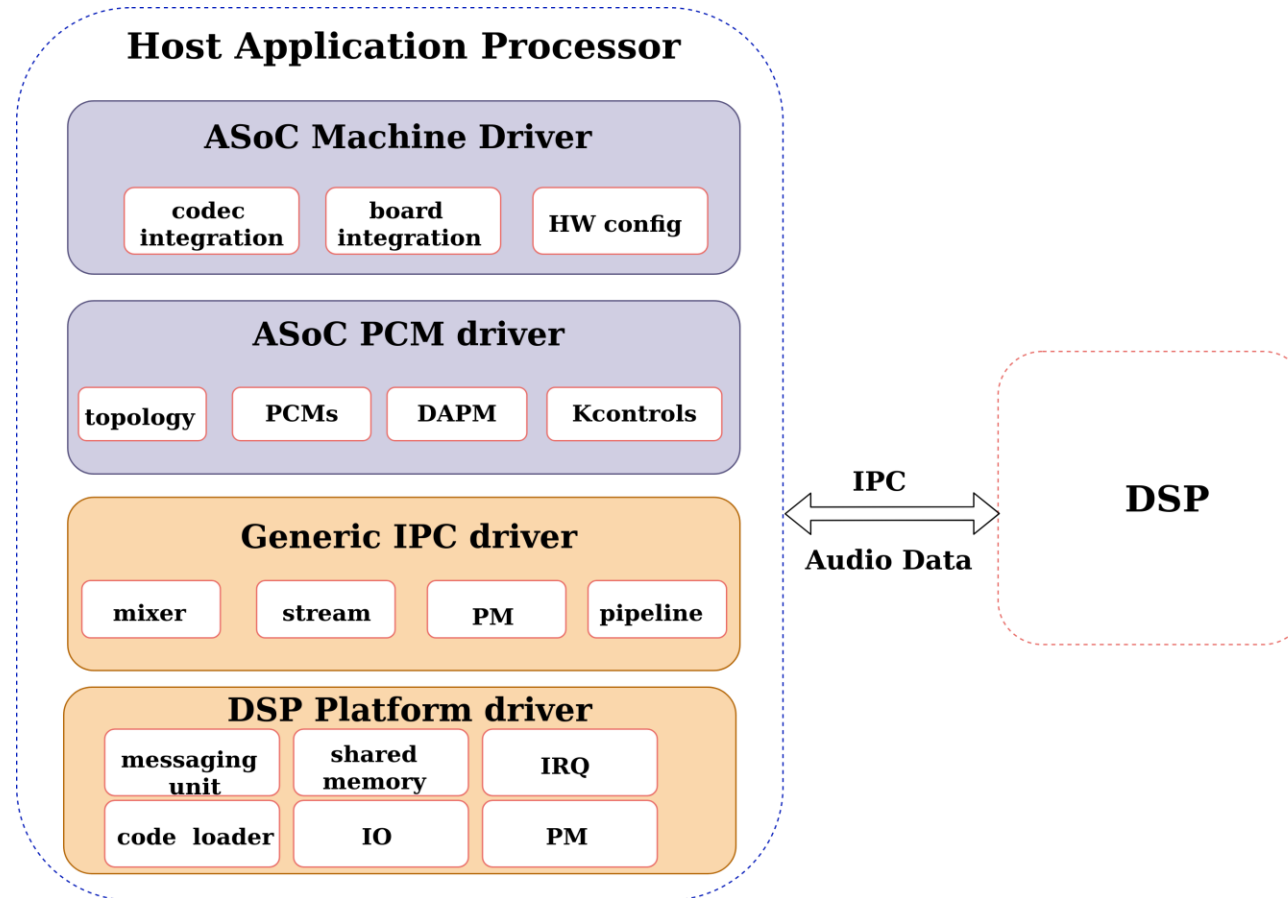
SECURE CONNECTIONS
FOR A SMARTER WORLD

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. © 2023 NXP B.V.



SOF LINUX DRIVER ARCHITECTURE



SOF LINUX DRIVER ALSA COMPONENTS

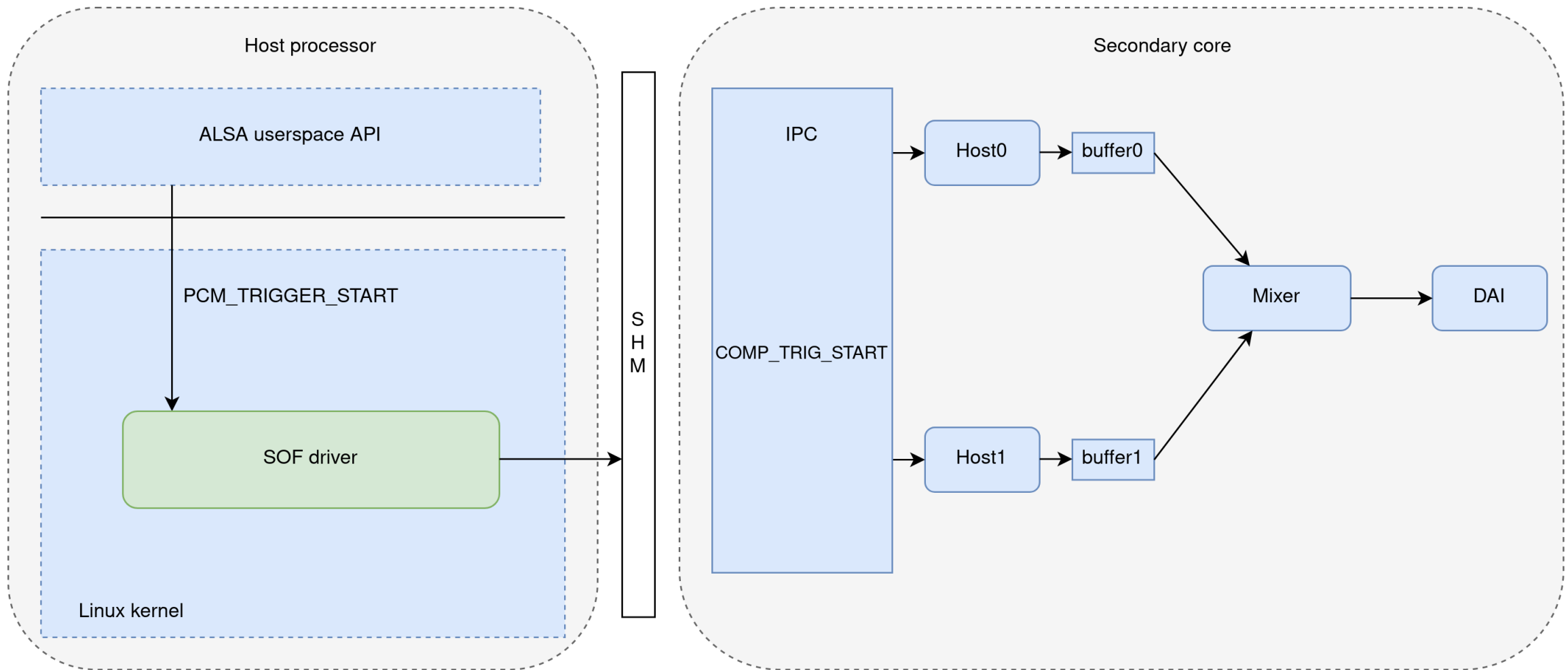
- Machine driver
 - Puts together codec, cpu dai into a sound card
- PCM driver
 - Helps implementing ALSA operation (pcm, compress, topology, kcontrols)
- Generic IPC driver
 - Allows to transfer ALSA commands to the FW
 - Handles FW notifications (e.g audio data consumed)
- DSP platform driver
 - Implements IPC transport protocol based on hardware IPs available

ALSA PCM OPS

```
1 struct snd_pcm_ops {
2     int (*open)(struct snd_pcm_substream *substream);
3     int (*close)(struct snd_pcm_substream *substream);
4     int (*ioctl)(struct snd_pcm_substream * substream,
5                 unsigned int cmd, void *arg);
6     int (*hw_params)(struct snd_pcm_substream *substream,
7                     struct snd_pcm_hw_params *params);
8     int (*prepare)(struct snd_pcm_substream *substream);
9     int (*trigger)(struct snd_pcm_substream *substream, int cmd);
10    snd_pcm_uframes_t (*pointer)(struct snd_pcm_substream *substream);
11    /* .... */
12 };
```

- open
 - Loads the SOF FW and topology
- hw_params
 - configures IPC PCM params
- trigger
 - stream actions: TRIGGER_START
- ioctl
 - copies audio data buffers

ALSA API INTERACTION



SOF IPC CUSTOM PROTOCOL

```
1  /* Global Message Types */
2  #define SOF_IPC_GLB_TPLG_MSG      SOF_GLB_TYPE(0x3U)
3  #define SOF_IPC_GLB_PM_MSG       SOF_GLB_TYPE(0x4U)
4  #define SOF_IPC_GLB_COMP_MSG     SOF_GLB_TYPE(0x5U)
5  #define SOF_IPC_GLB_STREAM_MSG   SOF_GLB_TYPE(0x6U)
6  #define SOF_IPC_GLB_GDB_DEBUG    SOF_GLB_TYPE(0xAU)
7
8  /* DSP Command Message Types */
9
10 /* stream */
11 #define SOF_IPC_STREAM_PCM_PARAMS SOF_CMD_TYPE(0x001)
12 #define SOF_IPC_STREAM_TRIG_START SOF_CMD_TYPE(0x004)
13 #define SOF_IPC_STREAM_TRIG_STOP  SOF_CMD_TYPE(0x005)
14 #define SOF_IPC_STREAM_TRIG_PAUSE SOF_CMD_TYPE(0x006)
15 #define SOF_IPC_STREAM_TRIG_RELEASE SOF_CMD_TYPE(0x007)
16
17 /* topology */
18 #define SOF_IPC_TPLG_COMP_NEW      SOF_CMD_TYPE(0x001)
19 #define SOF_IPC_TPLG_COMP_FREE    SOF_CMD_TYPE(0x002)
20 #define SOF_IPC_TPLG_COMP_CONNECT SOF_CMD_TYPE(0x003)
21 #define SOF_IPC_TPLG_BUFFER_NEW   SOF_CMD_TYPE(0x020)
22
23 /* PM */
24 #define SOF_IPC_PM_CTX_SAVE        SOF_CMD_TYPE(0x001)
25 #define SOF_IPC_PM_CTX_RESTORE    SOF_CMD_TYPE(0x002)
```

- firmware load
- topology load
- custom transport protocol
 - shared memory and doorbell interrupt
- Host AP sends commands
 - stream, topology, PM, debug, trace, probe
- Secondary core sends
 - stream notifications
 - FW ready

SOF UTILITIES

- sof-logger
 - User space application that reads the logs from shared memory
- rimage
 - tool for firmware image creation (encapsulate elf in a specific binary format)
- topology files
 - describes the Audio components
 - Based on this Linux kernel sends commands to the FW to create audio topology components
- sof-ctl
 - Used to send custom configuration to SOF FW
 - Encoded usually as a binary blob

Sound Open Firmware with Zephyr



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. © 2023 NXP B.V.



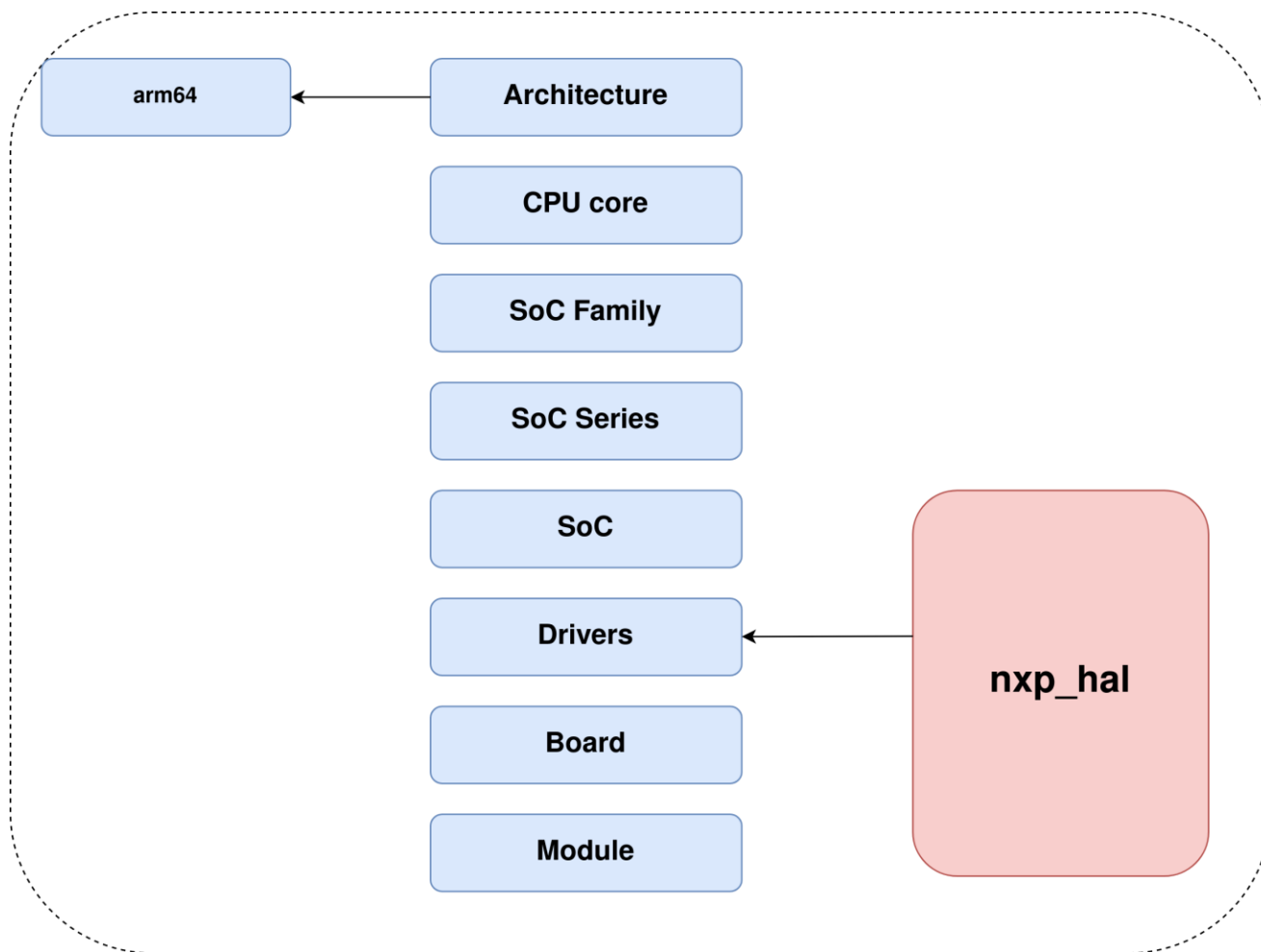
SOUND OPEN FIRMWARE DEVELOPMENT DIRECTION

- Permissive BSD/MIT licensed code
- Architecture agnostic written in C, initially supporting Cadence Xtensa DSPs
 - HIFI2, HIFI3 DSPs used by Intel
 - HIFI4 DSP used by NXP on i.MX8MP, i.MX8QXP, i.MX8QM
- Features selected at runtime using Kconfig
- Support for user defined Audio scenarios
 - Dynamically loaded pipeline topologies at runtime
- Modular design initially started with XTOS
 - Limitation: it only supports Xtensa architecture
- Gradually switching to Zephyr
 - Support for Xtensa
 - Support for arm64 and many more architectures

SOUND OPEN FIRMWARE ON IMX93_EVK_A55 BOARD

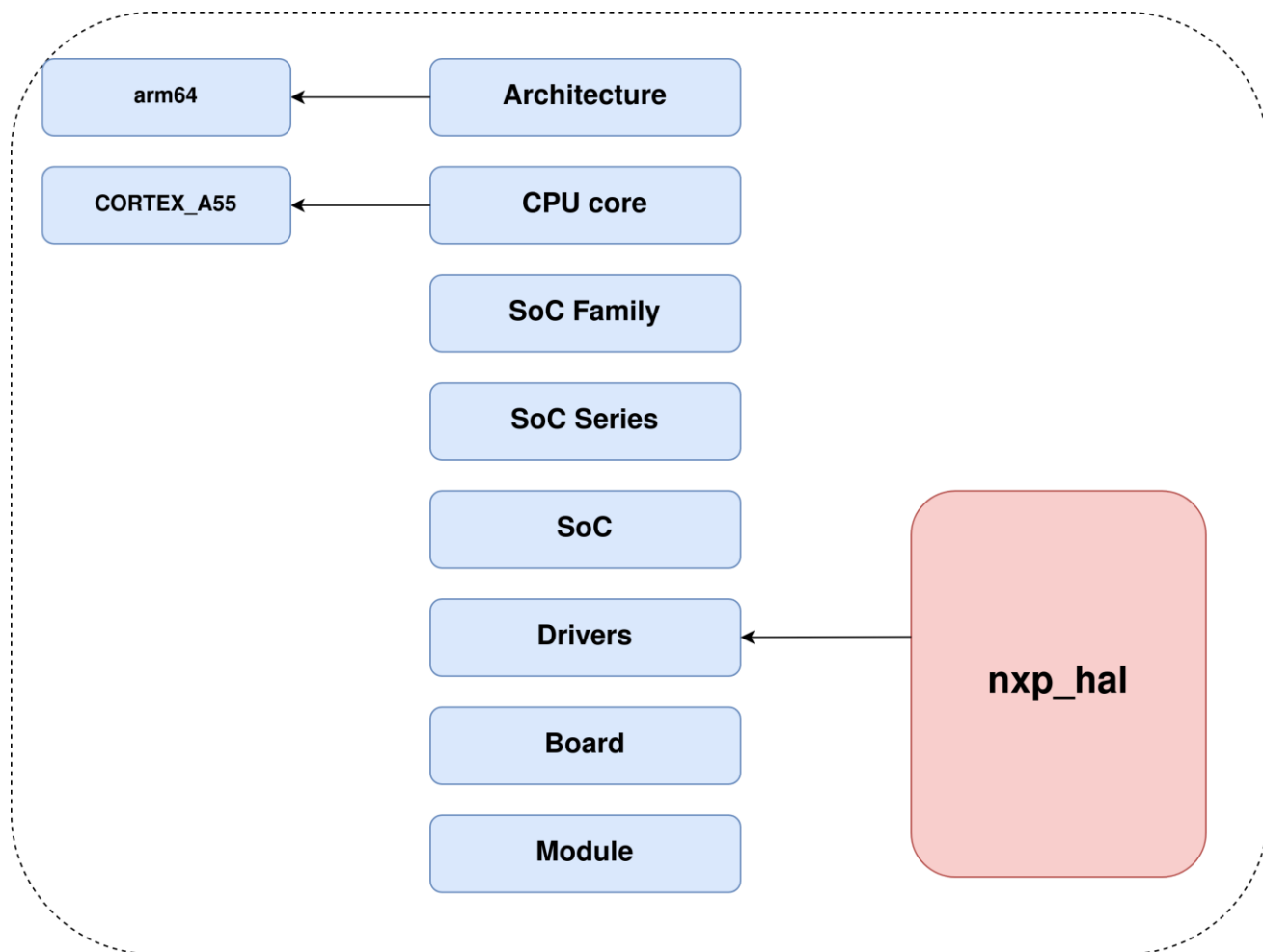
- Use Jailhouse to isolate one A55 core to run Zephyr
- Enable support for **imx93_evk_a55** board
- Enable SOF application in Zephyr on imx93_evk_a55 board
 - Use dts overlay to describe SOF memory regions (sram, sai, edma)
 - Use Kconfig fragment to annotate default board configuration (e.g DCACHE_LINE_SIZE, etc)
- Continue work on completely supporting Sound Open Firmware as a **native** Zephyr app
 - Use native API for OS primitives
 - Use native API for platform drivers
- End goal: Sound Open Firmware will only contain Audio logic
 - Everything else is provided by Zephyr

ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR



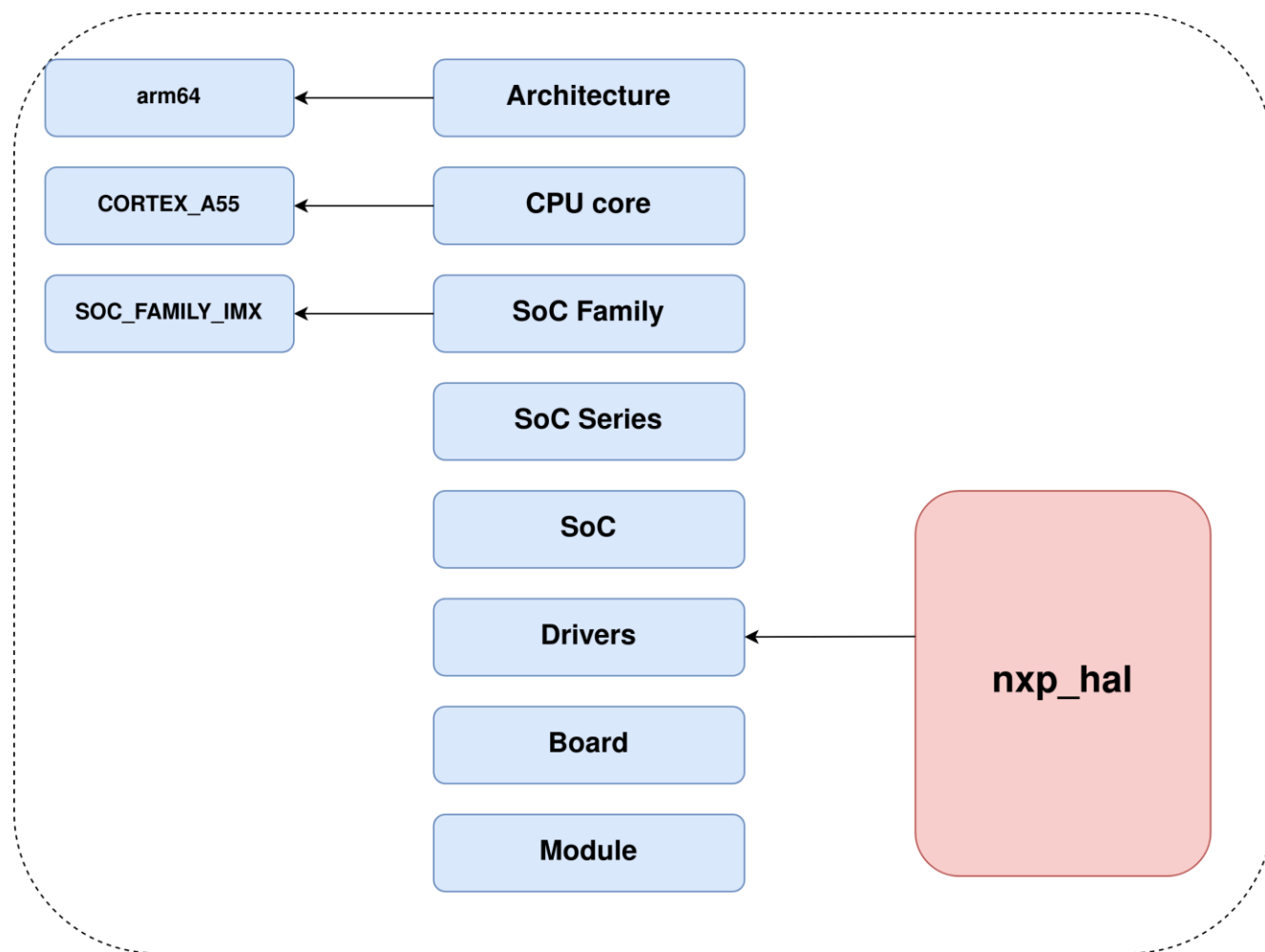
```
1 $ cat arch/Kconfig
2
3 config ARM64
4     bool
5     select ARCH_IS_SET
6     select 64BIT
7     select HAS_DTS
8     select HAS_ARM_SMCCC
9     select ARCH_HAS_THREAD_LOCAL_STORAGE
10    select BARRIER_OPERATIONS_ARCH
11    //...
12    help
13        ARM64 (AArch64) architecture
```

ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR



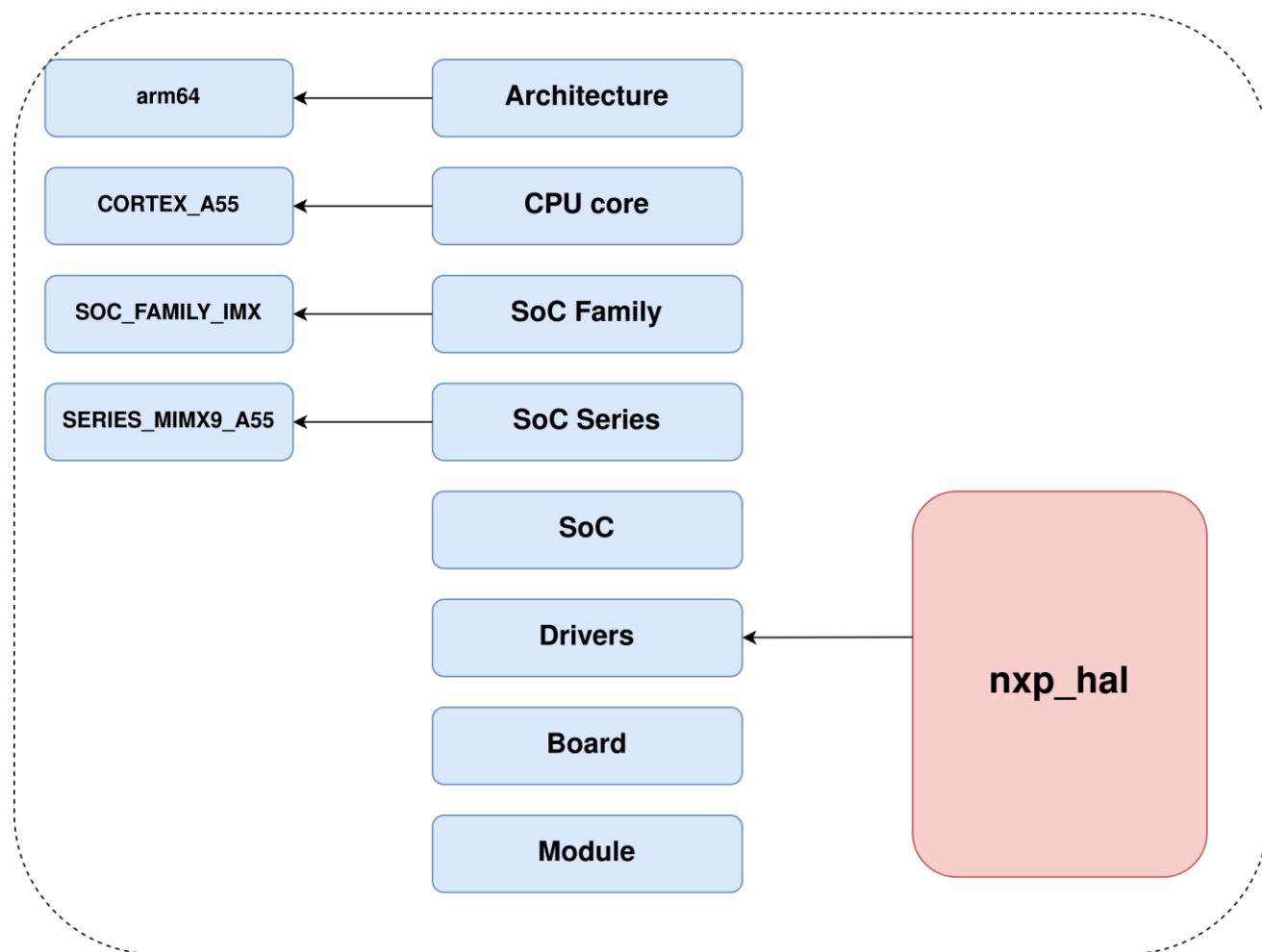
```
1 $ cat arch/arm64/core/Kconfig
2
3 config CPU_CORTEX_A55
4     bool
5     select CPU_CORTEX_A
6     select ARMV8_A
7     help
8         Use of a Cortex-A55 CPU
9
10 config ARMV8_A
11     bool
12     select ATOMIC_OPERATIONS_BUILTIN
13     select CPU_HAS_MMU
14     //....
15
16 config CPU_CORTEX_A
17     bool
18     select CPU_CORTEX
19     select SCHED_IPI_SUPPORTED if SMP
20     select CPU_HAS_FPU
21     select ARCH_HAS_SINGLE_THREAD_SUPPORT
22     select CPU_HAS_DCACHE
23     select CPU_HAS_ICACHE
24     //...
```


ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR



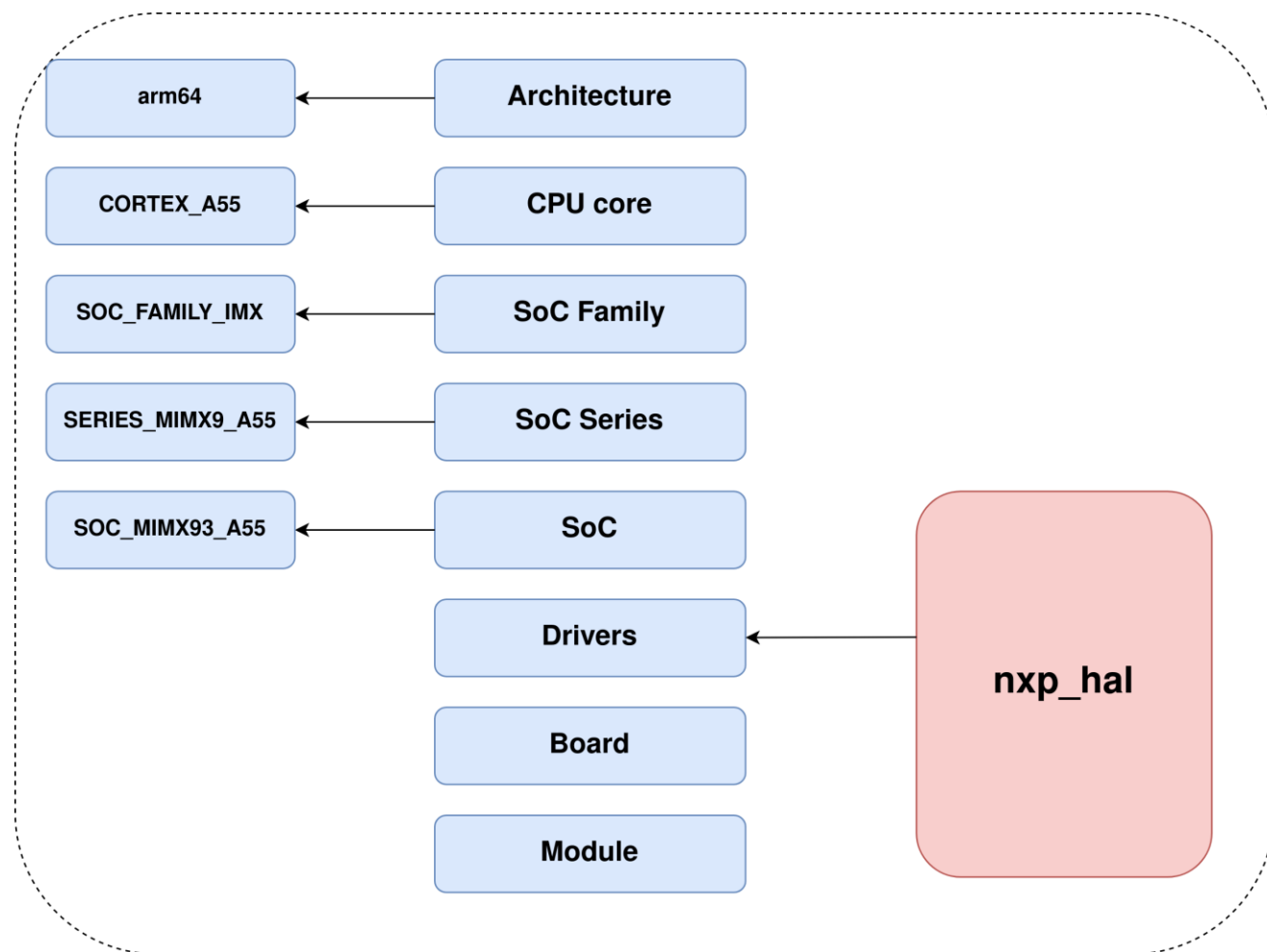
```
1 $ tree soc/arm64/nxp_imx/
2 |--- CMakeLists.txt
3 |--- Kconfig
4 |--- Kconfig.defconfig
5 |--- Kconfig.soc
6 |--- mimx8m
7 |--- mimx9
8
9 config SOC_FAMILY_IMX
10     bool
11
12 if SOC_FAMILY_IMX
13
14 config SOC_FAMILY
15     string
16     default "nxp_imx"
17
18 source "soc/arm64/nxp_imx/*/Kconfig.soc"
```

ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR



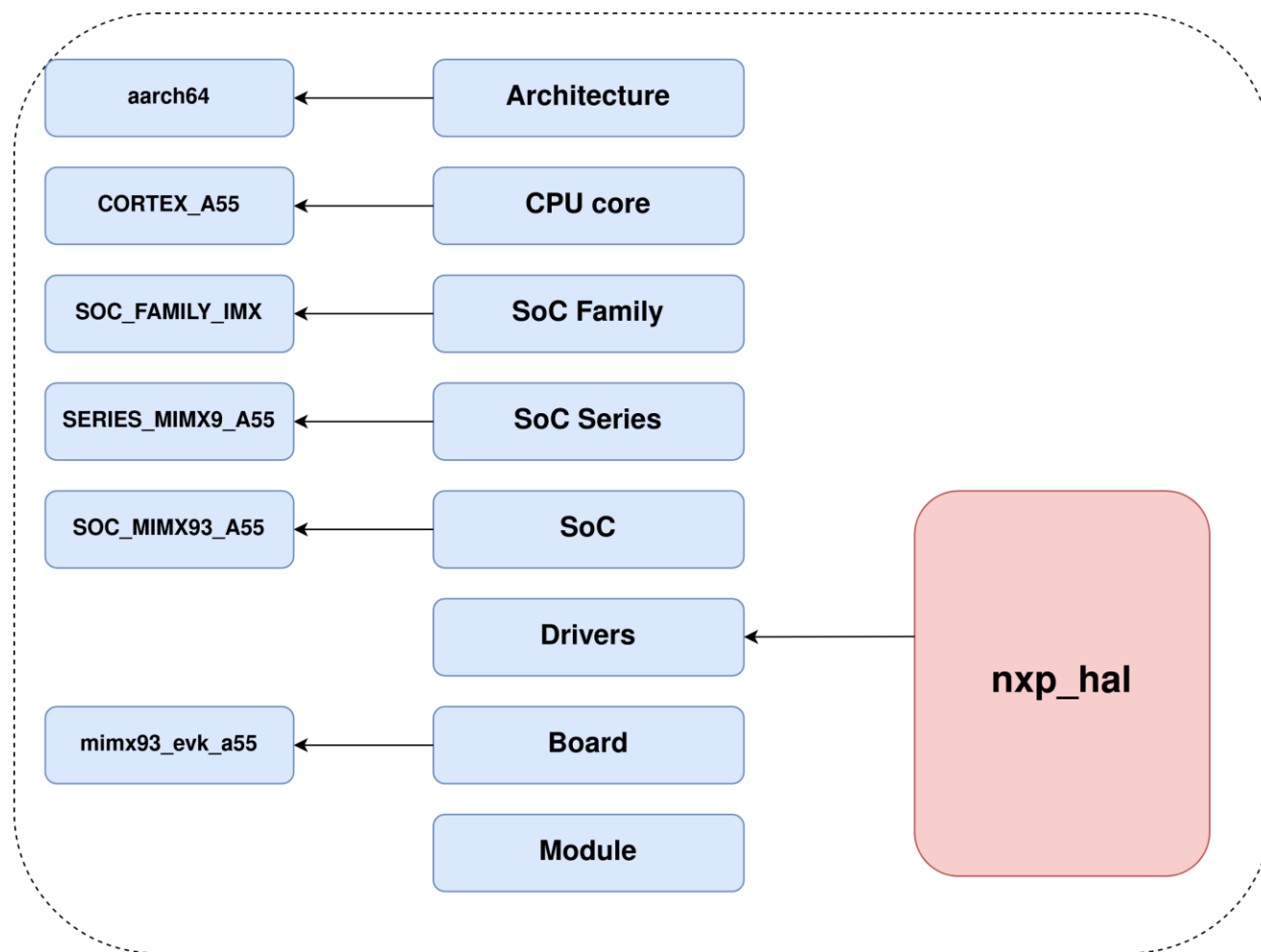
```
1 $ tree soc/arm64/nxp_imx/mimx9/
2 |— CMakeLists.txt
3 |— Kconfig.defconfig.mimx93
4 |— Kconfig.defconfig.series
5 |— Kconfig.series
6 |— Kconfig.soc
7 |— linker.ld
8 |— mmu_regions.c
9 |— pinctrl_soc.h
10
11 $ cat arm64/nxp_imx/mimx9/Kconfig.series
12
13 config SOC_SERIES_MIMX9_A55
14     bool "NXP i.MX9 A55 Core Series"
15     select ARM64
16     select SOC_FAMILY_IMX
17     help
18         Enable support for i.MX9 A55 Series.
```

ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR



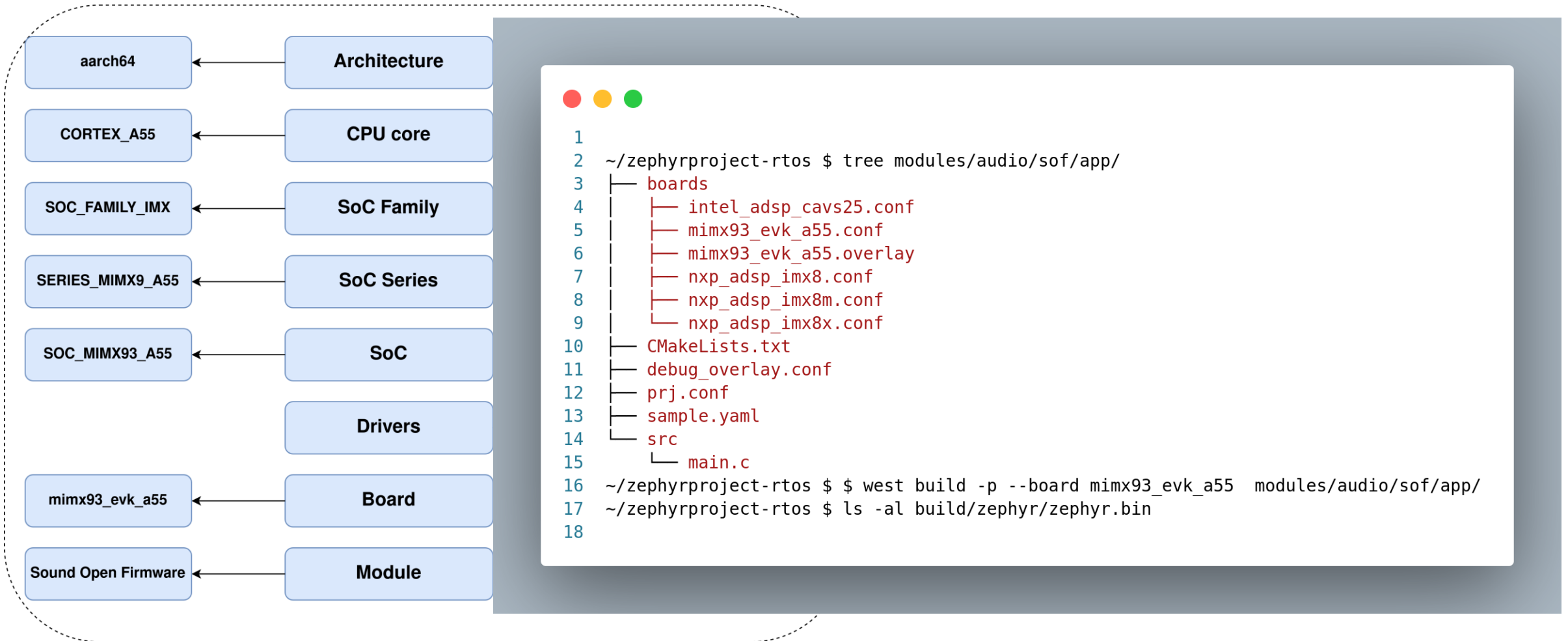
```
1 $ cat soc/arm64/nxp_imx/mimx9/Kconfig.soc
2 choice
3 prompt "NXP i.MX9 A55 Selection"
4 depends on SOC_SERIES_MIMX9_A55
5
6 config SOC_MIMX93_A55
7     bool "NXP i.MX93 A55"
8     select ARM64
9     select CPU_CORTEX_A55
10    select ARM_ARCH_TIMER
11    select HAS_MCUX if CLOCK_CONTROL
12    select HAS_MCUX_CCM if CLOCK_CONTROL
13    select HAS_MCUX_IOMUXC if PINCTRL
14
15 endchoice
```

ENABLE IMX93_EVK_A55 BOARD SUPPORT WITH ZEPHYR

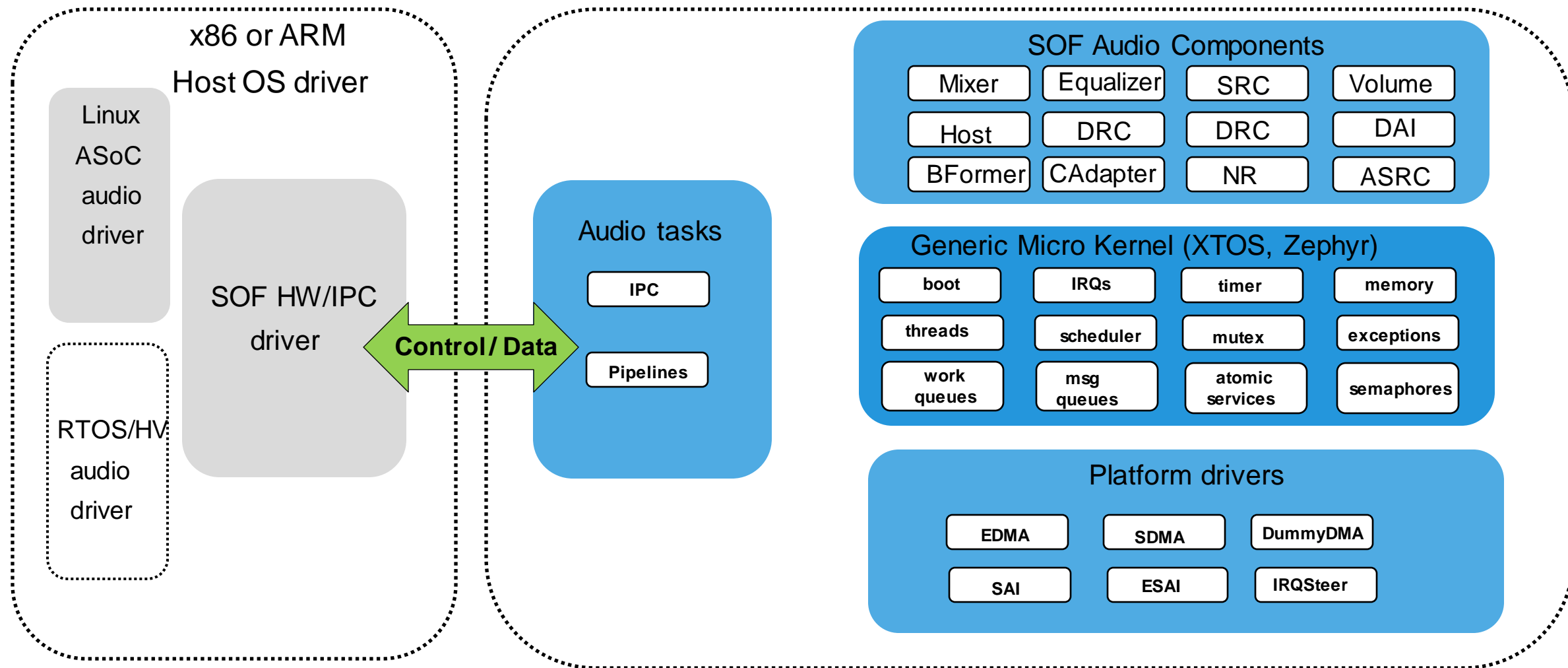


- cpu
 - arm,cortex-a55
- timer
 - arm,armv8-timer
- gic
 - arm,gic
- clock control module (CCM)
- iomuxc
 - nxp,imx-iomuxc
- lpuart
 - nxp,imx-lpuart

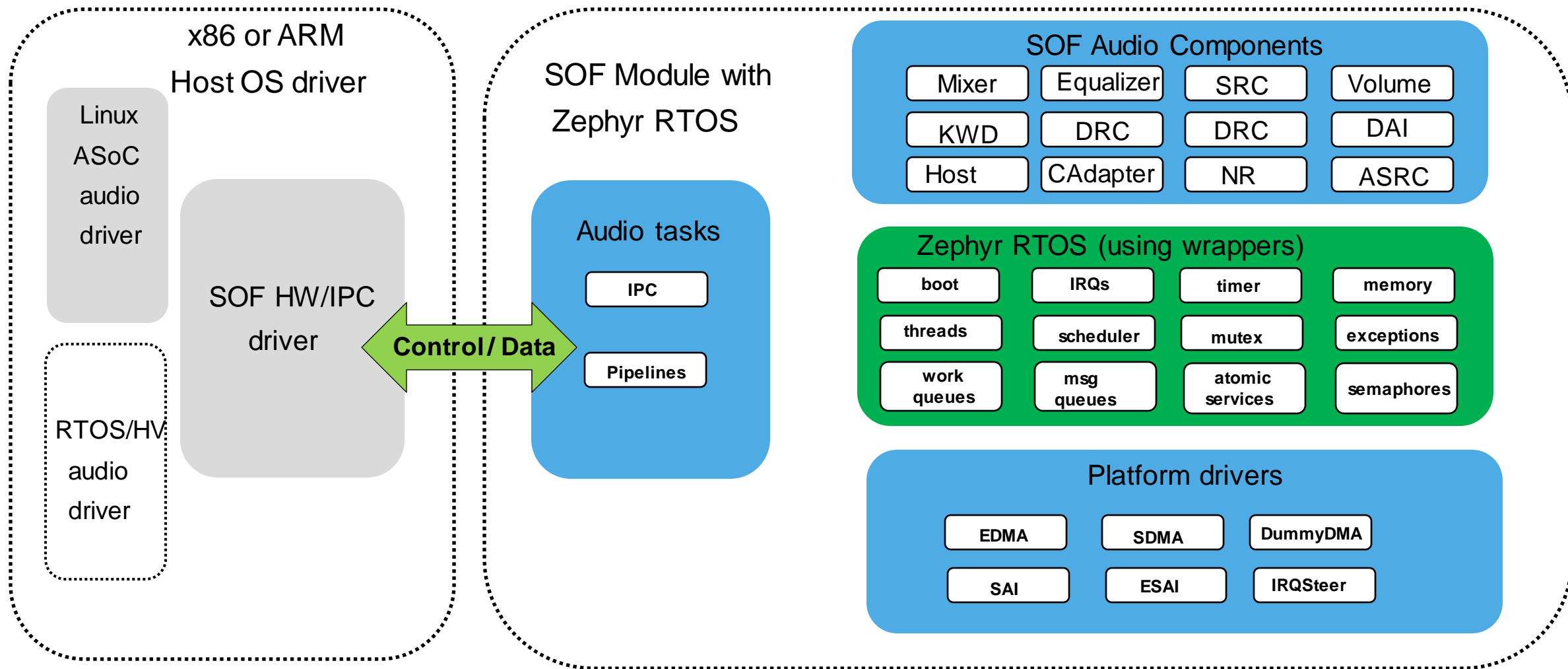
SOUND OPEN FIRMWARE APP USING ZEPHYR ON IMX93 BOARD



SOUND OPEN FIRMWARE ARCHITECTURE



SOF ARCHITECTURE WITH ZEPHYR RTOS



GENERIC MICRO KERNEL SUPPORT

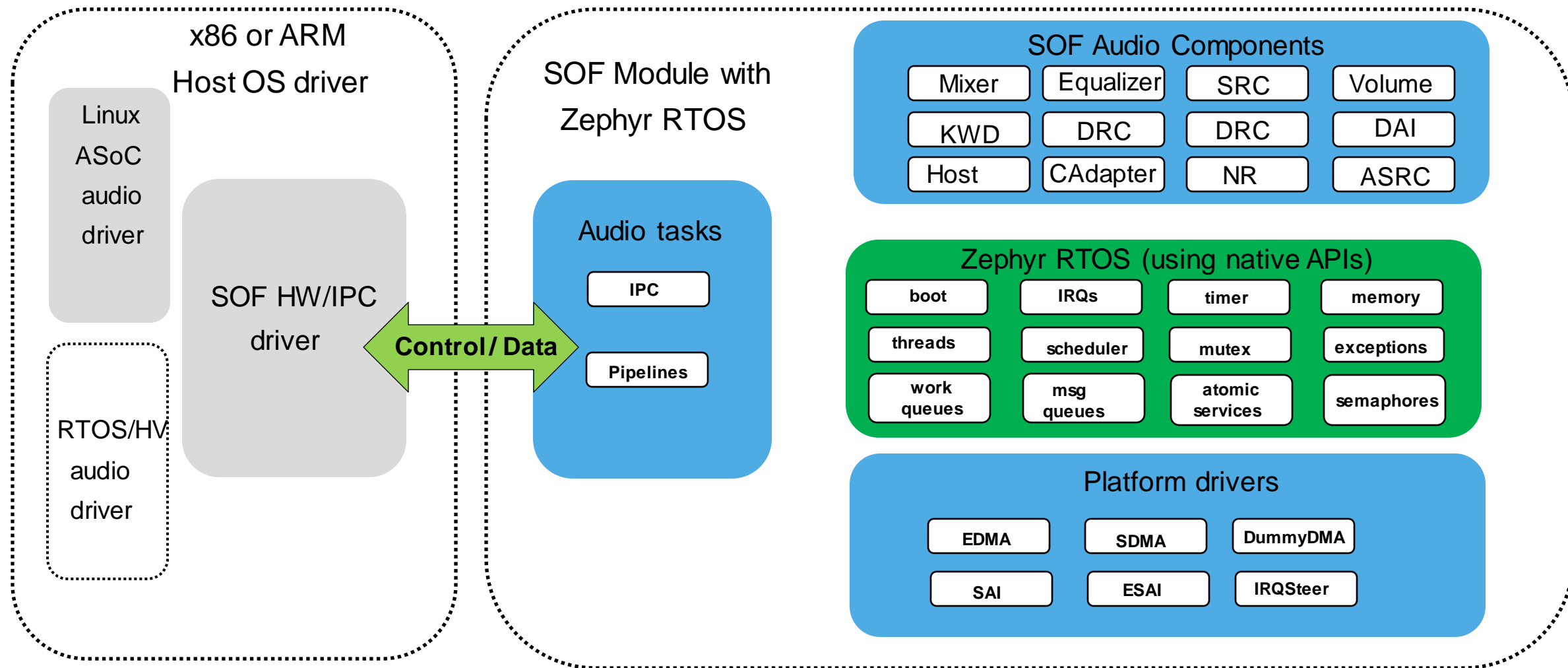


```
1 static inline int
2 interrupt_register(uint32_t irq, void(*handler)(void *arg), void *arg)
3 {
4 #ifdef CONFIG_DYNAMIC_INTERRUPTS
5     return arch_irq_connect_dynamic(irq, 0,
6                                     (void (*)(const void *))handler, arg, 0);
7 #else
8     return -EOPNOTSUPP;
9 #endif
10 }
```

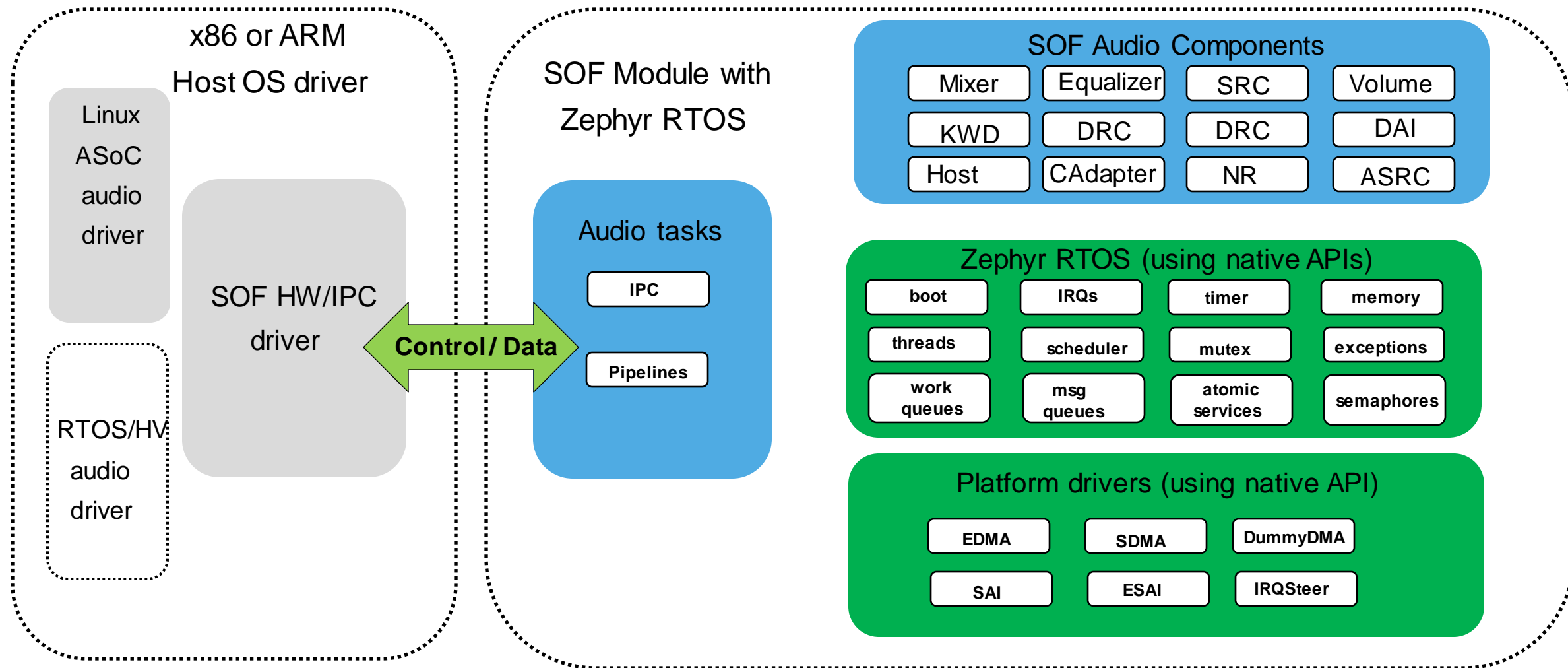


```
1 $ modules/audio/sof/zephyr/include/rtos/
2 |— alloc.h
3 |— atomic.h
4 |— bit.h
5 |— cache.h
6 |— clk.h
7 |— idc.h
8 |— init.h
9 |— interrupt.h
10 |— kernel.h
11 |— panic.h
12 |— sof.h
13 |— spinlock.h
14 |— string.h
15 |— task.h
16 |— timer.h
17 |— wait.h
```


SOF ARCHITECTURE – TRANSITION PHASE



SOF ARCHITECTURE – FINAL PHASE



DIGITAL AUDIO INTERFACE API

- New API introduced in Zephyr
- **include/zephyr/drivers/dai.h**
- High level audio driver abstraction
- NXP IPs
 - Synchronous Audio Interface (SAI)
 - Enhanced Serial Audio Interface (ESAI)

```
1 // include/sof/lib/dai-legacy.h
2 struct dai_ops {
3     int (*probe)(struct dai *dai);
4     int (*remove)(struct dai *dai);
5     int (*set_config)(struct dai *dai, struct ipc_config_dai *config,
6                     const void *spec_config);
7     int (*trigger)(struct dai *dai, int cmd, int direction);
8     int (*get_hw_params)(struct dai *dai,
9                         struct sof_ipc_stream_params *params, int dir);
10    int (*hw_params)(struct dai *dai, struct sof_ipc_stream_params *params);
11 }
12
13 // include/zephyr/drivers/dai.h
14 __subsystem struct dai_driver_api {
15     int (*probe)(const struct device *dev);
16     int (*remove)(const struct device *dev);
17     int (*config_set)(const struct device *dev, const struct dai_config *cfg,
18                     const void *bespoke_cfg);
19     int (*config_get)(const struct device *dev, struct dai_config *cfg,
20                     enum dai_dir dir);
21     int (*trigger)(const struct device *dev, enum dai_dir dir,
22                  enum dai_trigger_cmd cmd);
23     const struct dai_properties *(*get_properties)(const struct device *dev,
24                                                    enum dai_dir dir, int stream_id);
25 };
```

DMA API AND DRIVERS

- SOF DMA ops vs Zephyr DMA API
- Introduce DMA suspend / resume in Zephyr
- Add **host-zephyr.c** and **dai-zephyr.c** in SOF
- Port EDMA and SDMA drivers
 - dma_mcux_edma already in Zephyr
- Use device tree to describe DMA devices

```
1 //include/zephyr/drivers/dma.h
2 __subsystem struct dma_driver_api {
3     dma_api_config config;
4     dma_api_reload reload;
5     dma_api_start start;
6     dma_api_stop stop;
7     dma_api_suspend suspend;
8     dma_api_resume resume;
9     dma_api_get_status get_status;
10    dma_api_get_attribute get_attribute;
11 };
12
13 // include/sof/lib/dma.h
14 struct dma_ops {
15     int (*set_config)(struct dma_chan_data *channel,
16                     struct dma_sg_config *config);
17     int (*copy)(struct dma_chan_data *channel, int bytes, uint32_t flags);
18     int (*start)(struct dma_chan_data *channel);
19     int (*stop)(struct dma_chan_data *channel);
20     int (*pause)(struct dma_chan_data *channel);
21     int (*release)(struct dma_chan_data *channel);
22     int (*probe)(struct dma *dma);
23     int (*remove)(struct dma *dma);
24 };
25
26 // zephyr/drivers/dma/dma_mcux_edma.c
27 static const struct dma_driver_api dma_mcux_edma_api = {
28     .start = dma_mcux_edma_start,
29     .stop = dma_mcux_edma_stop,
30     .suspend = dma_mcux_edma_suspend,
31     .resume = dma_mcux_edma_resume,
32     /* ... */
33 };
```

CONTINUE ENHANCE SUPPORT FOR SOF IN ZEPHYR

- Messaging Unit
 - Already existing in Zephyr, **ipm_imx** but needs some adaptation to fit SOF IPC
- Interrupt Steer
 - Adapt to use interrupt controller API in Zephyr
- Clocks
 - Move clock management on FW side
- Power domains
 - Move PM domains on FW side
- Decouple Firmware from Host OS
 - Standalone run of firmware
 - Potential for enabling on i.MX RT series

Q & A



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. © 2023 NXP B.V.





SECURE CONNECTIONS
FOR A SMARTER WORLD