



Zephyr[®] Project

Developer Summit 2022

June 8-9, 2022

Mountain View, CA + Virtual



Zephyr® Project
Developer Summit 2022

June 8-9, 2022

Mountain View, CA + Virtual

Virtualization with Zephyr & Xen for embedded safety systems

Dmytro Firsov
EPAM Systems

Agenda

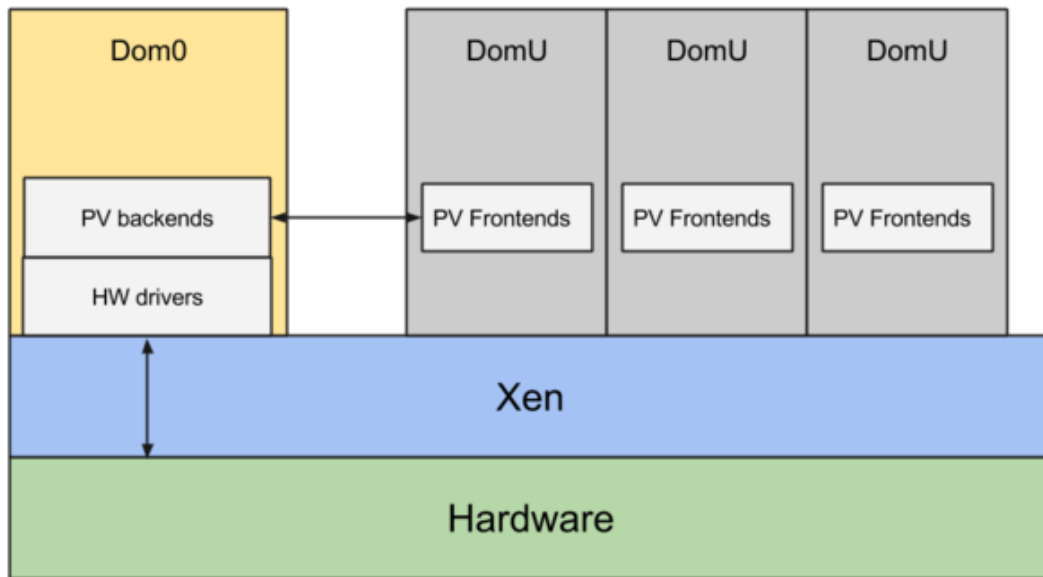
- Virtualization, Xen hypervisor
- Domain model in Xen
- Thin Domain-0 concept
- Zephyr as Xen control domain (Domain-0)
- Xen features upstream activities in Zephyr RTOS
- Development status for Zephyr as Xen control domain

Virtualization, Xen hypervisor

- Virtualization is the process, that allows to run virtual instance(s) of system (virtual machine) abstracting from actual computer system hardware;
- It helps to isolate software environment, run different OS on same machine simultaneously, divide huge computer systems to smaller parts and increase hardware utilization, etc.
- For creating, running and managing virtual machines the hypervisors are used.
- Xen is type 1, or bare metal hypervisor.
- Xen is a part of the Linux Foundation and supported on x86 and Arm platforms (todays speech will be about latter).
- Xen boots from a bootloader, starts privileged domain and manages CPU and memory resources for all created VMs.

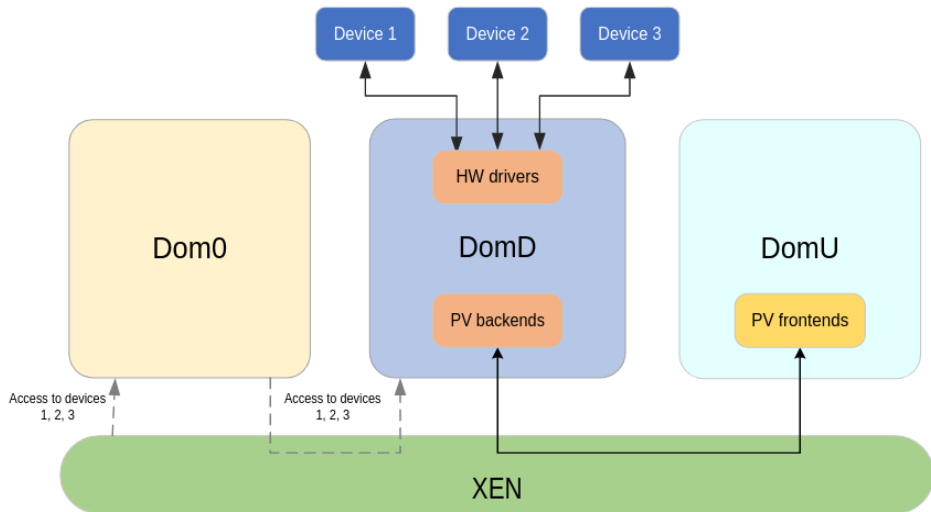
Domain model in Xen

- In classic Xen domain model, you have one privileged domain – Domain-0, all others are unprivileged domains – Domain-U.
- Domain-0 is started by Xen and works as control domain – it performs domain management.
- Also, Dom-Us do not have access to real hardware and work through the PV frontend drivers.
- This requires from Dom-0 to work also as a hardware domain and to deal with HW and PV backend drivers.





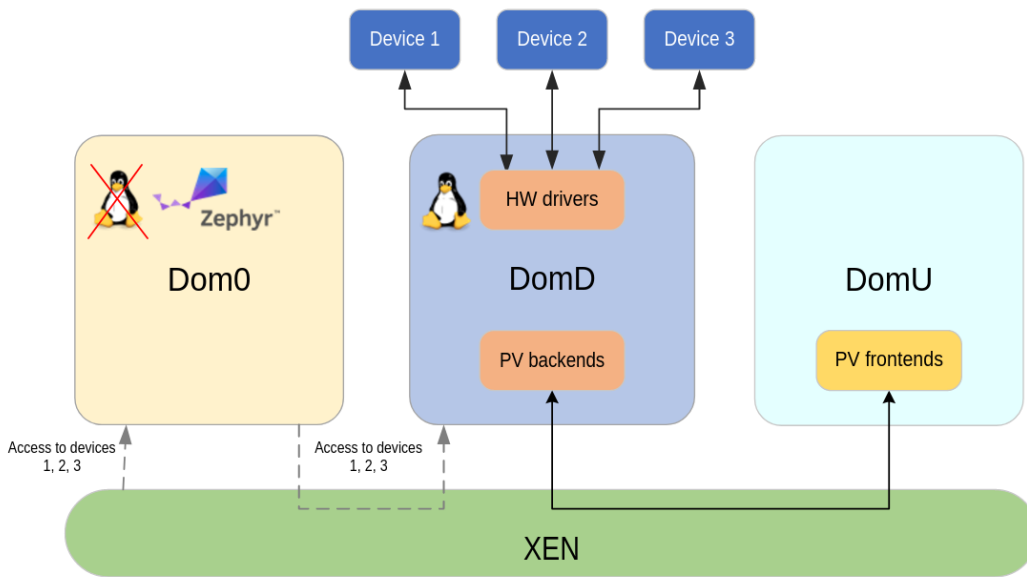
Thin Domain-0 concept



- This concept is aimed to reduce number of responsibilities for Dom-0.
- After boot, Dom-0 grants HW access for one specific domain, that becomes a hardware (driver) domain. It is called Domain-D.
- Dom-D deals with hardware and PV backend drivers.
- Domain-0 now works only as control domain and perform only domain management. So, it becomes light-weight.

Zephyr as Xen control domain (Domain-0)

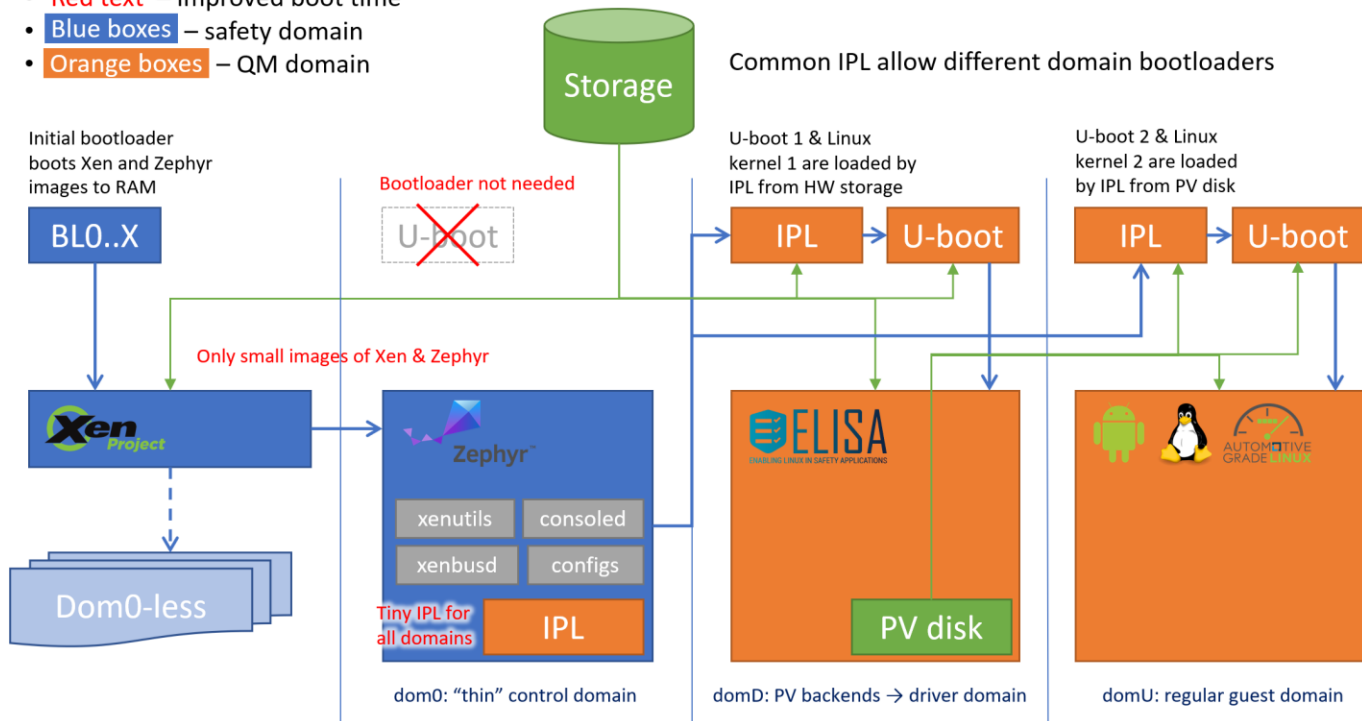
- Due to mentioned facts, we can use simpler (than Linux) OS for this purposes.
- Zephyr is a great choice for this purpose, because it is FOSS, part of LF (as Xen Project) and targeting Functional Safety use cases.
- Thin Domain-0 concept allows to minimize the changes needed to Zephyr.
- Some boot steps can be moved to external bootloaders.
- It is still needed to implement domain configuration & management tools in Zephyr.





Virtualization with Zephyr & Xen for embedded safety systems

- **Red text** – improved boot time
- **Blue boxes** – safety domain
- **Orange boxes** – QM domain

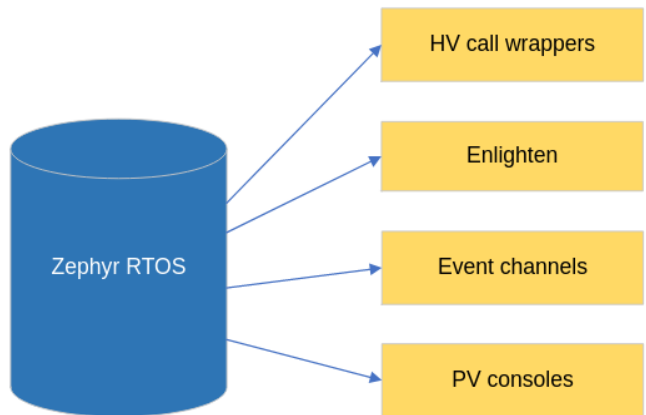




Zephyr as Xen control domain advantages

- Improves system boot time
- Reduces resources consumption
- Reduces overall system complexity
- Freedom of interference for functional safety
 - *when Xen and Zephyr will be FuSa compliant*

Xen features upstream activities in Zephyr RTOS



Xen support in Zephyr [was introduced](#) by my colleague, Volodymyr Babchuk, in January 2021 (as 'xenvm' board). Currently it supports only ARM64 architecture and runs as unprivileged domain. 'xenvm' provides some of basic Xen functionalities:

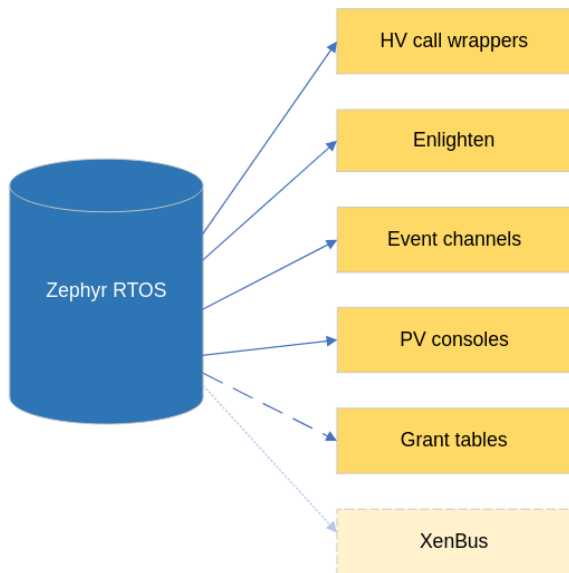
- [hypercall wrappers](#), that are needed for OS-to-hypervisor communication;
- [enlighten mapping](#) – special memory page with system info, provided to all domains by Xen;
- [event channels support](#) – Xen primitive for interdomain event interaction;

Also, Zephyr has 2 PV drivers for hypervisor console:

- [ring buffer based](#) – classic Xen domain PV console, that is used by all unprivileged domains;
- [consoleio based](#) – special console, that works via hypercalls – used by default only for Domain-0.



Xen features upstream activities in Zephyr RTOS



Some features are on review:

- [expanding](#) of event channels functionality – current implementation supports only Xen-predefined channels, these changes provide interface for allocation and interdomain binding. Needed for PV drivers' synchronization.
- [grant table driver](#) - one of Xen interfaces for memory sharing between domains, widely used for PV drivers' data exchange.

The big feature planned for upstreaming – XenBus driver. It is a complex Xen interface, that is used for a lot of purposes: receiving information about the domain, frontend/backend bindings, parameters exchange etc. It is crucial for establishing a connection between domains and PV driver configuration.

More detailed description of xenvm supported features you can find in [Zephyr docs](#).



Xen features upstream activities in Zephyr RTOS

Development of Xen support with Zephyr RTOS is quite challenging. Here are some of problems, that we faced so far:

- **Zephyr licensing conflicts**

Zephyr is Apache licensed, but lot of Xen code, drivers and utilities are under GPLv2, so we cannot take it as is. Another problem exists even for compatible licenses – Xen headers and some of Xen drivers are MIT licensed. For such licenses we need to get TSC approval every time.

- **Static configuration**

Zephyr does not support runtime configuration via device-tree (yet?), so some problems may occur during boot. For example, Xen will change memory regions for Dom0 (it may differ), and we have no way to tell Zephyr about it. Same issue is with relocation, Zephyr is compiled for specific start address and will crash when it will be changed by Xen.

- **Slow review for Xen-related PRs**

Some PRs may be unreviewed for months, and this slows development down. For some Xen PRs review is not a finish, because as I mentioned before some of them requires TSC approval due to licensing.

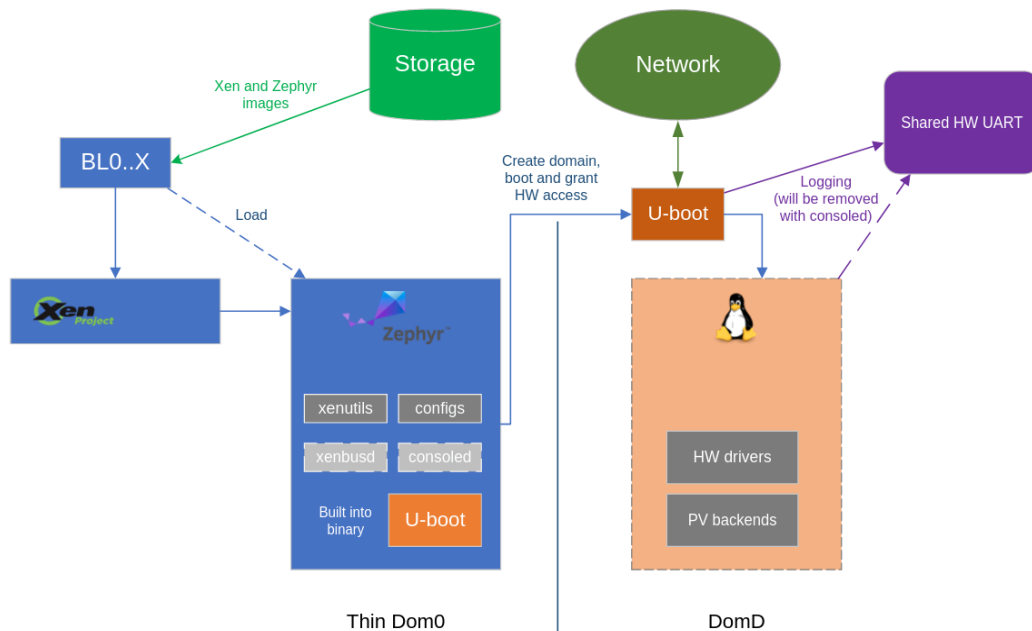
Development status for Zephyr as Xen control domain

Current implementation runs on Renesas H3 hardware and is available on GitHub:

- Zephyr RTOS is configured and used as Xen Dom0 - [sources](#).
- Development special tool for domain configuration & management was started. It is implemented as Zephyr app and called xenutils – [sources](#).
- U-boot already has Xen support on Arm, but it was impossible to enable Renesas related drivers to xenguest_arm64 (it is needed to use some shared peripherals). So new board config was added. Modified version of U-boot available [here](#).



Development status for Zephyr as Xen control domain



Development status for Zephyr as Xen control domain

- Zephyr Dom0 boots directly by Xen (no bootloader in-the-middle needed).
- Domain creation and destruction via Zephyr shell.
- In-code analogue of xen.cfg, lot of parameters for domains can be configured for domains, that are about to be created.
- Peripheral sharing – it implements iomem, irq and dtdev configuration . E.g., Ethernet device and one of the board UART are used for image loading and system logging in hardware domain.

Next targets are Linux booting, adding of xenbus driver, implementation of consoled and xenbusd Zephyr applications.



Zephyr® Project
Developer Summit 2022

June 8-9, 2022

Mountain View, CA + Virtual

Thank you!

Any questions?

Contact:
dmytro_firsov@epam.com