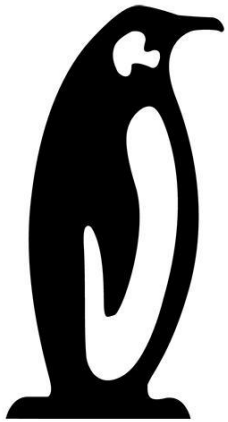
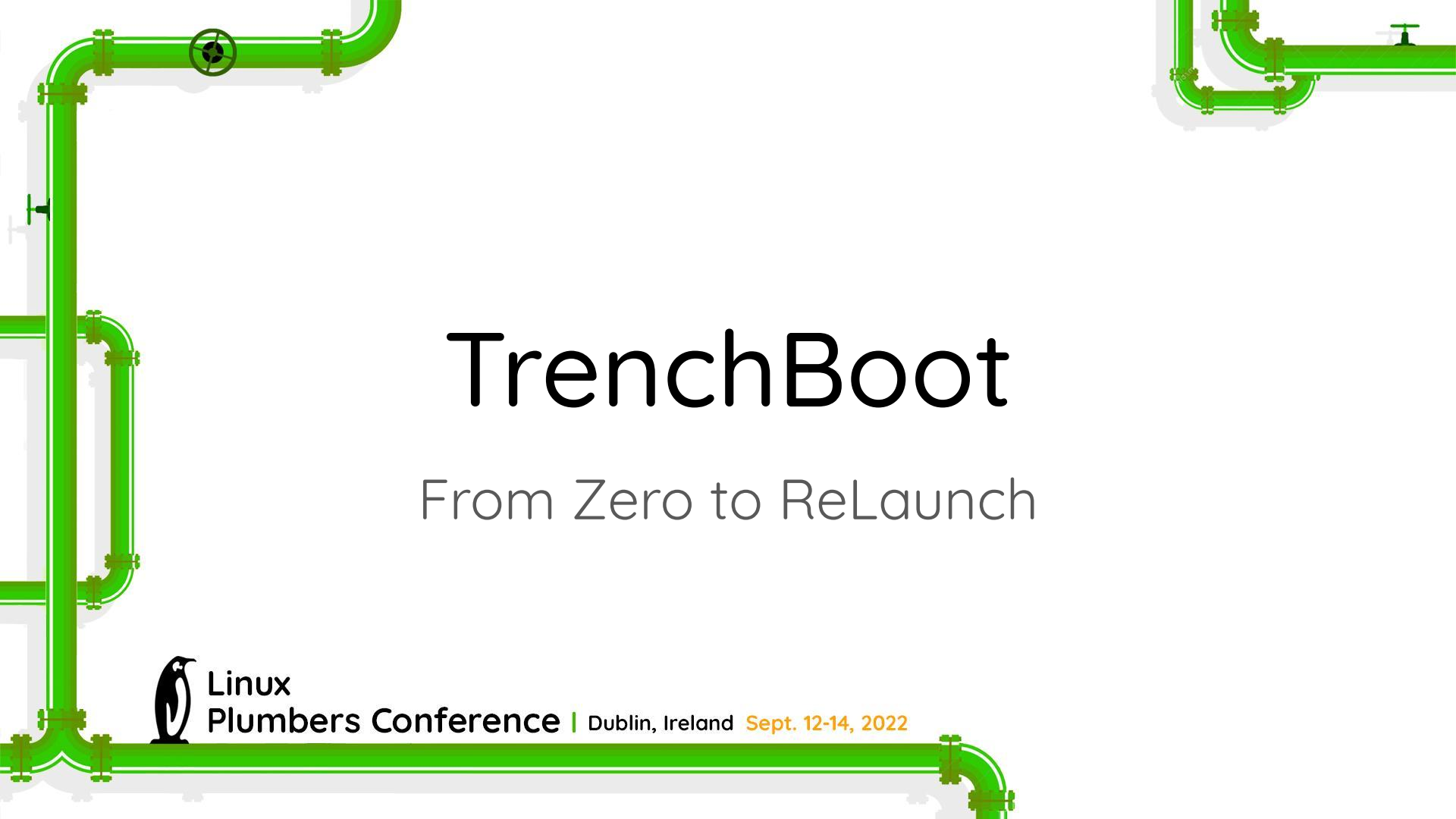


Linux Plumbers Conference

Dublin, Ireland September 12-14, 2022



TrenchBoot

From Zero to ReLaunch

Linux
Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022

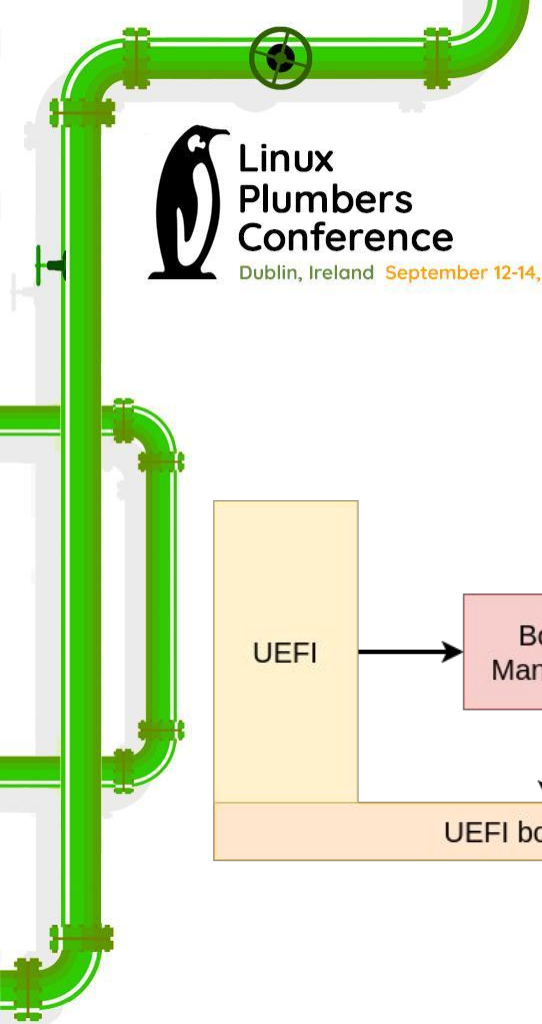
Introduction

- Update on TrenchBoot development
  - Review existing state
  - Introduction of SLRT and dl-stub to support efi-stub requirements
- Presenting the new Secure ReLaunch capability
  - Introduction to late-launch and the Secure ReLaunch implementation
  - How it is deployed
- Roadmap

Existing Secure Launch Series

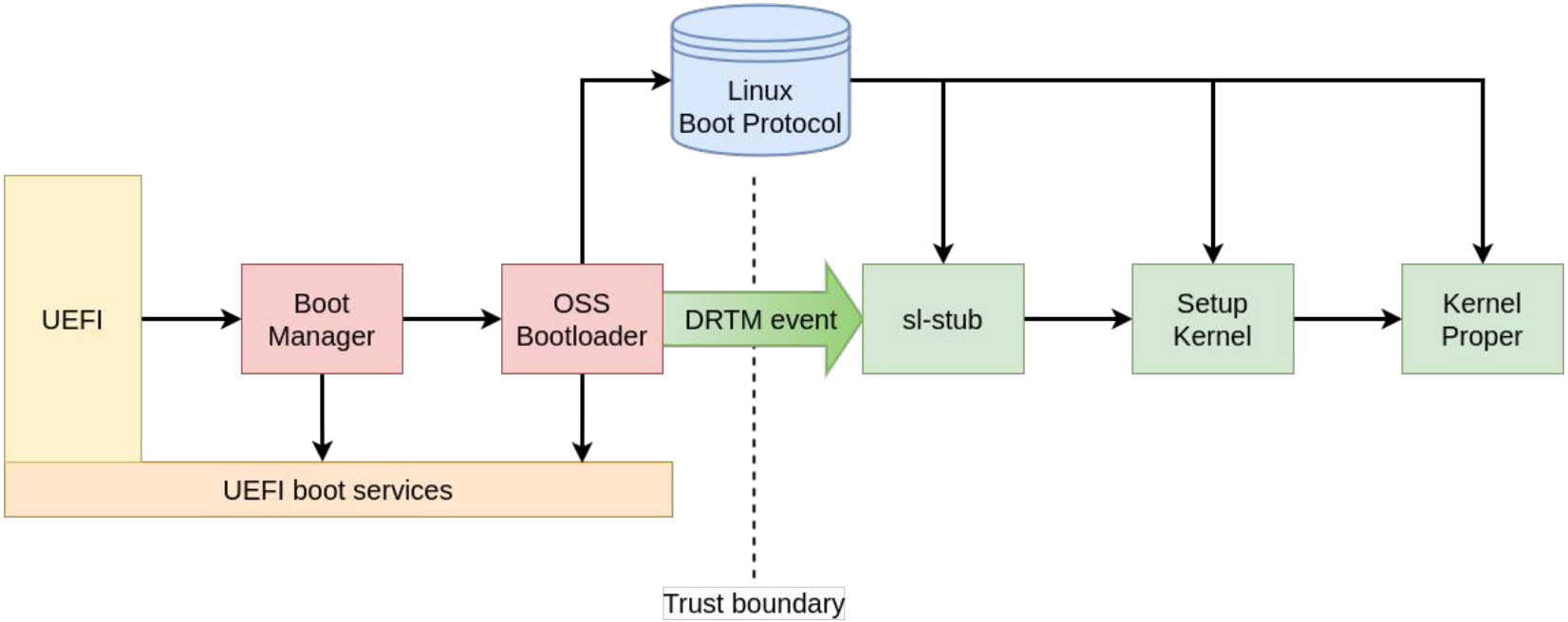
The development goals of the previous Secure Launch series was on how to get the Linux kernel running post dynamic launch.

- Make the world look right for the Linux kernel from a CPU perspective.
- Then capture as accurate of a record as possible of the Linux environment started by the dynamic launch.
- The series changes consisted of,
  - sl-stub - The kernel entrypoint that either the ACM (on Intel) or SKL (on AMD) would jump to, to start the kernel
  - Setup Kernel - The function sl_main() provides the core setup logic and recorder of the Linux environment constructed.
  - Kernel Proper - The function slaunch() provides final setup and validation while the slmodule provides limited runtime support.

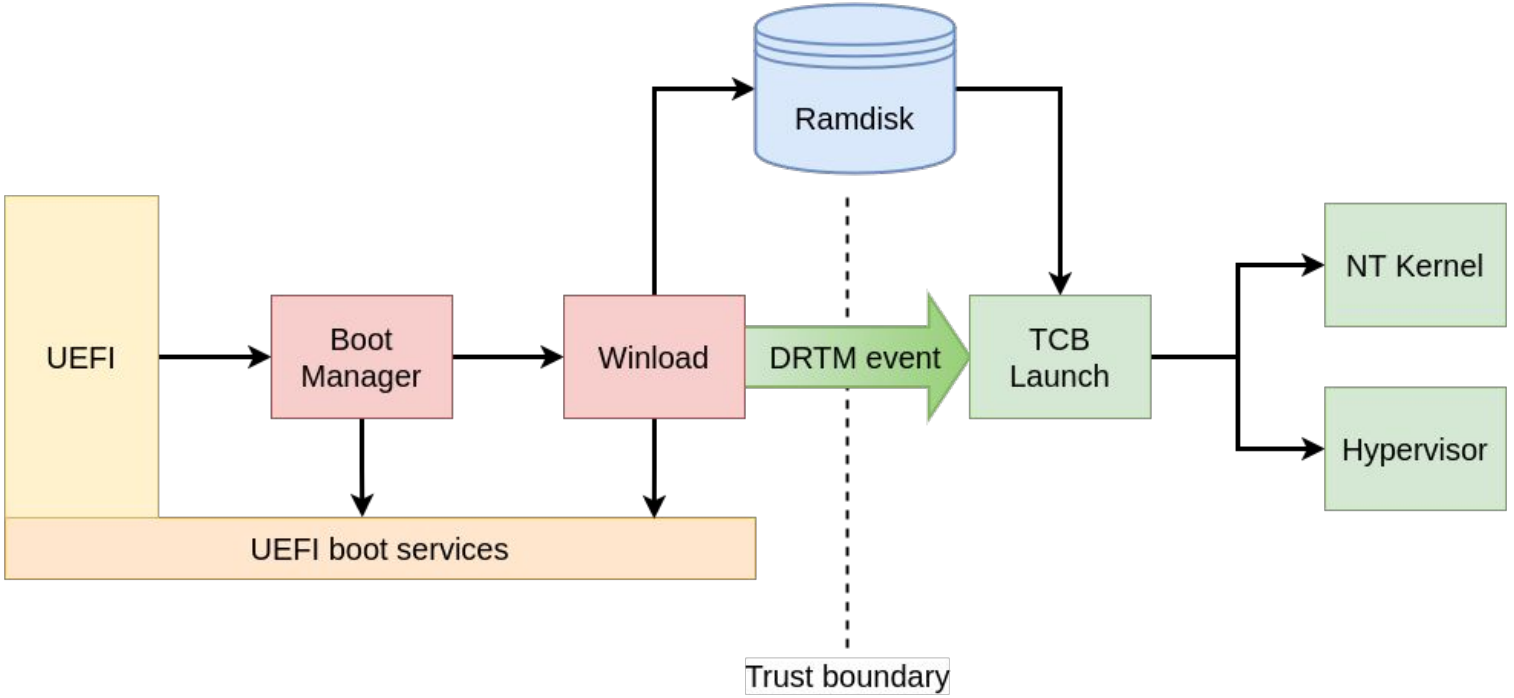


Linux Plumbers Conference
Dublin, Ireland September 12-14, 2022

Launch Flow for Existing Secure Launch



Providing Context: Microsoft Approach



UEFI and Dynamic Launch Compatibility

A challenge enabling full supporting for UEFI environments is that Linux requires the use of the efi-stub, a peer entrypoint to sl-stub for the Linux kernel.

- This requires cooperation and collaboration on how to handle the transition.
  - The CPU dynamic launch instruction cannot be invoked until after Exit Boot Services is called.
  - Efi-stub requires being ran before as well as being the entity that calls Exit Boot Services.
- Ultimately an approach was found that meet the requirements of both projects
  - Efi-stub developers proposed a hook function that could be called when efi-stub was finished.

Going Back to the Drawing Board

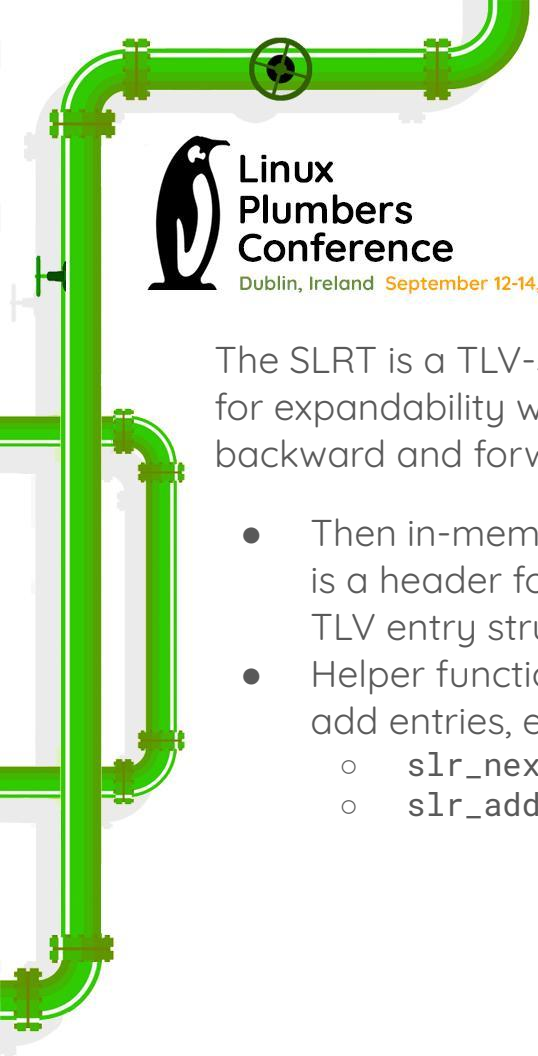
With the agreement that a hook would be used as a means for efi-stub to handoff for the dynamic launch to occur, now it must be determined how to make it all work.

- Several questions had to be answered,
  - What is responsible for providing the hook; where will the hook be located; how will it be located?
  - The hook will require some runtime information; what needs to provided; where will it be written; how will it be structured?
  - Information needs to be passed across the dynamic launch; where will this information be written; how will it be structured?
  - This shouldn't be Linux specific, how can the solution be designed such that it is kernel agnostic?

The dl-stub and Secure Launch Resource Table

The questions lead to the understanding that there must be a container to store information and a locatable code object that must be persistent independent of the system software.

- The result is the dl-stub and the Secure Launch Resource Table (SLRT)
  - The dl-stub will be a static binary that contains the minimal logic to quiescent the system and trigger the dynamic launch
  - The SLRT will hold all the relevant information,
    - Location of the dl-stub
    - Dynamic Launch meta-data
  - The SLRT and the dl-stub will be installed by the boot manager/bootloader.
    - On UEFI systems, the SLRT will be stored under a well-known GUID in the configuration table

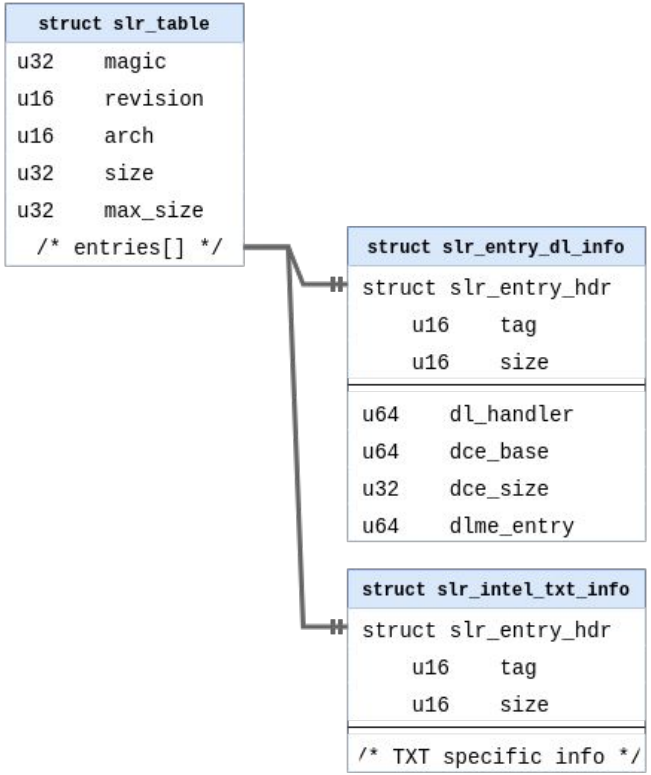


Linux Plumbers Conference
Dublin, Ireland September 12-14, 2022

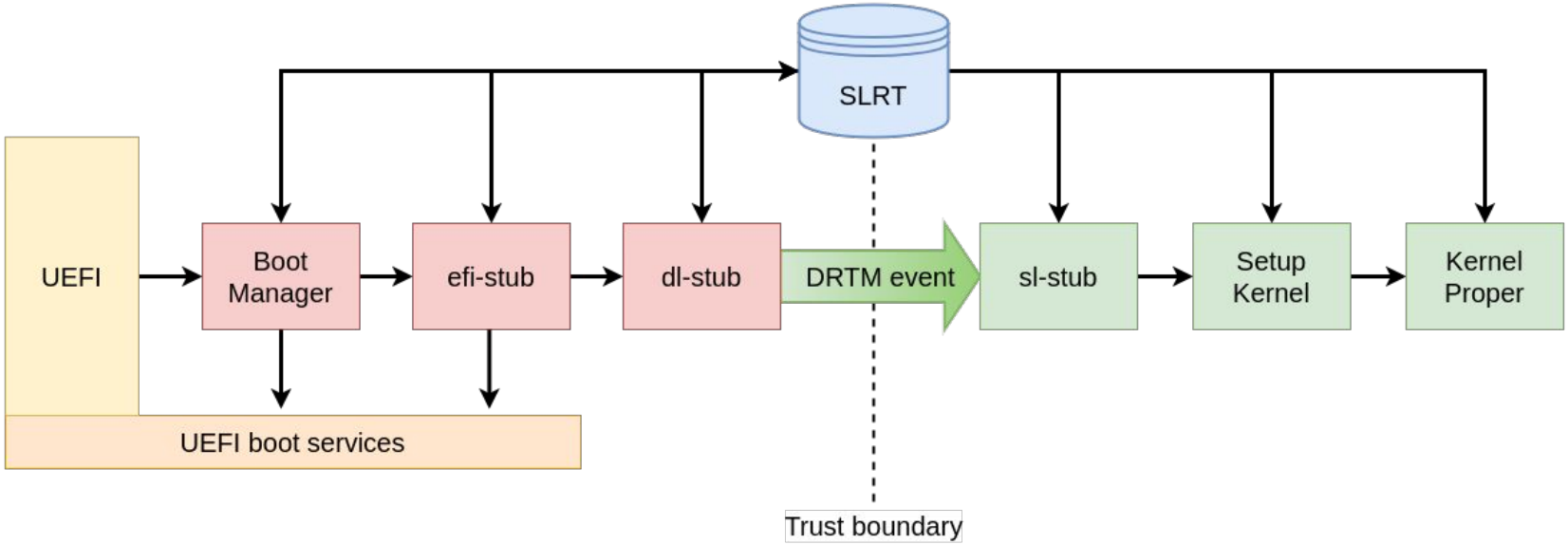
Details for the SLRT

The SLRT is a TLV-style table designed for expandability while enabling backward and forward compatibility.

- Then in-memory representation is a header followed one or more TLV entry structures.
- Helper functions to search and add entries, eg.
  - `slr_next_entry_by_tag()`
  - `slr_add_entry()`



Launch Flow for dl-stub



DRTM Late Launch

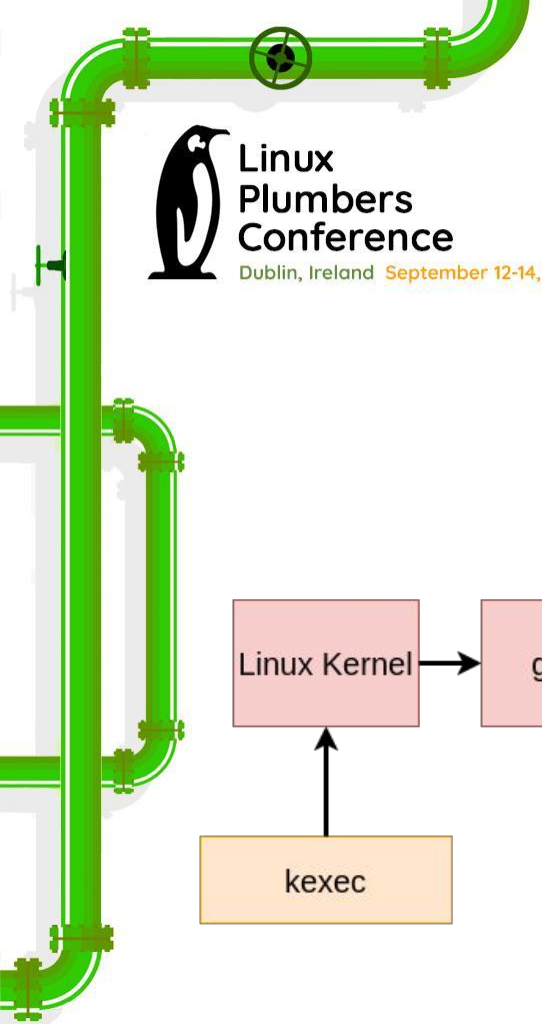
A Dynamic Launch can be done at any time during system lifecycle.

- There are two classifications for a Dynamic Launch.
  - Early Launch: when Dynamic Launch is used in conjunction with system firmware launch.
  - Late Launch: when Dynamic Launch is used by an Operating System to re-establish trust.
- Late launch is a unique and powerful feature of DRTM solutions.
  - At an arbitrary point in time a system can prepare for and initiate the Dynamic Launch Event.
  - This re-establishes the DRTM measurement and marks a point in time where the system is in a known good state.
- This process can be done any number of times driven by system policy.
- Note that a late launch is not a power cycle so certain state and configuration information can be saved across a late launch (e.g. paused VMs).
- TrenchBoot late launch for Linux is Secure ReLaunch.

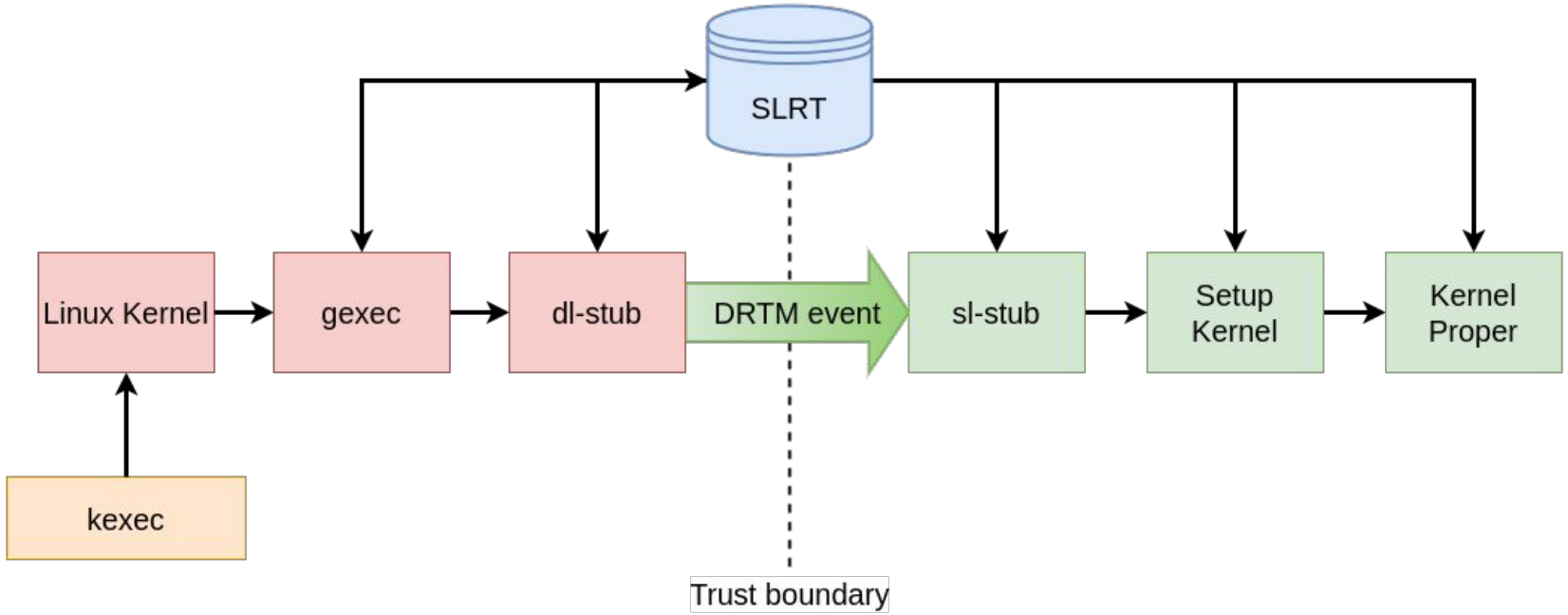
Secure ReLaunch Design

The goal is to introduce a late-launch capability for Linux with minimal changes through reuse.

- GRUB already contains the support for setting up and initiating the Dynamic Launch Event.
- A new platform will be added to GRUB called “gexec”.
- When configured for this platform, building GRUB will produce an ELF binary image which can be executed via kexec.
- A separate entry point will exist for gexec to capture information passed to the kexec’ed image (e.g. boot params on x86).
- The existing dynamic launch code with some modification will perform the relaunch.
- There will be no changes to kexec or the Linux kernel to support this.



Secure ReLaunch Flow



Deployment

The deployment of Secure ReLaunch will only require a build of GRUB with support for the new gexec platform.

- Build the gexec platform and build a ReLaunch bootable image:

```
$ ./configure --with-platform=gexec --target=x86_64
$ make && sudo make install
$ ./grub-mkimage -O i386-gexec -o gexec.img -p . -c relaunch.cfg
```

- The ReLaunch GRUB config file:

```
slaunch
linux /vmlinuz nokaslr intel_iommu=strict console=tty0 console=ttyS0,115200n8
initrd /initramfs.gz
```

Roadmap

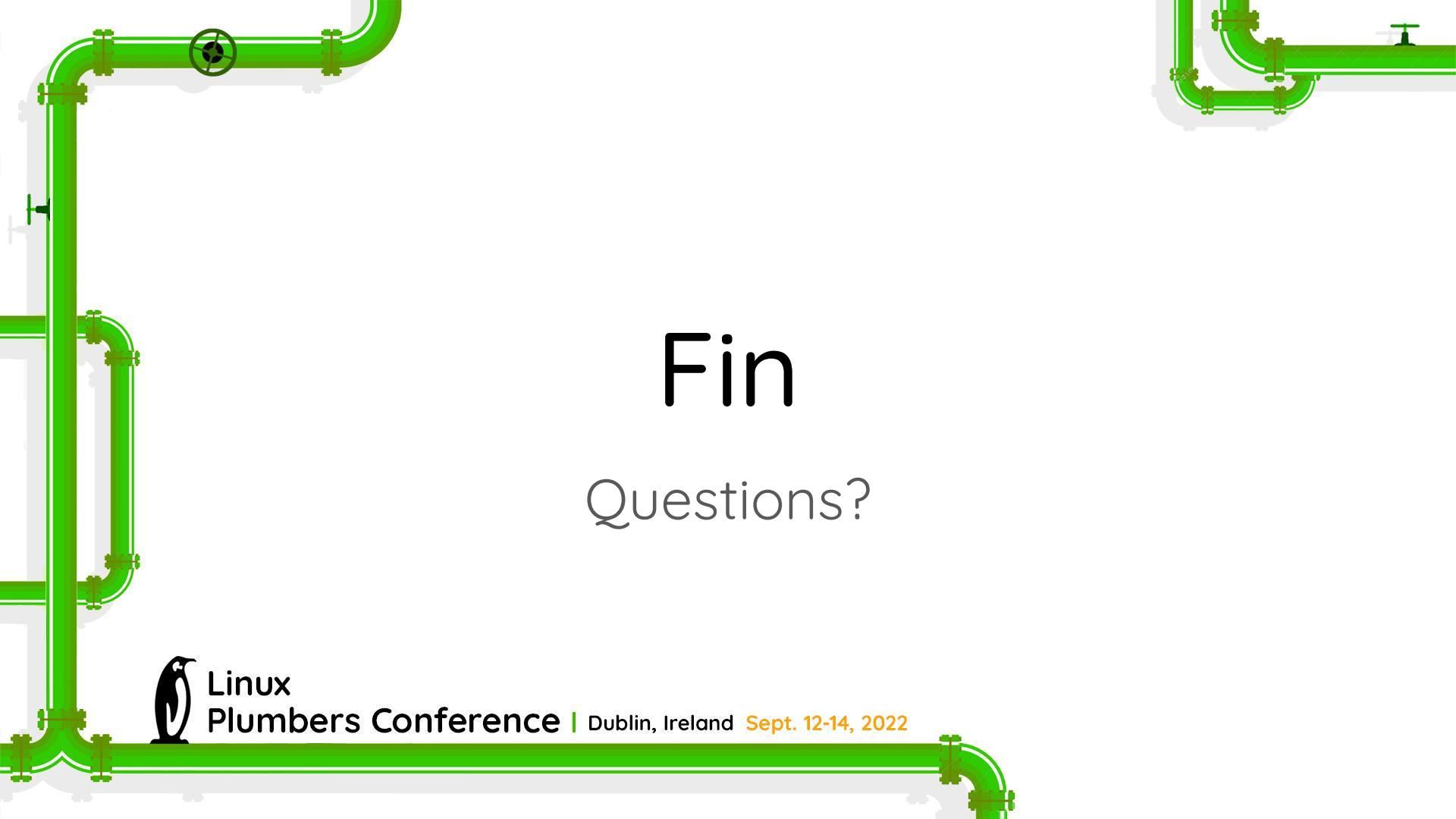
- Next planned Secure Launch kernel series (v6) will contain:
  - DL Stub support to allow direct EFI boot of kernel with Secure Launch
  - Secure Launch Resource Table (SLRT) definition and support code
- Initial GRUB patches need to be posted to the TrenchBoot project
- GRUB DL Stub support follows that coincides with Secure Launch kernel v6 series
- GRUB relaunch support/gexec will come after
- Future platform support:
  - AMD Secure Launch Support
  - ARM Secure Launch Support

Want to Contribute?

If you are excited to work on projects like TrenchBoot, Oracle Linux and Virtualization team is actively hiring. Look out for informational brochures around the conference center. You can also checkout the current openings:

Opening Listing: https://www.oracle.com/corporate/careers/
Job ids:

180033
180333



Fin

Questions?