LVC21-220: Generic image approach and LEDGE RP

Maxim Uvarov <maxim.uvarov@linaro.org>
What is LEDGE RP?

Linaro’s LEDGE RP is considered to be a reference platform for supporting industry standards for Linux operating system. Primary purpose is to build an operating system for IoT and EDGE devices observing specifications like UEFI, Secure Boot, EBBR, ACPI and etc.

- Uboot version XXX
- Or EDK2 Version XXX
- Optee version ZZZ
- Kernel version YYY
- Apps
- Apps
- Apps

TFA, OP-TEE, U-Boot - Trusted Substrate project

Linux kernel, userland - LEDGE RP project (Independence of Firmware)
Why do we do LEDGE RP?

- Open Source development environment
- Single image for ARM, ARM64, x86-64.
Things we are working on are..  
- Uboot loader UEFI support  
- QEMU  
- ARM Trusted Firmware-A  
- OP-TEE  
- Linux kernel  
- Security (fTPM, Selinux, IMA, Parsec etc)
Platforms

- QEMU ARM (machine=virt,secure=on)
- QEMU ARM64 (machine=virt,secure=on)
- Stm32mp157c-dk2 (32bit ARM)
- Synquacer (64bit ARM)
- Ti-am572x (32bit ARM)
- QEMU x86_64
Internals: Images (wic, certs, bios)

**Rootfs**: ledge-gateway-lava-ledge-qemuarm64*.rootfs.wic.gz

**Firmware**: Firmware.uefi-edk2.bin/EFI_VARS.bin and firmware.uefi.uboot.bin/Sec_vars.bin

**Certificates**: ledge-kernel-uefi-certs.ext4.img
WIC image

- ESP (vfat)
- Rootfs (ext4)

- ./dtb/
- ./EFI/BOOT/bootXXX.efi
- ./ledge-initramfs.rootfs.cpio.gz

- Boot0000
- root=UUID=6091b3a4-ce08-3020-93a6-f755a22ef03b console=ttyS0,115200

- initrd ledge-initramfs.rootfs.cpio.gz
- bootXXX.efi
Firmware

- EDK2/Tianocore
- Trusted firmware A FIP image (BL1, BL32 (optee), BL33(ubot))

```bash
qemu-system-aarch64 -cpu cortex-a57 -machine virt,secure=on -pflash firmware.uefi.uboot.bin -pflash uefi_vars.bin
```
Try to always upstream and use mainline versions of applications.
Images

**Ledge-iot** - the image for IoT devices. Package list and features are aligned to Fedora IoT

**Ledge-gateway** - image to support Edge gateways
We solved problems:

- fTPM as built-in OP-TEE OS TAs
- U-Boot EFI Secure Boot
- U-Boot signed Capsule Updates
- Runtime variable support in U-Boot
- EFI variables in secure media (RPMB)
- QEMU reboot in secure mode
- Selinux labeling
- Parsec and Rust applications support on Open Embedded
Build, CI, Docs ...

**Build and CI**

- [https://ci.linaro.org/job/ledge-oe/](https://ci.linaro.org/job/ledge-oe/)

**Binaries:**

- [http://releases.linaro.org/components/ledge/](http://releases.linaro.org/components/ledge/)

**Dev:**

- [http://snapshots.linaro.org/components/ledge/oe/](http://snapshots.linaro.org/components/ledge/oe/)

**Docs**

- [https://linaro.github.io/ledge-doc/](https://linaro.github.io/ledge-doc/)

**Source code**

- [https://github.com/Linaro/ledge-oe-manifest](https://github.com/Linaro/ledge-oe-manifest)
Security

OP-TEE OS (fTPM) → Linux Kernel (/dev/tpm0) → Parsec → App

RPMB Storage → QEMU emulation
Projects linked to LEDGE RP

Trusted Substrate - support of Embedded Base Boot Requirements

- UEFI Secure Boot
- UEFI Measured Boot
- UEFI random number generation
- UEFI update capsules
We are working on:

LEDGE RP
Stage 5

- A/B partition testing in LAVA
- RO partitioned rootfs (NAND friendly)
- LEDGE-RP + LUKS with authorized PCR policy
- Systemd-boot blessing
- Dynamic device tree in Kernel
- IMA Secure initrd load
- Keylime attestation
- Routing protocols (BGP,..) CI
- Authenticated Containers with Parsec
- AI/ML frameworks in CI
- Keyring IMA, Parsec helper
- Low power network / Wifi gatewaying
- OEMU ARM BSA ref machine
- FDO client SDK - Parsec user Wifi automated admission

XDP for VMs investigation
A/B partition support (via systemd)
Investigate U-Boot https
How to get involved?

https://www.linaro.org/engineering/edge-and-fog-computing/

Mailing list: team-ledge@linaro.org
Thank you

Accelerating deployment in the Arm Ecosystem