SIEMENS
Ingenuity for life

Demonstrating and Testing an Embedded Hypervisor with Debian

Jan Kiszka | MiniDebConf 2019

Agenda

SIEMENS
Ingenuity for life

About /me, about the project

SIEMENS
Ingenuity for life

Jan Kiszka <jan.kiszka@siemens.com>

- Member of embedded Linux team at Siemens Corporate Technology
- (In-house) consultant, architect, developer for OSS
- Focus on kernel, real-time, virtualization, embedded build systems
- Upstream contributor

https://github.com/siemens/jailhouse[-images]

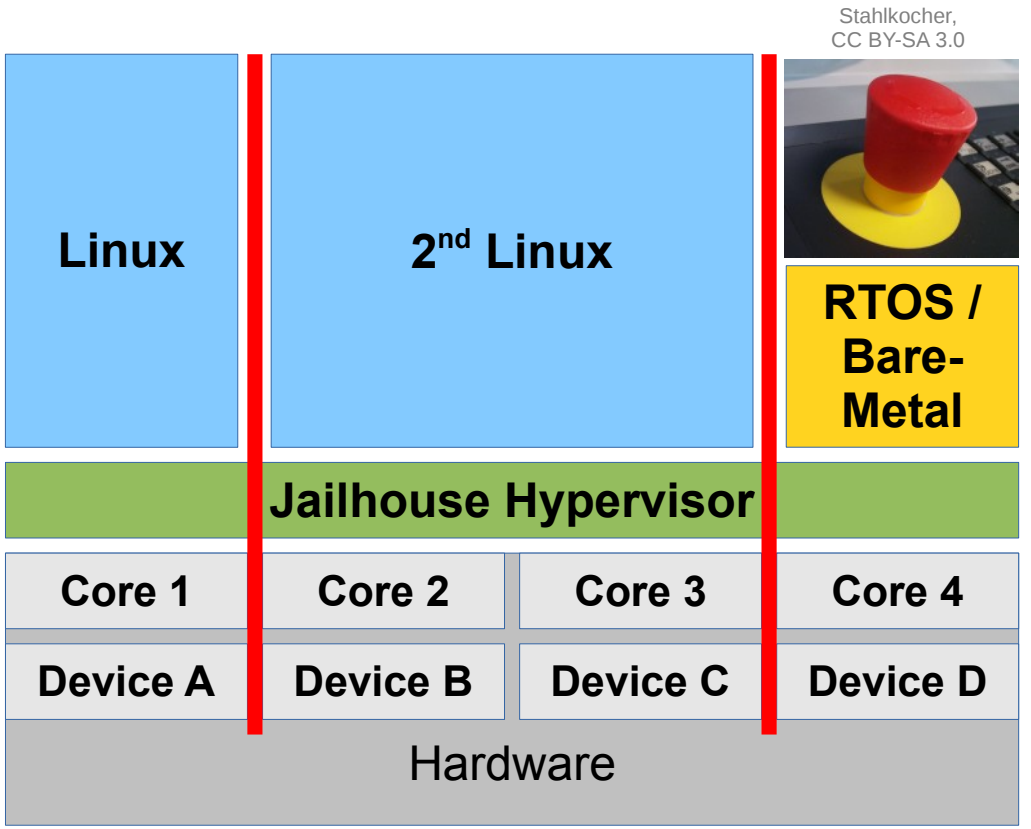
- Not a product of Siemens, rather an infrastructure component
- Started as open source project by Siemens
- Published for broader industrial usage and contributions

Jailhouse: Static Partitioning for Multicore Systems

SIEMENS
Ingenuity for life

Design Goals

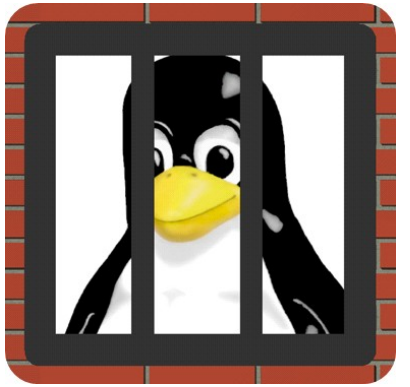
- Focus on maintaining static partitions
- No scheduling
- 1:1 resource assignment
- (Almost) no device emulation
- Keep runtime code base minimal
- Hard RT properties with minimal overhead
- Enable / simplify safety certification
- Open Source, Linux-friendly (GPLv2)
- http://jailhouse-project.org



Jailhouse Application Areas

SIEMENS
Ingenuity for life

- Machine/medical/railway control (RT, safety) besides less critical tasks (data collection, processing, monitoring, networking – Industrial IoT)
- DSP real-time workloads aside Linux (hardware simulation, software radio etc.) on high-performance CPUs
- ADAS, up to autonomous driving
- Time-sensitive networking, software switching, traffic inspection and management appliances, partitioned and co-located
- …



Jailhouse Hardware Support

SIEMENS
Ingenuity for life

x86

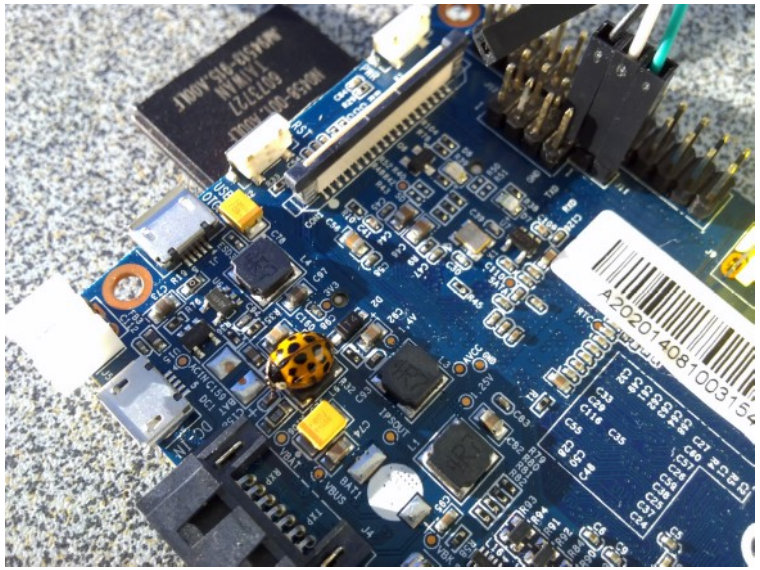
- Most Intel and AMD systems
- QEMU/KVM (via nested virtualization)

ARMv7

- Banana-Pi, Orange-Pi Zero, NVIDIA Jetson TK1, **emtrion emCON-RZ/G1x**
- Out of tree: **TI AM572x-EVM**

ARMv8

- AMD Seattle, LeMaker HiKey, Xilinx ZynqMP ZCU102, Ultra96, NVIDIA Jetson TX1/TX2, ESPRESSObin, MACCHIATObin, **NXP i.MX8MQ**
- Out of tree: **TI AM65xx EVM**
- QEMU (emulated)



Bold: backed by SoC/board vendor

Embedded means…

SIEMENS
Ingenuity for life

https://commons.wikimedia.org/wiki/File:Wooden_torture_chair_with_12_steel_blades,_China_Wellcome_L0058389.jpg

Demonstration and Test Images Needed

SIEMENS
Ingenuity for life

Requirements

- Directly bootable images for commodity boards
- UART console and some network
- Easily extensible with further packages
- Pre-configured for Jailhouse on the target
- Reproducible by users

Tasks

- Generate and pre-configure rootfs
- Build custom kernels
- Build Jailhouse against kernel
- Build custom bootloader where needed
- Generate SD-card images (for physical boards)

Use Case for Isar?

SIEMENS
Ingenuity for life

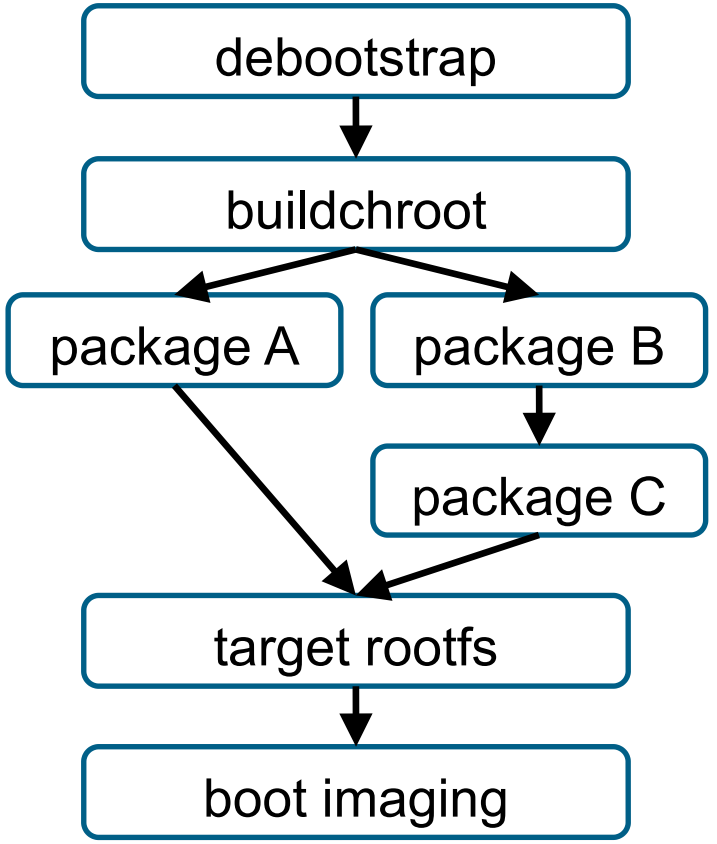
- **I**ntegration **S**ystem for **A**utomated **R**oot file-system generation
- Designed for creating embedded Debian images
  - **Binary** Debian feed
  - Yocto-like recipes, layers, configs (**bitbake**) for deviations and imaging
- Used at Siemens for an increasing number of embedded products
- Core of Mentor Embedded Linux (Debian flavor)
- https://github.com/ilbers/isar

https://en.wikipedia.org/wiki/Isar

Image Generation Sequence of Isar

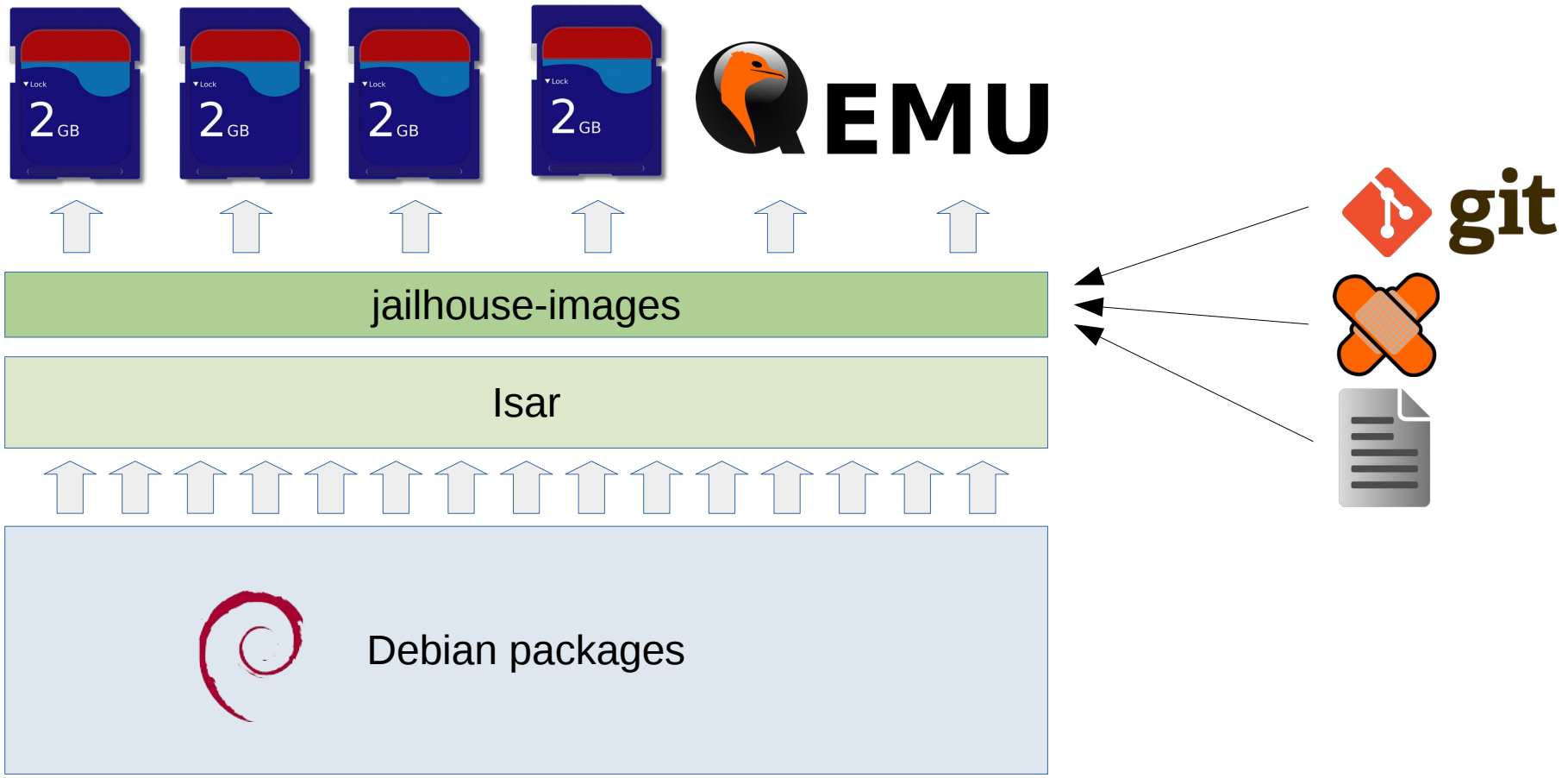
SIEMENS
Ingenuity for life

1. debootstrap Debian for target,
   also for host if cross-building
2. Create buildchroots (target and host)
3. Build custom Debian packages
   - pre-debianized packages
   - ad-hoc debianized packages
     (customizations/overlays, u-boot, …)
4. Assemble rootfs
   - debootstrap output
   - external packages
   - self-built packages
5. Run imager (typically wic)
   - Filesystem image generation
   - Partitioning
   - Bootloader installation and configuration



Layering Overview

SIEMENS
Ingenuity for life



The Birth of jailhouse-images

SIEMENS
Ingenuity for life

- Started early 2018 with x86 QEMU/KVM setup
- By now 8 targets:
  - 2 virtual
  - 6 physical
  - 3 architectures
- Prerequisite for building: docker (privileged)
- `git clone https://github.com/siemens/jailhouse-images`
- `./build-images.sh`
- `./start-qemu.sh <ARCH>`
- `dd if=target.img of=/dev/mmcblk0 …`

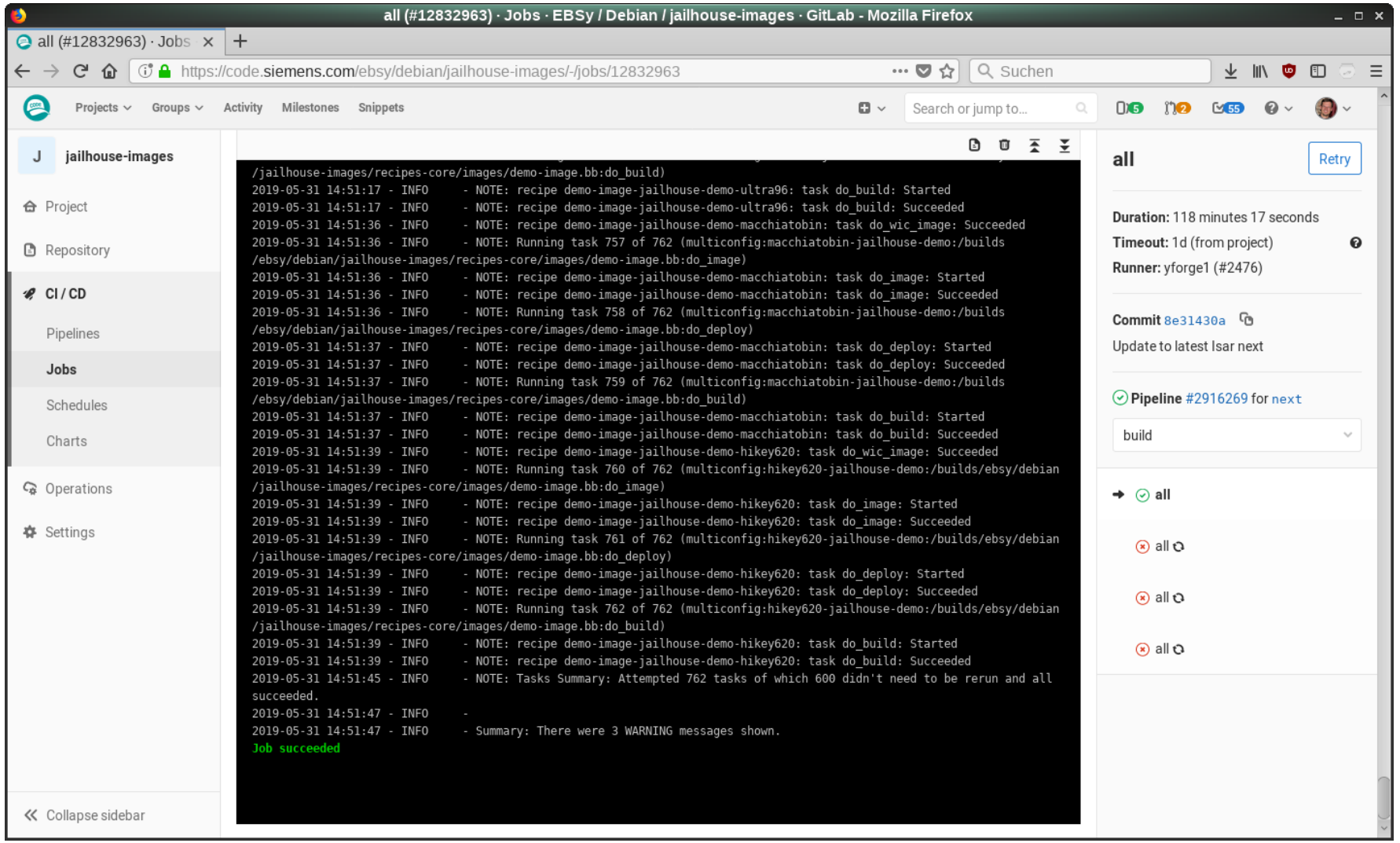
```
# ./build-images.sh
Available images demo images:
 1: QEMU/KVM Intel-x86 virtual target
 2: QEMU ARM64 virtual target
 3: Orange Pi Zero (256 MB edition)
 4: Intel NUC (NUC6CAY, 8 GB RAM)
 5: Marvell ESPRESSObin (1 GB edition)
 6: Marvell MACCHIATObin
 7: LeMaker HiKey (Kirin 620 SoC, 2 GB
edition)
 8: Avnet Ultra96
 0: all (may take hours...)

Select images to build (space-separated
index list): 1 2 4
```

SIEMENS
Ingenuity for life

Building...

Builds in gitlab-ci (custom runners)

SIEMENS
Ingenuity for life



Under The Hood

SIEMENS
Ingenuity for life

- jailhouse-images layer contains
  - 2 Jailhouse-enabled kernels (regular and PREEMPT-RT)
  - 2 Jailhouse packages (release and latest git)
  - 2 custom U-Boot versions
  - Buildroot-based initramfs for 2nd Linux cell
  - Image customizations (package list, settings, …)
  - 1 out-of-tree WIFI driver
  - Partitionings of target images (wic-based)
- Configuration management and (Debian-based) docker build image: kas https://github.com/siemens/kas
- Parallel build of independent tasks => 8 images in ~50 min.

```
# ./build-images.sh -all -latest
2019-06-07 15:15:00 - INFO     - kas 1.0 started
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - Using /jailhouse-images as root for repository
jailhouse
2019-06-07 15:15:00 - INFO     - /work/isar$ git cat-file -t
59c7dd2b8b3172d53de1c7a39fbd49751193559a
2019-06-07 15:15:00 - INFO     - Repository isar already contains
59c7dd2b8b3172d53de1c7a39fbd49751193559a as commit
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - Using /jailhouse-images as root for repository
jailhouse
2019-06-07 15:15:00 - INFO     - /work/isar$ git status -s
2019-06-07 15:15:00 - INFO     - /work/isar$ git rev-parse --verify HEAD
2019-06-07 15:15:00 - INFO     - 59c7dd2b8b3172d53de1c7a39fbd49751193559a
2019-06-07 15:15:00 - INFO     - Repo isar has already been checked out with correct
refspec. Nothing to do.
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - Using /jailhouse-images as root for repository
jailhouse
2019-06-07 15:15:00 - INFO     - /work/isar$ /tmp/tmpuhlsj1q9 /work/build
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - Using /jailhouse-images as root for repository
jailhouse
2019-06-07 15:15:00 - INFO     - /jailhouse-images$ git rev-parse --show-toplevel
2019-06-07 15:15:00 - INFO     - Using /jailhouse-images as root for repository
jailhouse
2019-06-07 15:15:00 - INFO     - /work/build$ /work/isar/bitbake/bin/bitbake -k -c
multiconfig:qemu-amd64-jailhouse-demo:demo-image multiconfig:qemu-arm64-jailhouse-
demo:demo-image multiconfig:orangepi-zero-jailhouse-demo:demo-image multiconfig:nuc6cay-
jailhouse-demo:demo-image multiconfig:espressobin-jailhouse-demo:demo-image
multiconfig:macchiatobin-jailhouse-demo:demo-image multiconfig:hikey620-jailhouse-
demo:demo-image multiconfig:ultra96-jailhouse-demo:demo-image
Parsing recipes: 100% ...
```

Example: The Jailhouse Recipe

SIEMENS
Ingenuity for life

- Performs “ad-hoc” Debian packaging
- All-in-one package (ugly / pragmatic)
  - Out-of-tree kernel module
  - Hypervisor binary
  - Command line tool
  - Python module
  - VM partition (cell) configurations
  - Inmates (bare-metal cell executables)
- Debianization (debian/, including templates) carried by recipe
- Latest stable Jailhouse + few patches
- Latest next branch from git

```
recipes-jailhouse/
└── jailhouse
    ├── files
    │   ├── 0001-configs-Add-support-for-
MACCHIATObin-board.patch
    │   ├── 0001-configs-Add-Ultra96-board.patch
    │   ├── 0001-configs-orangepi0-Permit-access-
to-another-clock.patch
    │   ├── debian
    │   │   ├── changelog
    │   │   ├── compat
    │   │   ├── control
    │   │   ├── jailhouse.install
    │   │   ├── postinst
    │   │   └── rules
    │   ├── linux-nuc6cay-demo.c
    │   └── nuc6cay.c
    ├── jailhouse_0.10.bb
    ├── jailhouse.inc
    └── jailhouse_latest.bb
```

Another Recipe Example: Custom Kernels

SIEMENS
Ingenuity for life

- Relies on linux-custom.inc by Isar
- Builds drop-in replacement for Debian kernel
- Internally uses upstream `make deb-pkg` plus some post-processing
- Actual recipe reduced to the minimum
  - Specify kernel source
  - Add patches if needed
  - Specify kernel config
  - $Profit

```
recipes-kernel/linux/
├── files
│   ├── 0001-arm64-dts-zcu100-revC-Give-wifi-some-
time-after-powe.patch
│   ├── 0001-ARM-dts-orange-pi-zero-Adjust-wifi-
settings.patch
│   ├── 0001-pwrseq_simple-Workaround-for-missing-
device-tree-ent.patch
│   ├── amd64_defconfig_4.19
│   ├── arm64_defconfig_4.19
│   ├── orangepi-zero_defconfig_4.19
│   └── preempt-rt.cfg
├── linux-jailhouse_4.19.46.bb
├── linux-jailhouse_4.19.inc
└── linux-jailhouse-rt_4.19.37-rt20.bb
```

And The Ugly: Bootloaders

SIEMENS
Ingenuity for life

- Life can be simple:
  `IMAGER_INSTALL += "u-boot-sunxi"`
  - Uses Debian upstream U-Boot package
  - ...but from buster
- Life can be hard:
  `u-boot-macchiatobin_2018.09-atf1.6.bb`
  - 5 different artifact sources
  - Patches
  - Board-specific build steps
- Life can be even harder:
  `u-boot-ultra96_2019.01-atf1.6.bb`
  - /me too lazy to reproduce 3 artifacts:
    pulled from OpenSuse contrib rpms

```
recipes-bsp/
├── arm-trusted-firmware
│   ├── arm-trusted-firmware_1.6.inc
│   └── files
│       └── 0001-tools-Fix-broken-object-
compilation-rules.patch
└── u-boot
    ├── files
    │   ├── 0001-zynqmp-Downgrade-to-PMUFW-
0.3.patch
    │   ├── macchiatobin-rules
    │   ├── ultra96.bif
    │   └── ultra96-rules
    ├── u-boot-macchiatobin_2018.09-atf1.6.bb
    └── u-boot-ultra96_2019.01-atf1.6.bb
```

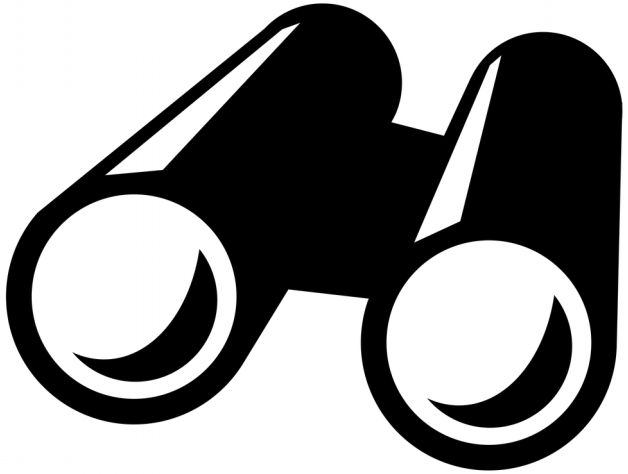
SIEMENS
Ingenuity for life

Booting...

Project Outlook

SIEMENS
Ingenuity for life

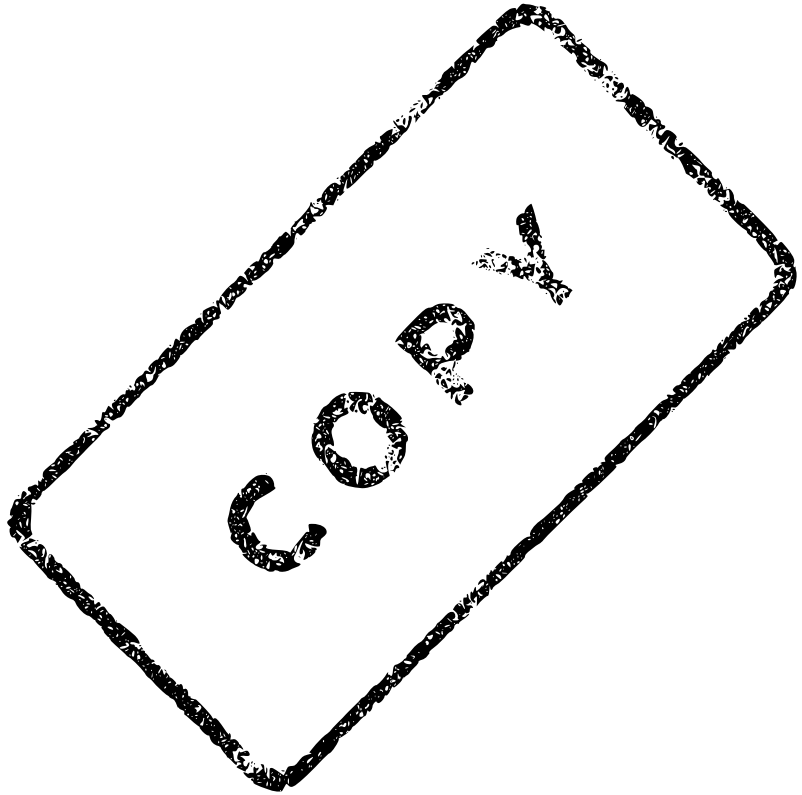
- Update (of course)
- More boards
  - TI AM65xx-based target
  - Xilinx ZCU102 (Ultrascale+ reference board)
  - Something iMX8-based?
- Automated testing, including on-device (LAVA)
- Integration with Isar QA?



Some Concept Copies

SIEMENS
Ingenuity for life

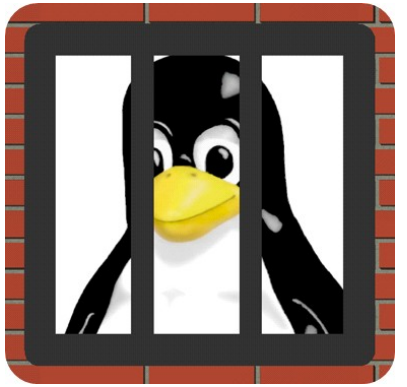
- Xenomai: Real-time Linux framework
  - Requires specific kernels with patches
  - May require board-specific tuning and/or patches
  - Currently lacks Debian upstream development
  - https://gitlab.denx.de/Xenomai/xenomai-images
- Civil Infrastructure Platform LTS system
  - https://gitlab.com/cip-project/cip-core/isar-cip-core
  - Still in early stages
  - Currently working on LAVA integration & CI build



Summary

SIEMENS
Ingenuity for life

- Embedded system building remains complex
  - Non-upstream changes
  - Adjustments to standard packages
  - Unpackaged components
  - Full, preconfigured images needed
- jailhouse-images: close-to-real-life use case
  - Builds directly usable images
  - Addresses many common integration issues
  - Exploits Isar build system
  - Benefits a lot from being Debian-based
- More users of this pattern to come, OSS and products
- Feedback welcome, including alternative modeling suggestions!



Thank You! Any Questions?

SIEMENS
Ingenuity for life

Jan Kiszka
Siemens Corporate Technology

E-mail:
jan.kiszka@siemens.com

siemens.com