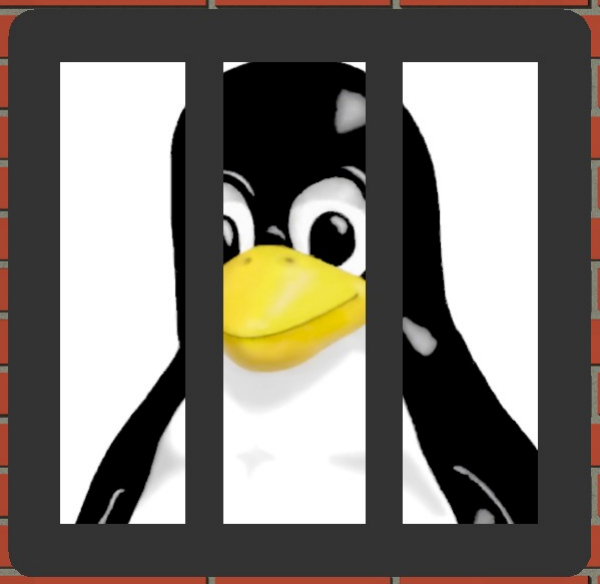
SIEMENS



Siemens Corporate Technology | October 2015

Data Plane Isolation via the Jailhouse Hypervisor

With material by Huawei European Research Center Munich

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Data Plane Isolation via the Jailhouse Hypervisor

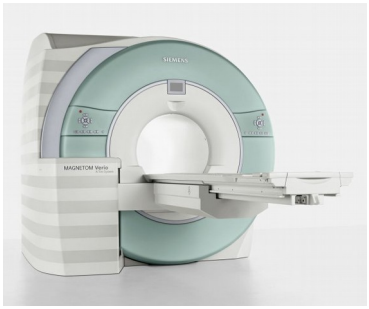
Agenda

Motivation
Jailhouse introduction & philosophy
Current status
Running multiple Linux instances (with demo)
Data plane isolation architectures
Summary

Unrestricted © Siemens AG 2015. All rights reserved

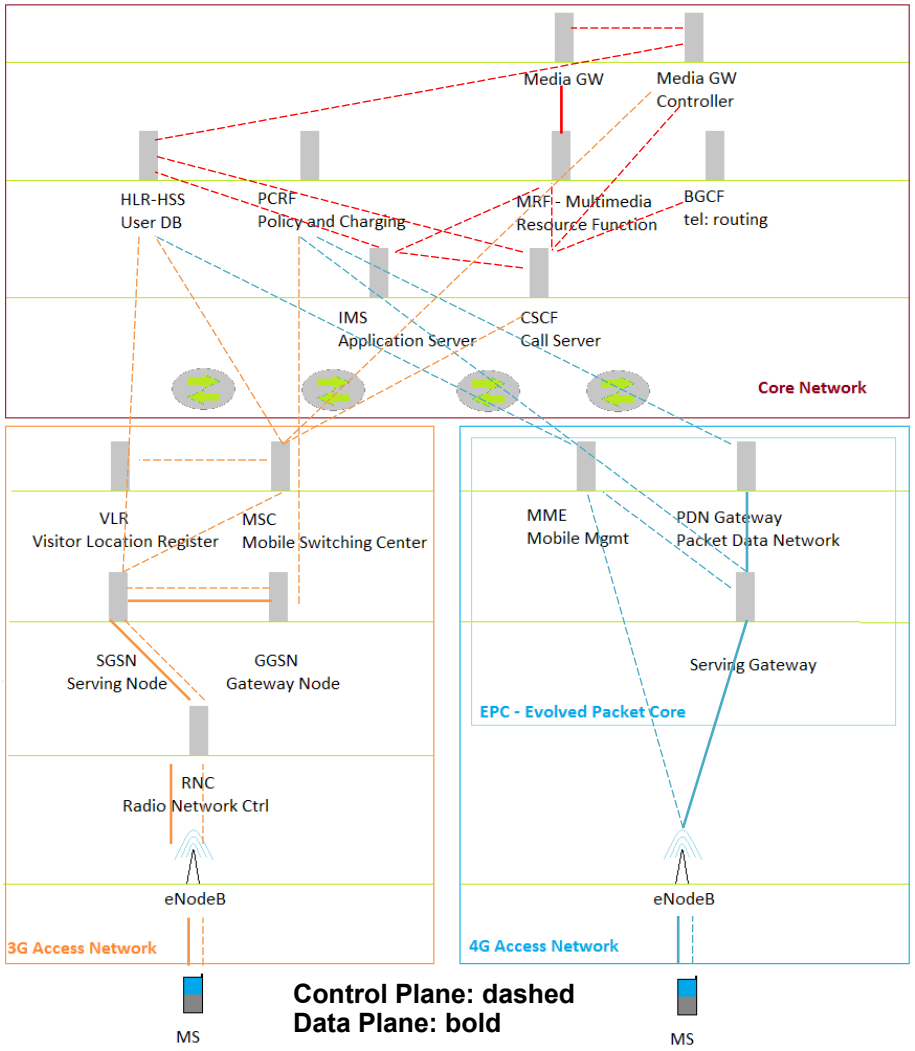
Original Drivers of Jailhouse

- **Low latency & high throughput**
- **Hard real-time**
- **Preexisting software**
- **Mixed criticality**



Unrestricted © Siemens AG 2015. All rights reserved

3G & 4G Telecom Network



Network Implementation follows industry standards (ITU-T, 3gpp, ETSI, …) which define the Network Elements and Network Interfaces which enable the Services.
Latest revisions of the specifications identify Network Functions and Reference Points, in order to better decouple the functions from the physical nodes necessary to provide them.

Specifications also set performance requirements for the Control Plane (signaling) and Data Plane (traffic) interfaces, according to the service needs of the evolving network and the emergence of innovative applications.

Reference: 3gpp 23.003
“Network Architecture” Release 12-13

Contributed by Huawei ERC Munich

Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Use Cases for the Telecom Domain

Telecom Network standardization sets requirements for the Control Plane and Data Plane interfaces, according to the needs of the evolving network and the emergence of innovative applications.

Application diversity

- *Current 3G/4G data plane must service different kinds of applications with very different performance requirements e.g.*
  - *Telecom control plane signaling*
  - *Web applications*
  - *Video stream*
  - *Voice*
  - *Games*

4G minimum service requirements:

- **D-Plane latency < 10ms for both downlink and uplink.**
- **Throughput potential 100Mpbs**

5G

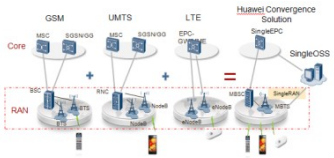
- New class of applications requiring 5G changes traffic patterns and performance requirements e.g:
  - Cloud storage: high uplink traffic
  - 3D games: high throughput and low latency
  - Tactile Internet: low latency
  - Augmented reality: low latency and high instant data volumes
  - Self driving car: high throughput, low latencies
  - IoT: low latency

- **By 2020:**
- **5x to 10x reduction in latency**
- **100 x throughput**

Heteregeneous Network Elements

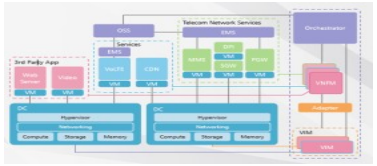
Network elements has different characteristics e.g.:

- Evolved Packet Core:
  - Gateway to mobile network
  - High RAS requirements
  - High volume of signaling and data traffic
  - Performance requirements derived from applications
- Base Stations:
  - Communicates with mobiles over the air interface
  - Lower throughput but very strict RT requirements on air interface



Cloudification

- Cloudification is moving telecom elements on COTS hardware and on top virtualization technologies
- Software-defined networking (SDN) can decouple control and data planes



- Programmability increases flexibility, speed and innovation but:
- **Increased SW layers like Virtualization should not affect too much on application performance**

Increased Throughput, reduced latency

Increased complexity

Contributed by Huawei ERC Munich

Unrestricted © Siemens AG 2015. All rights reserved

Requirements from Telecom Domain

- Consolidation of (Linux) services
- Enabling of open platforms
- Low latency & high throughput
- Avoid SMP scalability bottlenecks
- Strict quality of service needs

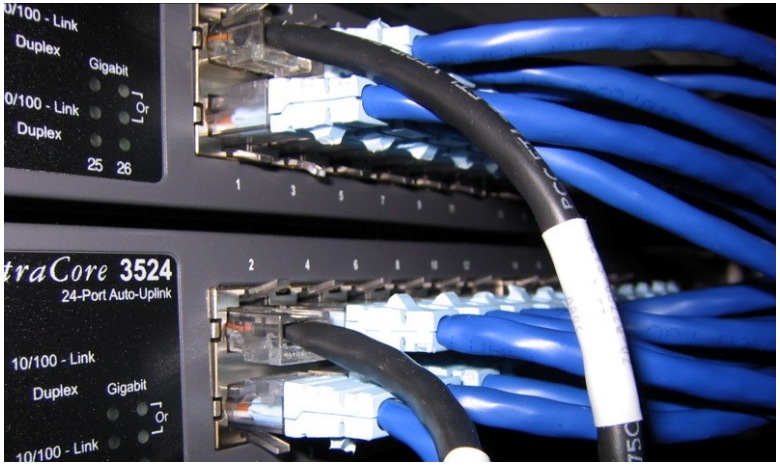


Image: Ben Stanfield, licensed under CC BY-SA 2.0

Unrestricted © Siemens AG 2015. All rights reserved

Linux to serve them all?

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Data Plane Runtime Environments

Linux

- Typically user-land network stacks
- DPDK (Data Plane Development Kit)
- ...

Classic RTOS

- Legacy stacks
- Minimal latency designs

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Linux Options Supporting Data Planes

Available features

- PREEMPT-RT
- isolcpus
- nohz_full
- KVM

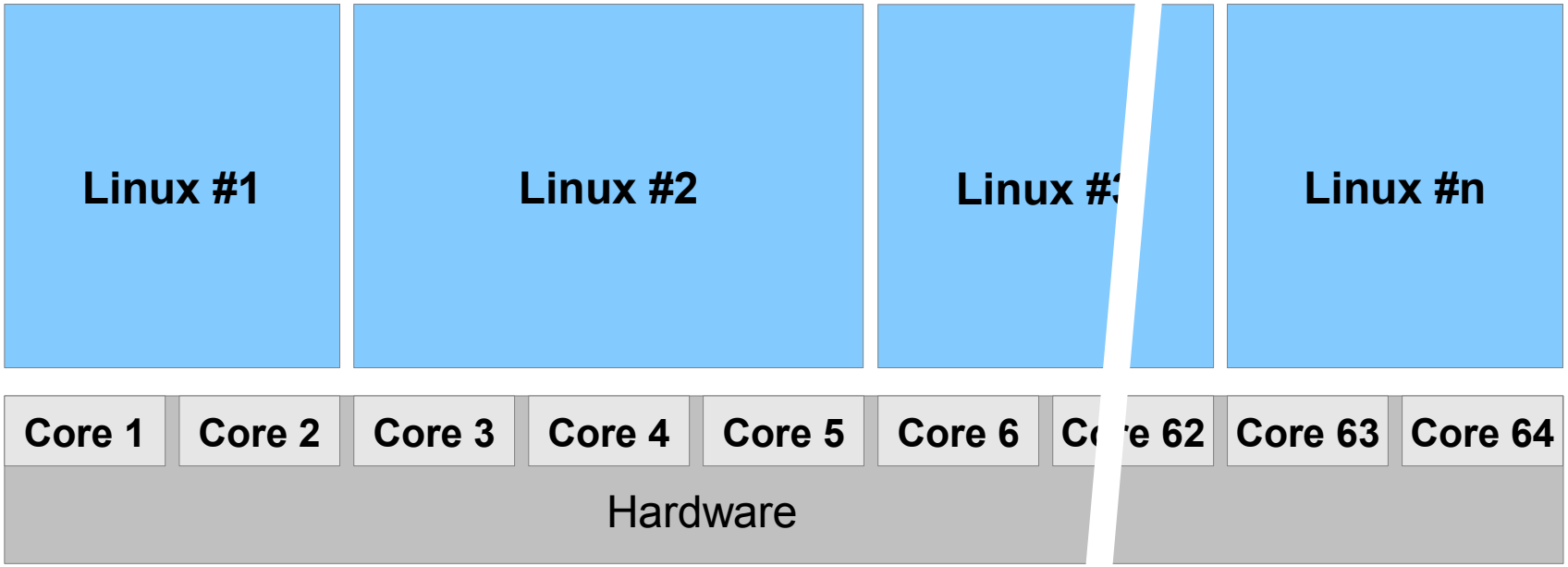
Imperfect CPU isolation

- 1 tick per second
- Latency side effects of
  - CPU hotplug
  - Module loading/unloading
  - TLB shoot-downs
  - ...

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Hard Partitioning of Linux?



Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Isolation via the Jailhouse Hypervisor

Agenda

- Motivation
- **Jailhouse introduction & philosophy**
- Current status
- Running multiple Linux instances (with demo)
- Data plane isolation architectures
- Summary

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

What is Jailhouse?

A tool to run

… real-time and/or safety tasks

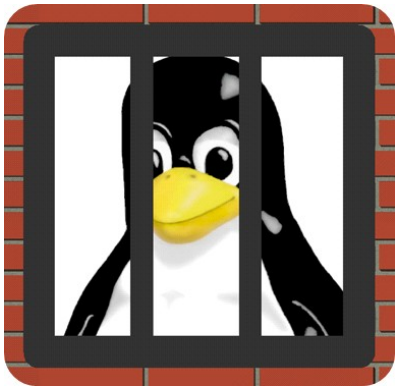
… on multicore platforms (AMP)

… aside Linux

It provides

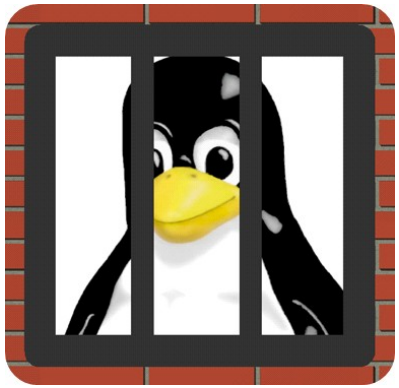
- **strong & clean isolation**
- **bare-metal-like performance & latencies**
- **no reason to modify Linux** (well, almost)

… and it's open source (GPLv2)



Unrestricted © Siemens AG 2015. All rights reserved

What is it not?



!=

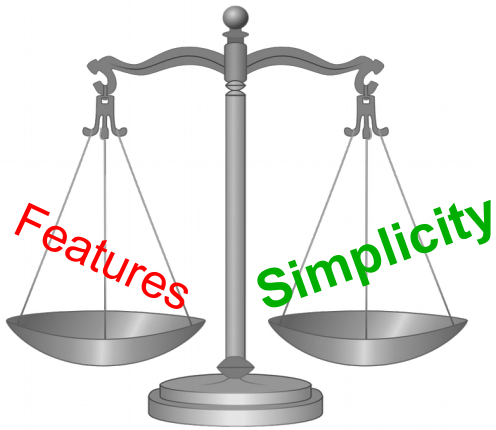
...

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

What makes Jailhouse different?

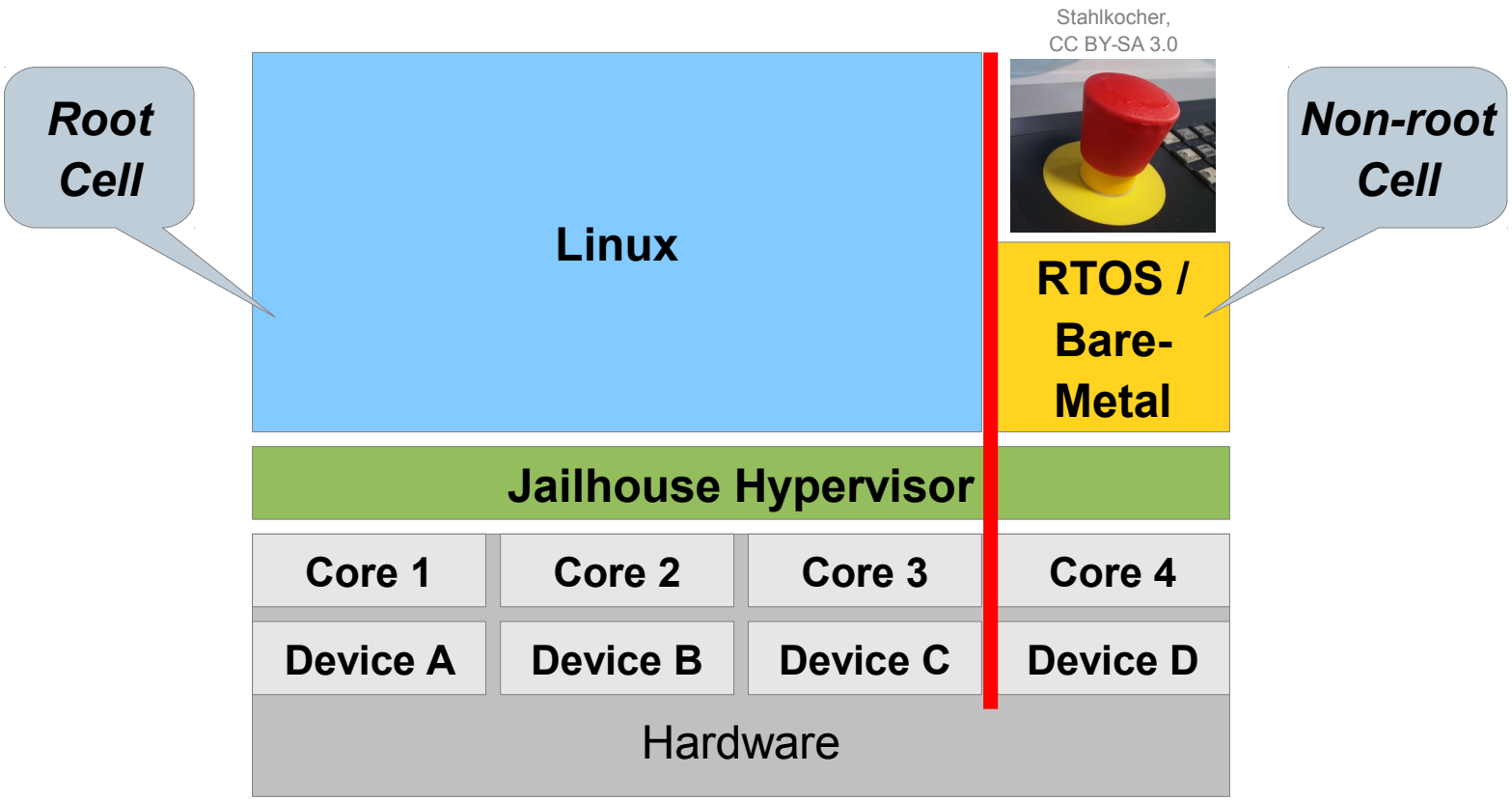
- **Use virtualization for isolation – *ok, nothing new***
- **Prefer simplicity over features**
  - Resource access control
    instead of resource virtualization
  - 1:1 resource assignment
    instead of scheduling
  - Partition booted system
    instead of booting Linux
  - Do not hide existence of Jailhouse
- **Offload work to Linux**
  - System boot
  - Jailhouse and partition (“cell”) loading & starting
  - Control and monitoring



Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

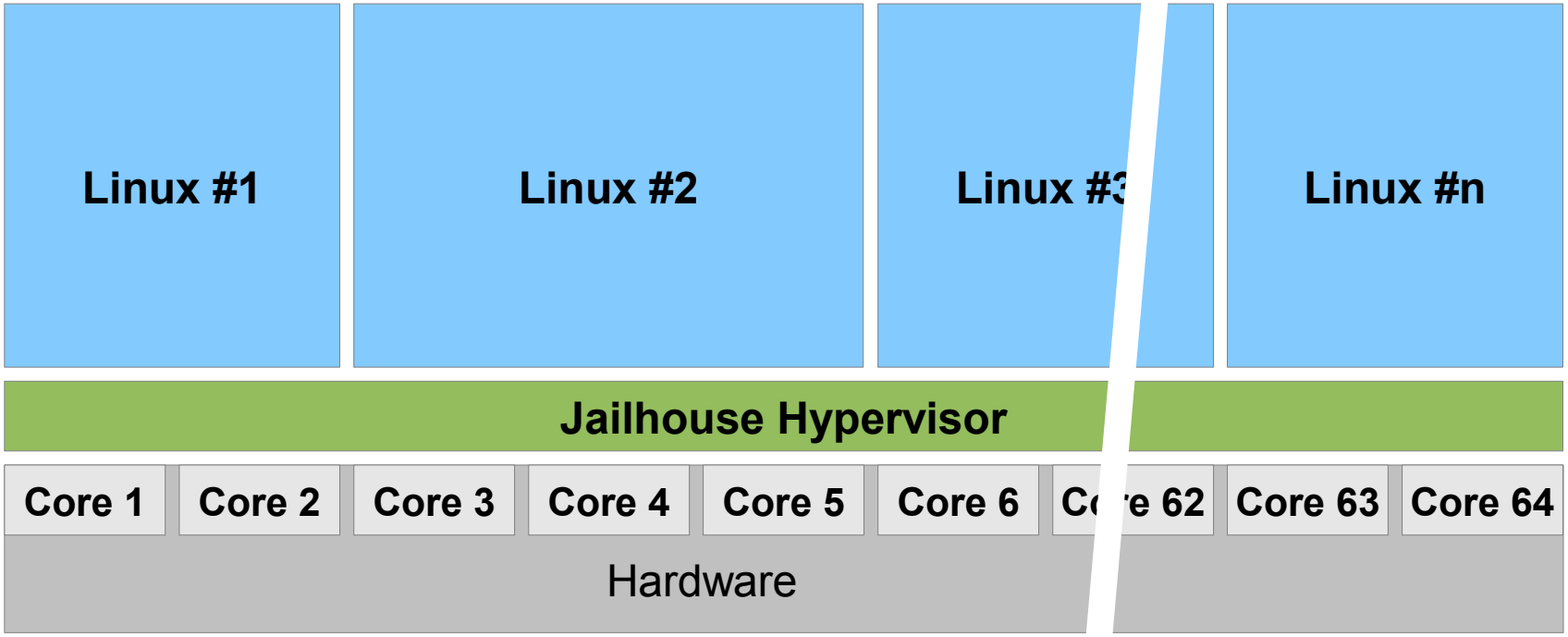
Asymmetric Multiprocessing with Jailhouse



Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Hard Partitioning of Linux



Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Isolation via the Jailhouse Hypervisor

Agenda

- Motivation
- Jailhouse introduction & philosophy
- **Current status**
- Running multiple Linux instances (with demo)
- Data plane isolation architectures
- Summary

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Jailhouse Status – x86

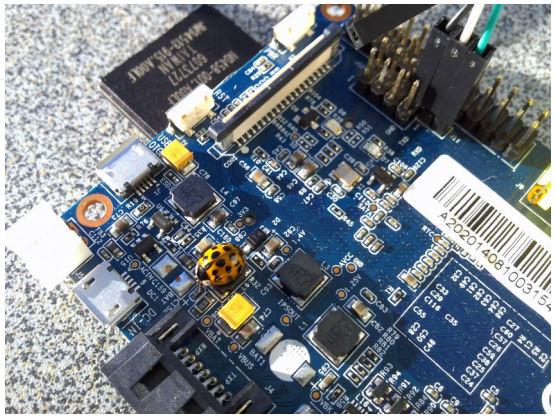
- **Initial focus on x86, first Intel, then AMD**
  - Requirement: VT-x / VT-d, AMD-V
  - AMD IOMMU support under review/testing
- **It's small!**
  - Currently ~8.5K lines of code (for Intel)
- **Direct interrupt delivery**
  - Zero VM exits, minimal latencies feasible
  - Max. timer IRQ latency (Xeon D-1540): **<1 µs**
- **Cache Allocation Technology**
  - Intel feature for partitioning L3 cache
  - Code ready to be merged, ongoing validation

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Jailhouse Status – ARM

- **ARMv7**
  - Runs in FastModel, on Banana-Pi, NVIDIA Jetson TK1
  - SMMU on to-do list
- **It's small too!**
  - Currently ~6.5k lines of code (for TK1)



- **ARMv8**
  - Work in progress by Huawei (ERC Munich)
  - Second revision posted, under review
  - Targets: ARMv8 Foundation Model, HiKey, ...

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Jailhouse Enhancements by Huawei ERC Munich

- Main focus is basic ARMv8 enablement
  - GICv2/v3
  - SMMU
  - Timers
  - Build system enhancements
  - Etc.
- Shared memory based virtio device for VM to VM communication
  - Generic solution also for other scenarios like NFV
- Performance tuning and optimizations

CCL 3.0 © Thomas Wolf, www.foto-tw.de

Contributed by Huawei ERC Munich

Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Isolation via the Jailhouse Hypervisor

Agenda

- Motivation
- Jailhouse introduction & philosophy
- Current status
- **Running multiple Linux instances (with demo)**
- Data plane isolation architectures
- Summary

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Linux As Non-Root Cell?

"Can you also run Linux partitions?"

"Would be too much patching of Linux."

"Why not using Xen PV interfaces? Linux already supports it."

"I try to get a linux kernel run as inmate on Jailhouse [on ARM]."

OK, let's think about this again...

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Looking at x86...

Regular PC

- All devices available
  (APIC, IOAPIC, PIT, PIC, UARTs, CMOS, RTC, PCI bridges...)
- Resource discovery via ACPI
- Sophisticated handover from BIOS / UEFI

Jailhouse non-root cell

- Only subset of devices
  - APIC (with certain addressing restrictions) → IPI, MSI, timer
  - IOAPIC pins (if assigned)
  - UART (if assigned)
  - PM Timer (as reference clock)
  - TSC (as fast-past clock)
  - Selected PCI devices, no bridges
- Basic resource discovery via communication region
  - PM timer address
  - Number of logical CPUs

Unrestricted © Siemens AG 2015. All rights reserved

A Simpler Solution Than Expected

So, we *do* need to patch a lot?

No!

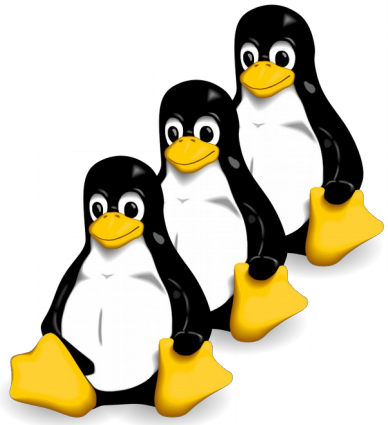
```
arch/x86/Kconfig                          |  10 +
arch/x86/include/asm/hypervisor.h         |   1 +
arch/x86/include/asm/jailhouse_para.h     |  27 ++
arch/x86/kernel/Makefile                  |   2 +
arch/x86/kernel/apic/apic_flat_64.c       |  12 +-
arch/x86/kernel/cpu/hypervisor.c          |   3 +
arch/x86/kernel/jailhouse.c               | 229 ++++++++++++++
arch/x86/kernel/smpboot.c                 |   7 +-
8 files changed, 286 insertions(+), 5 deletions(-)
```


Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Status of Non-Root Linux Cells

- **x86**
  - Assignment of MSI/MSI-X PCI devices working, no legacy INTx
  - SMP working
  - Inter-cell shared memory working
- **ARM**
  - Partly easier due to device tree description
  - Pitfall: shared resources (e.g. clock gate control on Banana Pi)
  - No publicly available reference setup so far
- **Common limitation: no virtual consoles yet**
  - To be built on top of ivshmem or its successor



Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Inter-Cell Communication

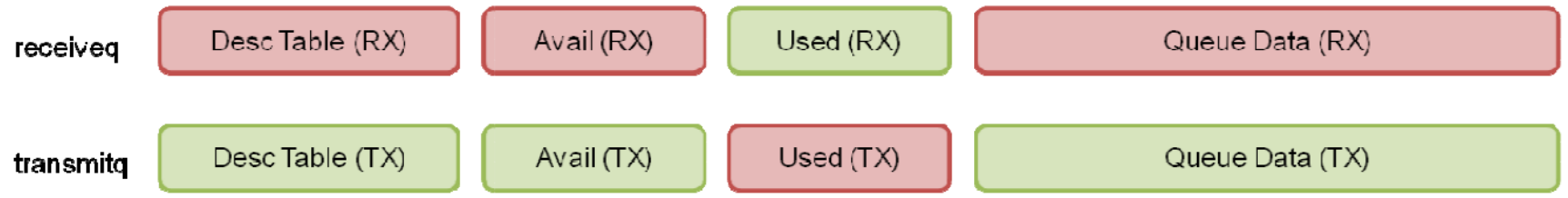
- **ivshmem**
  - Shared r/w RAM region of two cells
  - Signaling (MSIs)
  - No messaging layer on top yet
- **Why not... virtio, rpmsg, you-name-it...?**
  - Must not share everything
  - Minimize copying
  - Minimize hypervisor effort
  - Avoid dynamic page remappings
- **No perfect solution, need to find the least evil one**
  - Ongoing discussion with OPNFV and KVM communities
  - See our proposal at https://github.com/hw-claudio/virtio-peer/wiki

Unrestricted © Siemens AG 2015. All rights reserved

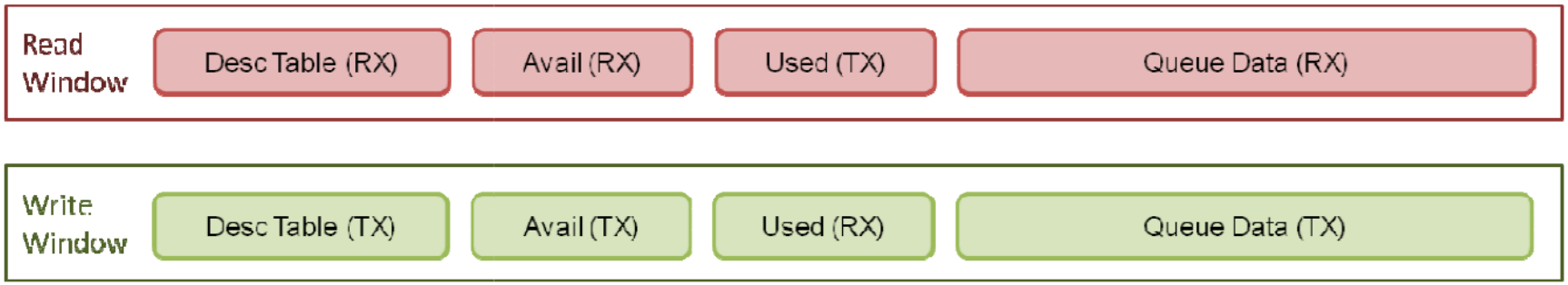
SIEMENS

Shared Memory Building Blocks for Virtio Queues

Virtqueue View



Window View



Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Isolation via the Jailhouse Hypervisor

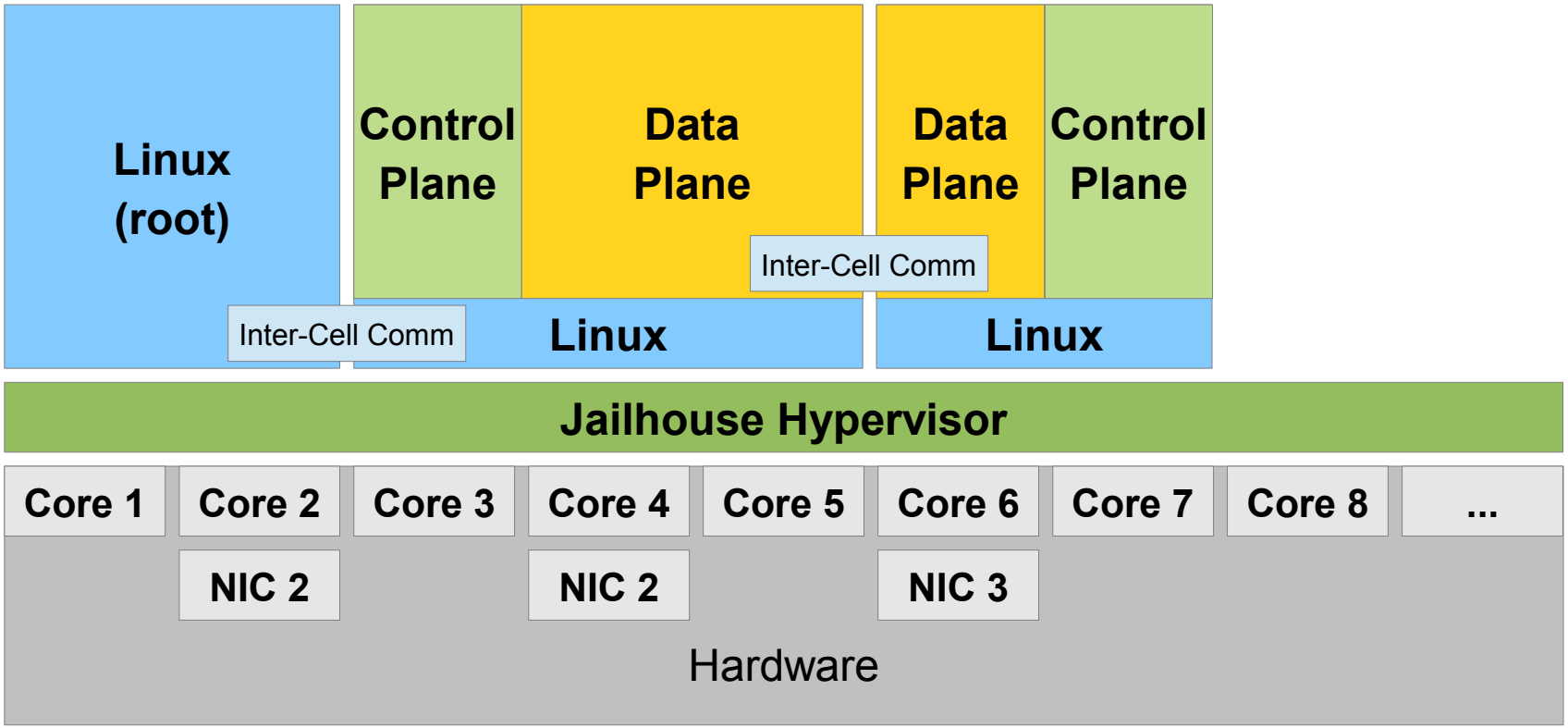
Agenda

- Motivation
- Jailhouse introduction & philosophy
- Current status
- Running multiple Linux instances (with demo)
- **Data plane isolation architectures**
- Summary

Unrestricted © Siemens AG 2015. All rights reserved

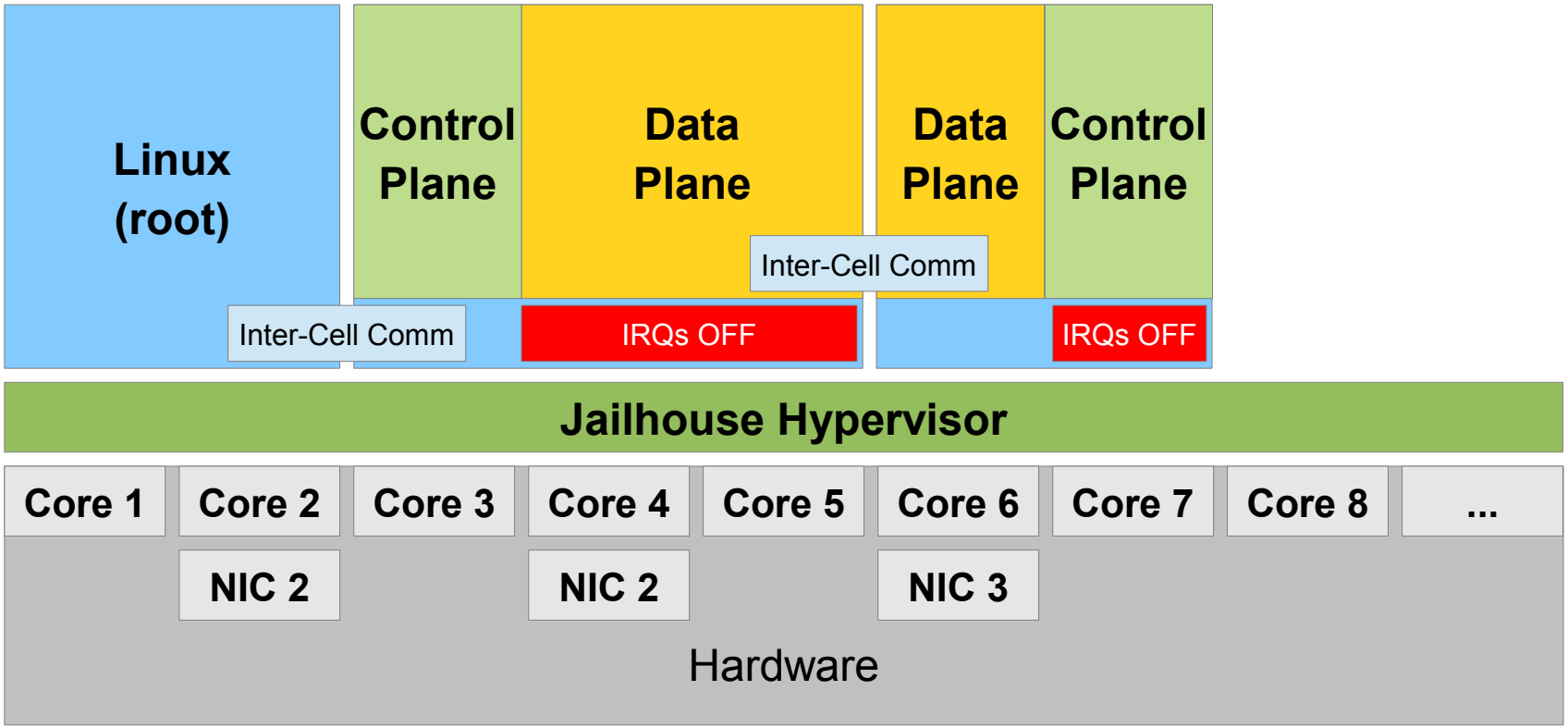
SIEMENS

Isolation of DPDK setups



Unrestricted © Siemens AG 2015. All rights reserved

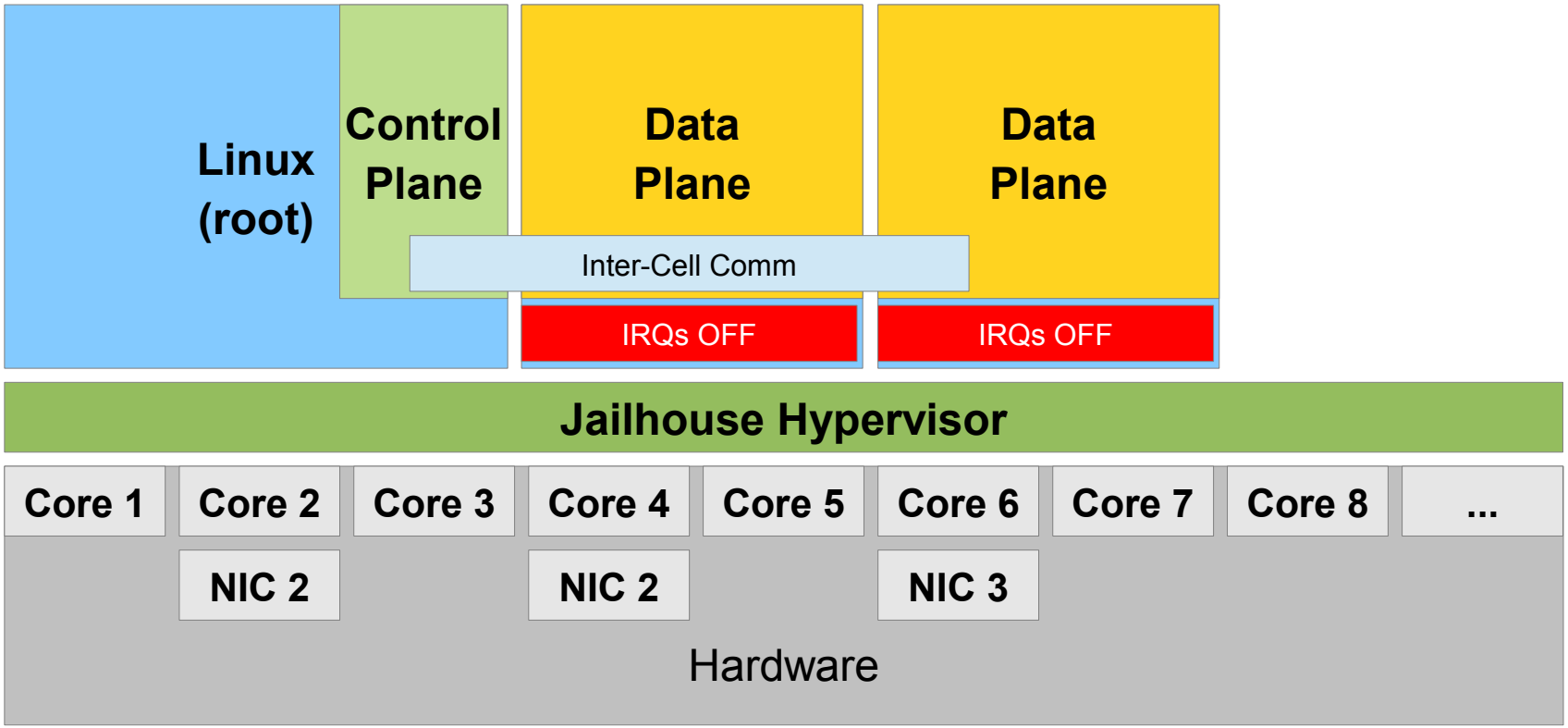
IRQ-free Data Plane



Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Data Plane-only Cells



Unrestricted © Siemens AG 2015. All rights reserved

Data Plane Isolation via the Jailhouse Hypervisor

Agenda

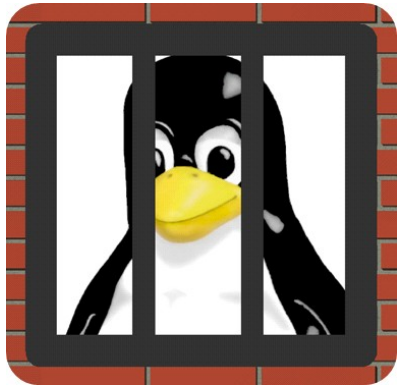
- Motivation
- Jailhouse introduction & philosophy
- Current status
- Running multiple Linux instances (with demo)
- Data plane isolation architectures
- **Summary**

Unrestricted © Siemens AG 2015. All rights reserved

SIEMENS

Summary

- **Jailhouse provides benefits for data plane scenarios**
  - Full CPU isolation
  - Minimal I/O latency
  - Strict separation
- **Jailhouse can run multiple Linux cells**
  - Small Linux patch set for x86
  - Feasible with customizations on ARM
- **Jailhouse is a community project**
  - Driven by users of different domains:
    automation, telco, automotive, …
  - Open for new use cases,
    while keeping focus on simplicity and compactness



Unrestricted © Siemens AG 2015. All rights reserved

Any Questions?

Thank you!

https://github.com/siemens/jailhouse

Jan Kiszka <jan.kiszka@siemens.com>

Unrestricted © Siemens AG 2015. All rights reserved