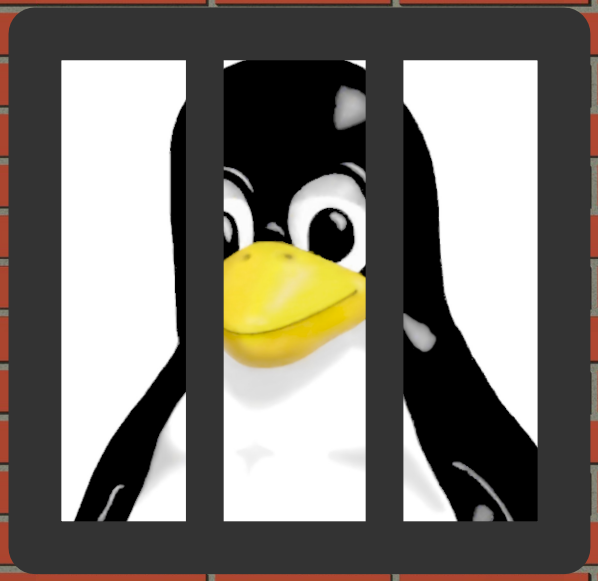
SIEMENS



Siemens Corporate Technology | October 2013

Static System Partitioning and KVM

Static System Partitioning and KVM

Agenda

Motivation & requirements

Jailhouse – a new partitioning approach

Combining partitioning and virtualization

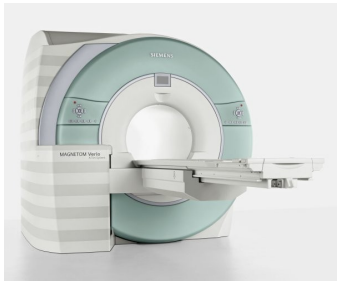
Going open source

Summary and outlook

Gimme all your CPUs!

The need for full resource dedication

- **High-speed control tasks (>10 kHz)**
  - Every µs overhead reduces achievable frequency
  - Small, infrequent disturbances can have significant impact
    => Cache pollutions
    => Deadline misses
- **High-performance computing**
  - Long-running tasks don't want interruptions
    => Keep caches hot

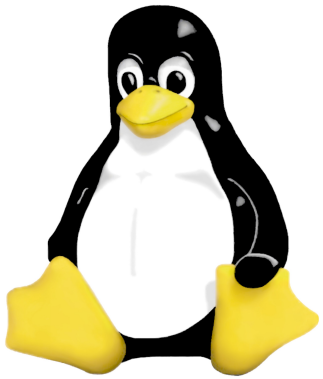


SIEMENS

CONFIG_NO_HZ* – the solution?

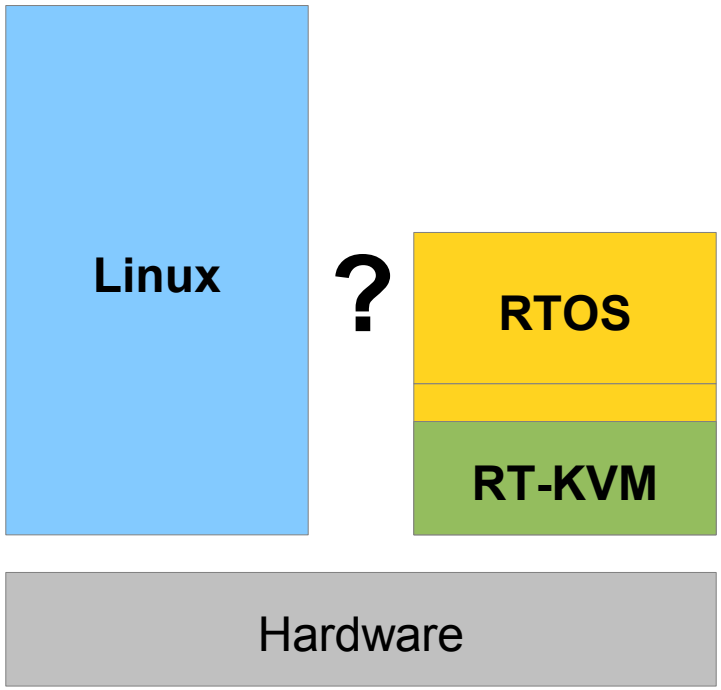
CPU domination with Linux

- **Goal: dominate CPU with a single task**
  - No interrupts, including timer ticks
  - No housekeeping work (RCU, load measurement etc.)
  - Standard application programming model
  - But do not break Linux!
- **Important steps made in upstream**
  - Reduce ticks to 1 HZ if only one task present
  - Offload RCU work to other CPUs
- **But...**
  - not yet 100%
  - more tasks/interrupts may have to run (on_each_cpu...)



SIEMENS

What if you need asymmetric multiprocessing?



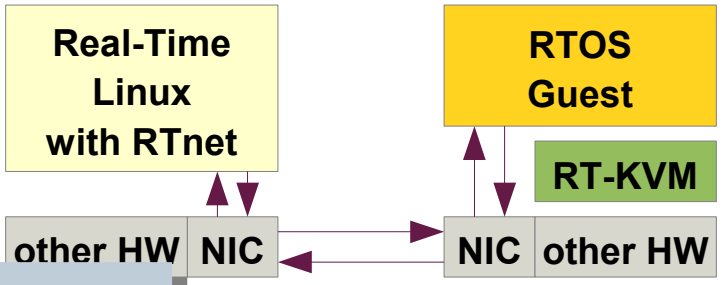
Latencies Achievable in KVM-only Setups

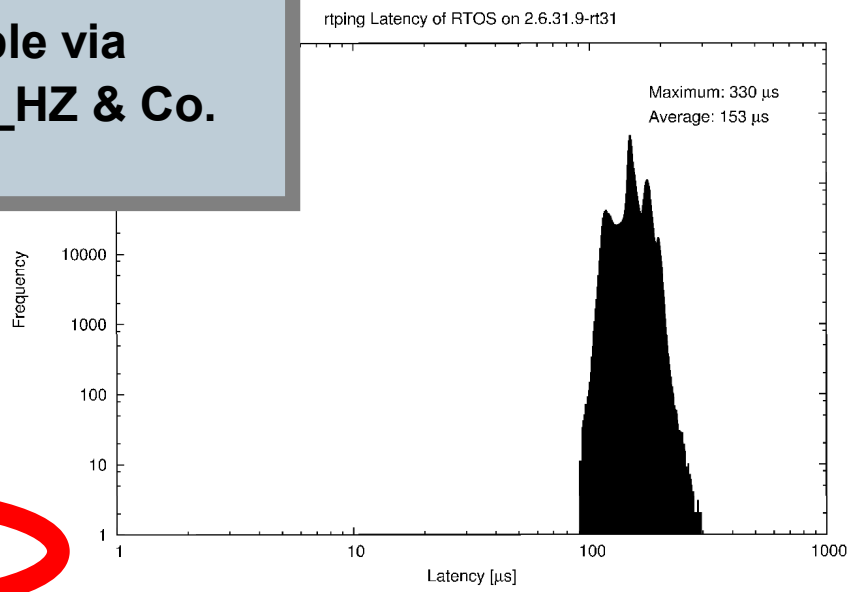
Measuring I/O latency of an RT Guest

- **Host setup**
  - KVM on x86 PREEMPT-RT Linux
  - Virtual machine on **dedicated core**
  - Intel NIC (E1000 family) as I/O device, directly assigned to guest
  - Permanent disk I/O load
- **Guest setup**
  - Proprietary RTOS
  - Real-time network stack
- **Measurement setup**
  - Linux/Xenomai (native installation)
  - Real-time network stack RTnet
  - Periodic ICMP ping messages sent to target
  - Record round-trip latency (error <50 µs)

=> Worst-case latency after 16h: 330 µs

Improvable via CONFIG_NO_HZ & Co.





SIEMENS

Small is Beautiful

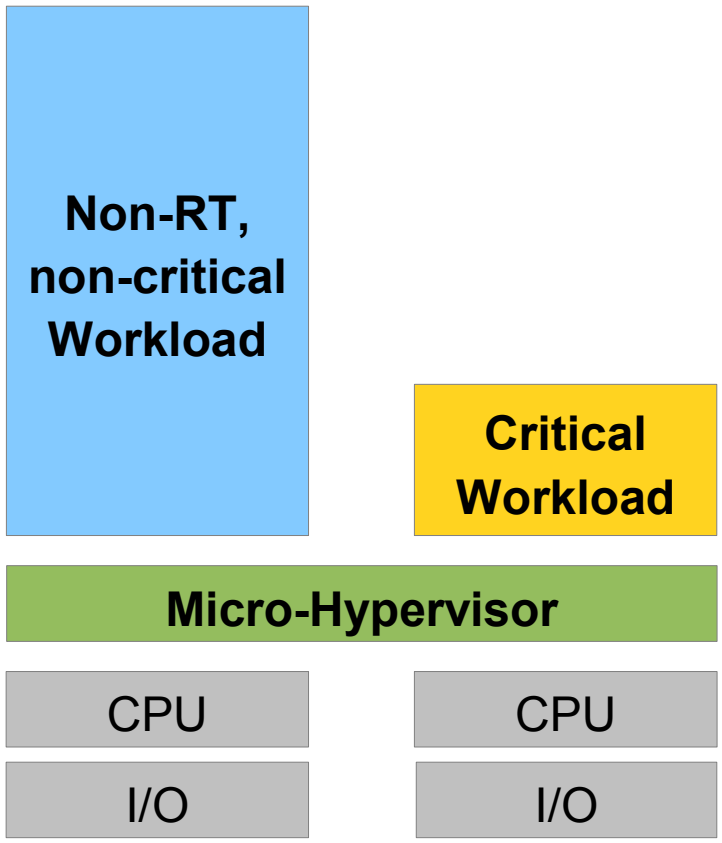
Validation efforts correlate with code sizes

- **Demanding security & safety scenarios**
  - Often require certification (Common Criteria, IEC 61508, …)
  - Need to look closely at hardware & software
    - Review / testing
    - (Formal) validation
- **The larger your system, the higher your effort**
  - Split critical from non-critical components
  - Keep critical components small
- **Virtualization can help with segregation**
  - ...if it remains simpler than non-critical parts

1st Approach: Micro-Hypervisor

Small, bare-metal hypervisor separates workloads

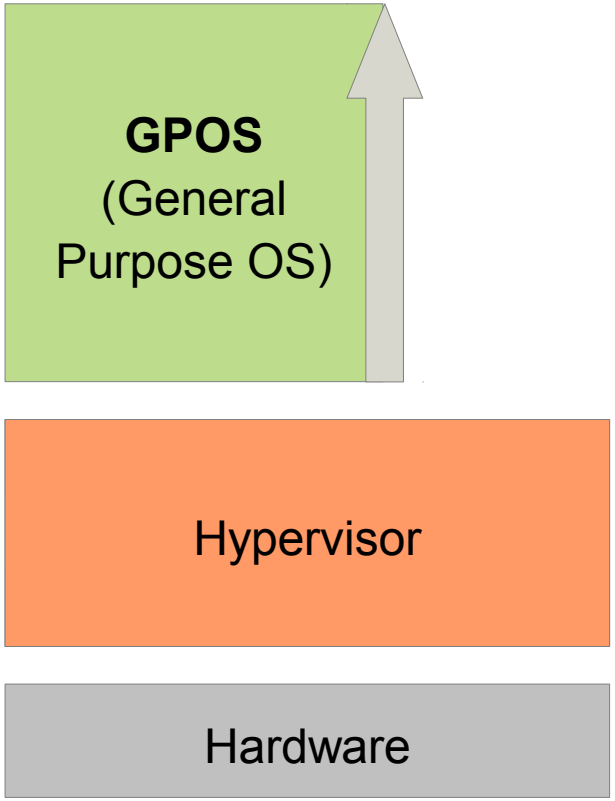
- **Focused on guest isolation**
  - Spatial
  - Temporal
- **Reduced complexity (& features)**
  - Reduces validation effort
  - Reduces guest latencies
- **No standard available yet**
  - Niche market
  - Many commercial hypervisors
  - Few open source projects
    - Hardware restrictions
    - Not targeting industrial use



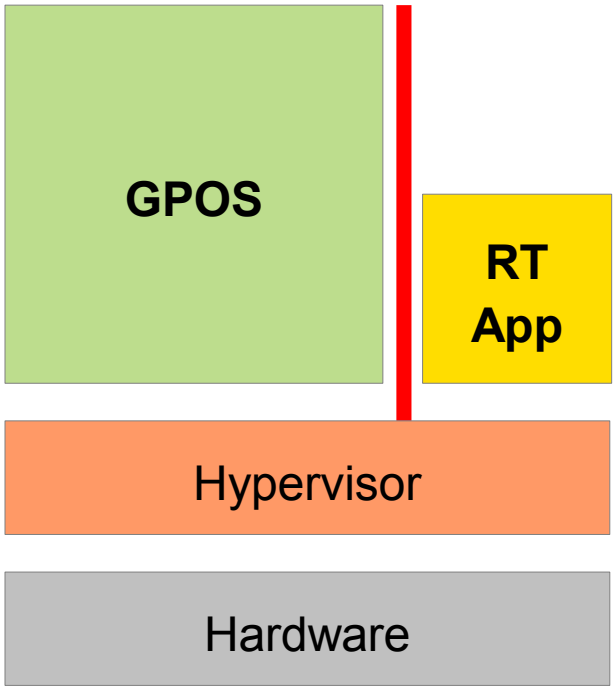
SIEMENS

A bare-metal hypervisor has to boot its guest

Classic type-1 hypervisor boot-up



1. Boot phase



2. Operational phase

Static System Partitioning and KVM

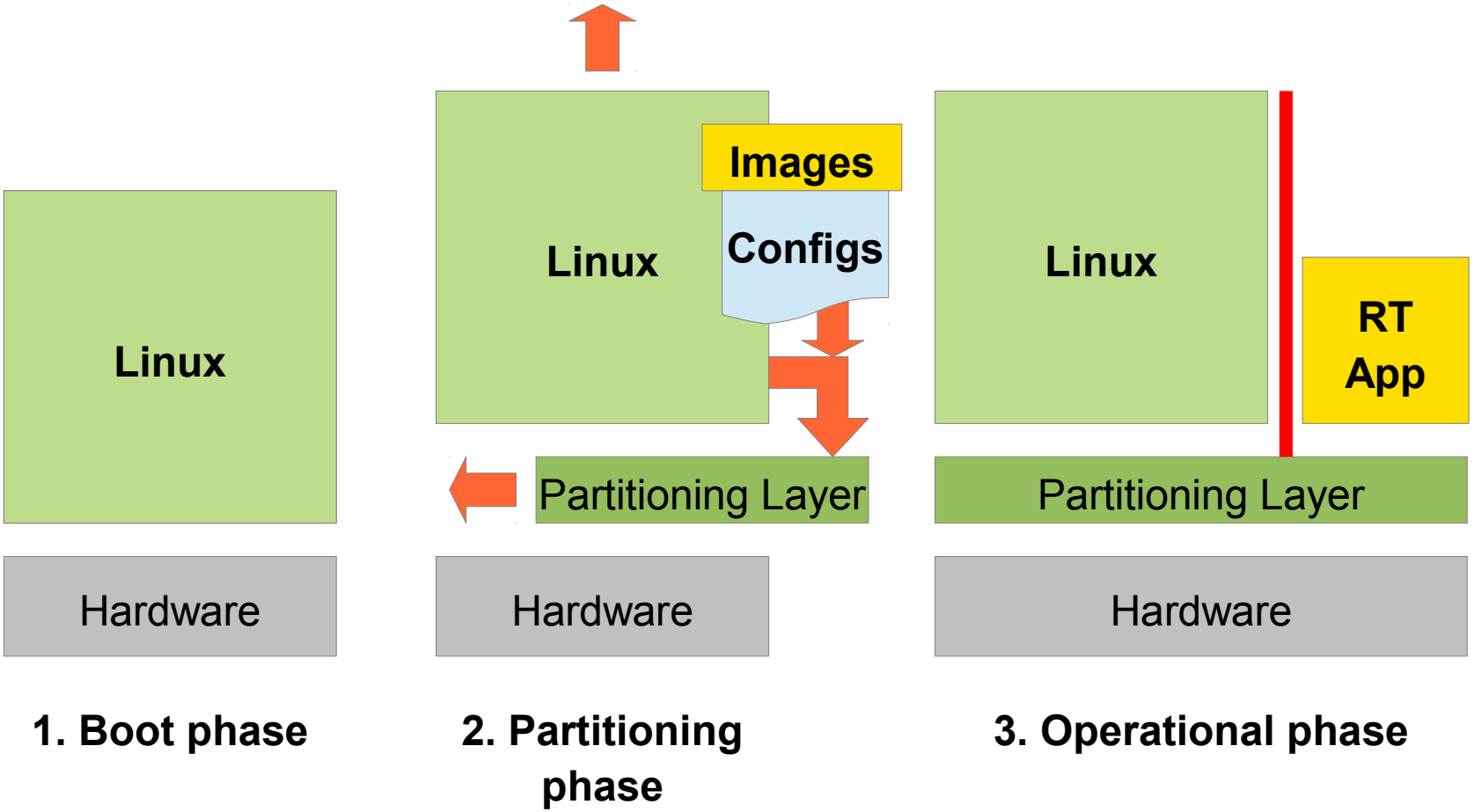
Agenda

- Motivation & requirements
- **Jailhouse – a new partitioning approach**
- Combining partitioning and virtualization
- Going open source
- Summary and outlook

SIEMENS

What about postponing the hypervisor start?

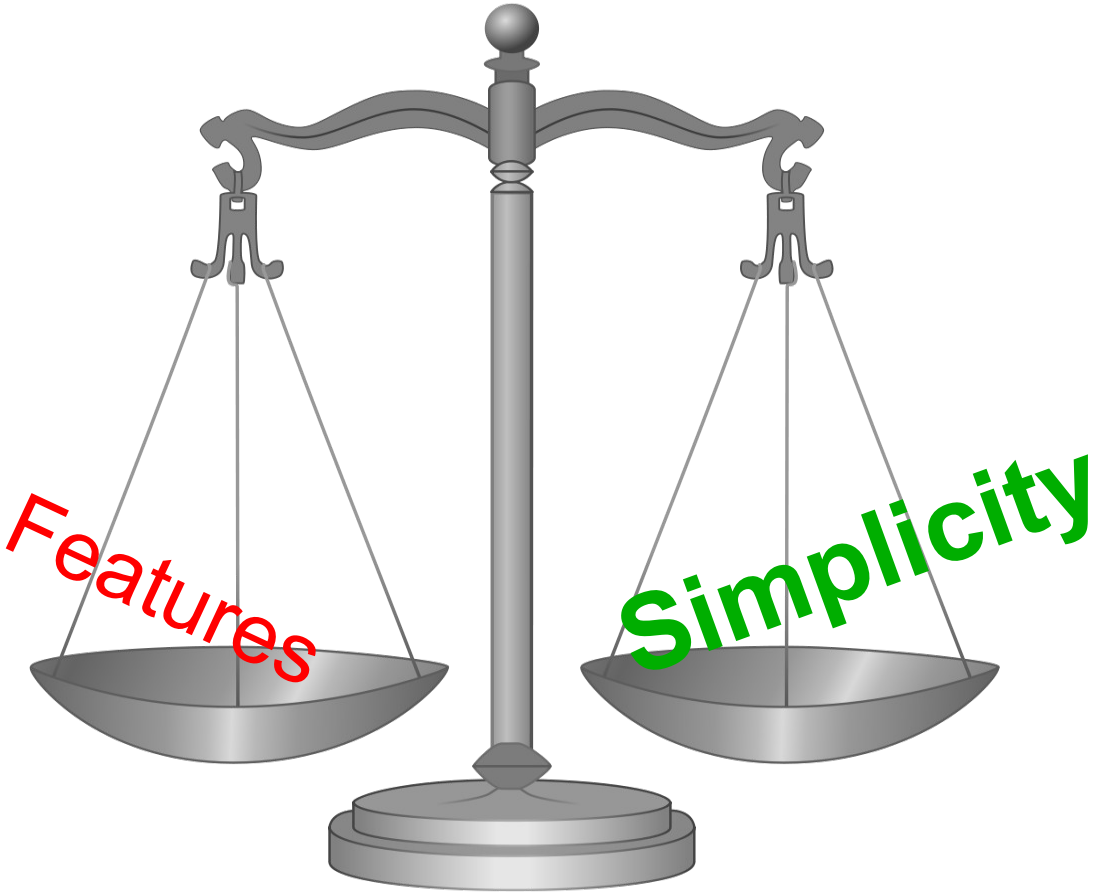
Basic concept of late partitioning



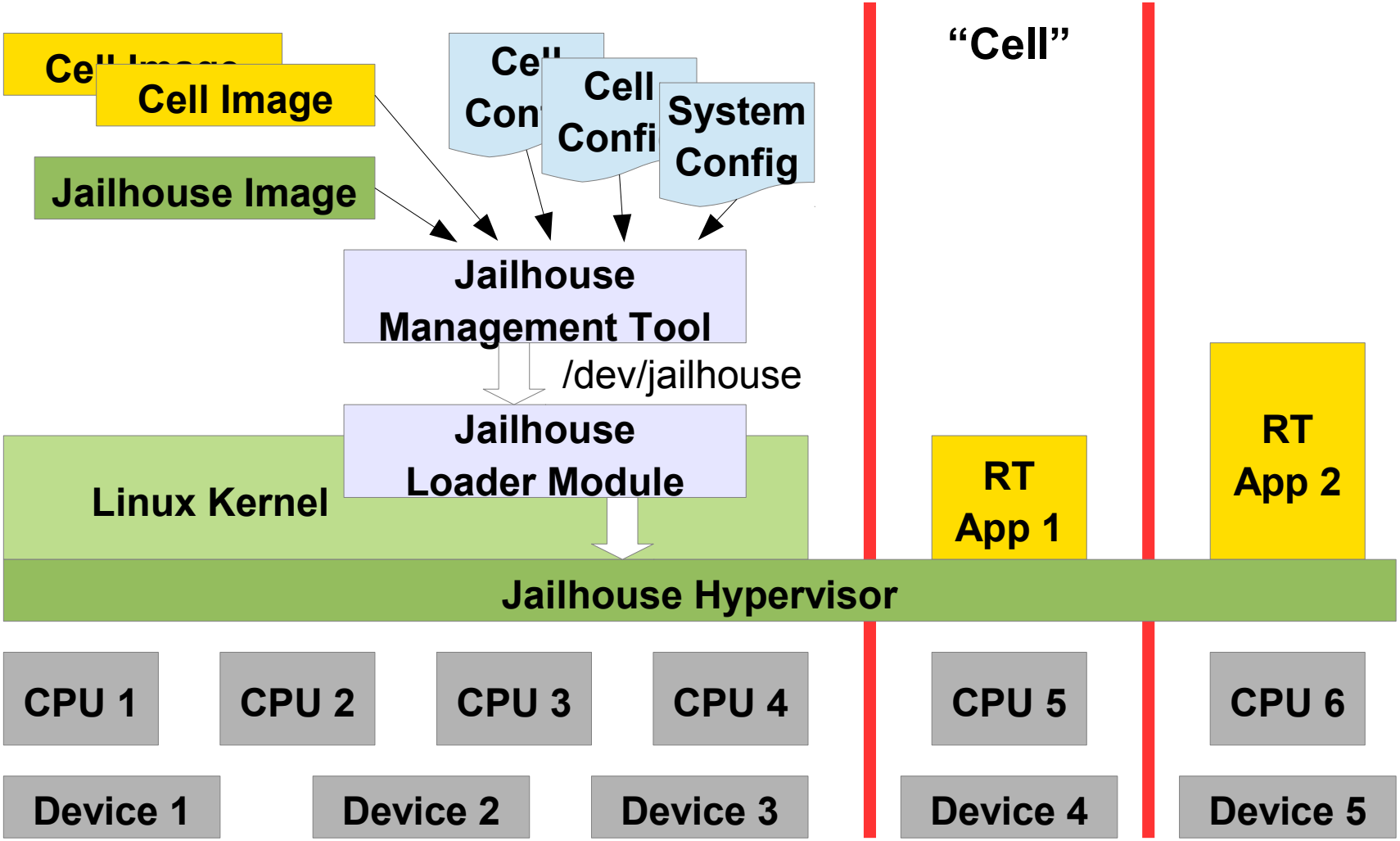
SIEMENS

Choosing the Right Balance

Jailhouse focuses on simplicity



Jailhouse Architecture



SIEMENS

Access Control instead of Virtualization

Limits of exclusive resource assignment

- **Intercept and filter access to sensitive resources**
  - Physical addresses (unless hardware filters)
  - I/O interrupt & IPI destination programming
  - Cross-cell impact (e.g. system reset)
- **1:1 resource assignment**
  - No overcommitment, no scheduling

    => Better predictability, less complexity
- **Do not hide hypervisor existence**
  - No emulation of lacking resources
  - Expose assigned resource (widely) unmodified
  - Linux won't notice (already booted), other cells need awareness

Jailhouse does not overlap with KVM

You need more? Use KVM!

SIEMENS

Linux is Our Friend

Reuse Linux for management tasks

- **Bootstrap**
  - System boot-up, hardware pre-configuration
  - Hypervisor loading and configuration
    - `jailhouse enable CONFIG-FILE`
  - RT partition creation & image loading
    - `jailhouse cell create CONFIG-FILE IMAGE-FILE`
    - Linux unplugs resources for new cell (CPU, devices, memory)

=> Reduced hypervisor complexity

=> UNIX-like look & feel

SIEMENS

Linux is Our Friend (2)

Reuse Linux for management tasks

- **Operation**
  - Reconfigurations (while in non-operational mode)
    - `jailhouse cell destroy NAME`
  - Monitoring, logging etc.
  - Shutdown
    - `jailhouse disable`

=> Reduced hypervisor complexity

=> Short turn-around times, less reasons to reboot

SIEMENS

Prototyping on x86

Jailhouse on Intel x86

- **Initial focus on Intel**
  - VT-x with EPT, unrestricted guest mode, x2APIC
  - VT-d with interrupt remapping
- **Direct interrupt delivery feasible**
  - Keep IRQs off while in hypervisor
  - Use NMIs + preemption timer for hypervisor IPIs
- **Minimalistic MMIO**
  - Enables IO-APIC, xAPIC, PCI mmconfig interception
  - Simple, unoptimized, slow-path only use cases
- **Work in progress**
  - Device assignment, management
  - Interrupt access control

SIEMENS

Prototyping on x86 (2)

Jailhouse development inside QEMU/KVM

- **Bootstrap development done inside QEMU/KVM**
  - Unbeatable turn-around times
    - <30 s from code fix over recompilation and deployment to execution
  - Source-level debugging of hypervisor
- **Found and fixed several nVMX deficits & bugs**
  - Direct IRQ delivery
  - nEPT stabilization
  - Unrestricted guest mode
  - Preemption timer
- **Unfortunately no virtual VT-d available yet...**

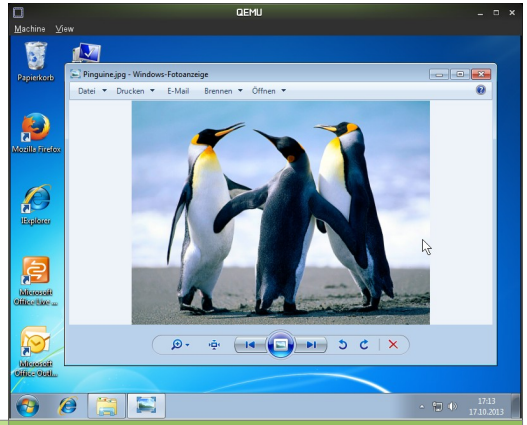
Static System Partitioning and KVM

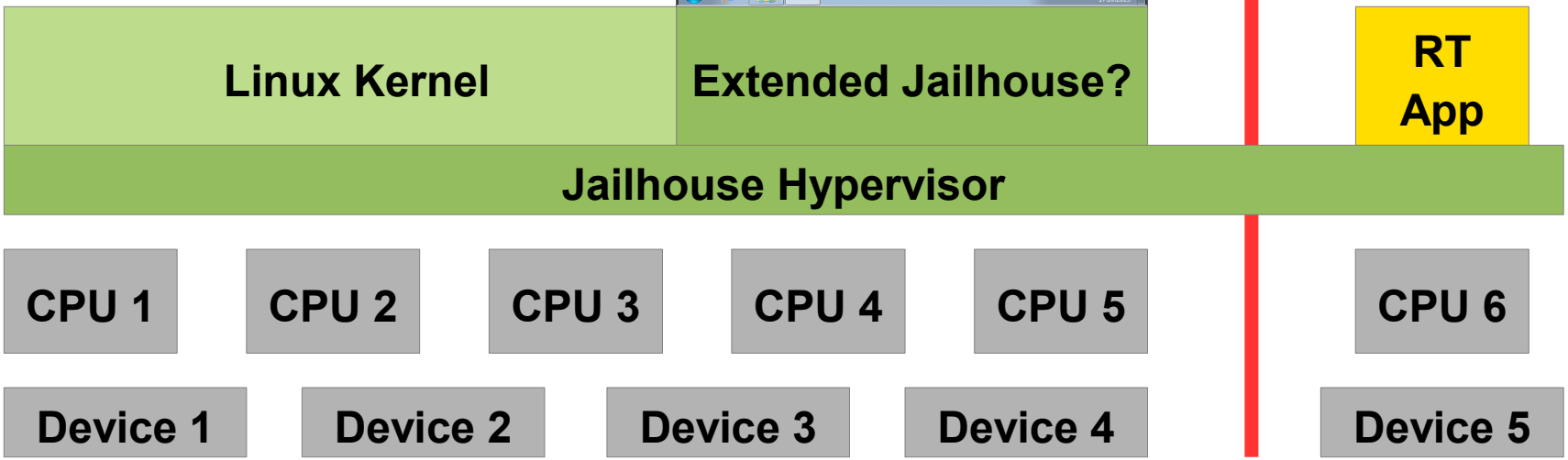
Agenda

- Motivation & requirements
- Jailhouse – a new partitioning approach
- **Combining partitioning and virtualization**
- Going open source
- Summary and outlook

What if more than Linux should run?

Hosting non-Linux guests

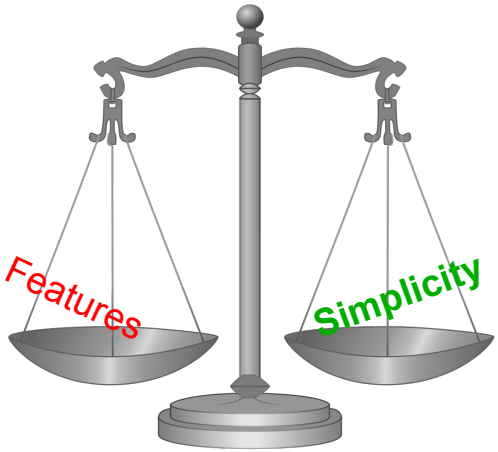




How to minimize the complexity increase?

Nested virtualization will be more beneficial

- **Full OS boot over Jailhouse**
  - Less overhead for guest
  - Requires more device emulations
  - Requires more accurate virtualization
  - Requires virtual BIOS
  - ...
- **Enable KVM over Jailhouse**
  - Overhead of monitoring privileged KVM operations
  - Can focus on CPU virtualization features
  - No need to virtualize/emulate, just validate
  - Gain (almost) all features of QEMU/KVM, benefit from its stability

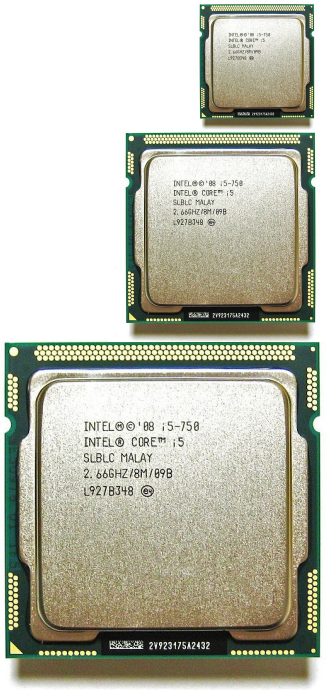


SIEMENS

Nested Virtualization on Diet

Enabling Intel x86 KVM over Jailhouse

- **Execute VMX instructions on behalf of KVM**
- **Monitor (shadow) VMCS accesses**
  - Valid fields?
  - Physical addresses with limits?
  - Unsupported features disabled?
- **We don't care if KVM crashes its CPU**
  - ...as long as it doesn't affect other cells
- **Deny EPT in 1st prototype**
  - Slow but simple
- **General need to establish feature restrictions**
  - Pragmatic: load KVM after Jailhouse
  - Rediscover features on Jailhouse detection

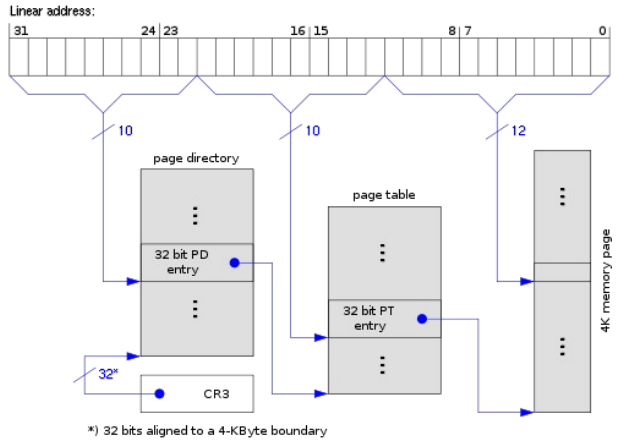


SIEMENS

Optimization: Nested EPT

Monitoring of Extended Page Table usage by KVM

- **1:1 mapping – no shadowing required**
- **Monitoring concept**
  - Full validation walk on new EPT
    - Note EPT internally as valid
  - Trap writes to known EPTs
    - Check if page belongs to known EPT (drop write-protection if not)
    - Declare EPT invalid if entry becomes invalid through write
    - Execute write
  - Use KVM's EPT while running its guest
- **Last resort: para-virtualization**



Static System Partitioning and KVM

Agenda

- Motivation & requirements
- Jailhouse – a new partitioning approach
- Combining partitioning and virtualization
- **Going open source**
- Summary and outlook

SIEMENS

Why Open Source?

Benefits of maintaining Jailhouse as open source

- **“Just a few lines of code, easily maintainable.”**
  - Hardware-assisted virtualization is non-trivial
    => Many-eyes principle
  - New CPUs and hardware features will keep us busy
    => Attract contributors, including silicon vendors
- **Broaden the usage**
  - Higher test coverage, faster stabilization
  - Additional use cases => more contributors
- **Close cooperation with Linux kernel**
  - Enable upstream changes of Linux (if required)
  - Keep the door open for integration
- **GPL: Preserve openness**

Static System Partitioning and KVM

Agenda

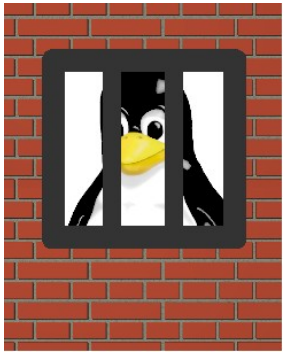
- Motivation & requirements
- Jailhouse – a new partitioning approach
- Combining partitioning and virtualization
- Going open source
- **Summary and outlook**

SIEMENS

Jailhouse – Static Partitioning as Linux Feature

Summary

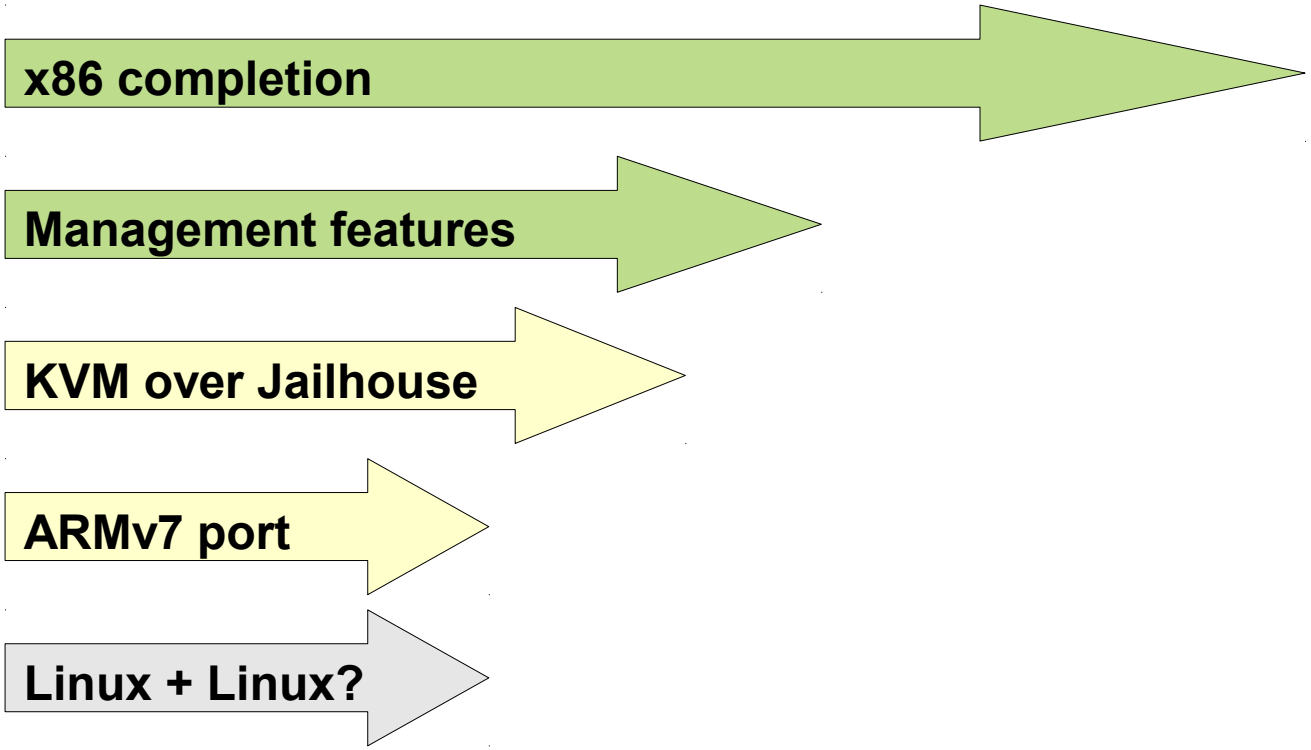
- **Need for critical workload isolation**
  - Undisturbed from non-critical system parts
  - Low-latency access to I/O
  - Reduce validation efforts
- **Jailhouse provides building block for partitioning**
  - Allows full CPU isolation
  - Reduced to the minimum (goal: <10k lines of code)
  - Linux-based to reuse handy infrastructure
  - Optionally combine with KVM for full virtualization



SIEMENS

What is next?

Outlook



Follow / join the development!

https://github.com/siemens/jailhouse

Any Questions?

Thank you!

Jan Kiszka <jan.kiszka@siemens.com>