

Virtualization: from Mobiles to Super Computers



Marco Cornero
STMicroelectronics

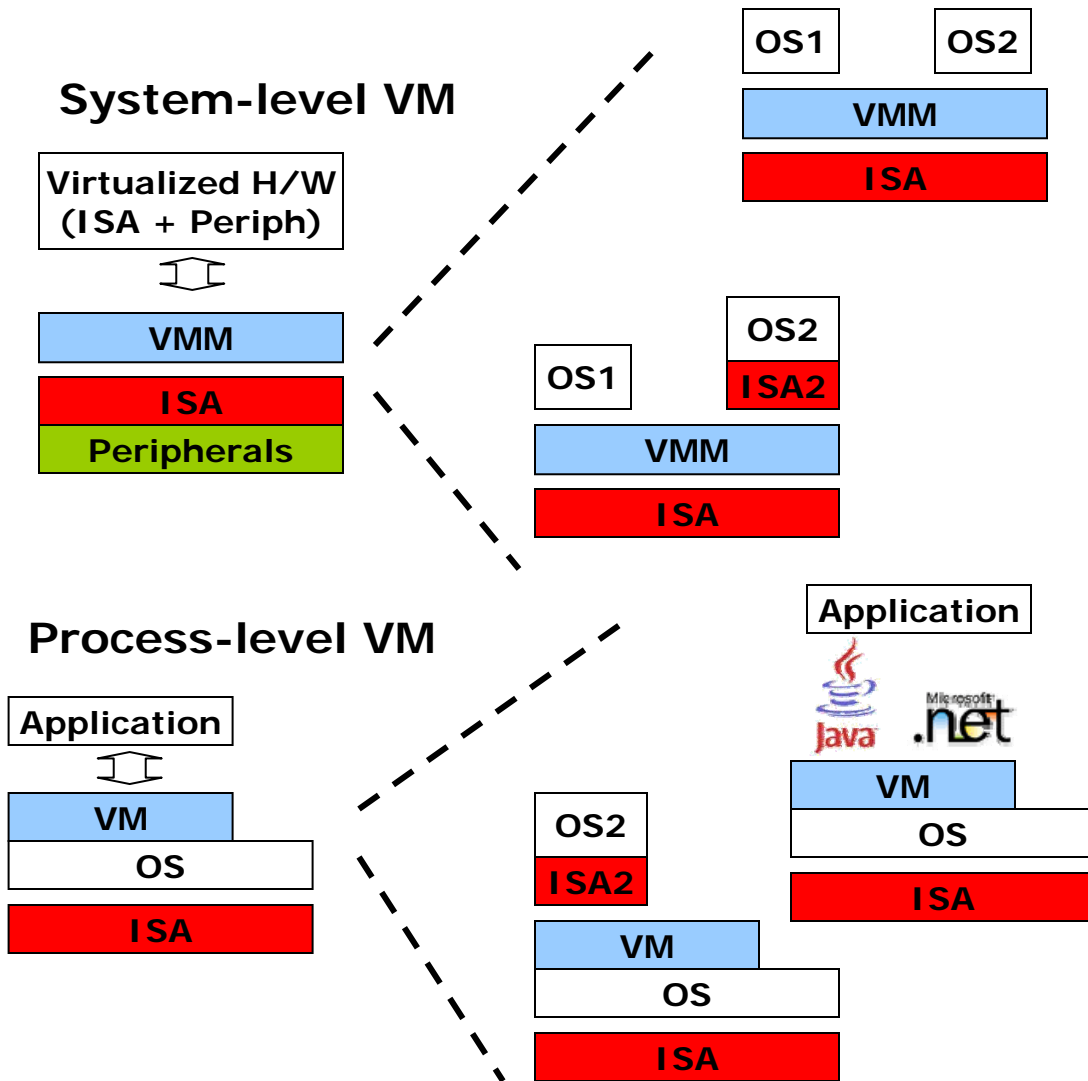
Director Compilers,
Operating Systems
and Applications



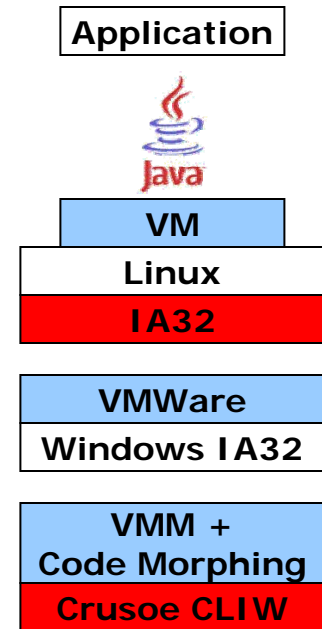
Acknowledgments

- ▣ Presentation prepared in cooperation with several HiPEAC members. Special thanks to
 - ▣ Koen De Bosschere, Jonas Maebe, Lieven Eeckhout (UGent), Bjorn De Sutter (IMEC), Marisa Gil (UPC)

Virtualization at Different Levels for Many Uses



A Concrete (realistic!) example



Main Benefits of Virtualization

- ▣ More Efficient and Flexible Use of Resources
- ▣ Isolation
 - ▣ Sandboxing, Security
- ▣ Hide legacy ISA Ballast
- ▣ Portability
 - ▣ Microprocessors
 - ▣ Complete systems
- ▣ Composability
- ▣ Optimization opportunities
 - ▣ E.g. Dynamic ISA translation to exploit new features
 - ▣ Exploitation of dynamic information
- ▣ Modeling

Virtualisation is Everywhere



ANDROID



Embedded



Desktop/Server



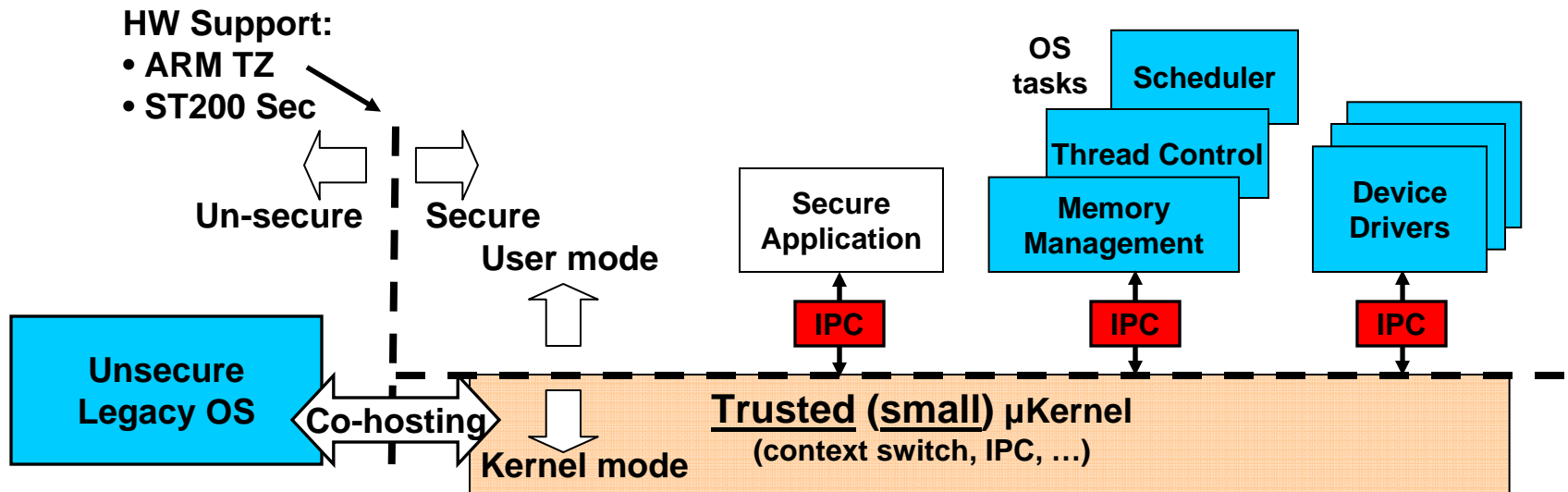
VMware Fusion



High Performance

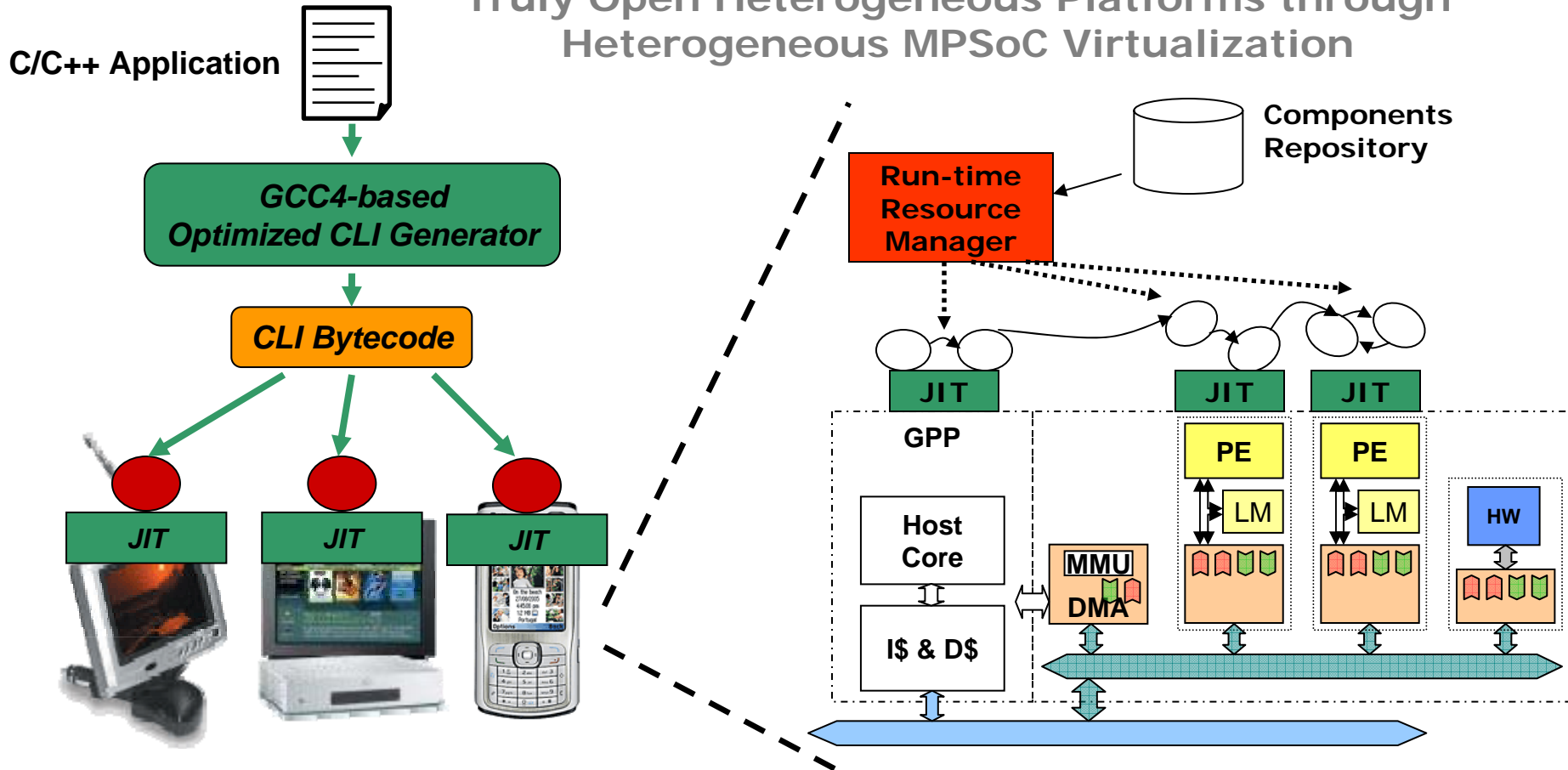
Use Case Examples in Embedded Systems (Already Sponsored by EU Projects)

Side-by-Side OS Execution for Security (e.g. with L4 micro-kernel)



Use Case Examples in Embedded Systems (Already Sponsored by EU Projects)

Truly Open Heterogeneous Platforms through
Heterogeneous MPSoC Virtualization



EU Academic Projects Example (UGent)

Modeling

- Dealing with non-determinism (what gets compiled/optimized, when)
- Relating higher and lower layer events
- Mapping models for one VM to another (on different HW, for different intermediate languages, ...)
- Interactions between virtualisation layers (HW resource abstractions & process VMs)
- Predicting effects of multi-tasking
- Example: predict performance of Java apps running on Jikes RVM (x86 & Power4)

Vertical Instrumentation

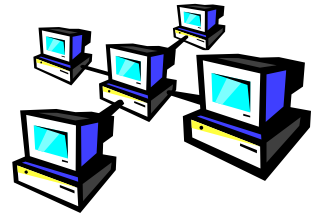
- Relating high level to low level events (e.g. memory accesses to Java objects -> object lifetime)
- Example: Jikes RVM + Diota instrumentator (x86)

Virtualization in the Embedded World

Key Applications and Main Challenges

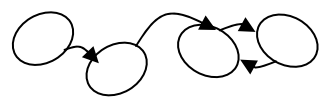
Ubiquitous Computing

- Smooth integration with external resources
 - Extension of resource manager
 - Fully virtualized off-loading to the network resources



MPSoC Programming Models

- H/W is a highly reconfigurable heterogeneous Multi-Proc.
- Programming models
- Resource management, QoS
- Virtual communication channels



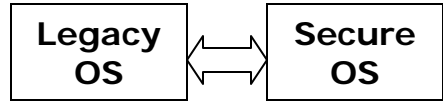
High-Level Language VM's

- Efficiency, predictability
- Interaction with low-level API's
 - Open, efficient

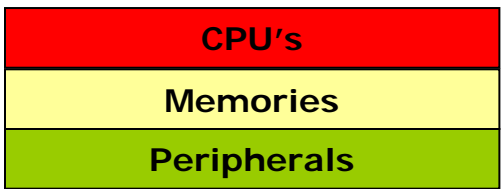
Strong HW/SW Cooperation

- Secure/non-secure HW/SW switching, resource isolation
- Strong reliability in secure mode
- QoS

Appl. Level Processor ISA Virtualization



Virtual Machine Monitor



- ISA to ISA Dynamic Binary Translation
- Bytecode to ISA Translation
 - Which bytecode?
 - JIT compilation
 - Bytecode Generation ⇔ JIT Compilation tradeoff

Strong HW/SW Cooperation (Codesigned VM's)

- Proc. Instruction-set
- Memory Management Units
- Virtualization-friendly Peripherals



Virtualization for HPC (embedded or not)

Effective use of multi-processors REQUIRES virtualization at different levels

- CPU in space and time
- Memory usage and bandwidth
- Cache partitioning
- I/O scheduling and partitioning

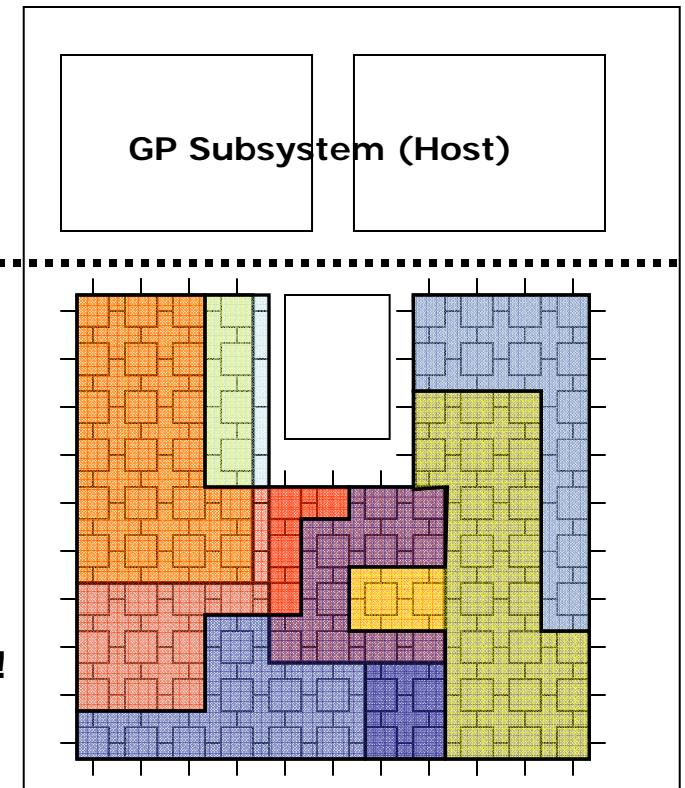
Use case in future embedded High-Performance Grid

Grid Virtualization for **Data Flow Intense Fabric** Resource Sharing

REQUIRED
(not feasible otherwise)



Logical Subsystems are now really logical!



Conclusions

- Virtualization is a key fundamental and transversal technology with huge economical impact, from deeply embedded systems to HPC
 - Flexibility
 - Resource sharing and optimization
 - Composability
 - Etc.
- EU benefits
 - It is among the key enabling technologies for any advance future computing system or subsystem
 - EU industry must develop/use it, simply because it cannot do without...
 - A lot of challenges still open for academia
 - EU is currently weak in virtualization
 - We should increase awareness of importance of this field
 - We need training, both in academia and industry
 - Several exciting challenges in front of us
 - Adaptive virtualization (H/W is reconfigurable!), virtualization of H/W accelerators, security, etc.
 - Market is already crowded, but applicability is so wide that there is space for much more
 - Good market opportunities!
 - Several open source environments, making it easy to cooperate
 - Java (several), CLI (Mono, DotGNU, GCC), Google Dalvik, LLVM and many others

Thank You!

