NICTA

Virtualizing Embedded Systems Why Bother?

Gernot Heiser

NICTA and University of New South Wales

Sydney, Australia

Australian Government

Department of Broadband, Communications and the Digital Economy

Australian Research Council

NICTA Funding and Supporting Members and Partners

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

UNSW THE UNIVERSITY OF NEW SOUTH WALES

NSW GOVERNMENT Industry & Investment

Victoria The Place To Be

THE UNIVERSITY OF MELBOURNE

THE UNIVERSITY OF SYDNEY

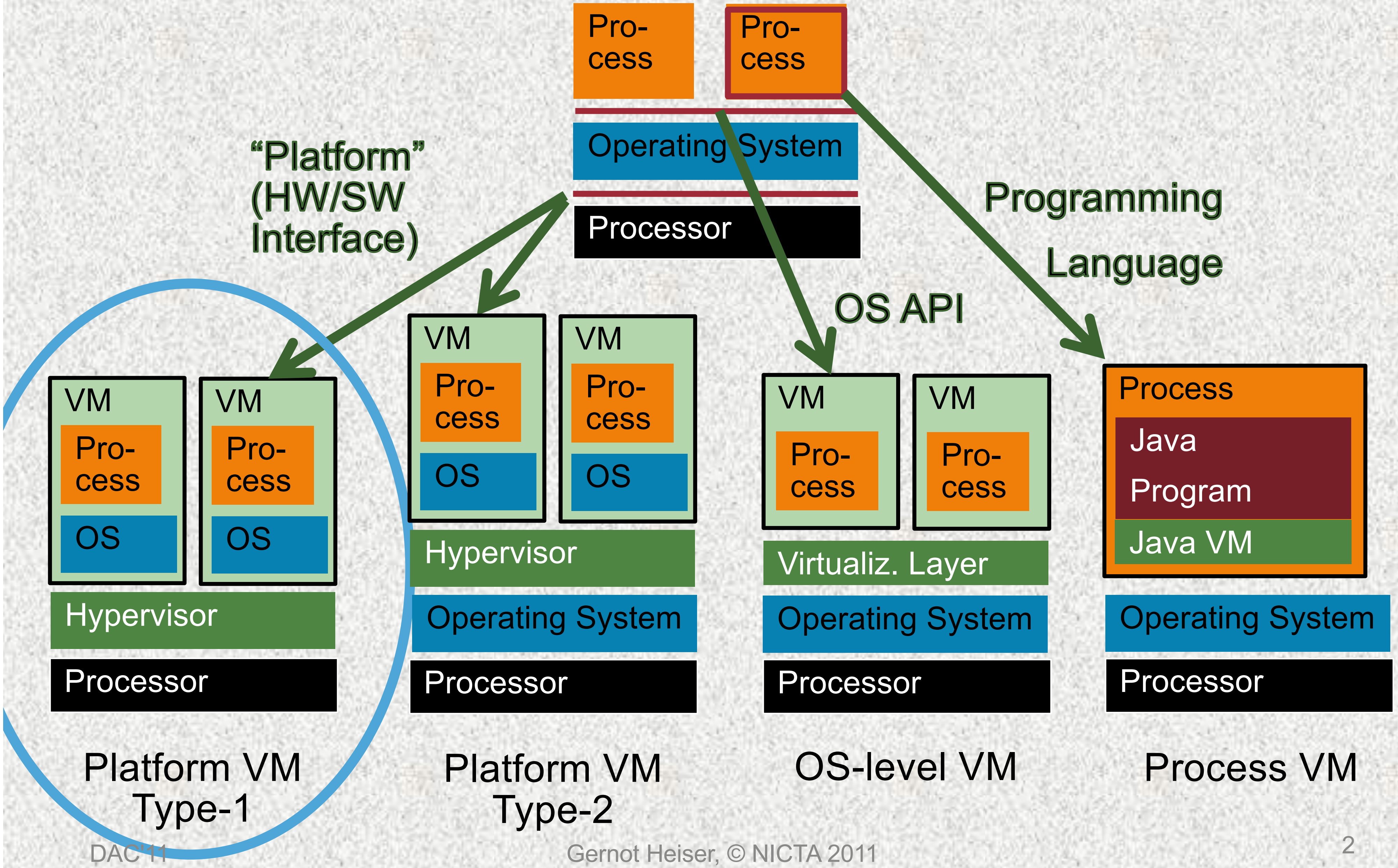
Queensland Government

Griffith UNIVERSITY

QUT Queensland University of Technology

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Types of Virtualization

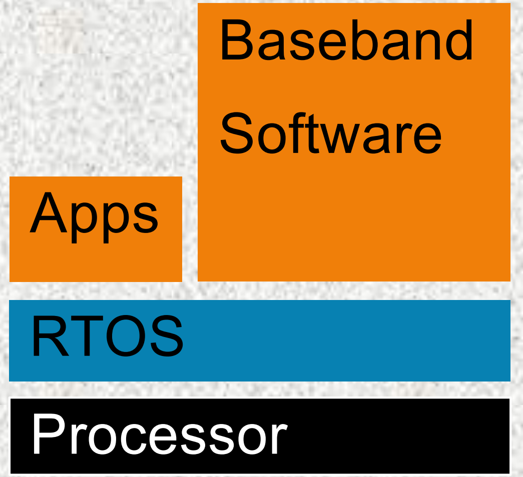


Traditional System Virtualization Uses

- Server consolidation
  - Hardware & energy savings with QoS isolation
  - Migration, checkpointing, debugging
  - Concurrent use of multiple OSes (or OS versions)
- Security
  - Partitioning to limit scope of compromises
  - Sandboxing untrusted apps

So, what does it do for embedded systems?

Mobile Phones



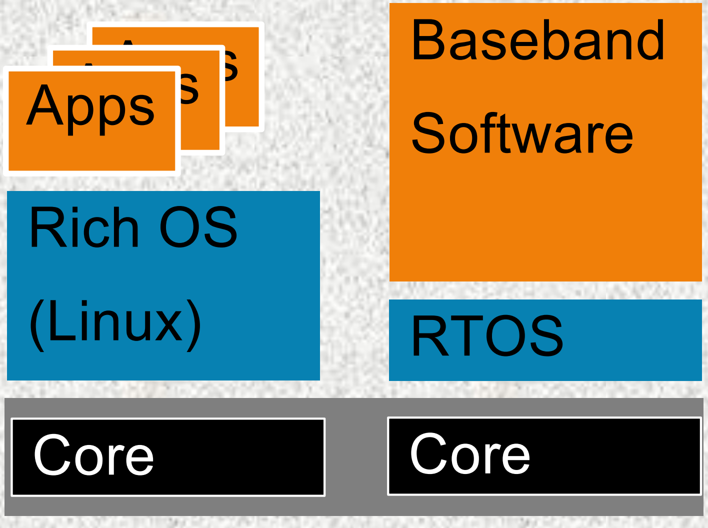
Baseband Software

Apps

RTOS

Processor

Dumb phone



Apps

Rich OS (Linux)

Baseband Software

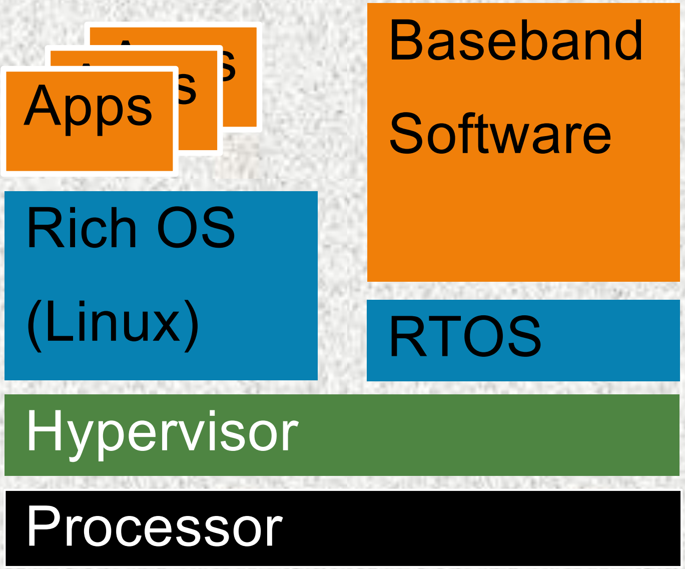
RTOS

Core

Core

Smartphone

Heterogenous Operating Systems!



Apps

Rich OS (Linux)

Baseband Software

RTOS

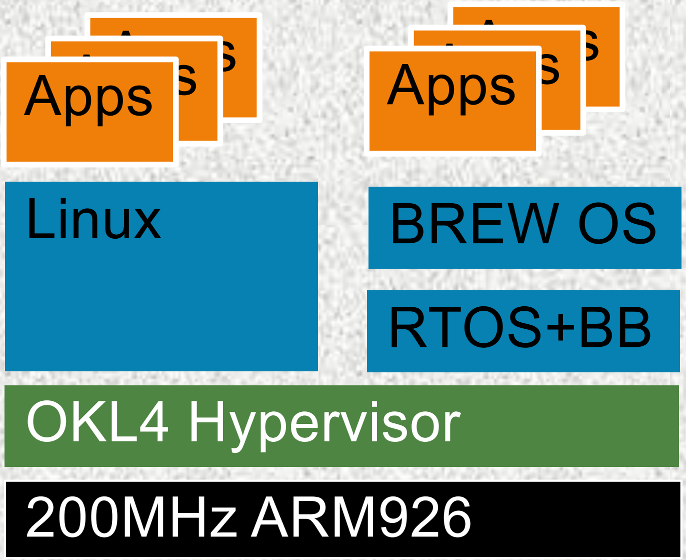
Hypervisor

Processor

Consolidated phone

Consolidated Phone: Motorola Evoke

- Linux+BREW OS
- Linux+BREW apps
- Seamless UI integration
- Released April 2009



Apps

Apps

Linux

BREW OS

RTOS+BB

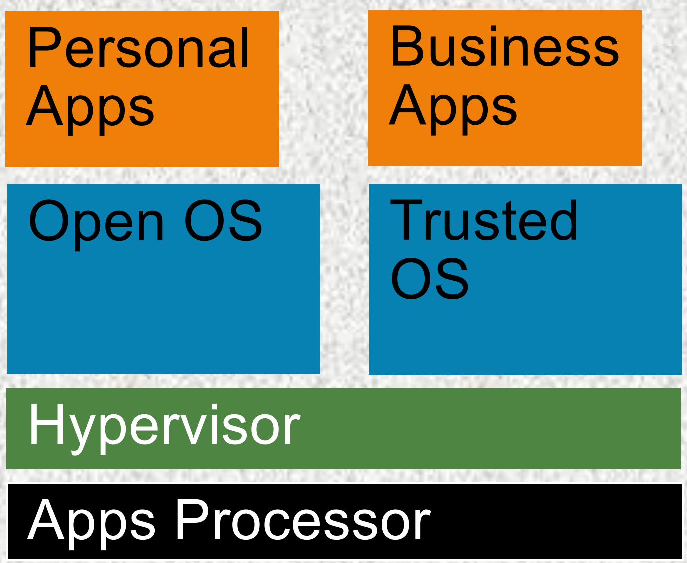
OKL4 Hypervisor

200MHz ARM926

Dual-Persona Smartphone

- Phones increasingly used to access business data
  - Companies lock down phones, no arbitrary apps
  - Employees end up carrying two phones

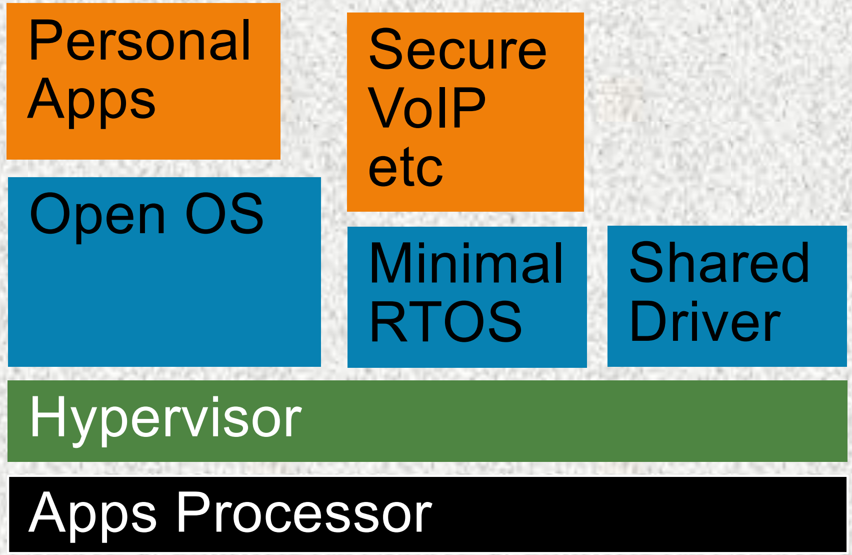
- Integrate two virtual phones into one physical
  - Locked-down business phone
  - Open personal phone



- Will reach market soon

Secure Communication on COTS Phone

- Secure phones are expensive (small product runs)
- Strong push for COTS devices in defence etc
- Virtualization provides secure comms on standard smartphone
- Encrypt voice, data and tunnel through open OS
- Hypervisor guarantees isolation



- Small “trusted computing base”
- Presently under evaluation by various agencies

Beyond CE: Critical Next To Untrusted

- General trends across industry verticals:
  - Growing functionality ⇒ growing complexity
  - Tight integration of critical functionality with complex communication / UI / entertainment software
    ⇒ growing security/safety threats
- Examples: medical devices, transport

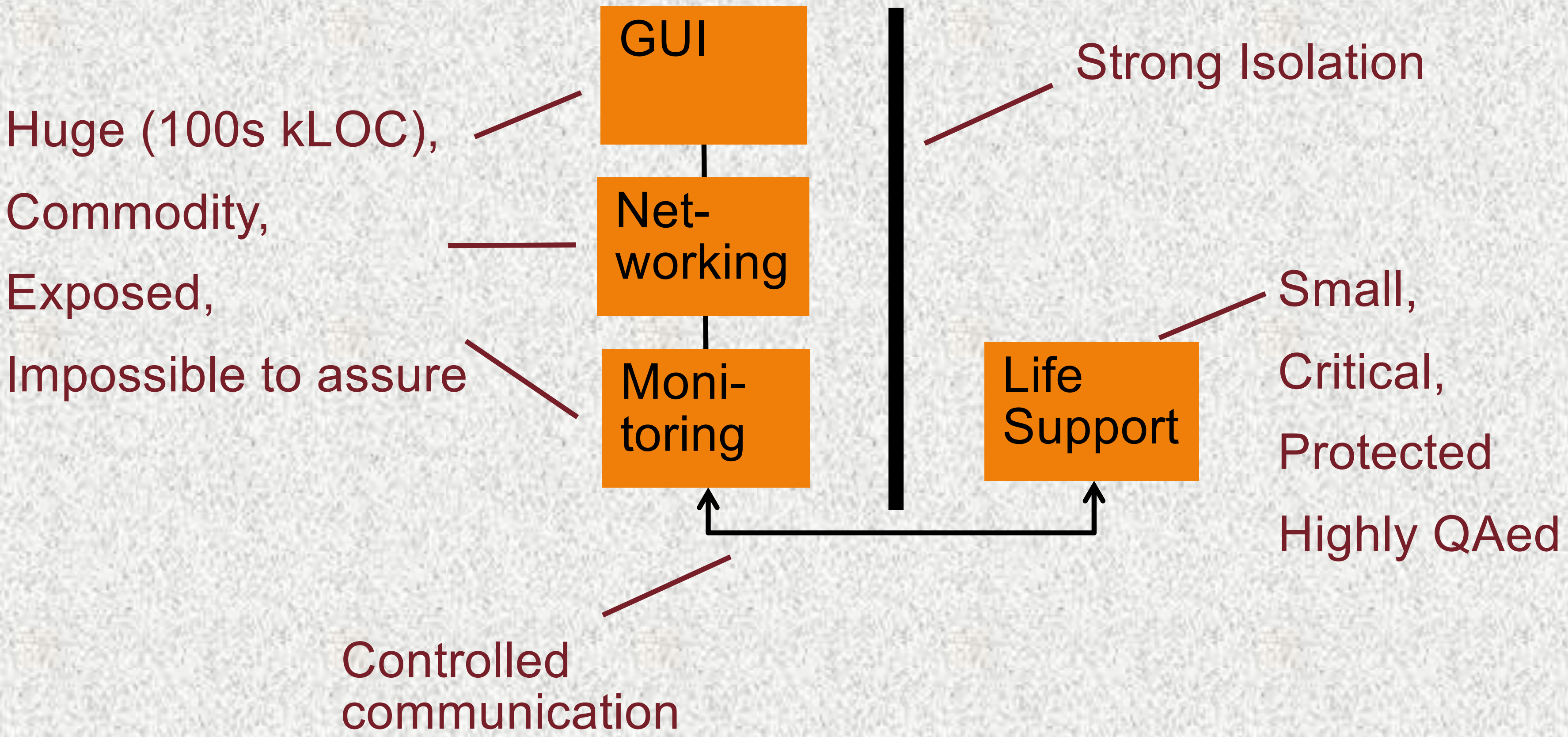
Wearable/Implanted Medical Devices

- Remotely monitored
  - eg pace makers
- Patient-operated
  - eg insulin pumps
- Complex software stacks
  - User interfaces
  - Network drivers and protocol stacks

⇒ Significant safety risks: eg Pacemaker hack [2008]

- Remote control of a pacemaker

Needed: Strong Protection



Huge (100s kLOC),

Commodity,

Exposed,

Impossible to assure

Strong Isolation

Small,

Critical,

Protected

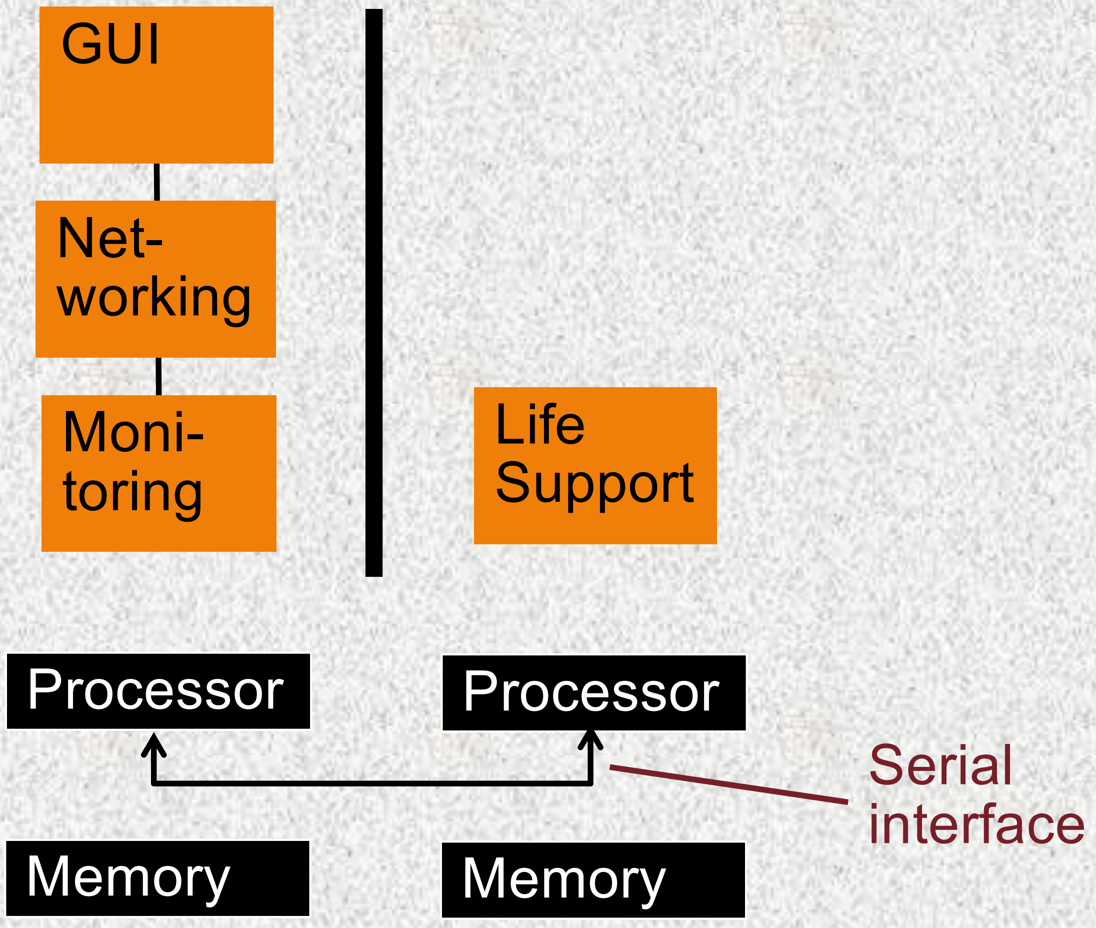
Highly QAed

Controlled communication

Needed: Strong Protection

Physical separation

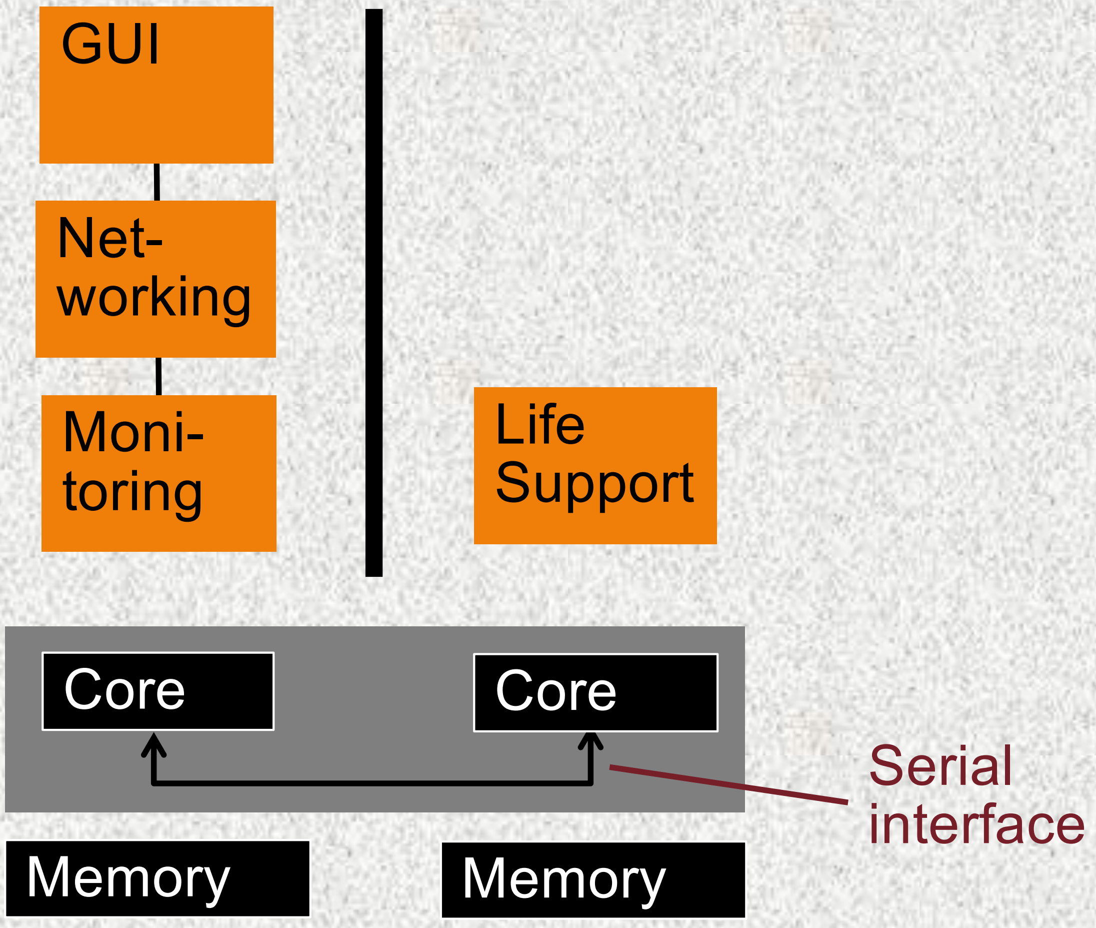
- Strong isolation
- Serial communication
- Cost:
  - BOM
  - Space
  - Low Bandwidth



Needed: Strong Protection

Physical separation on multicore

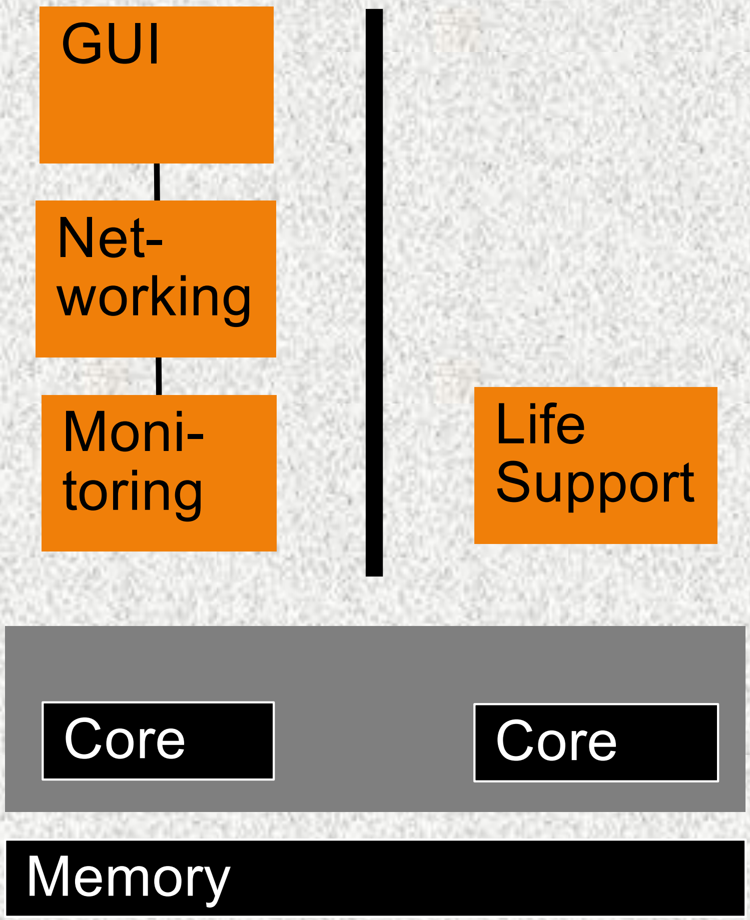
- Strong isolation if separated memory
- Cost:
  - BOM
  - Space
  - Medium Bandwidth



Needed: Strong Protection

Multicore w/o physical separation

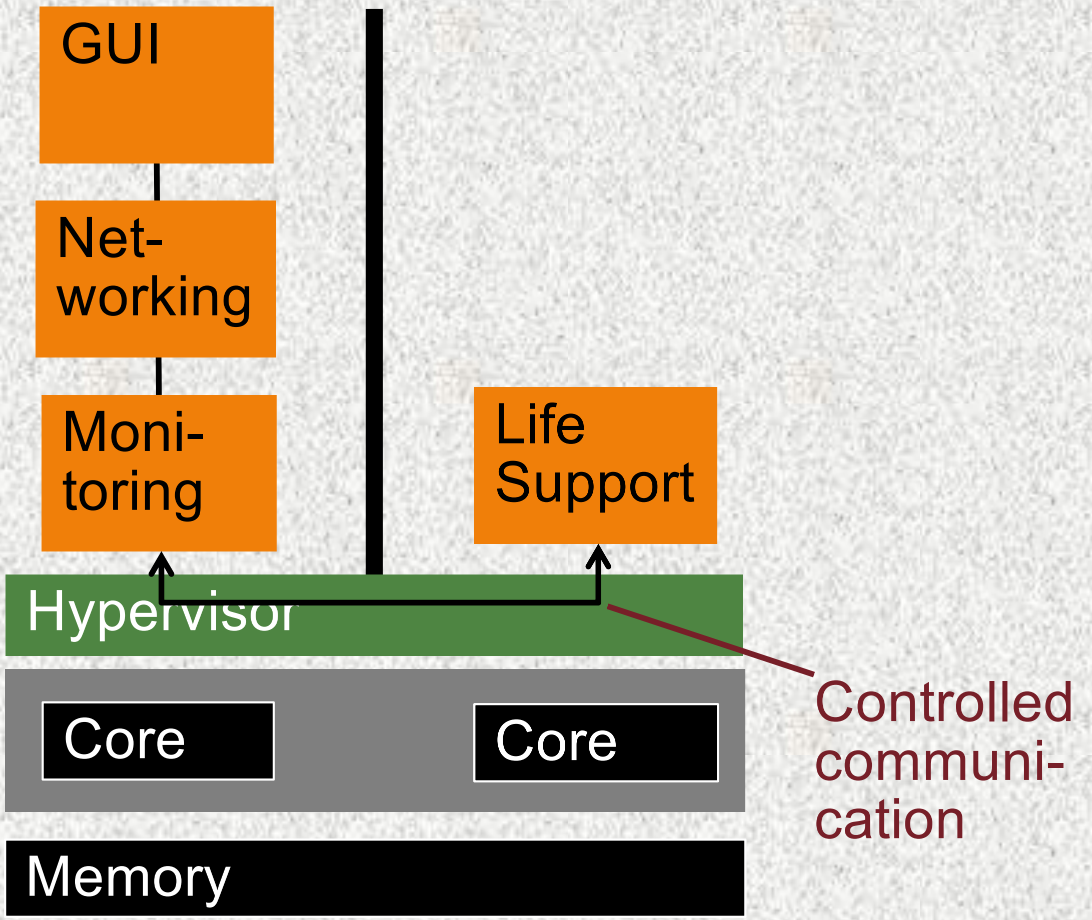
- High-bandwidth shared-memory communication
- No isolation!



Needed: Strong Protection

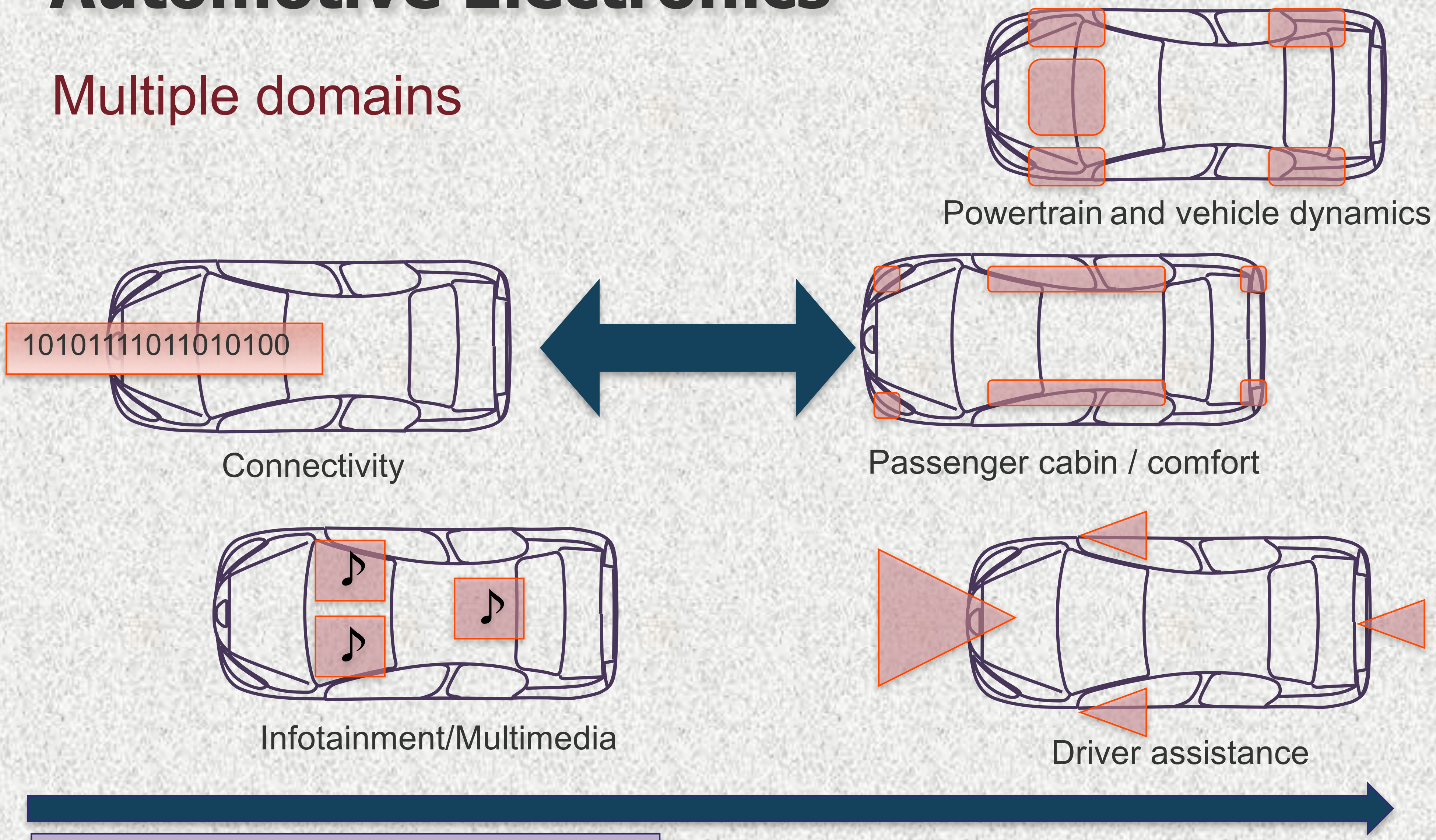
Single- or Multi-core with virtualization

- Strong isolation
- High-bandwidth shared-memory communication
  - Fine-grained control by hypervisor
- Min space, BOM



Automotive Electronics

Multiple domains



Powertrain and vehicle dynamics

Connectivity

Passenger cabin / comfort

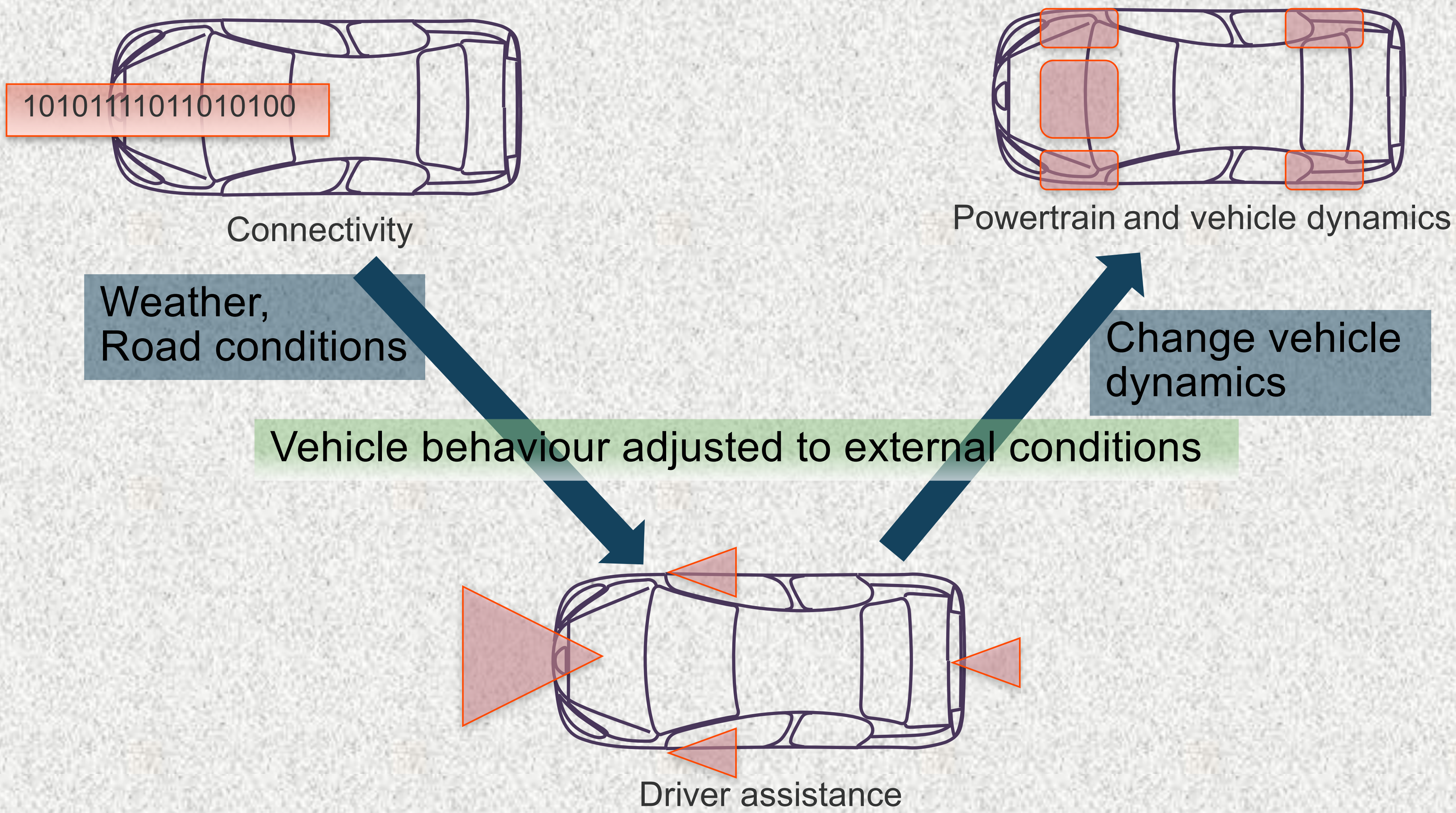
Infotainment/Multimedia

Driver assistance

Consumer Electronic Standards
Short life cycle

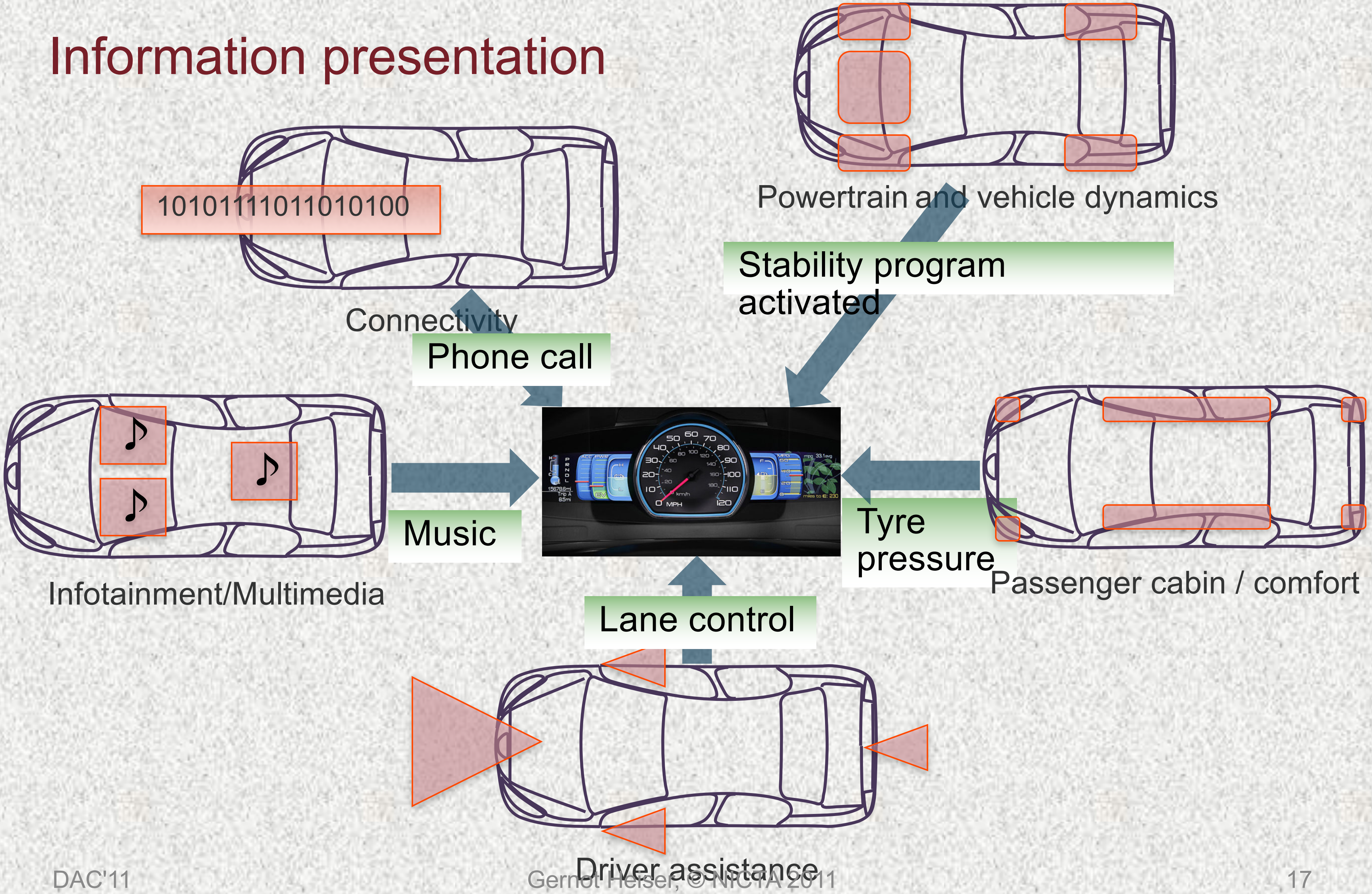
Car-industry controlled,
Long life cycle

Integration Across Domains



Integration Across Domains

Information presentation



10101111011010100

Connectivity

Powertrain and vehicle dynamics

Stability program activated

Phone call

Music

Infotainment/Multimedia

Tyre pressure

Passenger cabin / comfort

Lane control

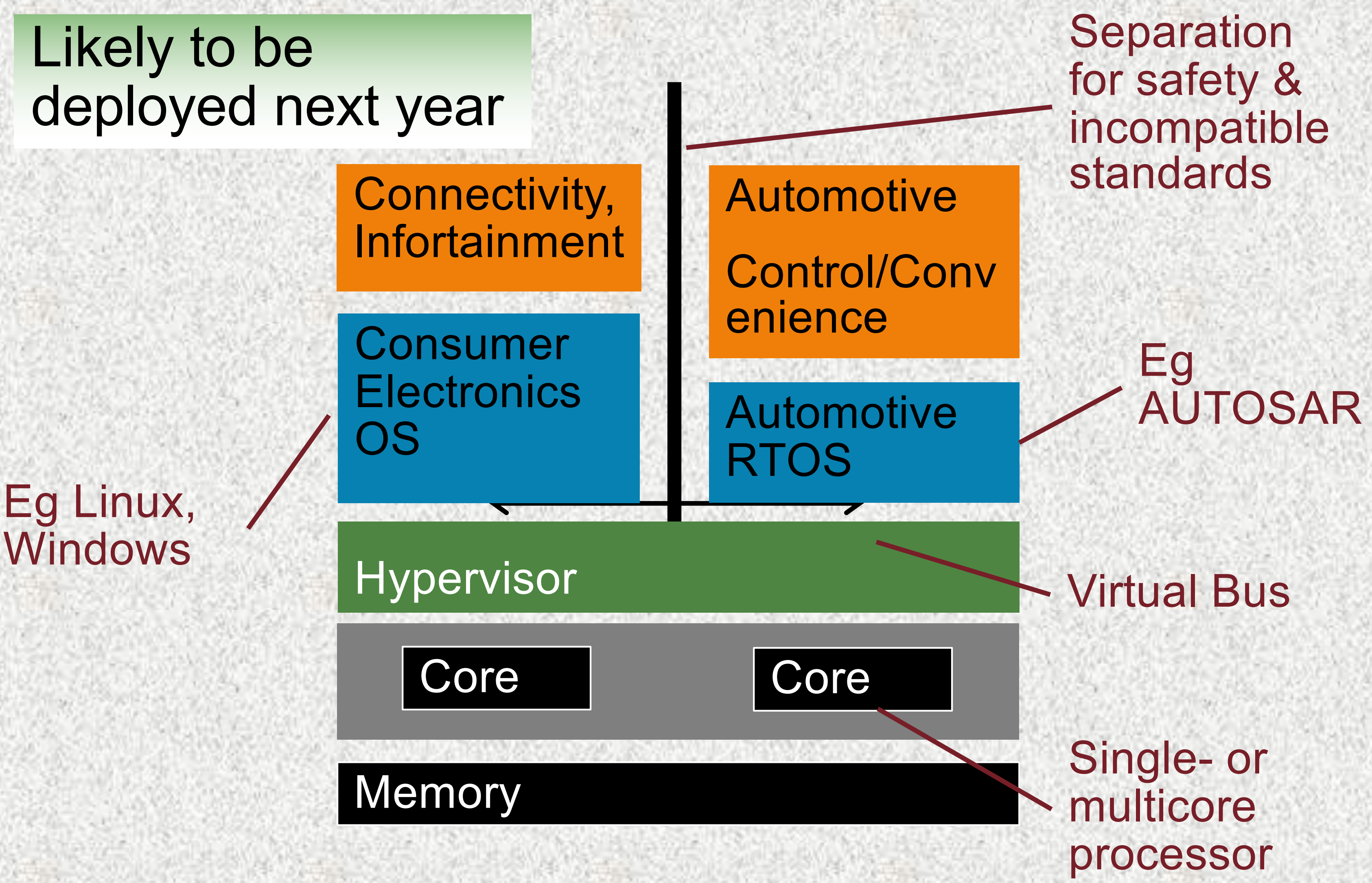
Driver assistance

Electronic Control Units (ECUs)

- BOM cost: $10s – $100s
- Dominated by balance-of-system
  - CPU is 10–20% of cost
- Expensive packaging
  - Heat resistant
  - Grease resistant
  - Acid resistant
  - Vibration resistant
  - Multiple communications interfaces
- Co-locating functionality saves $$, weight, space…

Functionality & Cost ⇒ Integration

Likely to be deployed next year



Virtualization Overheads

Example: netperf networking benchmark on Linux

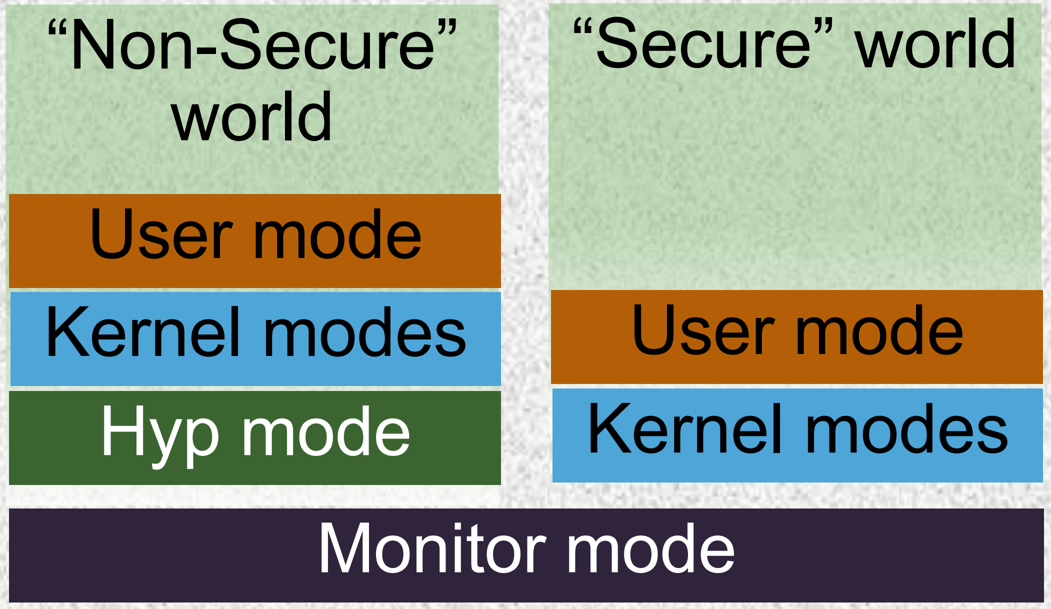
Type	Benchmark	Native	Virtualized	Overhead
TCP	Throughput	651 [Mib/s]	630 [Mib/s]	3 %
	CPU load	99 [%]	99	0 %
	Cost	12.5 [µ/KiB]	12.9 [µs/KiB]	3 %
UDP	Throughput	537 [Mib/s]	516 [Mib/s]	4 %
	CPU load	99	99	0 %
	Cost	15.2	15.8 [µs/KiB]	4 %

OKL4 Microvisor on Beagle Board (500 MHz Cortex A8 ARMv7)

Coming Up: Hardware Support

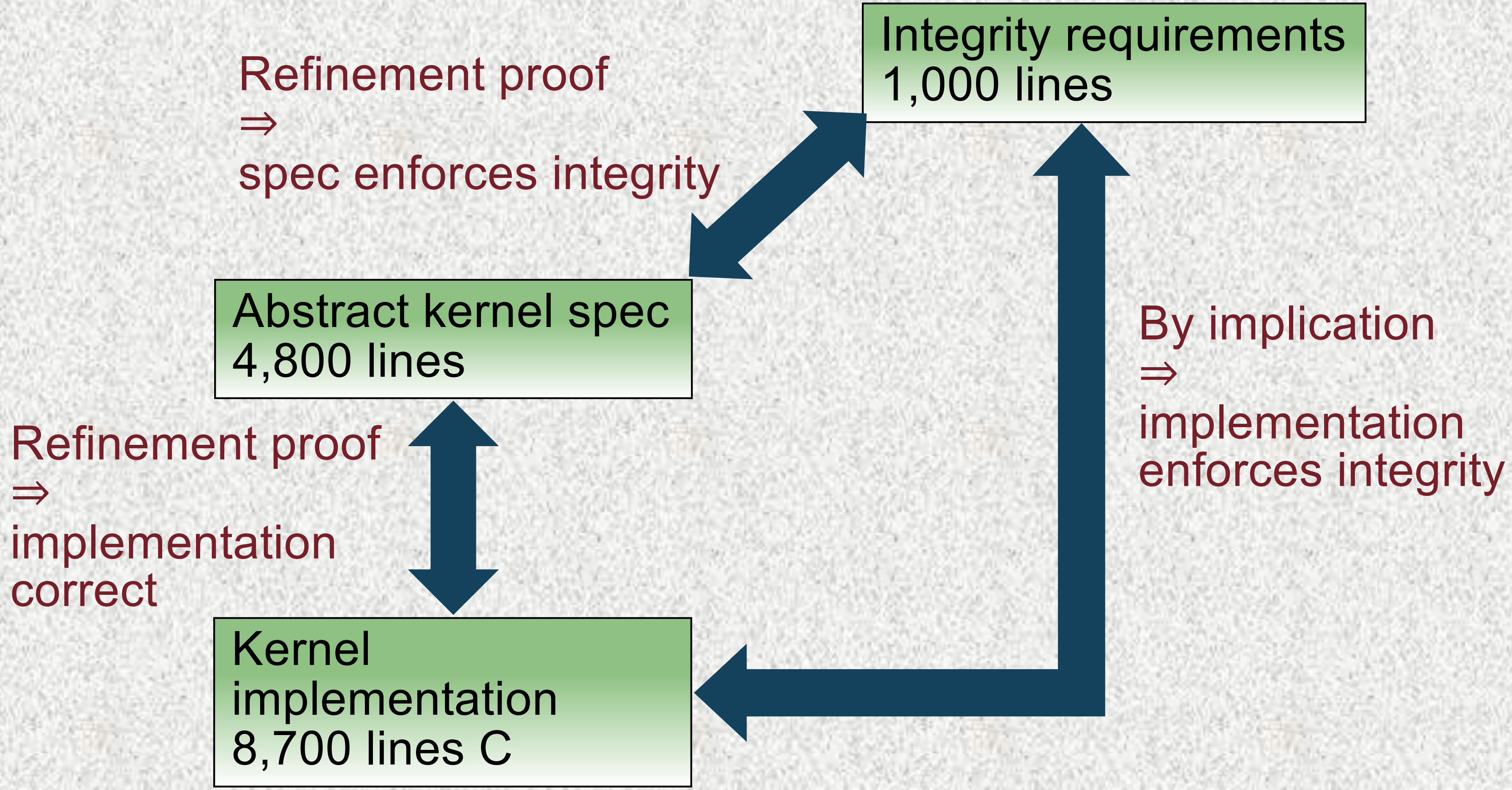
ARMv7 virtualization extensions announced Q3/2010

- Anticipate Si samples in 2011, products in 2012
- New “hyp” mode, various acceleration mechanisms



Future of Hypervisors: seL4 Microkernel

- Q: Can you trust separation by the hypervisor?
- A: Yes: we have proof!



Integrity requirements
1,000 lines

Refinement proof
⇒
spec enforces integrity

Abstract kernel spec
4,800 lines

By implication
⇒
implementation
enforces integrity

Refinement proof
⇒
implementation
correct

Kernel
implementation
8,700 lines C

Summary

- Virtualization in embedded systems is real
- Drivers:
  - Hardware utilization / multiplexing
  - Isolation for security
  - Isolation for safety
- Opportunity: small size enables correctness proofs
  - unprecedented trustworthiness