PHILIPS

Modular Communication-Centric MPSoC Architectures

Pieter van der Wolf
Philips Research

MPSoC'06
August 14 - 18, 2006

Outline

- Trends
- Key problems
- 5 key elements of solution
- Conclusion

Trends (Industry/Business)

- Uncertain and diversified markets
  - Late / changing product specs, short product life cycles
  - Different customers / tiers have different requirements
  - Winner takes all, high risks
- De-verticalization and specialization in industry
  - Drive towards open architectures and standards
    - For integration of own IP and IP from specialized providers in ecosystem
  - System solutions providers differentiating through:
    - Product architectures and integration skills & technology
    - Unique IP

Trends (Technical)

- Increasing complexity
  - At application, system, HW, and SW level
  - Increase in required know-how
    - Multitude of functions, standards and technologies to master
    - Single company cannot excel in all domains
  - Increasing NRE for SoC designs
    - Not compensated by higher volumes / higher margins
    - Shift in emphasis from BoM to NRE
  - Impacts product quality
    - Exploding costs for validation

Key Problems

- Cost of system design &integration is getting out of hand
  - Hundreds of man-years for HW/SW system
  - Diversity of products
- Product lead times too long and unpredictable
  - Missing market opportunities because of TTM
- Loosing grip on product quality (non-functional properties)
  - Exploding costs for validation
  - Degrading reliability and predictability
- High cost of ownership of hardware and software
  - Large portfolios of IPs
  - Key IPs require continuous innovation

PHILIPS

5 Key Elements of Next Generation Platforms

1. Modular systems based on coarse-grain subsystems

inter

2. Scalable & efficient communication infrastructure

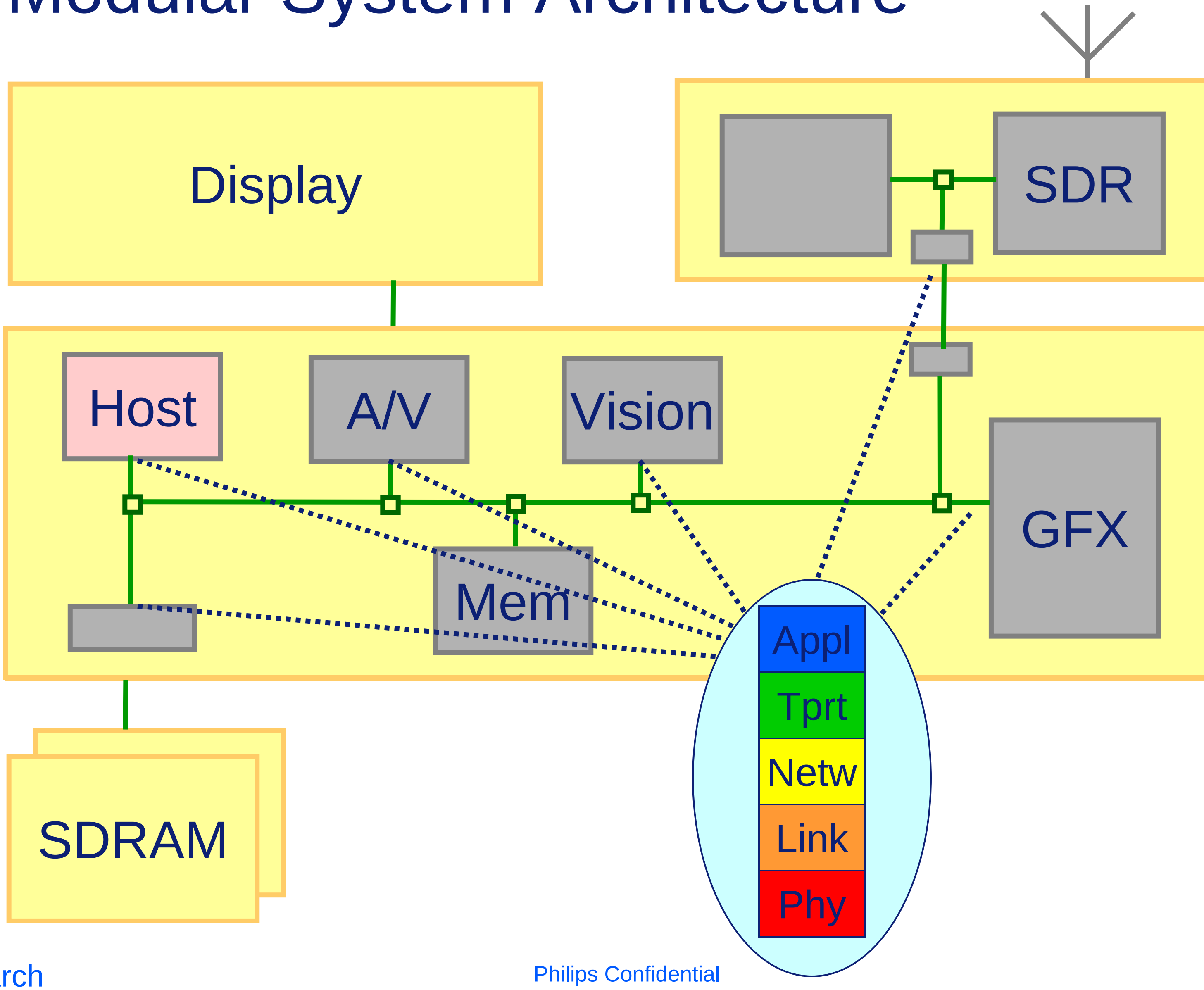
3. Open standards for communication interfaces

4. Controlled sharing of mem, interconnect, power, ..

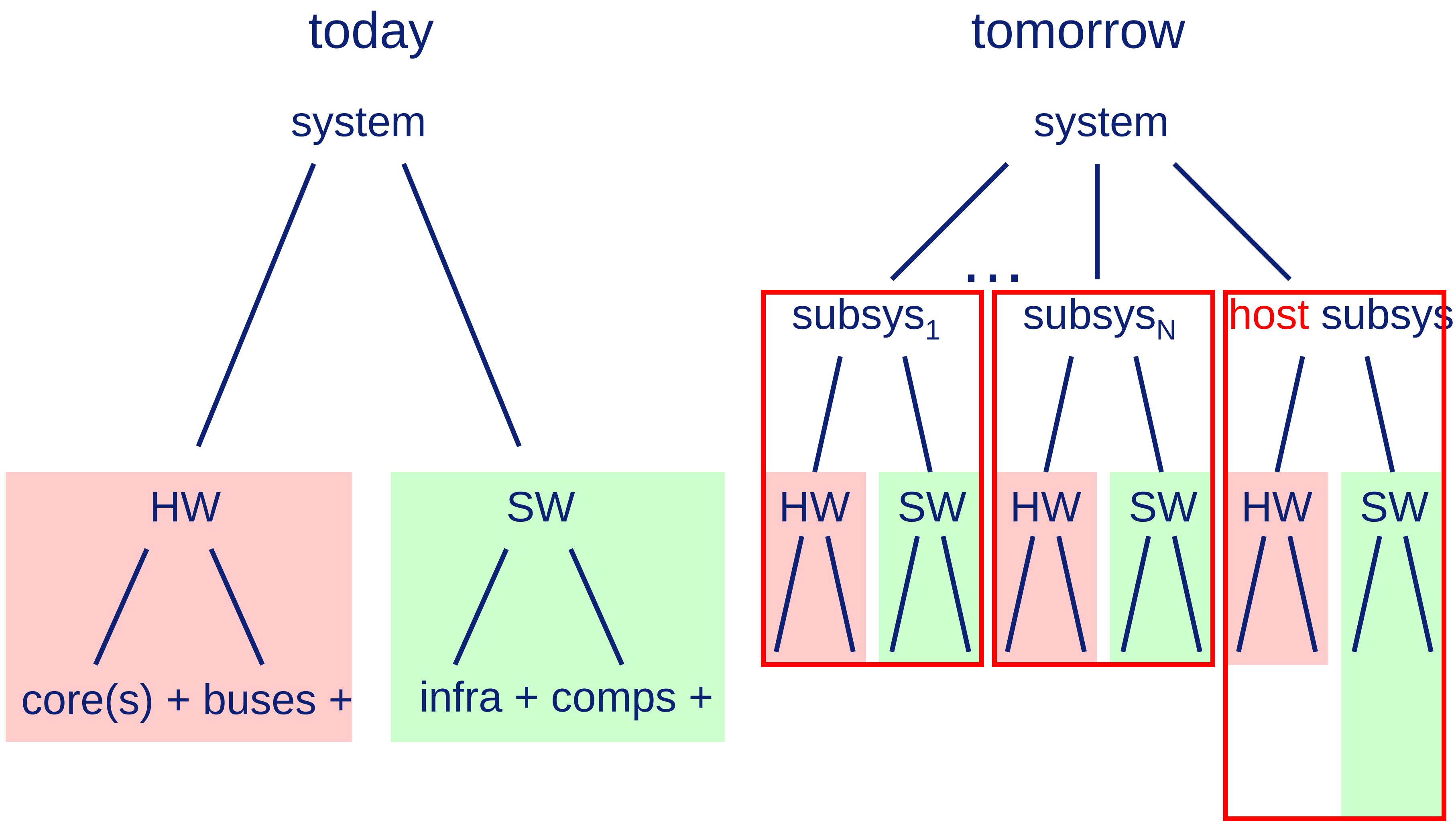
intra

5. Programmable architectures for subsystems

1. Modular System Architecture



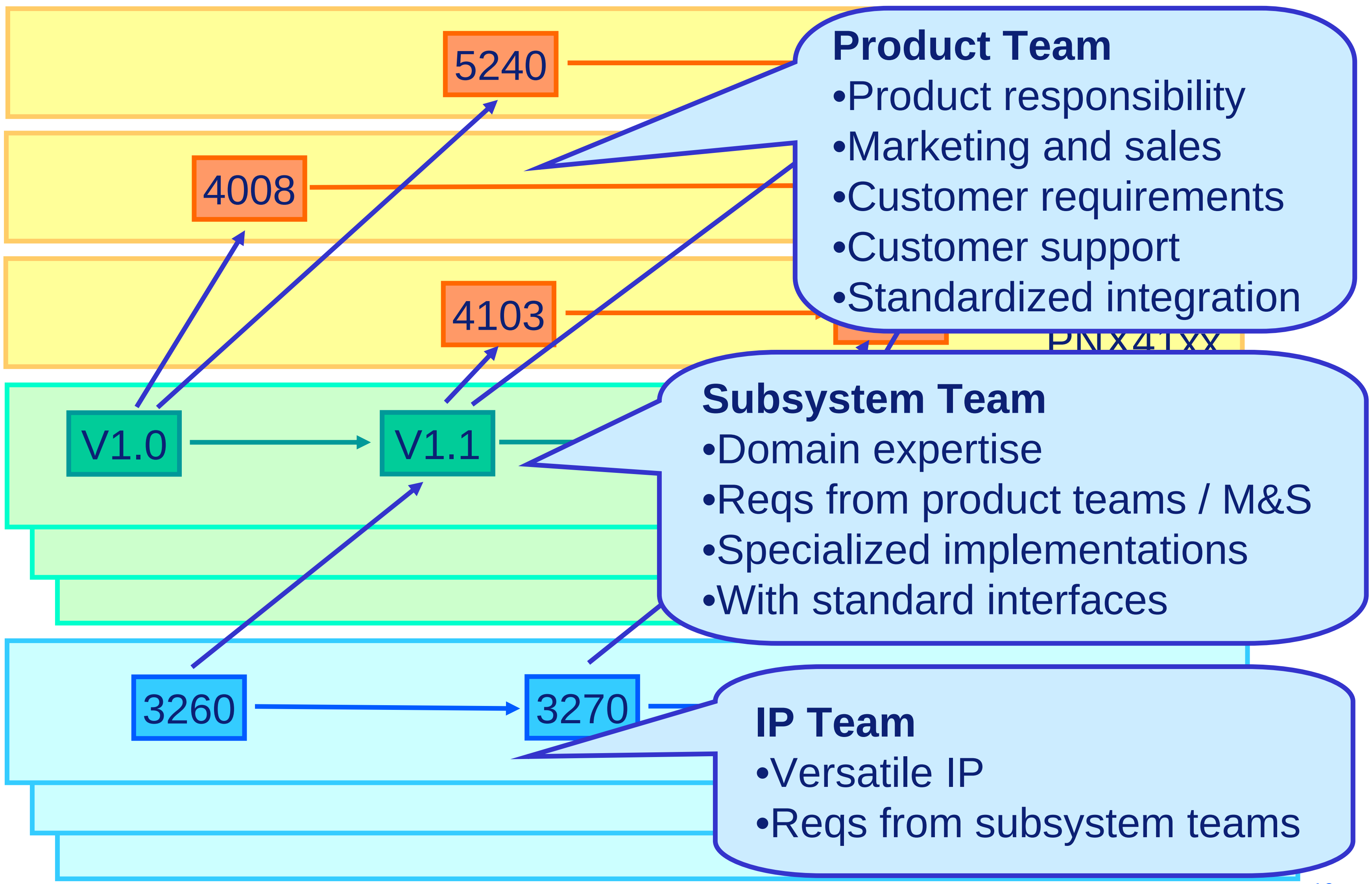
System architecting



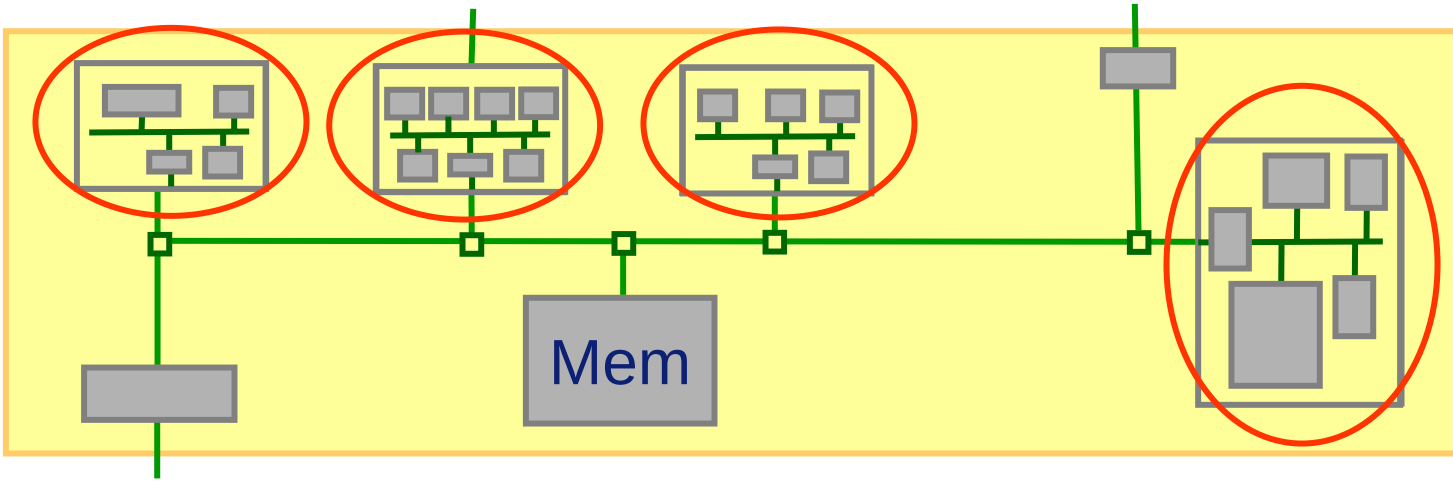
SubSystems are...

- Technically: units of
  - Functionality — data sheets
  - Low latency (“islands of ~”) — clock domains
  - Integration, optimization, and validation — building blocks
  - HW-SW integration — deeply embedded firmware
  - Interoperability — well-defined interfaces
- Organizationally: units of
  - Roadmapping — independent life-cycles
  - Expertise — modems, video, 3D
  - Organization — sites & teams
  - Standardization — “OpenGL-compliant 3D”
  - De-verticalization — make-or-buy

Organization: Products, Subsystems and IP



1. Modular systems with coarse-grain subsystems

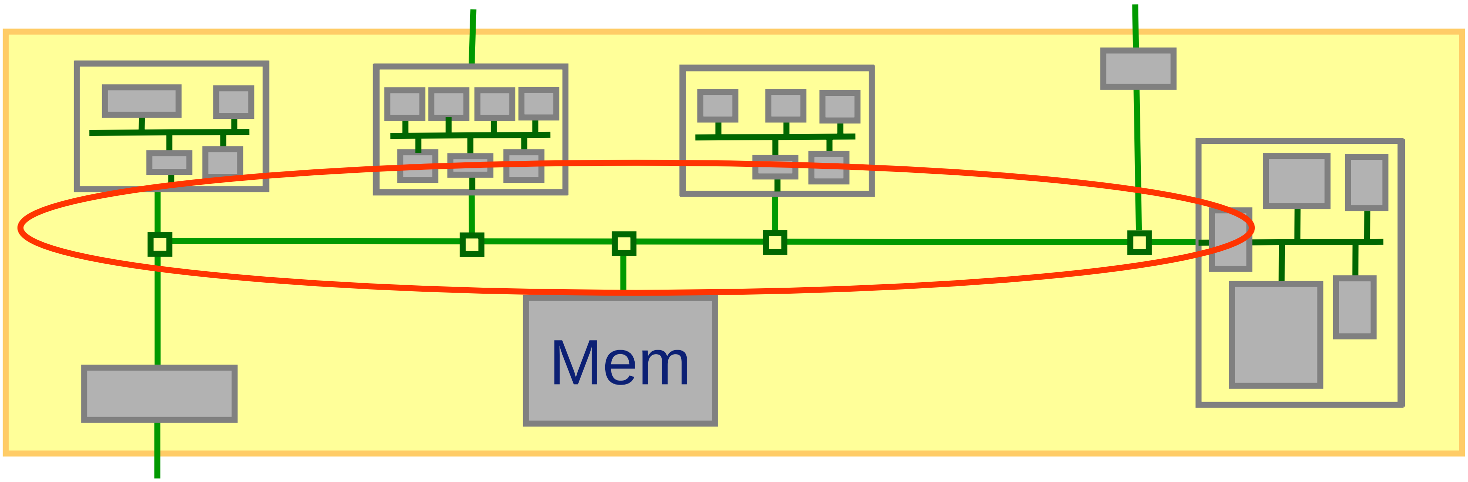


- Towards more autonomous, self-contained functional subsystems
- "Black box" with high-level interface to infrastructure
- Private internal SW/HW structure; legacy friendly
- Pre-integrated / pre-validated
- Limited sharing of resources: carefully trade BoM for NRE

Benefits:

- Specialized system know-how, implementation technology, life-cycles
- Autonomy and modularity ease integration of subsystem
- Optimized islands of low latency

2. Scalable and efficient communication infrastructure

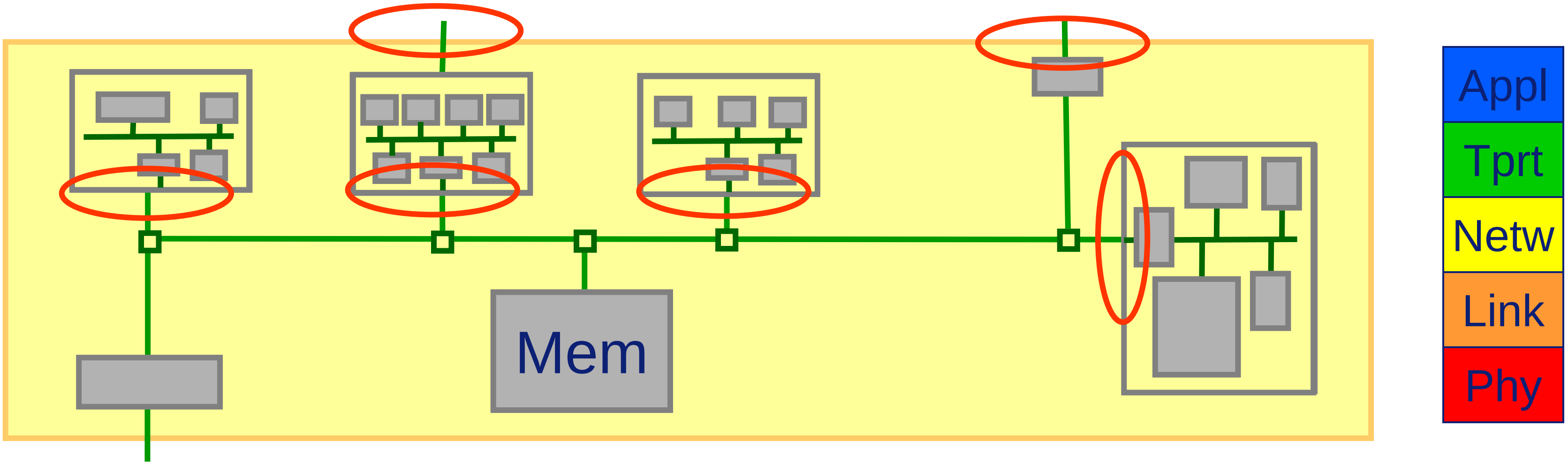


- Scalable network with QoS support and multiple traffic classes
- … including on-chip and off-chip memory traffic
- Avoid bandwidth bottlenecks and offer low latency access to code/data
- Transparent chip boundaries for easy repartitioning

Benefits:

- Scalability supports product families / product roadmaps
- QoS support for predictable system performance
- On-chip memory helps avoid bottleneck to off-chip SDRAM

3. Open standards for communication interfaces

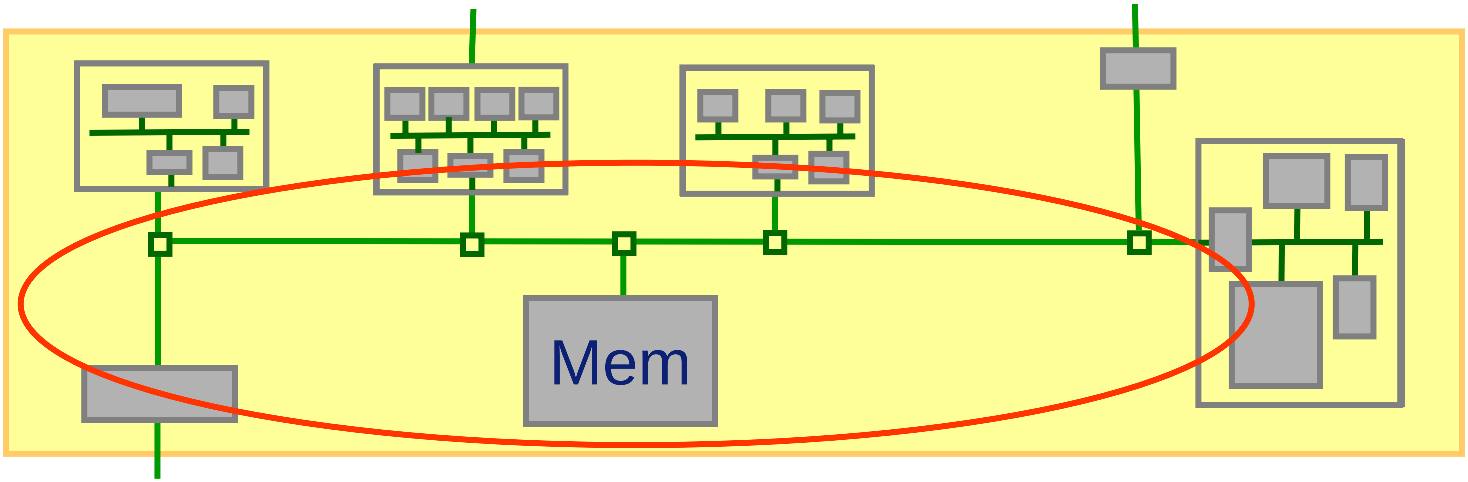


- Standardized protocols for interoperability and easy mix&match
- Separate inter and intra subsystem architectures and implementation
- High-level control and communication interfaces (SW and HW)
- Note: Also support cost-effective integration of legacy IP

Benefits:

- Interoperability of subsystems from different sources, easy mix&match
- Use of industry standards lowers entry barrier for new customers
- High-level interfaces offer stability and freedom of implementation

4. Controlled sharing of resources

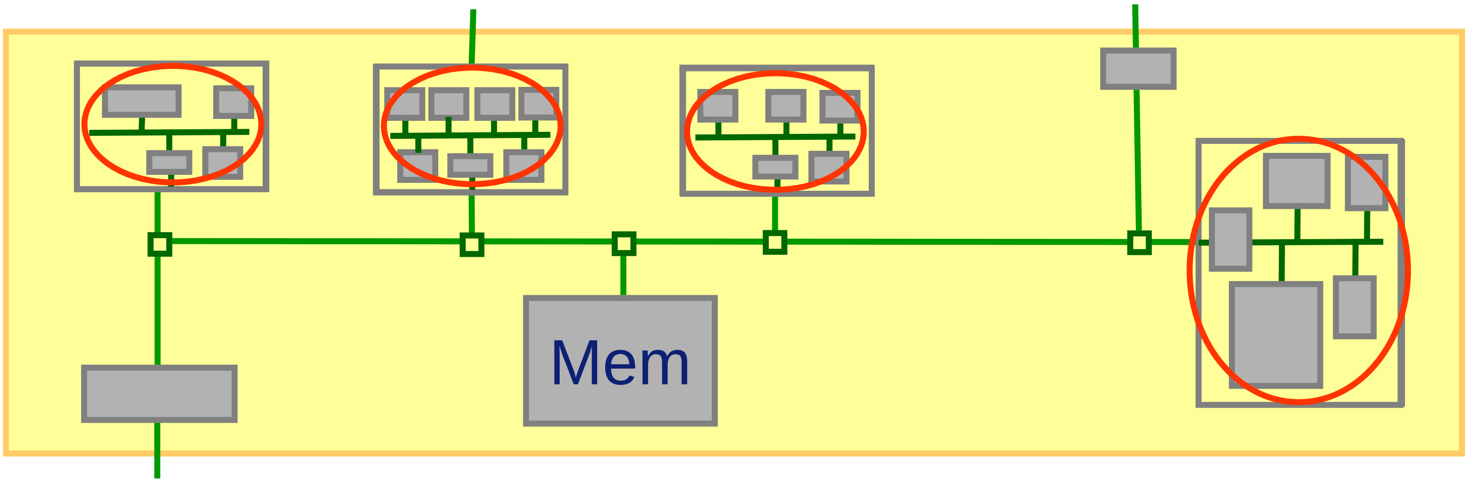


- Controlled resource sharing: memory, SDRAM, bandwidth, power
- Virtualization of resources (resources are virtually “private”)
- Models and tools for systematic integration

Benefits:

- Allows sharing for cost reduction and enhanced flexibility
- “Controlled” to prevent unintended interference among subsystems
- Supports compositional design (easy to add/remove functions)
- Predictable system performance

5. Programmable architectures for subsystems



- Specialized multi-processor architectures and software architectures
  - Examples: cache-coherent SMP, hard/soft RT
- Use of industry standard OS's, compiler frameworks, tools, etc.
- Constraint: at interfaces compliant with inter-subsystem standards

Benefits:

- Specialized technology can be used for differentiating subsystems
- Flexibility can be targeted towards multi-xx subsystem functionality
- Standards compliancy guarantees easy integration; faster innovation

Conclusion

Subsystems (HW + SW):

= extra layer in system hierarchy

= extra unit of re-use & integration

- with well-defined interoperability interfaces
- based on programmable architectures

connected

- by a standardized / open communication infrastructure
- with controlled sharing of memory, interconnect, power

etc

Benefits (w.r.t key problems)

- ↓ design / integration costs
- ↓ lead times & ↑ predictable process
- ↓ cost of ownership
- improved grip on product quality

Key Research Topics

- Memory architectures
  - SDRAM bandwidth bottleneck
  - Latencies of memory accesses
- On/off-chip networks
  - For range of traffic types, with QoS guarantees
    - Including protocols for control, power management, debug
  - Transparency of chip boundaries (MIPI)
- Predictable system integration
  - Virtualization of interconnect and memory
    - Reduce interference via shared resources
  - Method for reasoning about system performance
- Power consumption
  - Use case driven power management
- ....

PHILIPS

PHILIPS

sense and simplicity