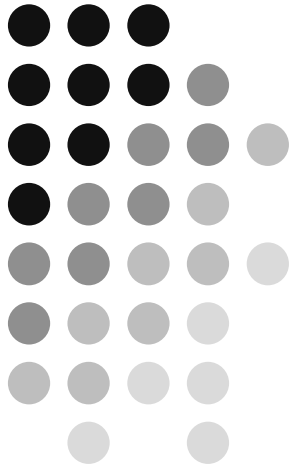
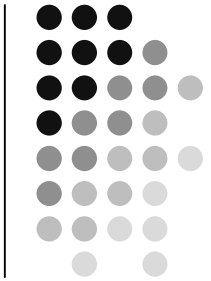
PreMaDoNA
kickoff 15 oct 2004

Project Overview



Henk Corporaal

Agenda



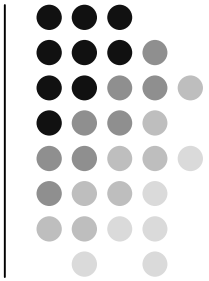
• 15.00 Opening and Overview	Henk Corporaal Bart Mesman
• 15.30 Implementation and Demonstrator	Patrick Groeneveld
• 15.40 Project Management	Bart Mesman
• 15.55 Application track	Peter de With Gerard de Haan
• 16.05 Simulation track	Sander Stuijk Orlando Moreira
• 16.15 *5 minutes coffee/thea break*	
• 16.20 Network Architecture Layer	Jef van Meerbergen Kees Goossens Marco Bekooij
• 16.30 Design Flow	Bart Theelen Bart Mesman
• 16.40 Resource Management and QoS	Jef van Meerbergen Milan Pastrnak Peter Poplavko
• 16.50 Discussion	
• 17.00 Reception	
• 17.30 Close	

PreMaDoNA

Predictable Matching
of Demands on
Networked Architectures

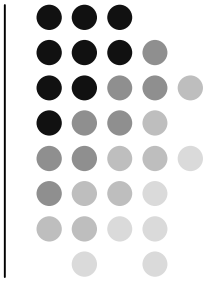
- Partners:
  - Philips Research
  - Philips Semiconductors
  - LogicaCMG
  - TU/e

Overview



- Problem statement
- NOC
- Predictable design
  - QoS management
  - Design flow
  - Architectural support for predictable design
- Work packages
- Links with other projects (intern; extern)

Problem statement



- Observations
- Problem
- Solution

Observation 1: The 3 Cs

- Convergence of *3 Cs*

  computers, communications and consumer electronics

- The computer enters the *3rd fase*

  computing power - networking - intelligent processing

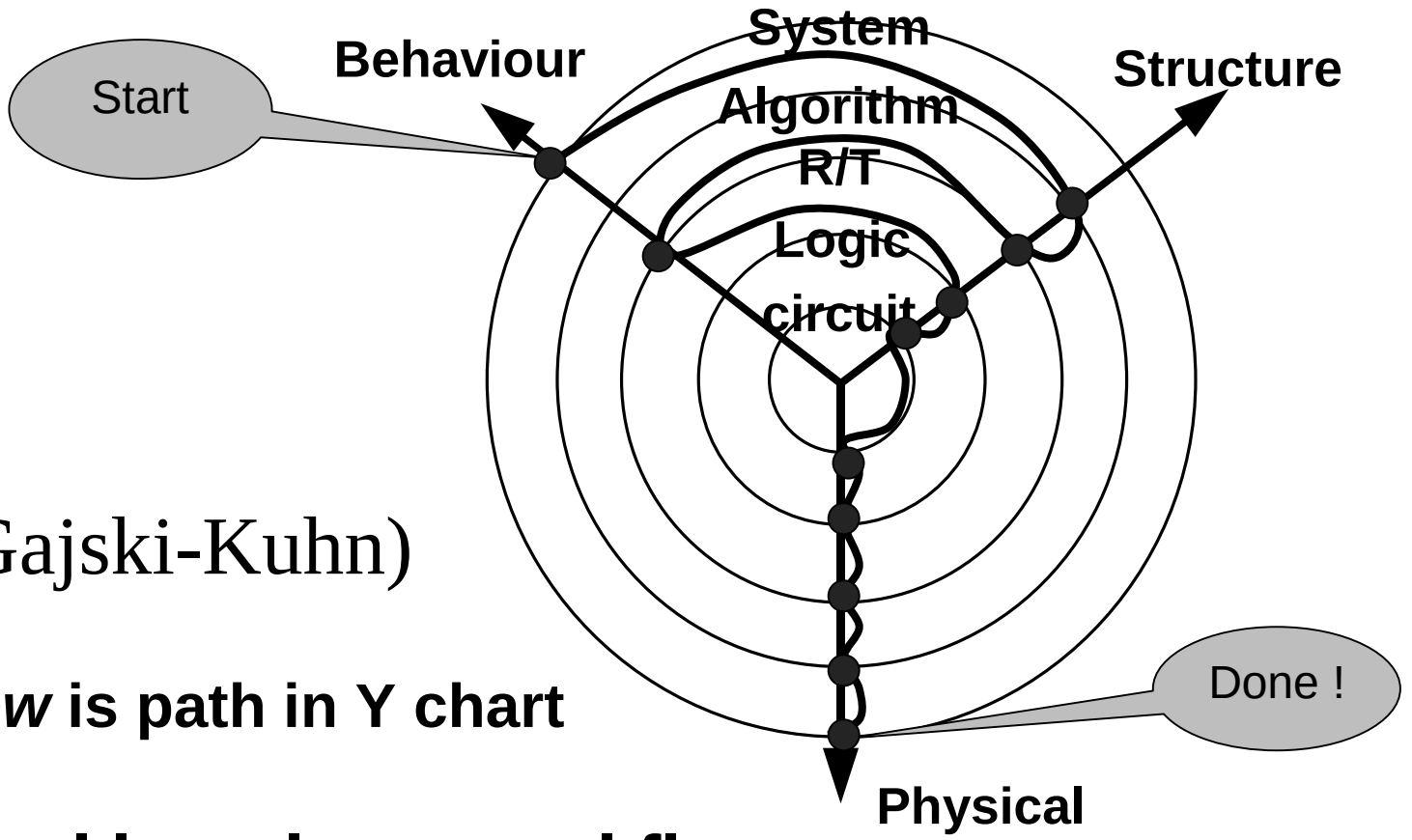
- The world is *1 network*

  wherever, whenever, all information and communication available

We get a smart environment

- **with short time-to-market**
- **with many wearable and wireless sub-systems**

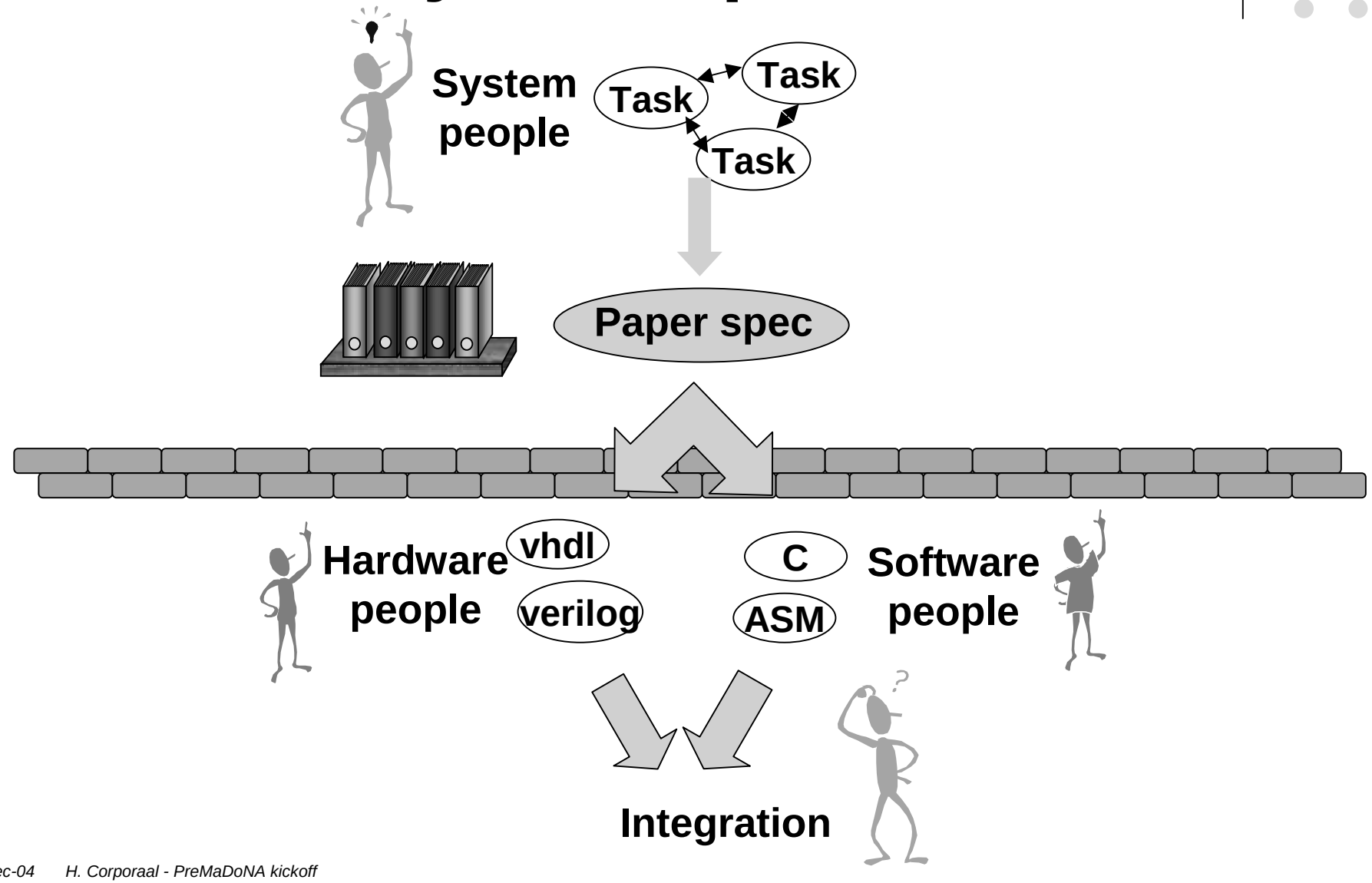
Observation 2: Current HW design practise



Y-Chart (Gajski-Kuhn)

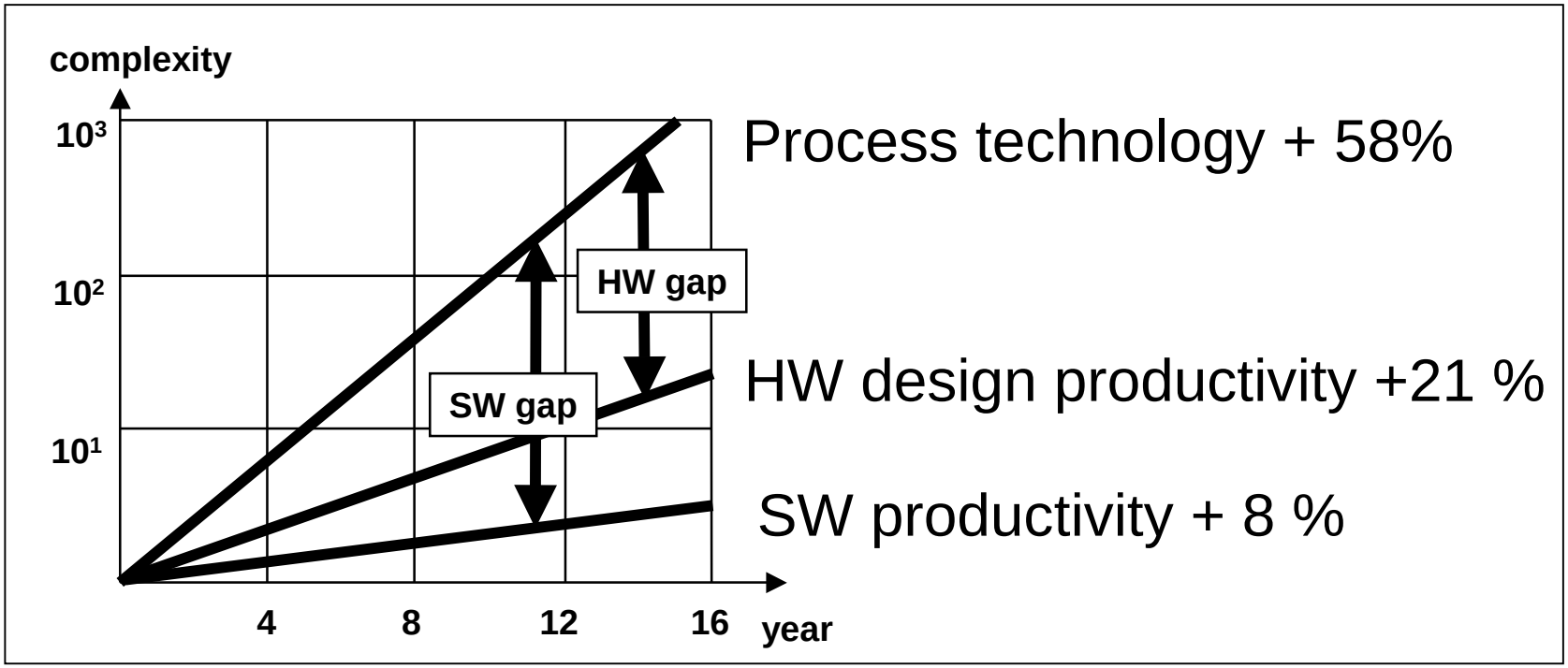
- ***Design Flow*** **is path in Y chart**
- **Till RT-level largely manual flow**

Observation 3: Informal system specification



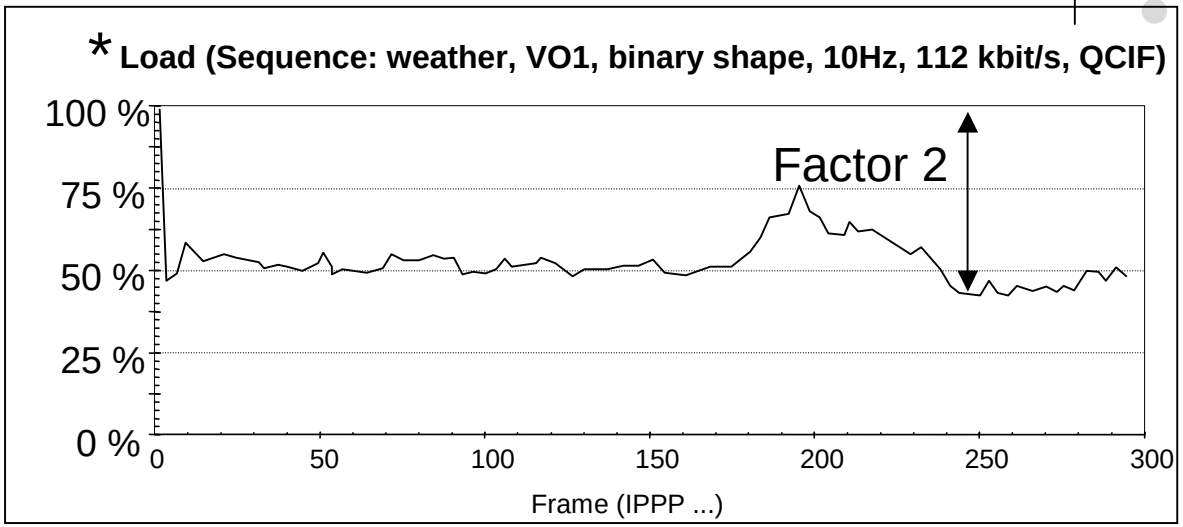
Observation 4: Design productivity

- Yes, we can fabricate the ICs, but …
- Can we design them ?
- Can we program them ?



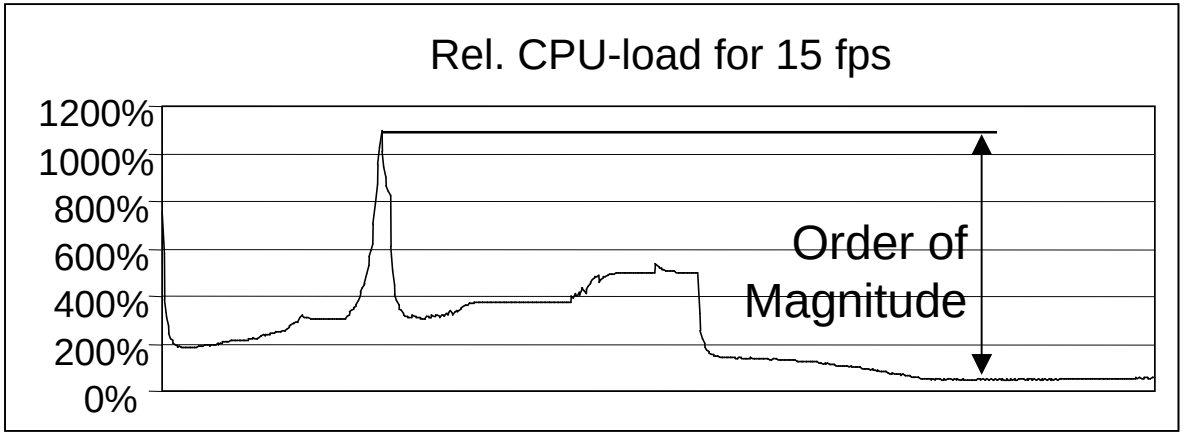
Obervation 5: More dynamism

Video

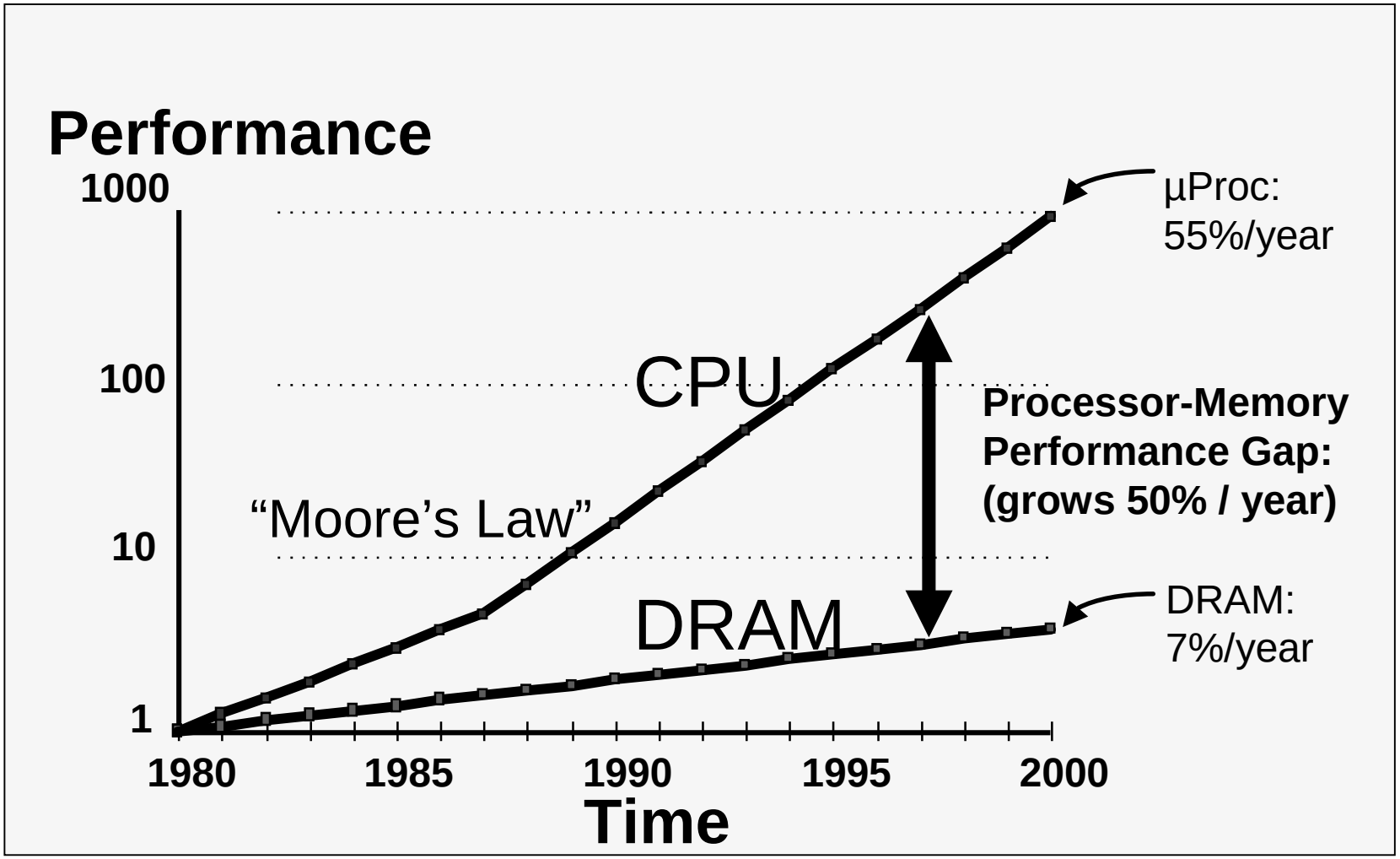


* **P. Kuhn, G. Diebel, "Complexity Analysis of the MPEG-4 VM 8.0," ISO/IEC JTC1/SC29/WG11/MPEG97/m2862, Fribourg, October 1997**

3D

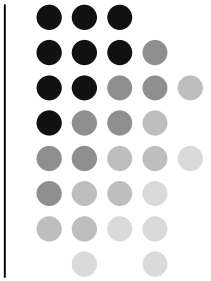


Observation 6: Memory gap
“data where are you?”



[Patterson]

What’s the design problem?

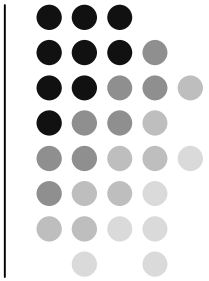


Given an incredible complex system, finish the design

- Yesterday
- with Zero power
- with Zero cost
- with QoS guarantees

- **At sufficient performance !**

Solution ?

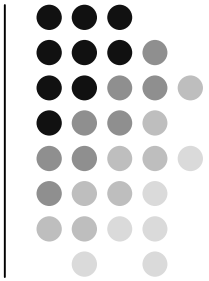


1. Platforms
   - HW and SW IP reuse
   - Standardization (interfaces)
   - Scalability and Flexibility

2. Advanced Design Flow for Platforms
   - Raise abstraction level
   - Tool support
   - Modeling of Power, Cost, Performance

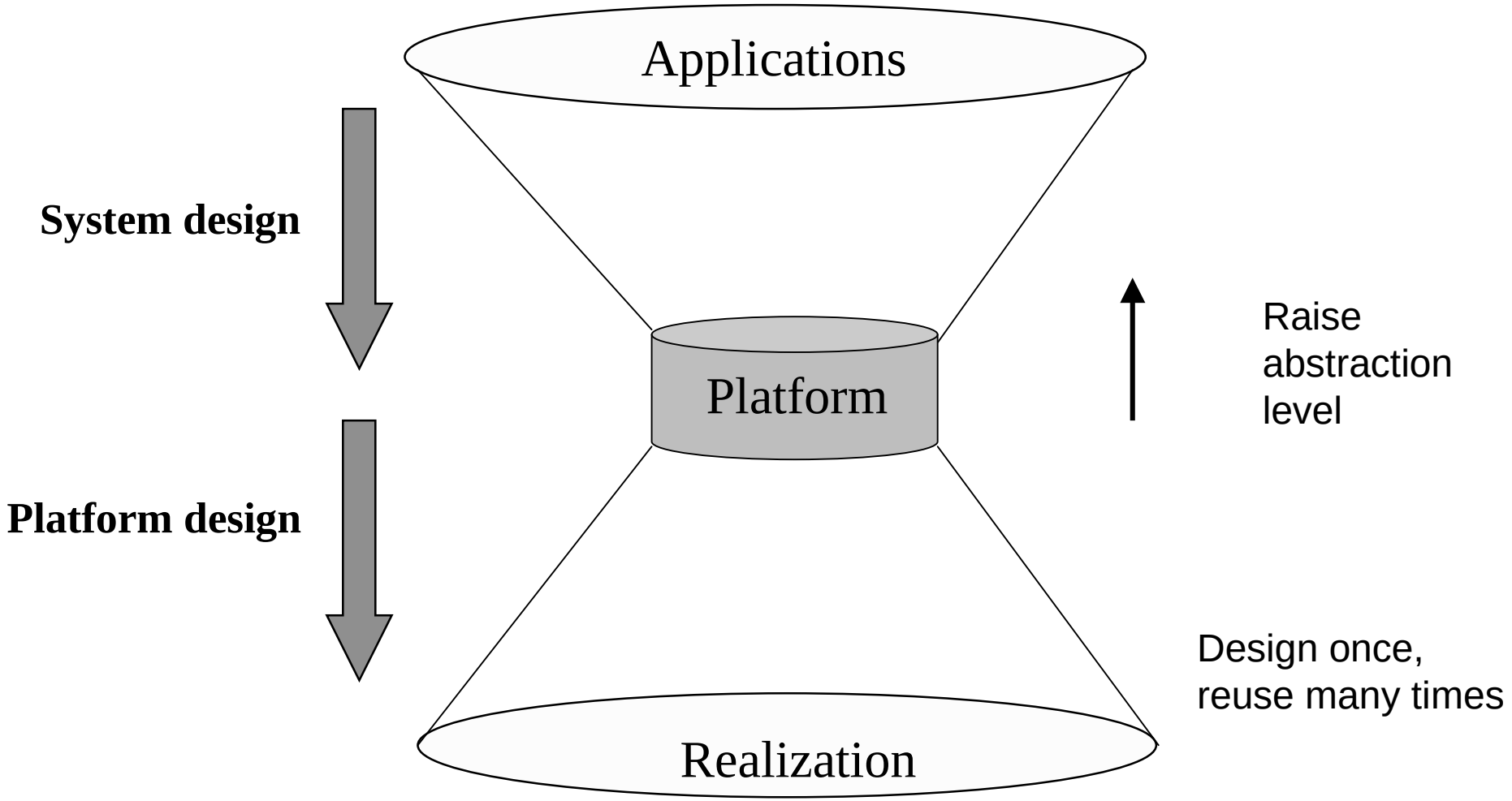
3. Predictability
   - QoS (quality of service) hooks
   - Reason about design properties at all levels

Platform characteristics

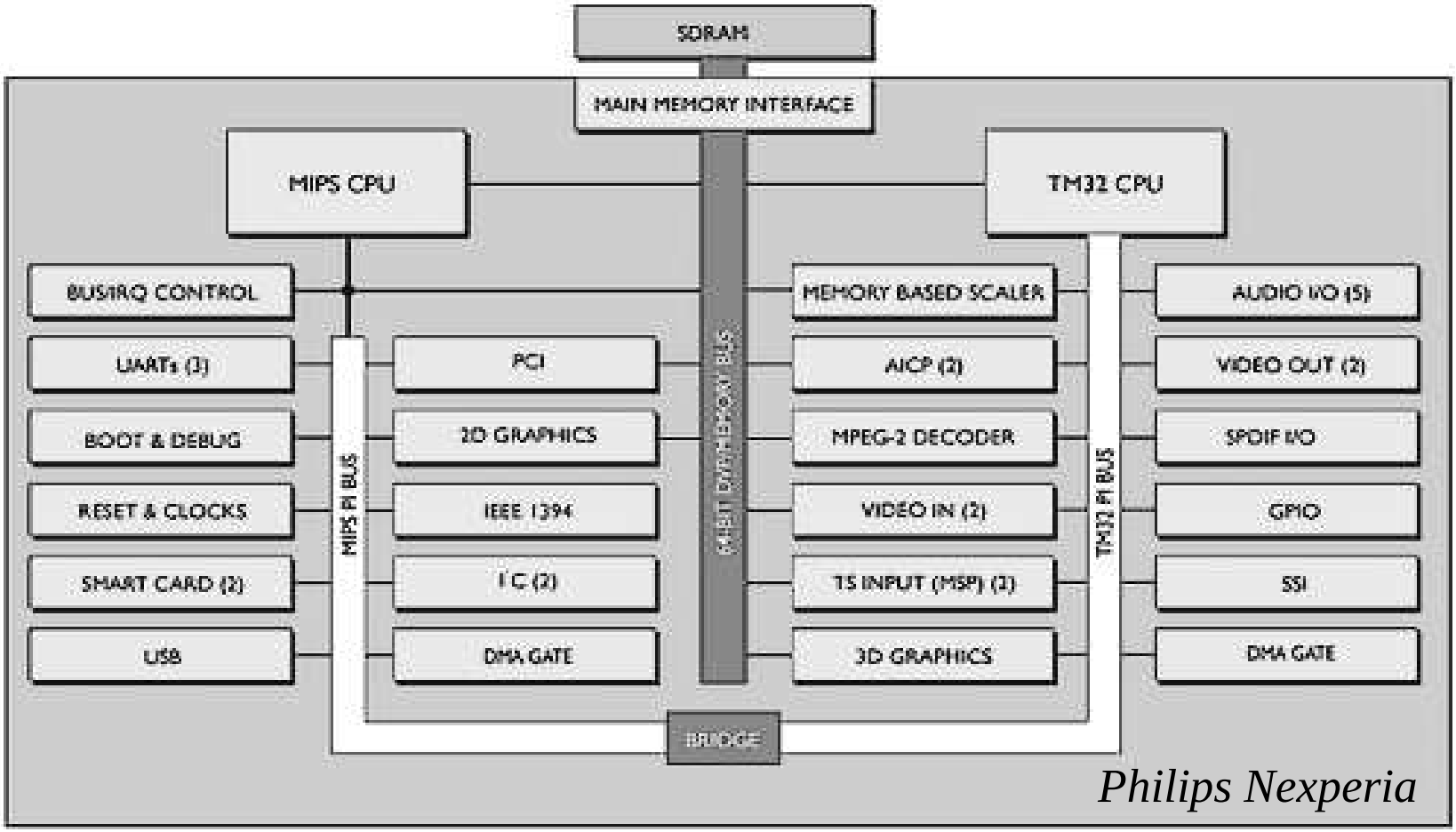


- Programmable
  - One or more processor cores
- Reconfigurable
- Scalable and flexible
- Memory hierarchy
  - Exploit locality
- Separate local and global wiring
- HW and SW IP reuse
  - Standardization (on SW and HW-interfaces)
  - Raising design abstraction level
- Reliable
- Cheaper

Platform and platform design

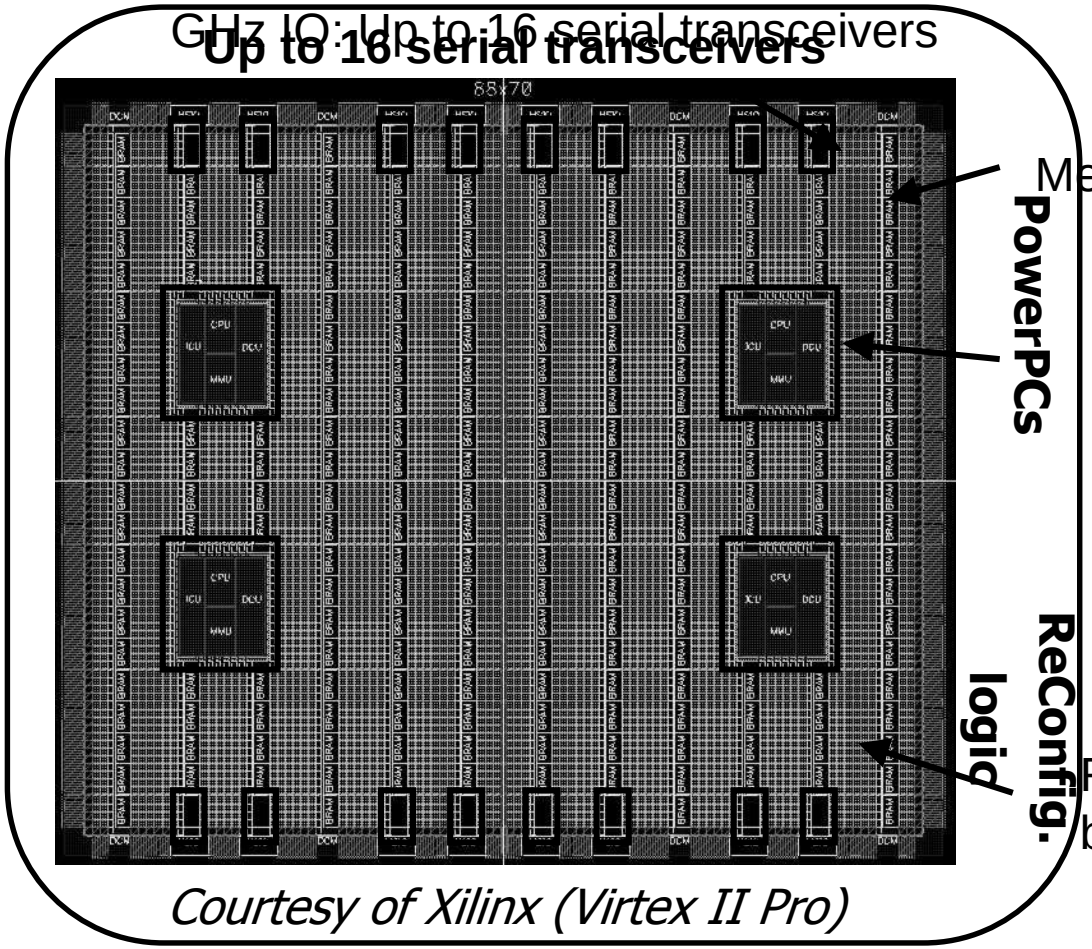


Example Platforms: Bus-based: Philips Nexperia



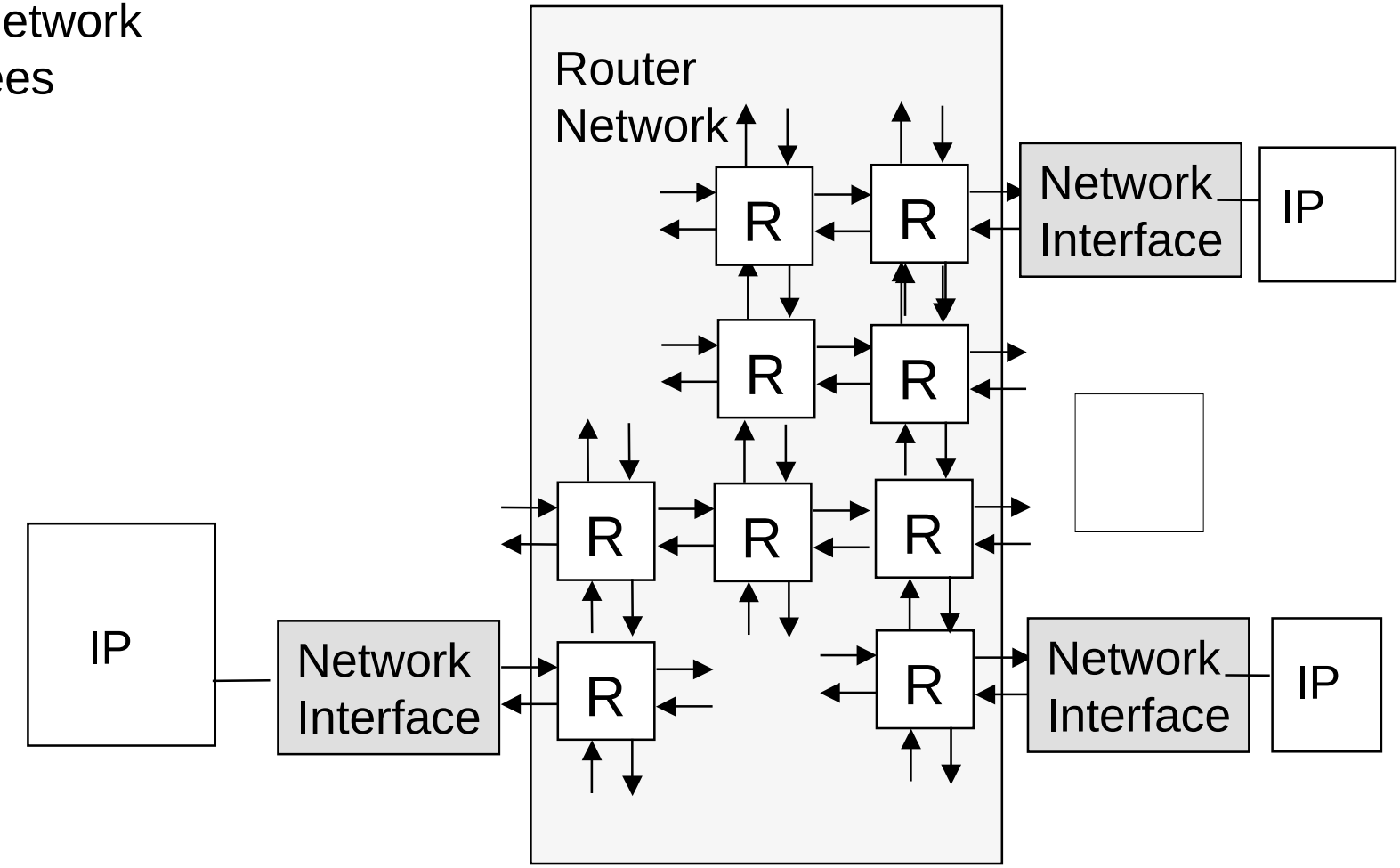
Philips Nexperia

Reconfigurable logic based: Xilinx Virtex II-Pro



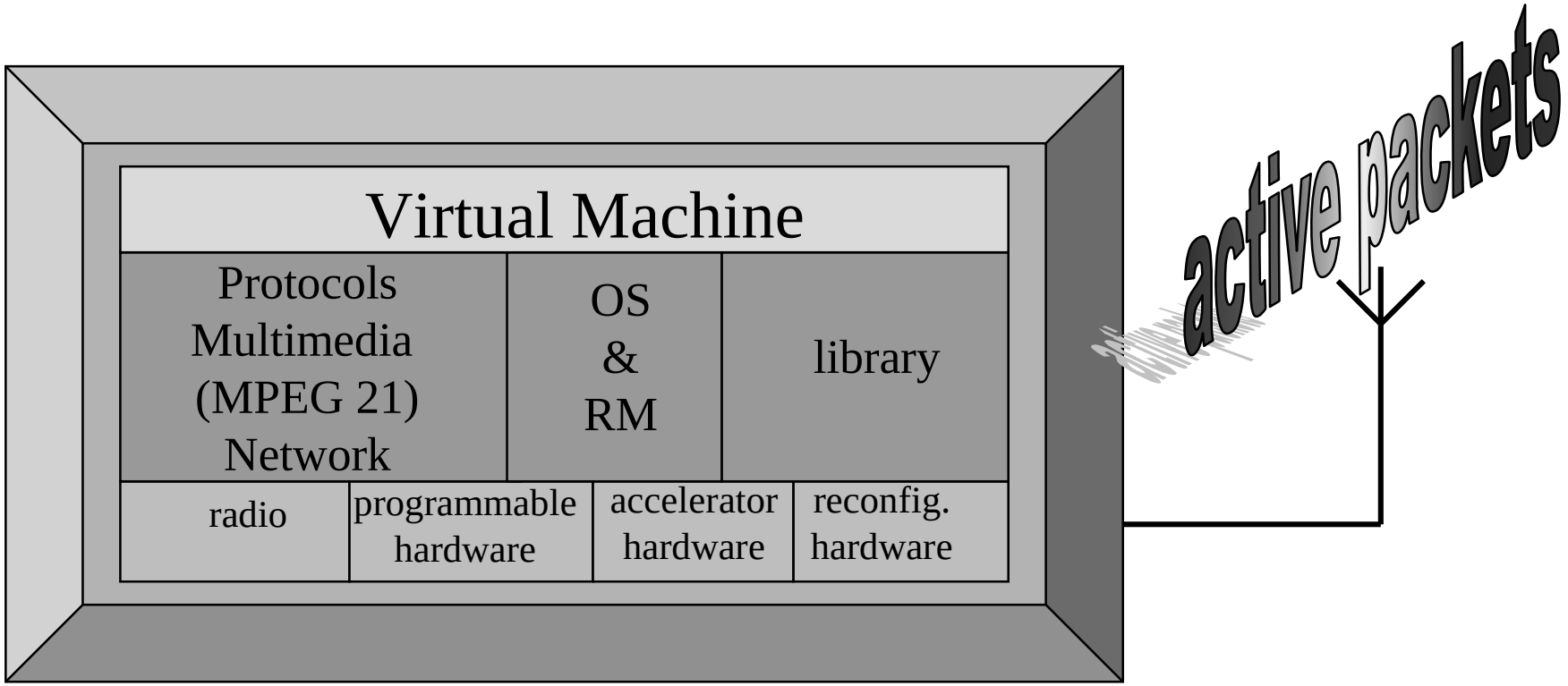
NoC based: Philips AETHEREAL

Built-in network guarantees

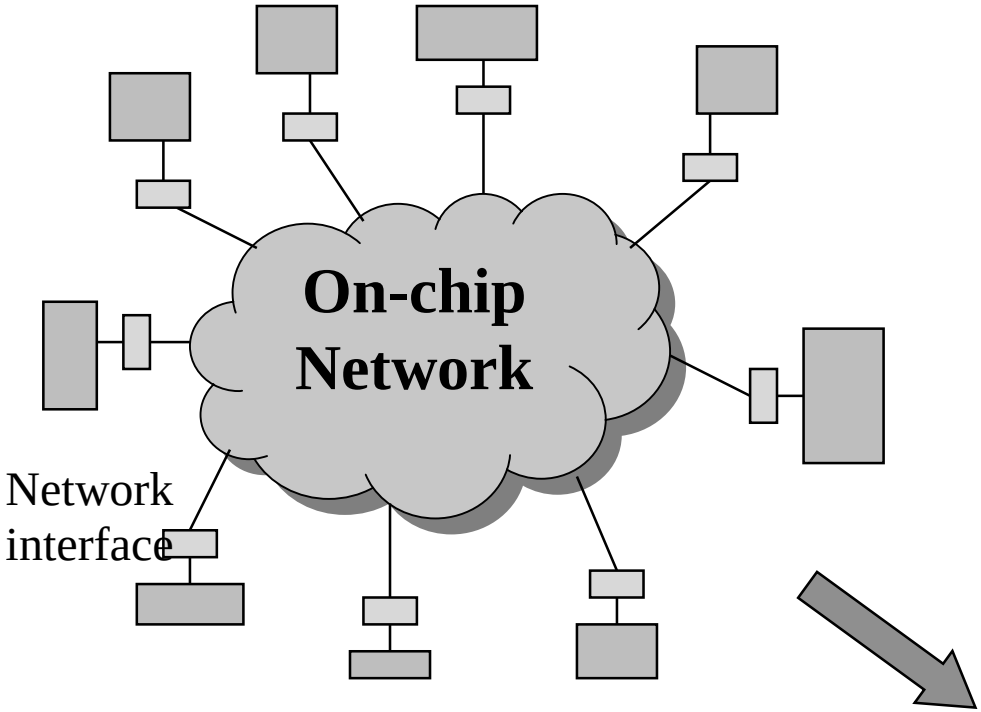


Future platforms
it's not only about HW !

Example: Smart Networked Devices

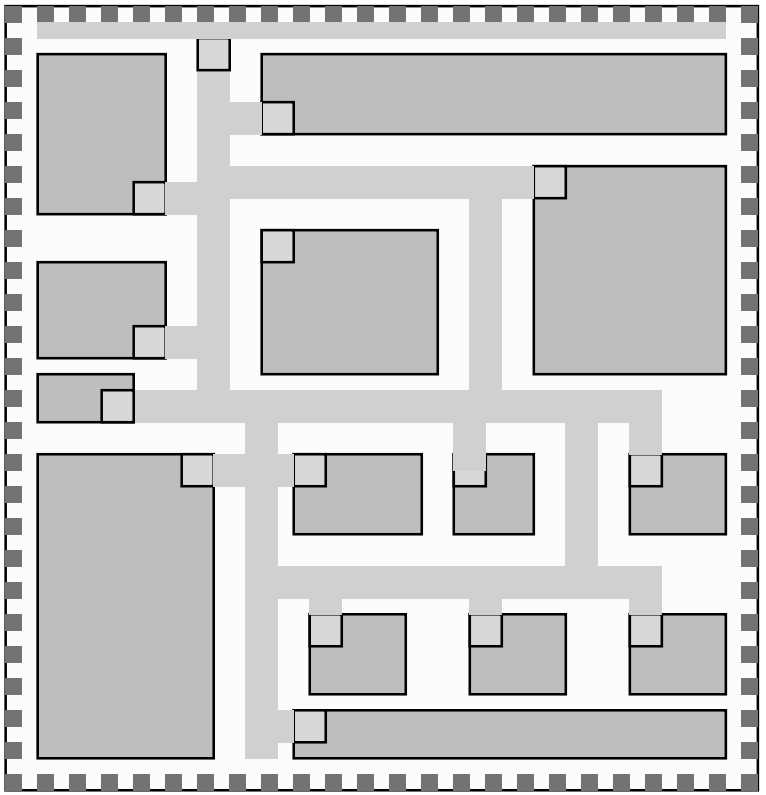


Future platforms: realization

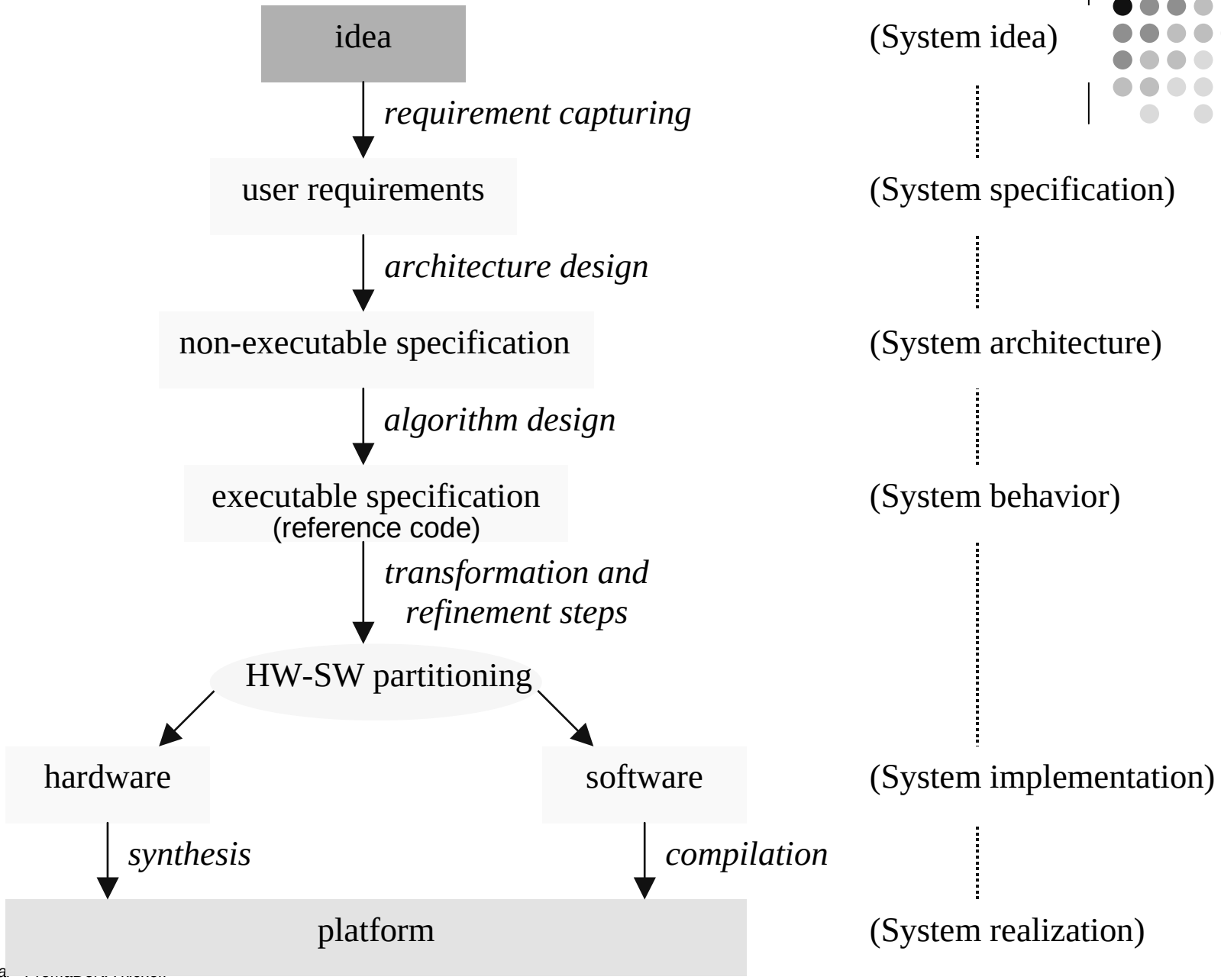


- IP - Isles:
  - 32 RISC microprocessor ~ 20 Kgates
  - MPEG decoding ~ 100 Kgates
  - Wavelet filtering ~ 40 Kgates
  - SRAM
  - DRAM
  - FPGA block

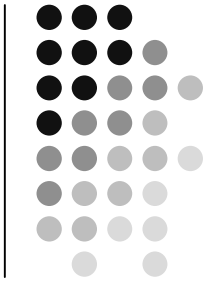
NoC realization

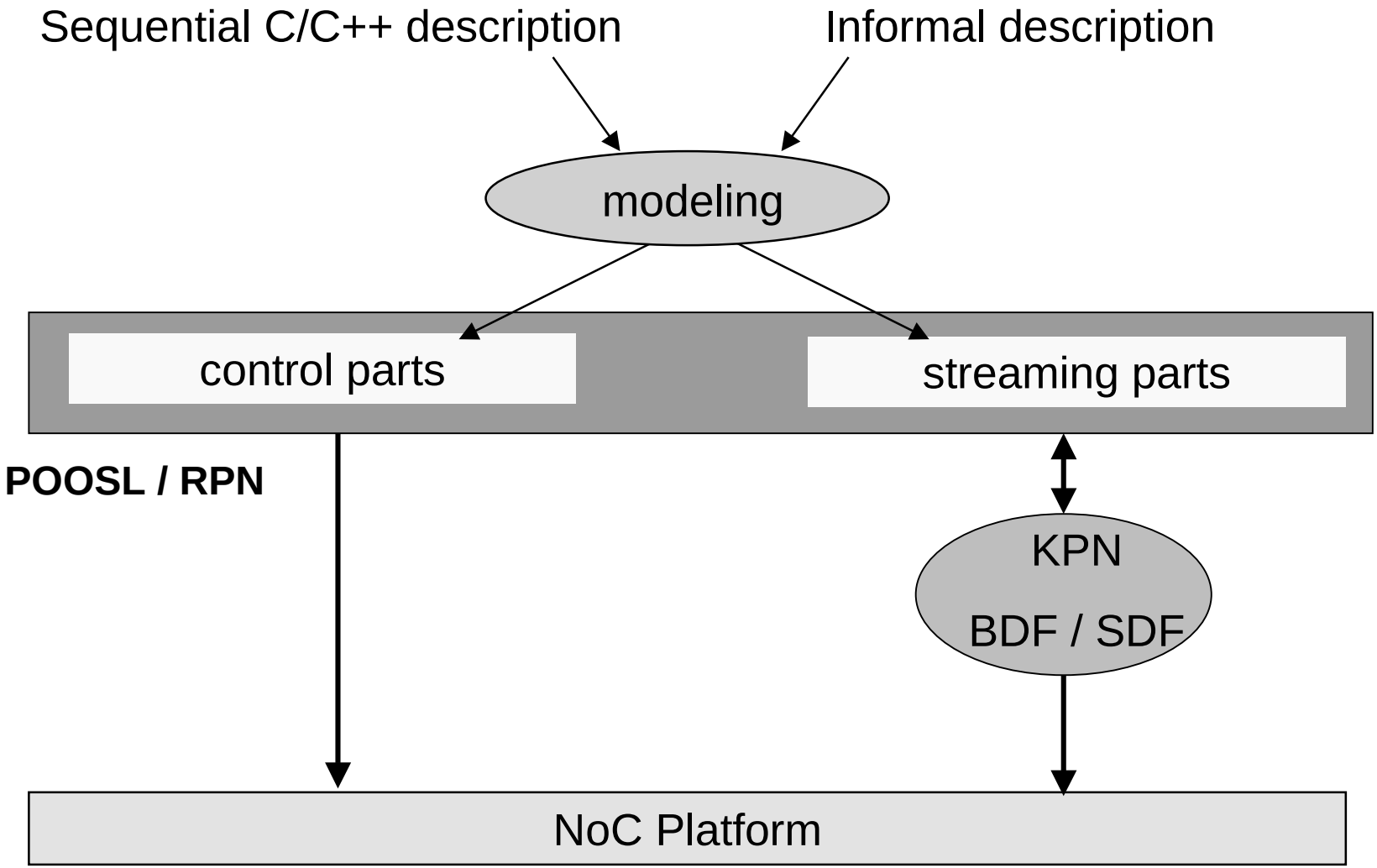


System Design trajectory

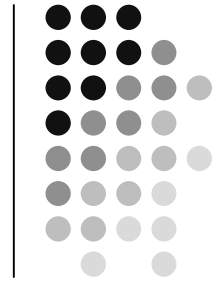


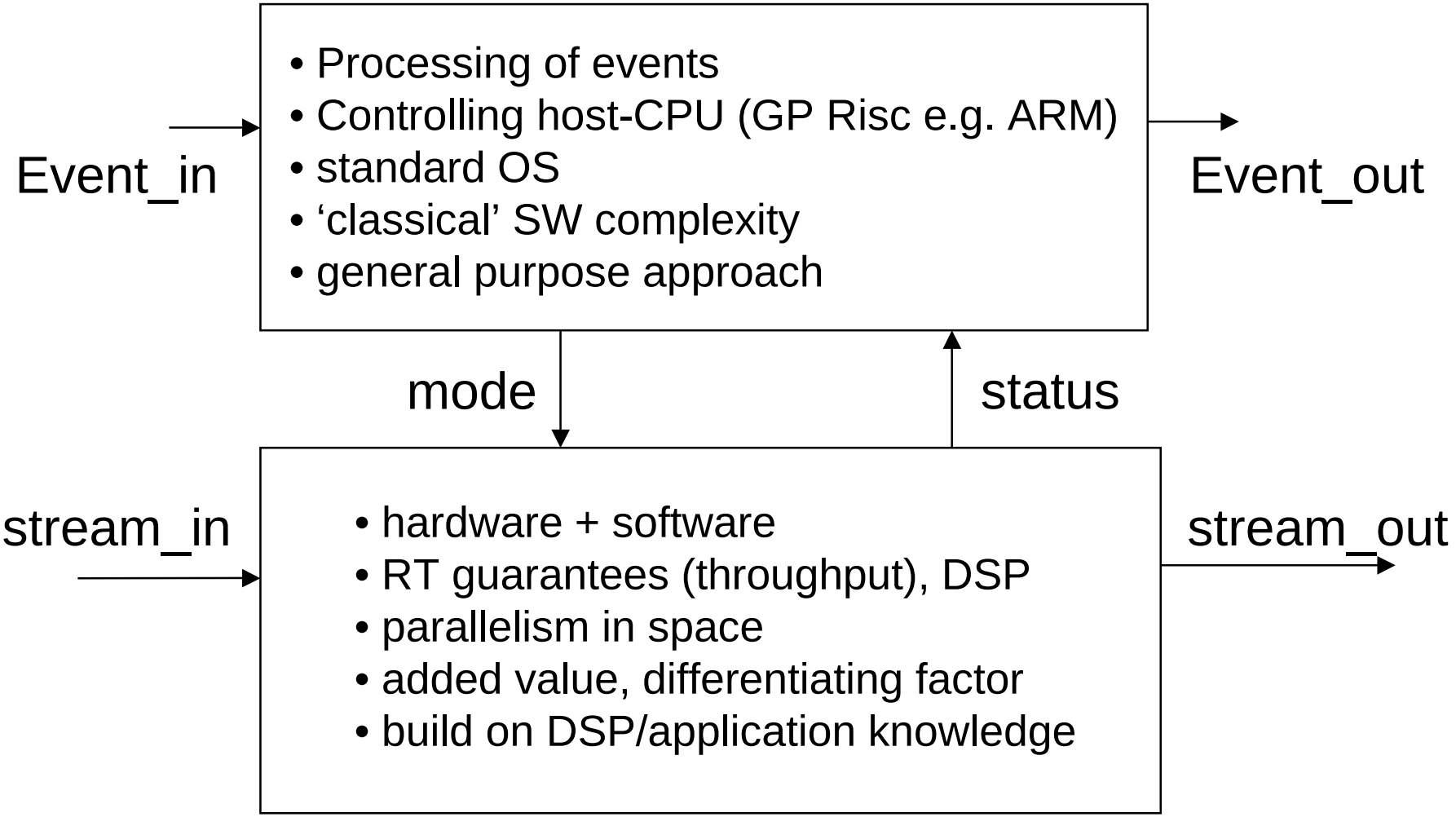
System design trajectory: specifics



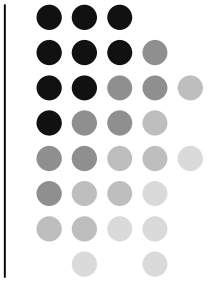


System level: events and streaming





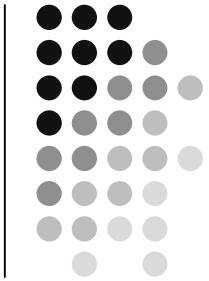
Predictability: What is needed ?



Deal with dynamism

- Changing application set
- Changing application behavior:
  - Scenario switches
  - Less dramatic load changes
    - e.g. changing number of objects / scene

Predictability



Architecture level aspects

- Deterministic control of all shared resources should be possible
  - caches
    - software control
  - shared memories
  - network
    - guaranteed throughput

Predictability

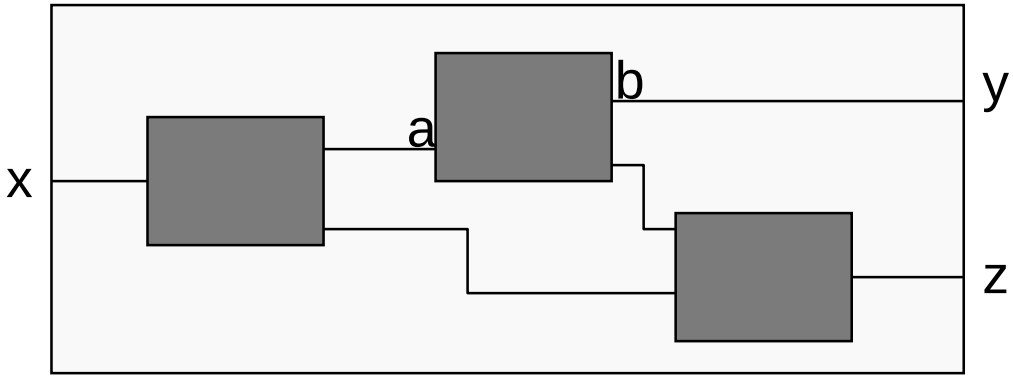
Design flow aspects

- Design time
  - Determine of upper bounds on time and resources

    ➔pareto curves
  - Scenario discovery:
    - separate your application in parts for which upper bounds not too far from worst case

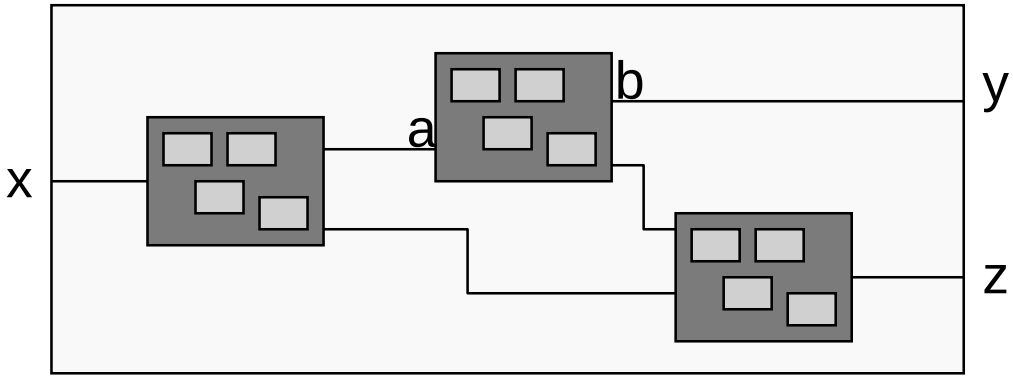
Predictability: Compositionality

High level design



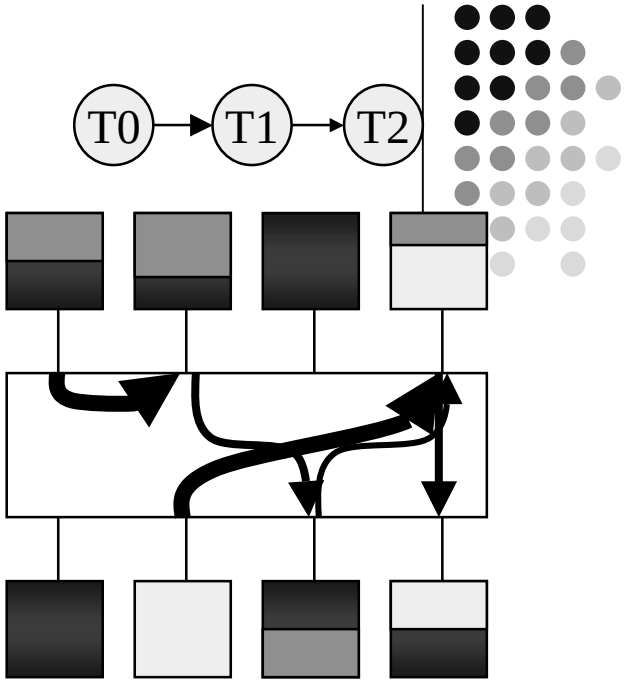
P(x,y) if [P(a,b),...] !

Low level design



P(x,y) if [P(a,b),...] ?

Mapping multiple jobs



Multiple jobs can be active simultaneously.

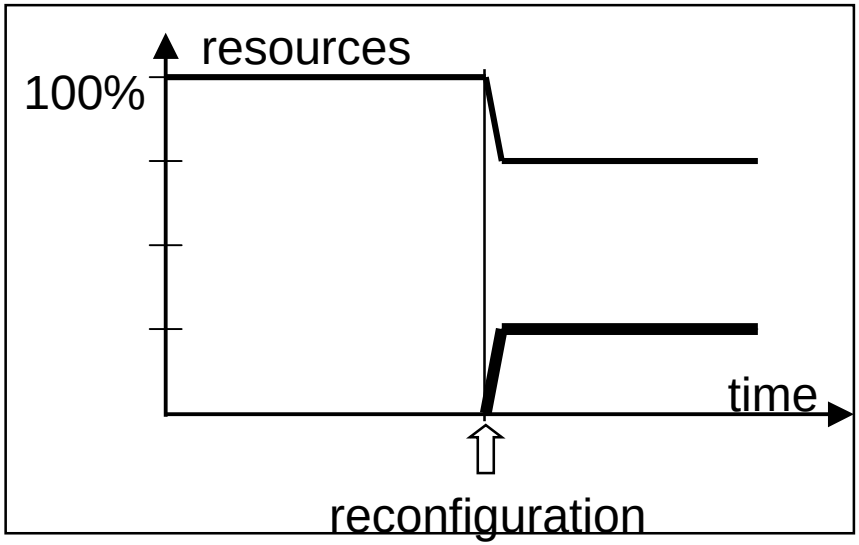
When can a second job start ?

Are the requested resources available ?

If not, can the quality level be lowered ?

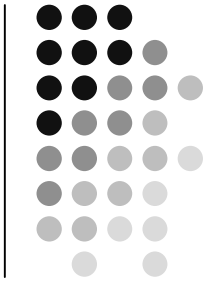
If not, can other jobs go for a lower quality ?

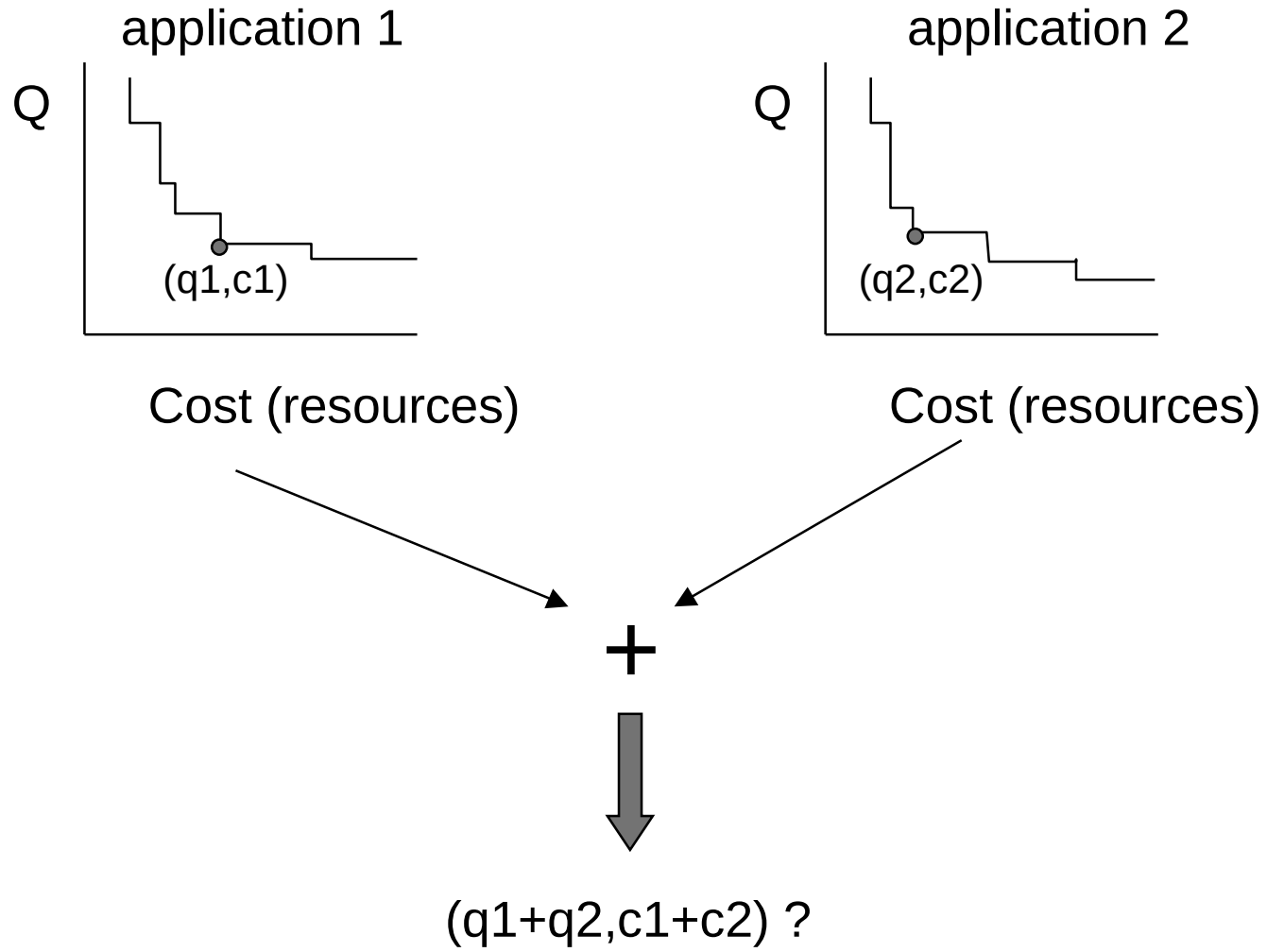
If yes, independent from other jobs ?



How to give guarantees?

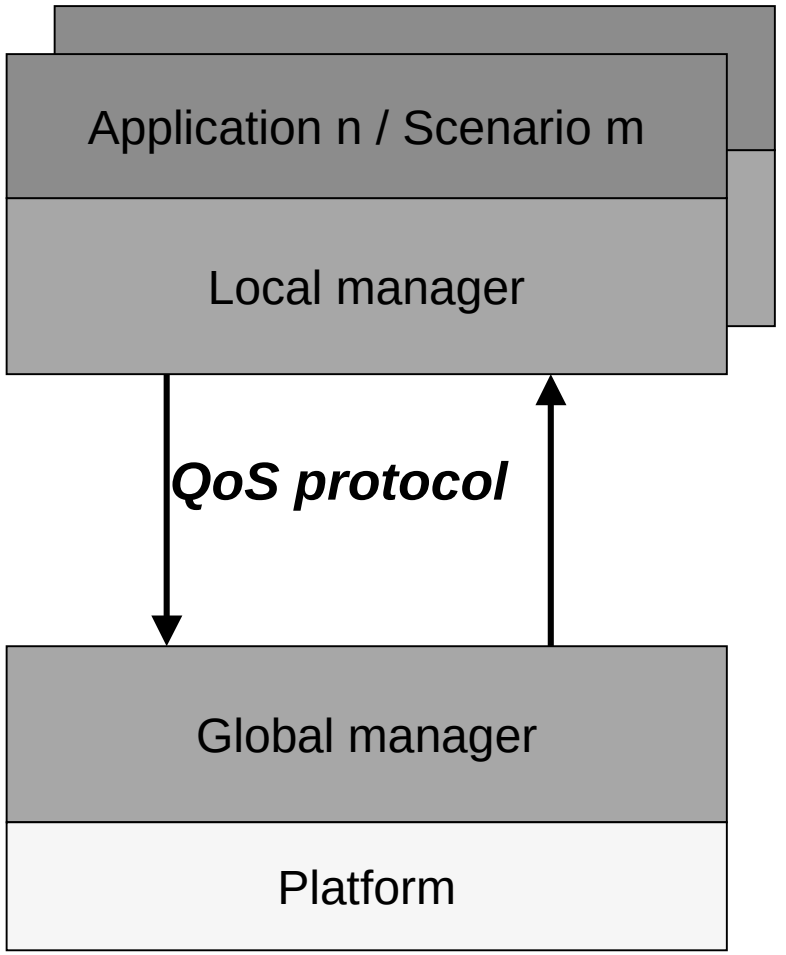
Predictability: Composability
Can we add Pareto points?



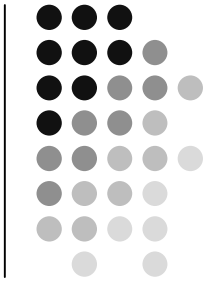


Predictability

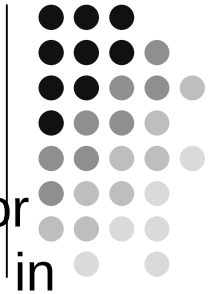
- Run time aspects
  - Scalable applications
  - QoS management



4 levels



- Level 3: System
  - Multiple applications running simultaneously
- Level 2: Application
  - started/stopped by user
  - e.g. DVD player
- Level 1: Job
  - e.g. SDF graph for the sound-part of a DVD player
- Level 0: Task
  - e.g. SDF actor

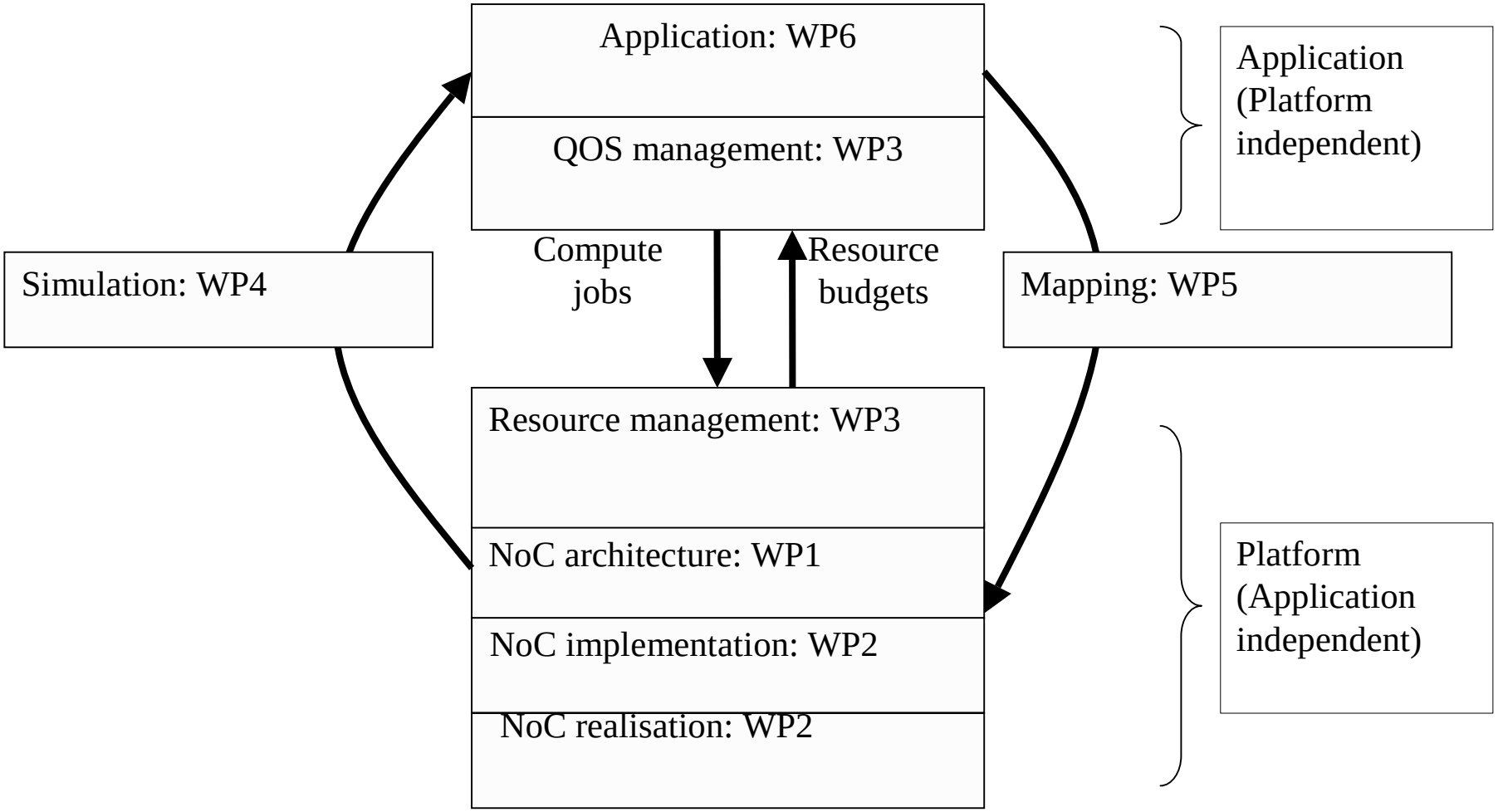


PreMaDoNA proposes a solution based on the removal and/or software control of unpredictable elements in the architectures in combination with a predictable mapping methodology that supports reasoning about throughput.

An FPGA demonstrator will prove that

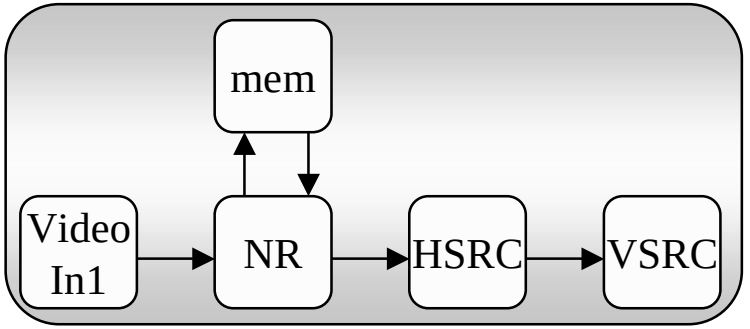
1. That our predictable design methodology saves a lot of design iterations.
2. That NoCs can be used as an adequate target for real-time video applications.
3. That dynamically application demands can be matched with the available NoC resources. Video quality should gracefully reduce when resources are limited.

Relations between WPs



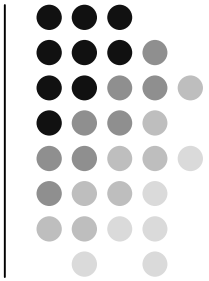
Application characteristics

- Task level: well-known compute intensive kernels
- Job level: data dependent dynamic applications (jobs)
- Application level: multiple jobs



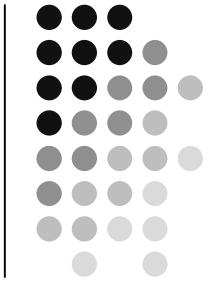
- System level: mutiple applications
- Streaming dominates the cost

Application package



- Case studies
  - Video Scaler (Bart Barenbrug; Philips Research)
  - FM radio (Caracas architecture; Philips Semiconductor)
  - Arbitrary Shape (Object) Texture decoder

Links to related projects



- Internal:
  - Epicurus: Demons, FAME, Promes, Betsy, ….

- External:
  - Philips Research: Aethereal and Hydra
  - Philips Semiconductor
  - Progress projects: SCALP, Artemisia, SmartCam, IMEC MPSoC, M4, Matador, SLI
  - and many others….

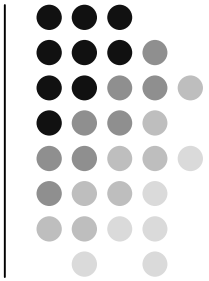
PreMaDoNA

Predictable Matching of Demands on Networked Architectures

Being able to design NoC-based real-time systems in a predictable way, such that we can guarantee non-functional requirements,

while being able to dynamically match quality versus available resources.

Solution based on the removal and/or software control of unpredictable elements in the architectures in combination with a predictable mapping methodology that supports reasoning about non-functional properties



THE END