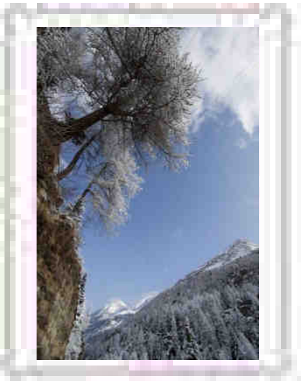


Aoste scientific presentation

Inria Sophia-Antipolis & Rocquencourt

with University of Nice-Sophia/CNRS UMR I3S

Robert de Simone

Our main objective

FORMAL MODEL-DRIVEN ENGINEERING

FOR

OPTIMIZED ALLOCATION

OF

EMBEDDED APPLICATIONS

ONTO

MODERN ARCHITECTURES

Motivations

- Small-scale execution platforms become increasingly parallel
  - *From HPC clusters and data-centers,*
  - *From graphics and embedded processors*
  - *From networks processors*

  … To the mainstream current user of smartphones, tablets, notebooks

- Applications deal increasingly with external physical environment
  - *From avionics and automotive*
  - *From mobile phone platforms*
  - *From robotics*

  … To general Cyber-Physical systems and Internet of things

Compilation becomes a nightmare, hand-made, error-prone and machine-dependent (huge step back!)

Motivations

- Small-scale execution platforms become increasingly parallel
  - *From HPC clusters and data-centers,*
  - *From graphics and embedded processors*
  - *From networks processors*

  … To the mainstream current user of smartphones, tablets, notebooks

- Applications deal increasingly with external physical environment
  - *From avionics and automotive*
  - *From mobile phone platforms*
  - *From robotics*

  … To general Cyber-Physical systems and Internet of things

Use of formal models and methods to provide recipes for better mapping (allocation + scheduling) ?

Homogeneous or heterogeneous parallel execution platforms

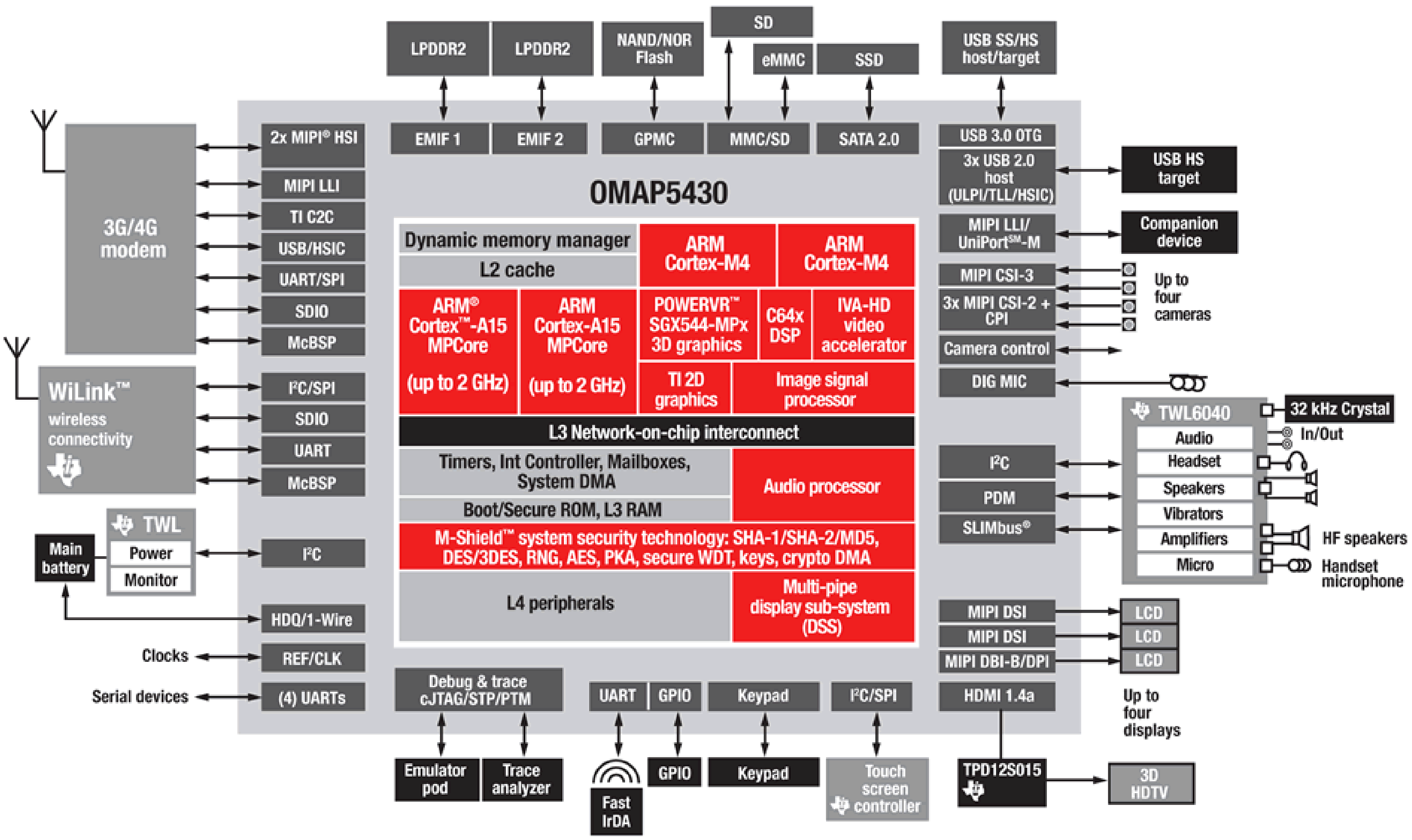
- Embedded platforms
  - Mobile phone **virtual platforms**
  - Automotive/avionics local networks
  - **Sensor networks**
- Massively Parallel Processor Arrays

  (from HPC and Cloud Computing)
  - **Manycore systems-on-chip** (with on-chip networks)
  - → **Kalray MPPA 256**, SThorM (was PF 2012), Tilera, Cavium, Intel…
- Hardware Accelerators
  - **GP-GPU** *hybrid architectures (from graphics processing)*

Witnessing convergence between embedded, HPC, and PC/phone/tablet architectures

MODERN ARCHITECTURES

TI OMAP5430 SoC



OMAP5430

LPDDR2
LPDDR2
NAND/NOR Flash
SD
eMMC
SSD
USB SS/HS host/target

EMIF 1
EMIF 2
GPMC
MMC/SD
SATA 2.0

3G/4G modem

2x MIPI® HSI
MIPI LLI
TI C2C
USB/HSIC
UART/SPI
SDIO
McBSP

WiLink™ wireless connectivity

I²C/SPI
SDIO
UART
McBSP

Main battery

TWL
Power
Monitor

I²C
HDQ/1-Wire

Clocks
REF/CLK

Serial devices
(4) UARTs

Dynamic memory manager
L2 cache
ARM Cortex-M4
ARM Cortex-M4

ARM® Cortex™-A15 MPCore (up to 2 GHz)
ARM Cortex-A15 MPCore (up to 2 GHz)
POWERVR™ SGX544-MPx 3D graphics
C64x DSP
IVA-HD video accelerator
TI 2D graphics
Image signal processor

L3 Network-on-chip interconnect

Timers, Int Controller, Mailboxes, System DMA
Audio processor
Boot/Secure ROM, L3 RAM

M-Shield™ system security technology: SHA-1/SHA-2/MD5, DES/3DES, RNG, AES, PKA, secure WDT, keys, crypto DMA

L4 peripherals
Multi-pipe display sub-system (DSS)

USB 3.0 OTG
3x USB 2.0 host (ULPI/TLL/HSIC)
USB HS target
MIPI LLI/ UniPort℠-M
Companion device
MIPI CSI-3
3x MIPI CSI-2 + CPI
Up to four cameras
Camera control
DIG MIC

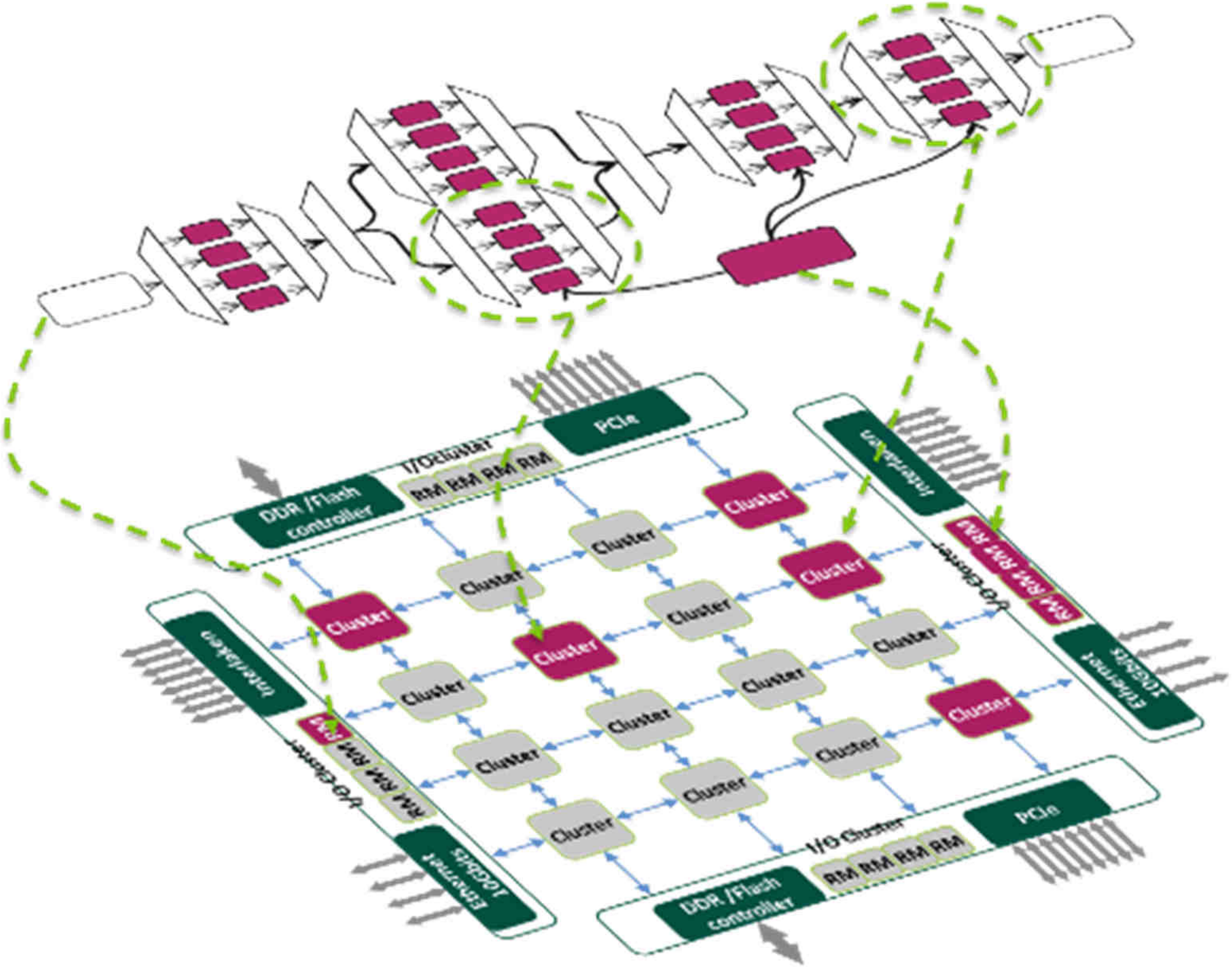
I²C
PDM
SLIMbus®

TWL6040
32 kHz Crystal
Audio
In/Out
Headset
Speakers
Vibrators
Amplifiers
HF speakers
Micro
Handset microphone

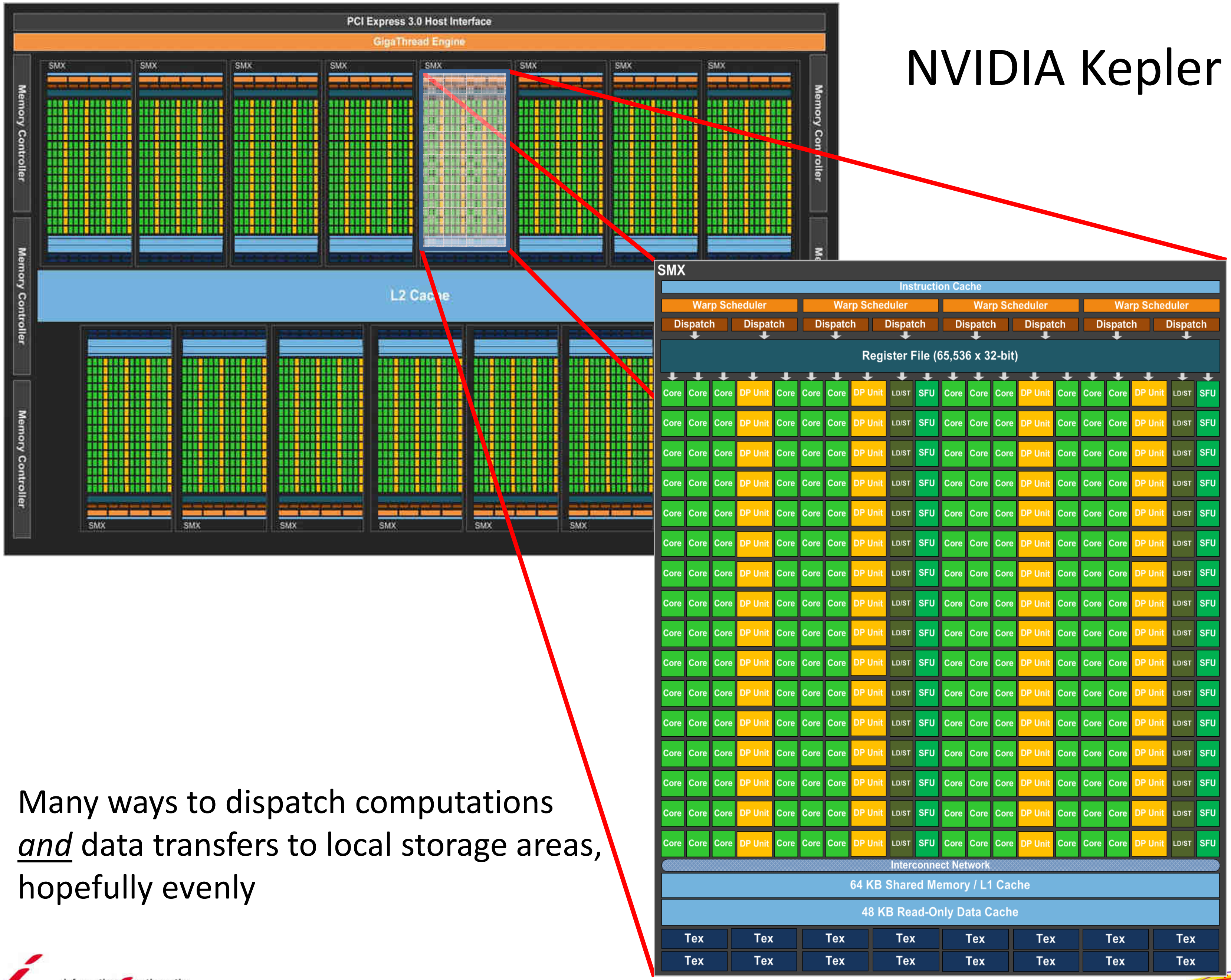
MIPI DSI
MIPI DSI
MIPI DBI-B/DPI
LCD
LCD
LCD
Up to four displays
HDMI 1.4a
TPD12S015
3D HDTV

Debug & trace cJTAG/STP/PTM
Emulator pod
Trace analyzer
UART
Fast IrDA
GPIO
GPIO
Keypad
Keypad
I²C/SPI
Touch screen controller

Kalray MPPA-256



NVIDIA Kepler GPU



Many ways to dispatch computations *<u>and</u>* data transfers to local storage areas, hopefully evenly

Dataflow streaming applications, with control of the environment

Cyber-Physical Systems domain

Strong interactions with physical environment, interfaces

In our case mobile phones, automotive, avionic domains

→ **Esterel, SCADE**, …

Multimedia signal/image processing

→ **Nested Loops with affine bounds** (static control) parallel compilation

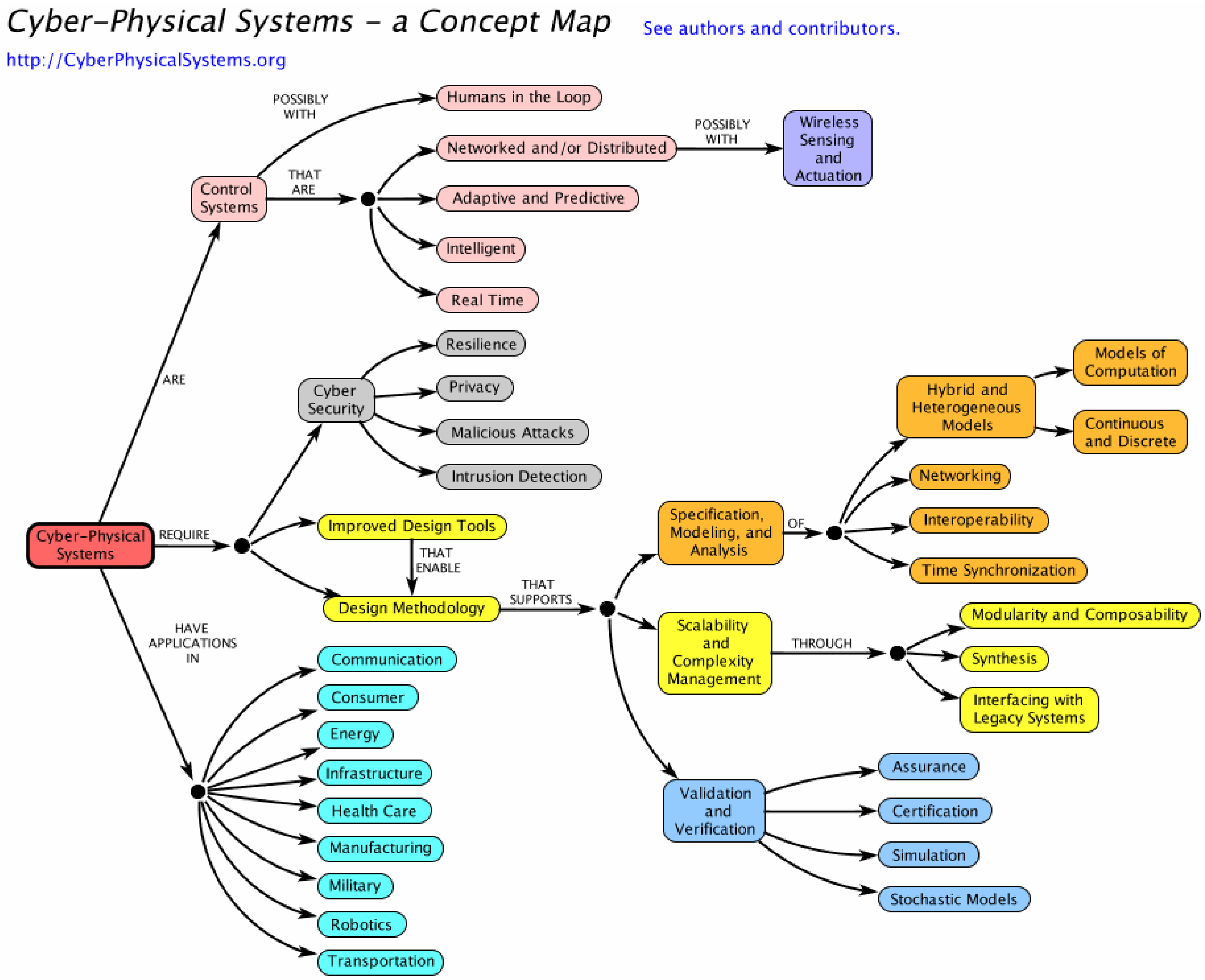
→ **Dataflow process networks** (**Kahn, Petri**, many variants)

Concurrency important (task-level, data-level parallelism)

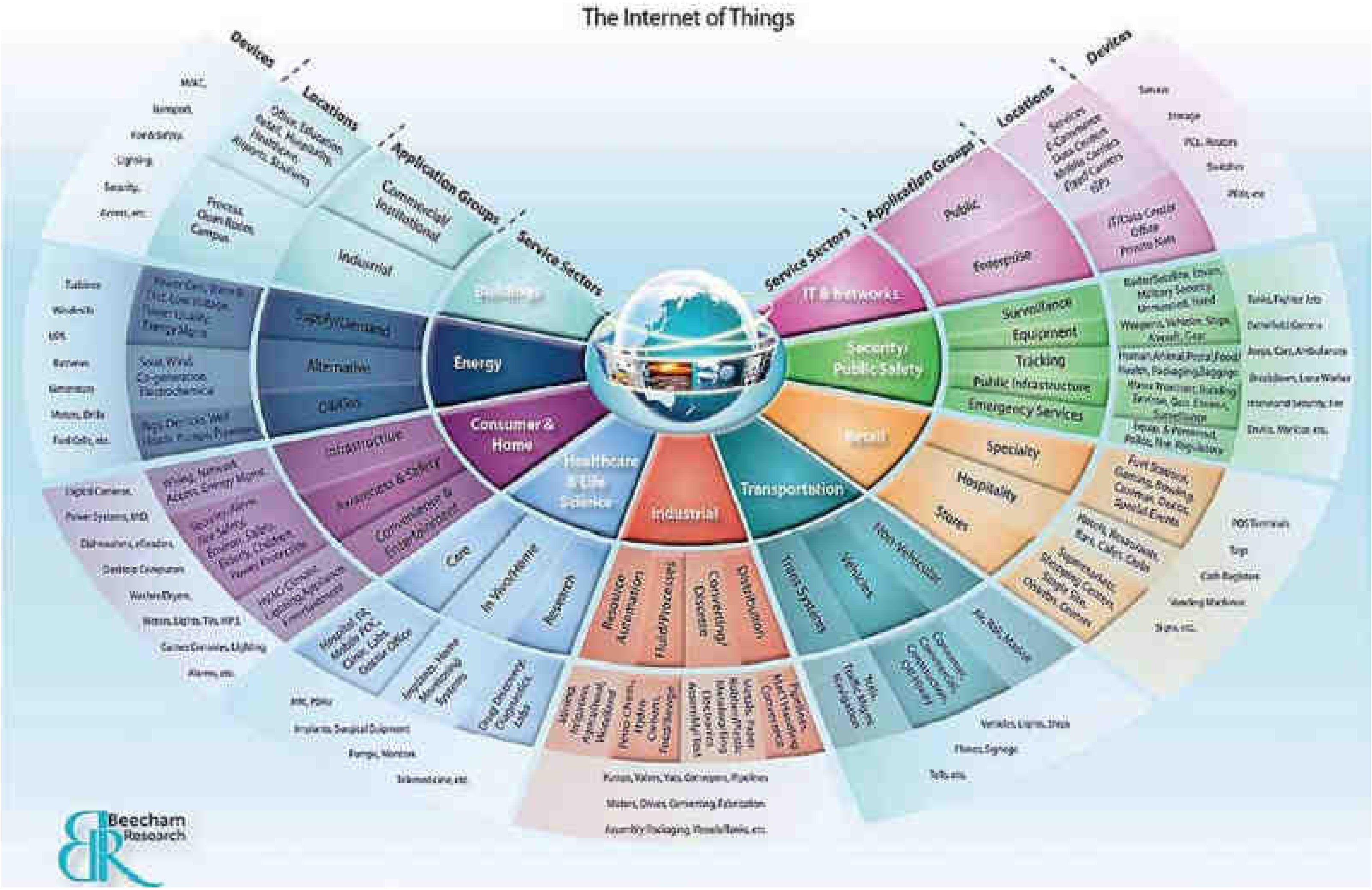
EMBEDDED APPLICATIONS

Abstract descriptions as Models of Computations

CyberPhysical domain



Internet of Things span



*Compilation **no longer automatic** on fancy parallel platform*

Issues:

- Mapping/distribution (spatial) of functions/tasks onto resources
- Scheduling (temporal) to share resources across several application functions

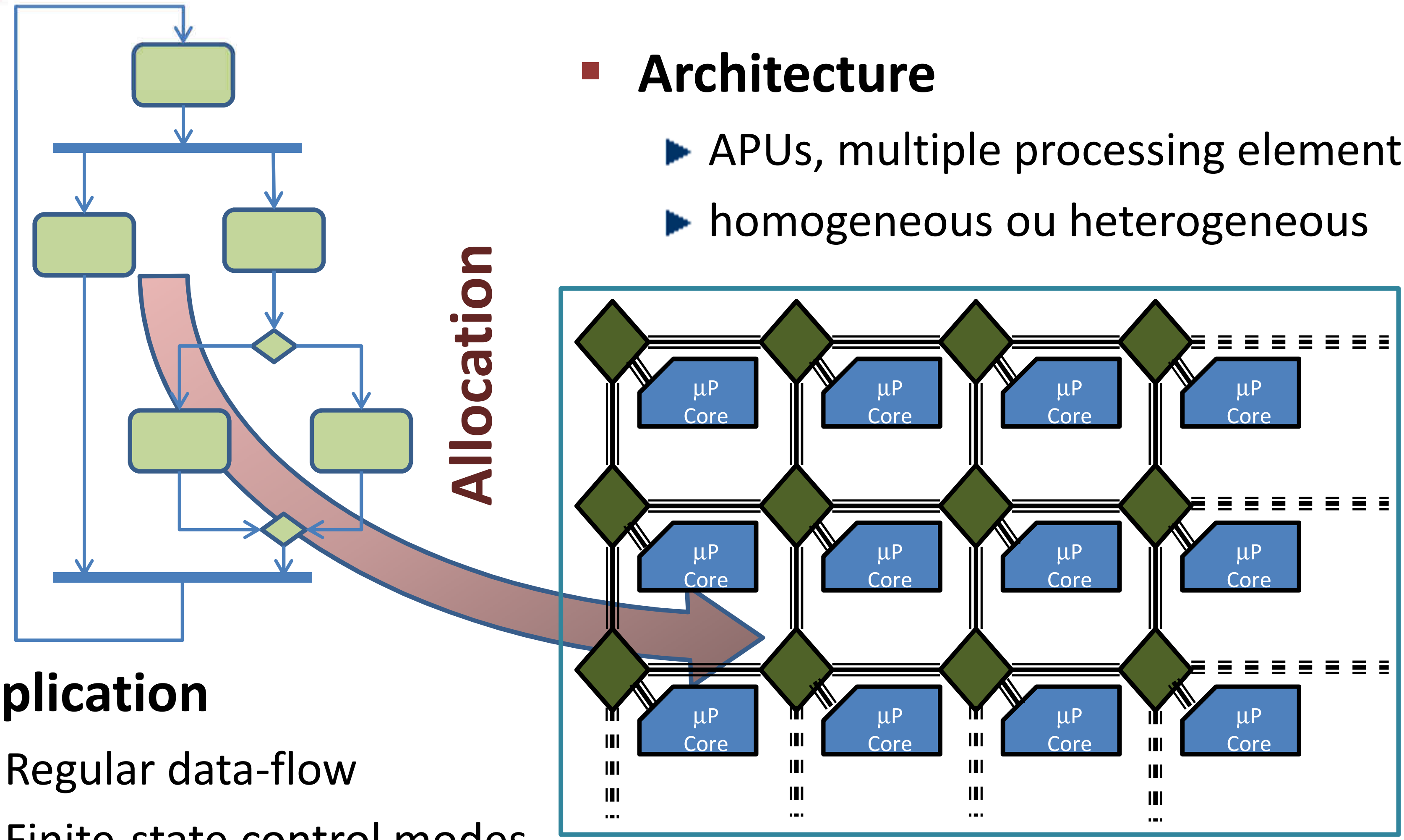
Become as important on data communications & memory transfers as on computations

OPTIMIZED ALLOCATION

→ Careful modeling of interconnect, timing aspects

Embedded Software/Hardware ?

- **Architecture**
  - APUs, multiple processing elements
  - homogeneous ou heterogeneous



- **Application**
  - Regular data-flow
  - Finite-state control modes

Need for mathematical properties to support analysis

Dataflow PNs, Synchronous languages, Nested loops w/ static bounds are both

- ✓ **executable languages** and
- ✓ **formal Models of Computation and Communication** (MoCCs),

with rich theory for analysis

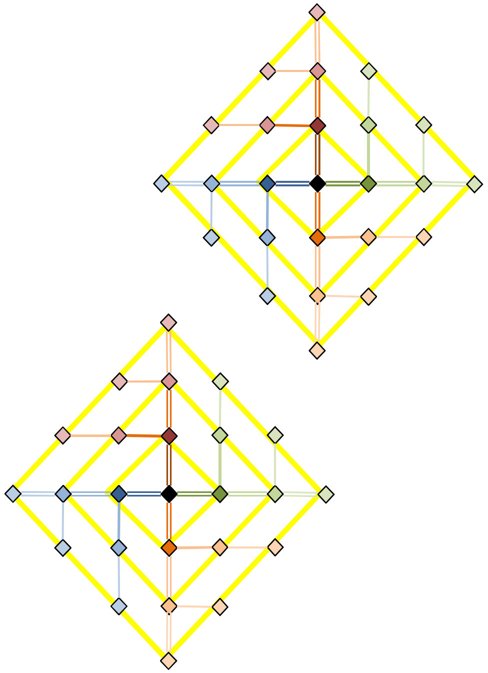
Optimization techniques to best fit applications to architectures

FORMAL MODEL-DRIVEN ENGINEERING

Theoretical results on static regular scheduling and routing

- **KRG** model with k-periodic routing/switching words
- Latency-Insensitive Design in SoC modeling
- Smooth schedules and even data distributions (mechanical/Christofffel words)
  → **K-Passa** tool
- Full axiomatisation for network topology transformations
- …

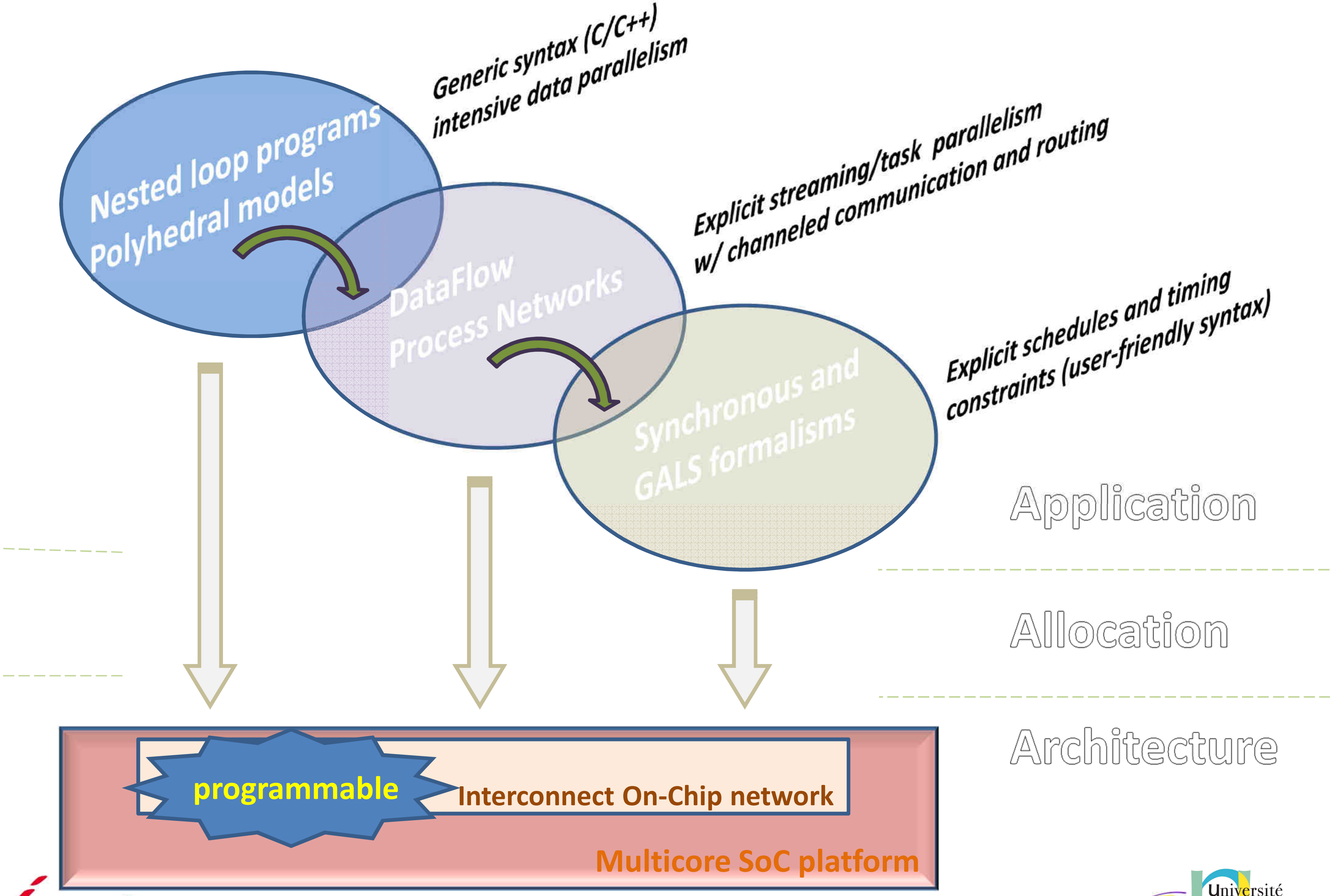
All2All diffusion routing schemes (cellular automata)



- At any local router (synchronously)

from South ▶	N	^{N}W	N	^{N}W	N
from East ▶	W	^{S}W	W	^{S}W	W
from North ▶	S	^{S}E	S	^{S}E	S
from West ▶	E	^{N}E	E	^{N}E	E

Compiling applications *also* on configurable NoC routers


Nested loop programs
Polyhedral models
Generic syntax (C/C++)
intensive data parallelism
DataFlow
Process Networks
Explicit streaming/task parallelism
w/ channeled communication and routing
Synchronous and
GALS formalisms
Explicit schedules and timing
constraints (user-friendly syntax)
Application
Allocation
Architecture
programmable
Interconnect On-Chip network
Multicore SoC platform

Engineering such a design flow

We use **MDE environments** and tools to represent application, platform and allocation models

→ **OMG UML profile MARTE**

The resulting framework also allows abstract application modeling to tune target hardware design

Platform-based design

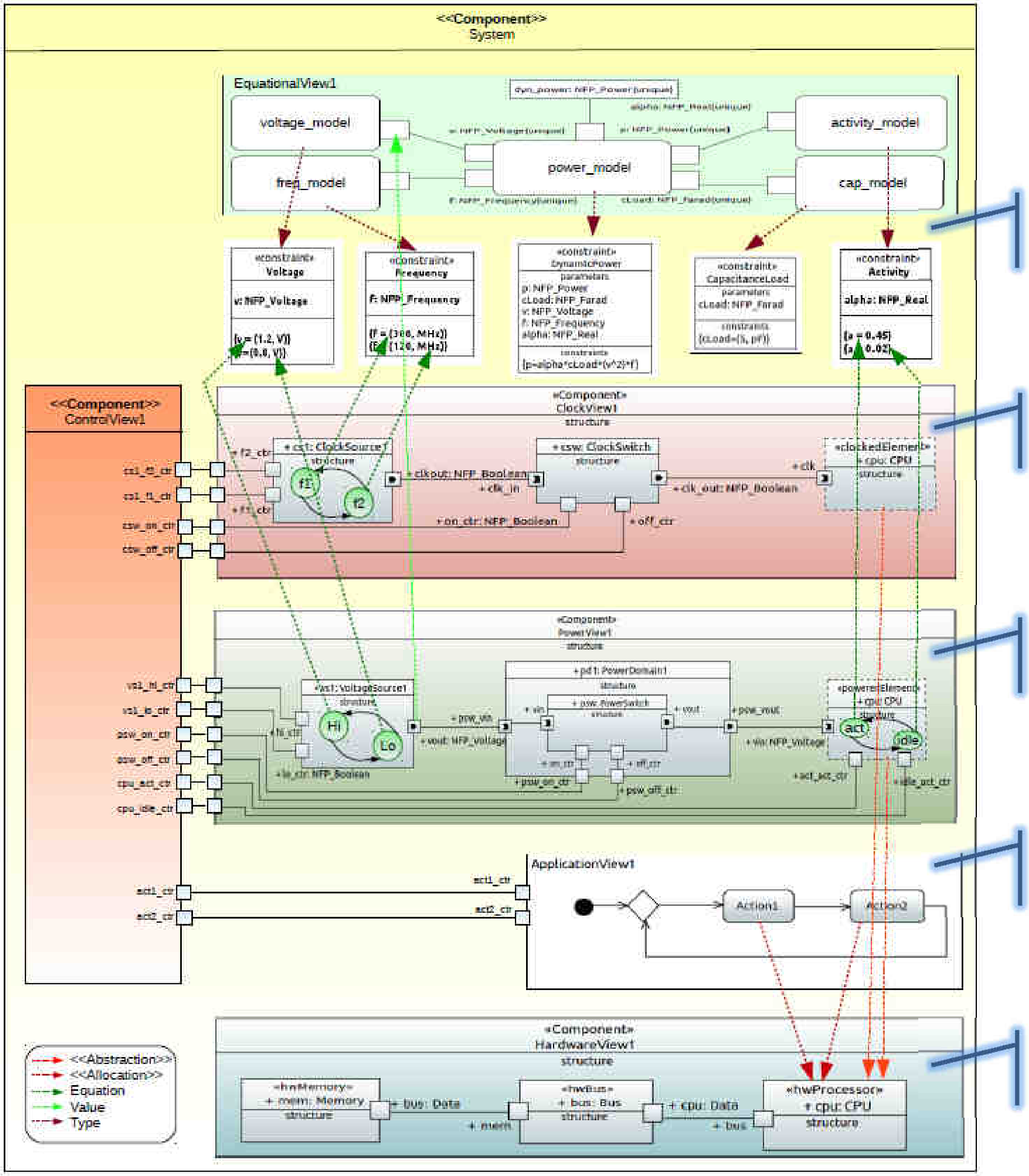
FORMAL MODEL-DRIVEN ENGINEERING

MARTE Time model:

- Multiple logical clocks (signals), multiform time
- **CCSL** (Clock Constraint Specification Language) to assert timing requirements in a formal setting, as integral part of the model
- Translation to formal models (variants of Buchi automata) for schedulability
- **TimeSquare** simulation environment
- Many formal connections with existing formalisms:

AADL, AutoSar/TADL, Spin/Promela, UppAal, nu-SMV,…

Multiview



Physical laws

Clock/perform

Power/energy

Appli (behav)

Archi

Thank you for attention !

Questions ?