

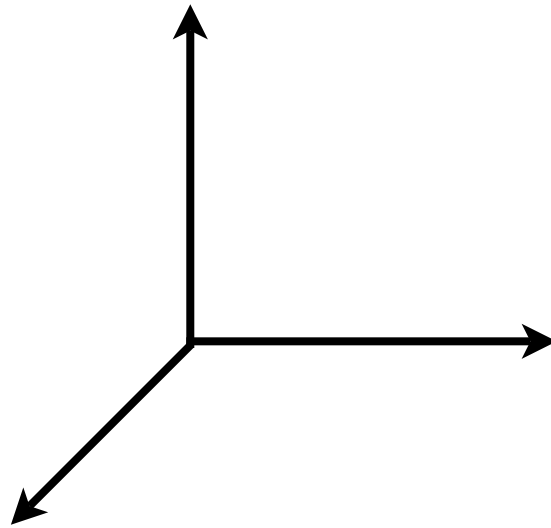
# BTV Roadmap

Jonas Maebe  
Koen De Bosschere

November 27<sup>th</sup>, 2008



# Computing Evolution



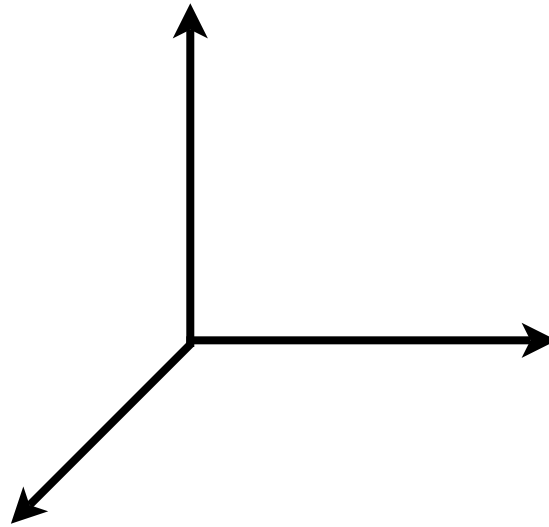
# Computing Evolution

## **Hardware**

Multi-cores

FPGAs

Accelerators



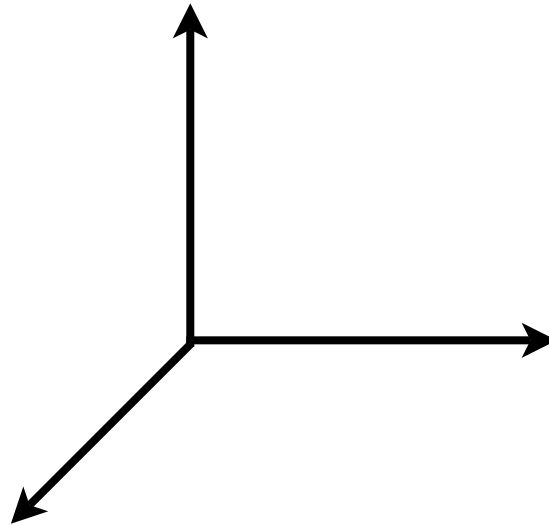
# Computing Evolution

## **Hardware**

Multi-cores

FPGAs

Accelerators



## **Software**

Abstractions

Software design techniques

Layers

# Computing Evolution

## Hardware

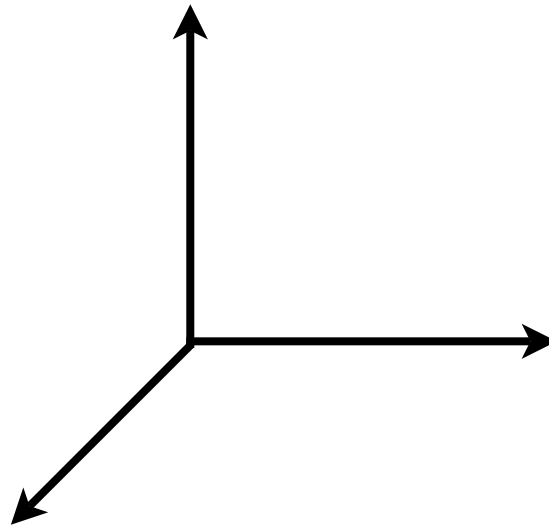
Multi-cores  
FPGAs  
Accelerators

## Requirements

Security  
Reliability  
Networking

## Software

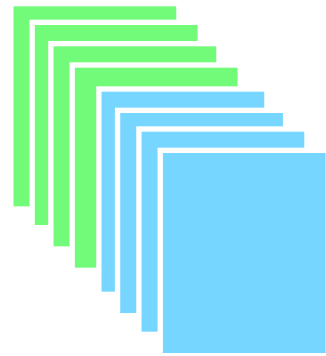
Abstractions  
Software design techniques  
Layers



# Software Evolution



# Software Evolution



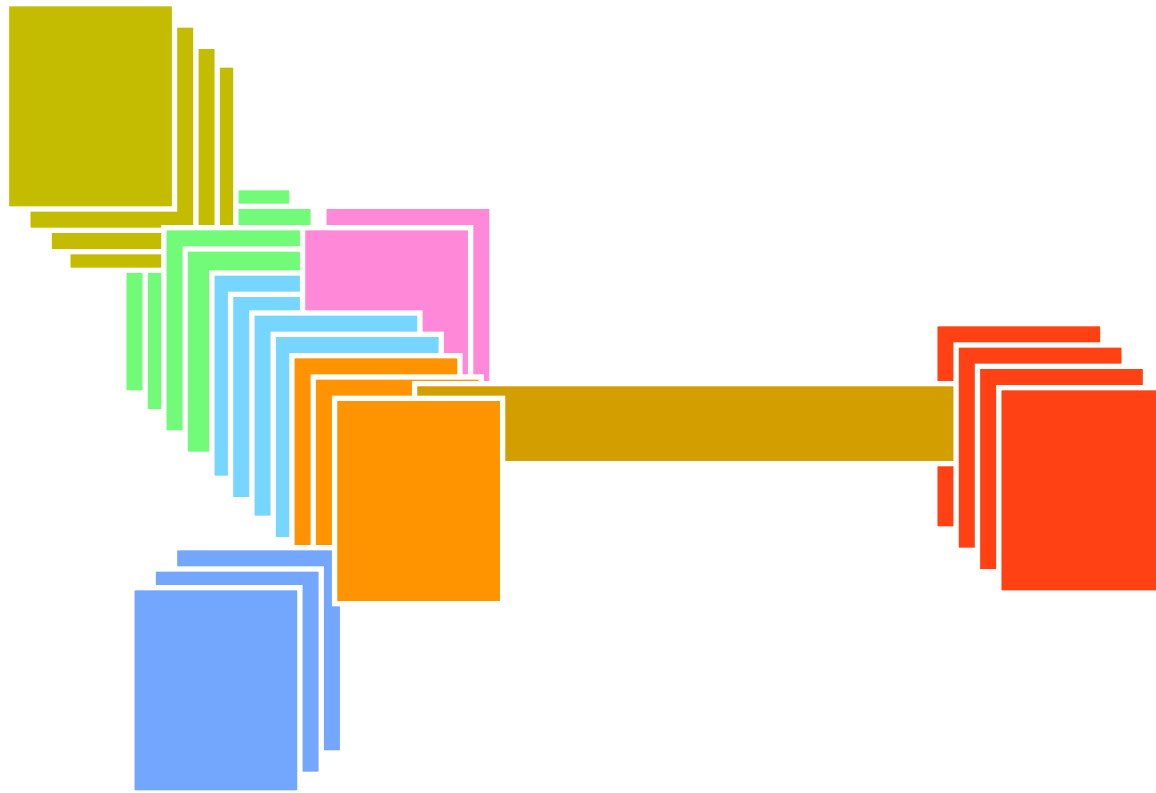
# Software Evolution



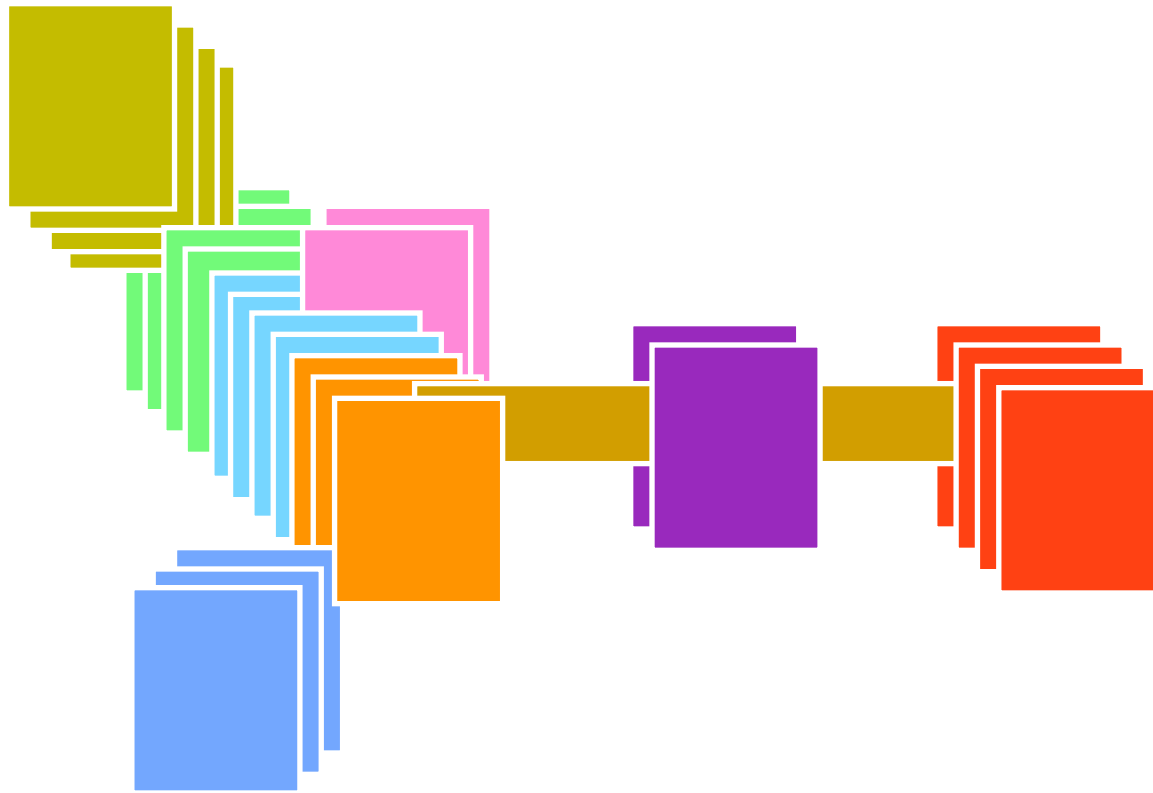
# Software Evolution



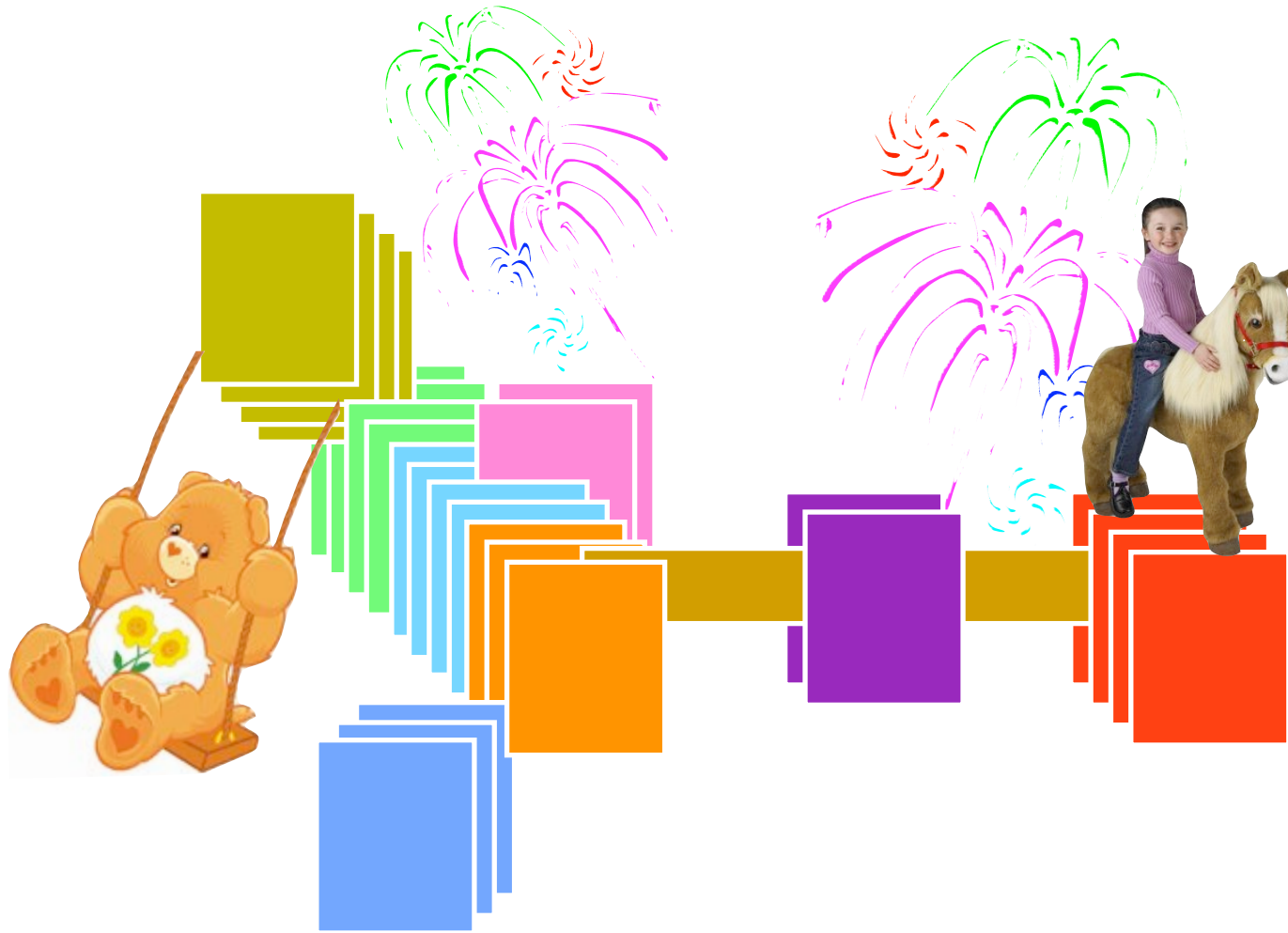
# Software Evolution



# Software Evolution

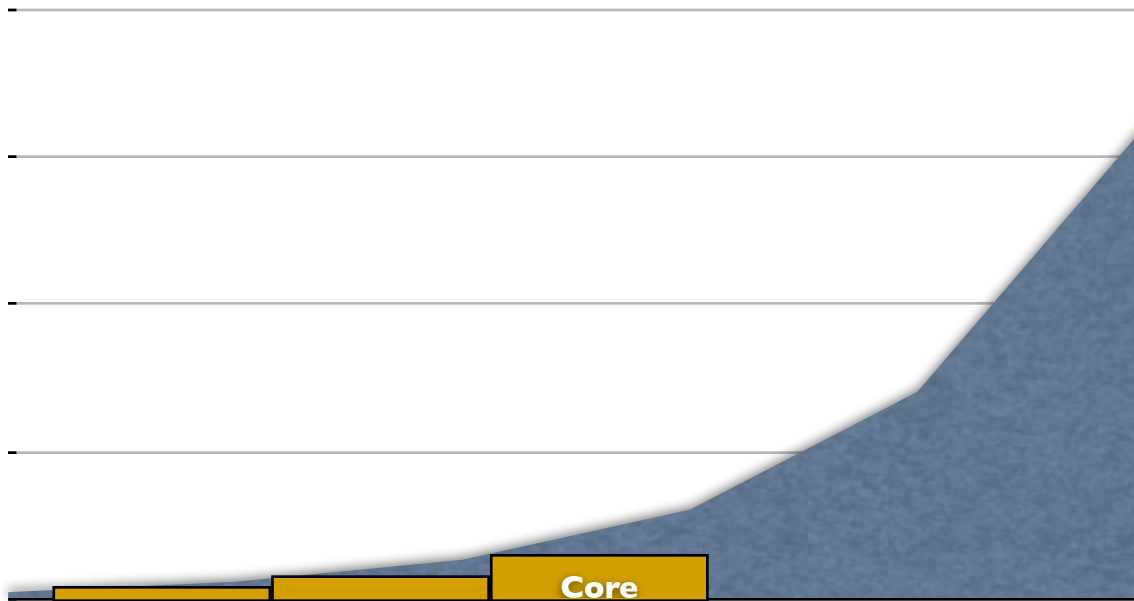


# Software Evolution



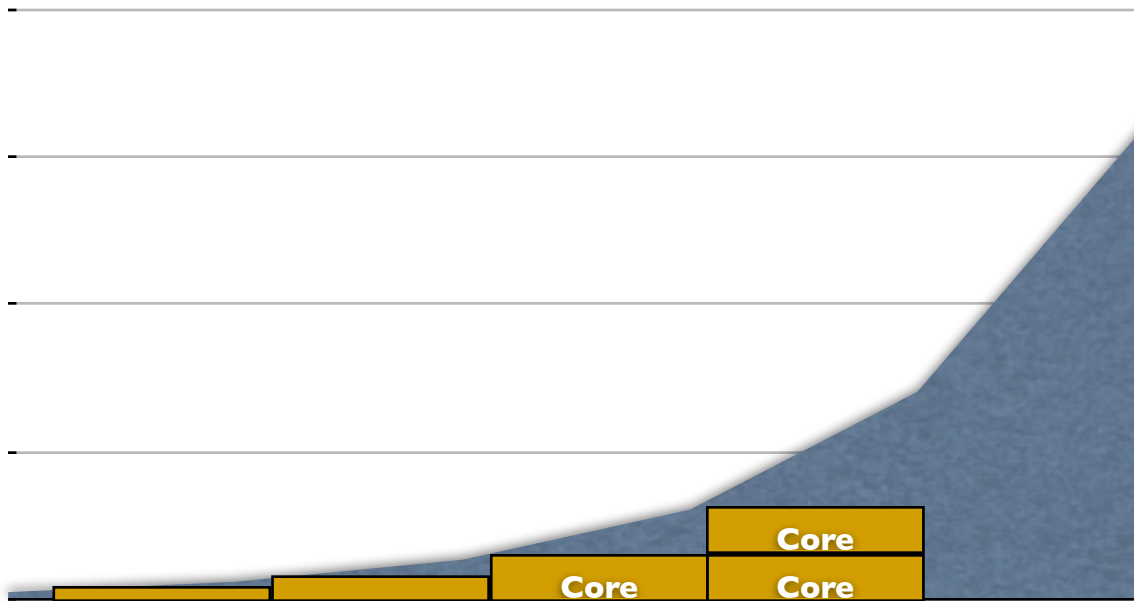
# Hardware Evolution

■ Transistor count



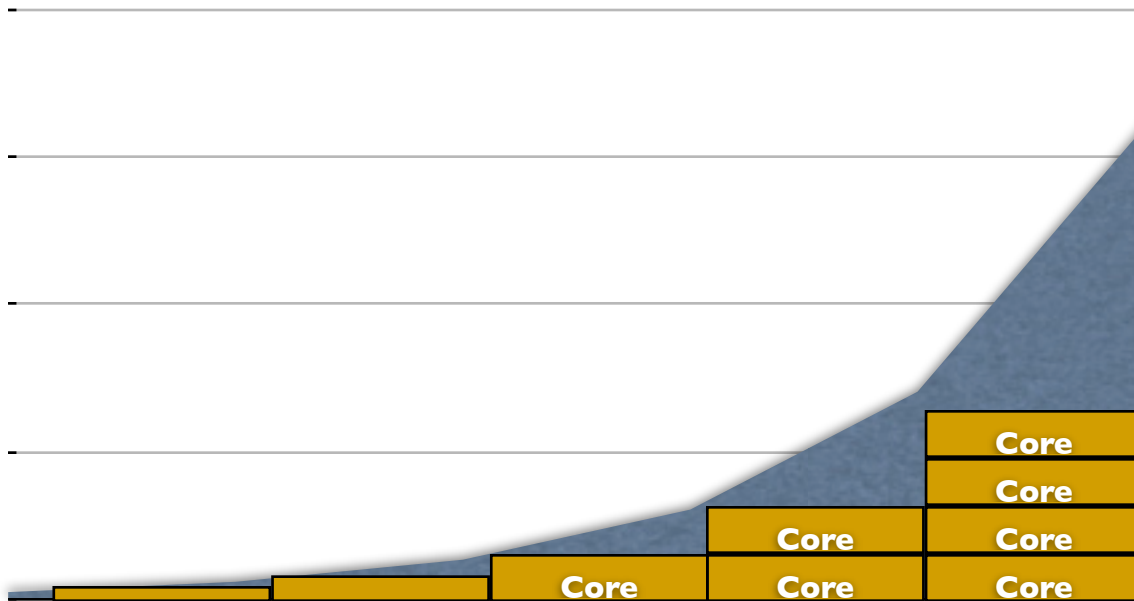
# Hardware Evolution

■ Transistor count



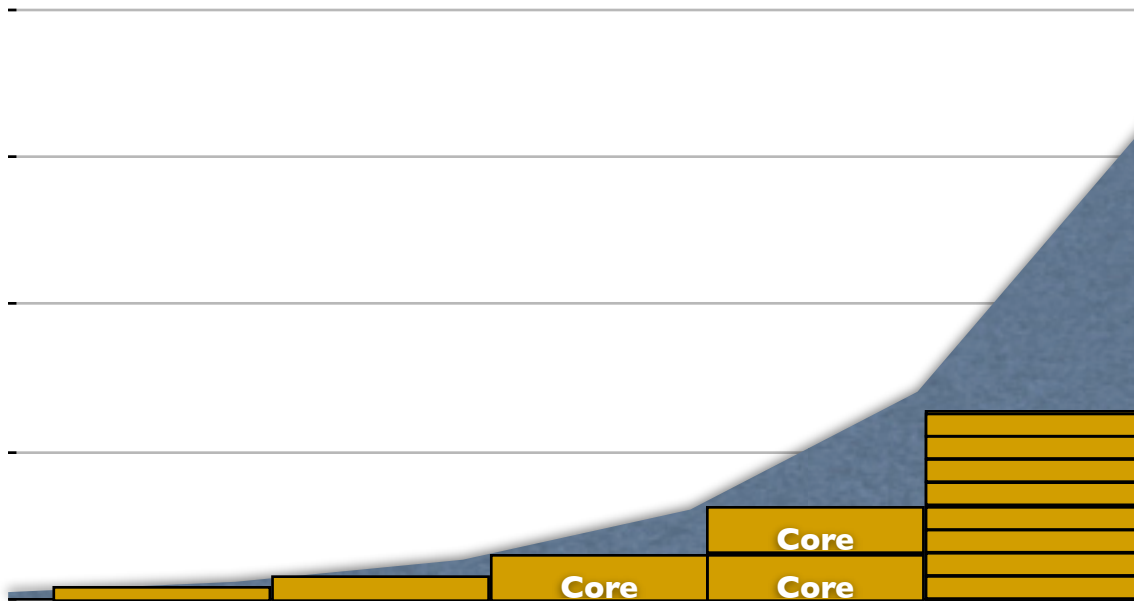
# Hardware Evolution

■ Transistor count



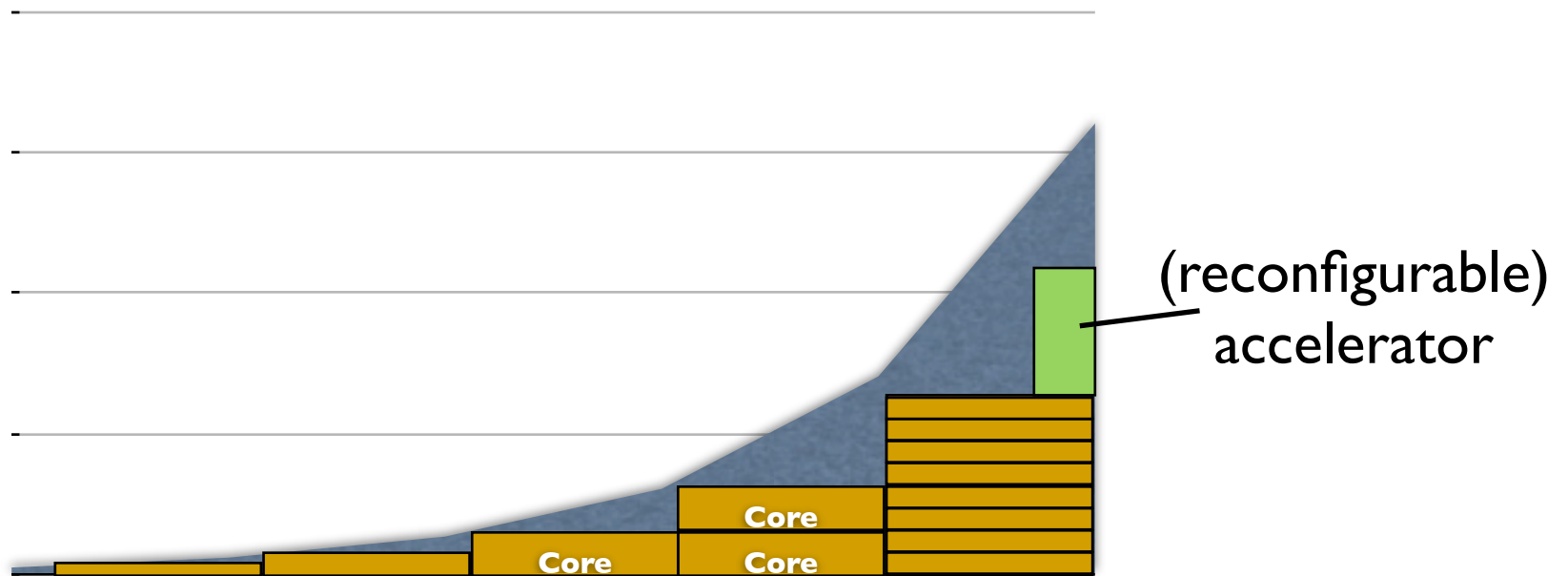
# Hardware Evolution

■ Transistor count



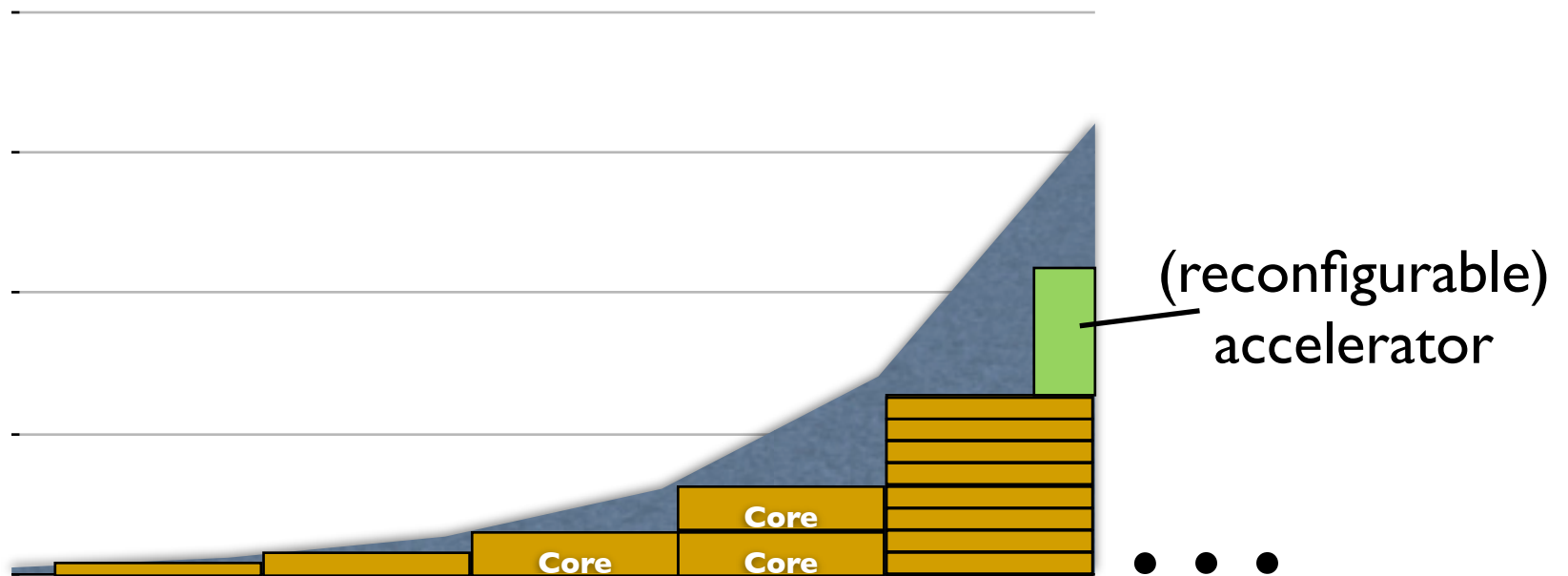
# Hardware Evolution

■ Transistor count



# Hardware Evolution

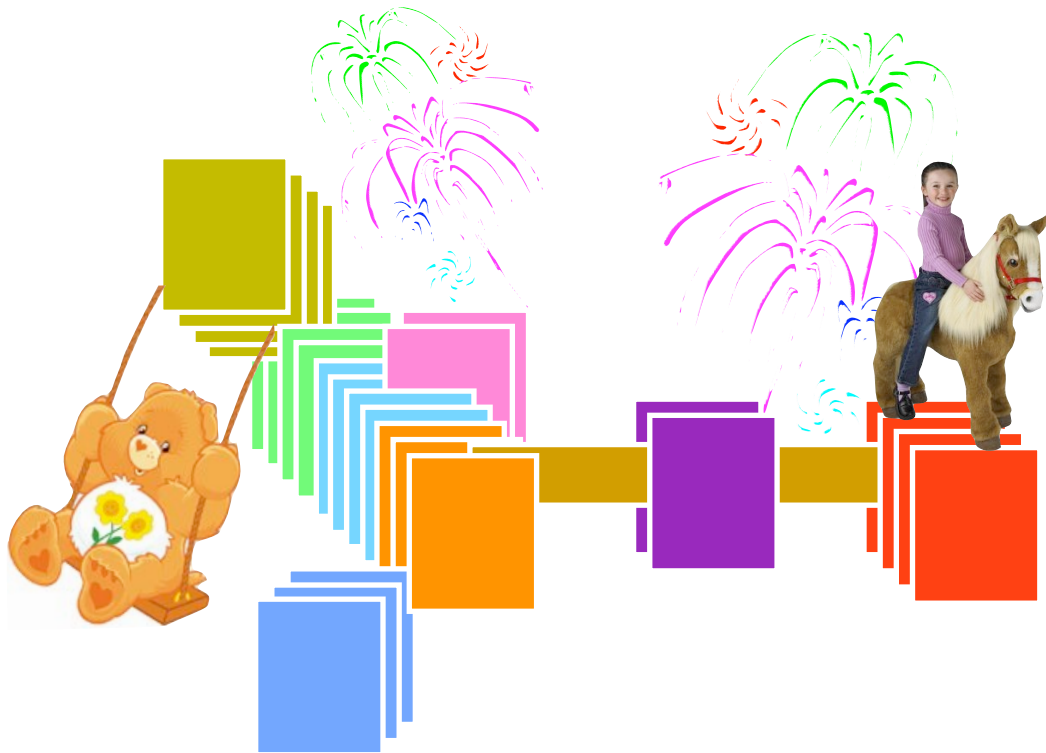
■ Transistor count





# Software/Hardware Complexity

# Software/Hardware Complexity



# Software/Hardware Complexity



# Software/Hardware Paradox

- Past
  - Software simple & easy to adapt
  - Hardware (functionally) largely static
- Present
  - Software increasingly harder to adapt
  - Hardware changing and becoming malleable

# Overall Big Challenges

- Allow hardware to evolve without breaking software
- Make optimal use of new and different hardware

**So, how does BTV fit in?**

# BTV Advantages: Hardware

- Platform-independence
- Adaptability to hardware variability
- Resource management

# BTV Challenges: Hardware

- Hardware support for BTV
- Efficiently dealing with hardware variability
- Virtualising everything

# BTV Advantages: Software

- Decouple design/execution time, cross layer optimisation
- Tool chain flexibility
- Observability/flexibility of the programs

# BTV Challenges: Software

- Generic meta-information about client programs (incl. multi-threading!)
- Active cooperation between clients and hosts (two-way initiatives)

# BTV Advantages: Requirements

- Vendor neutrality
- High availability
- Ubiquitous computing
- Security

# BTV Challenges: Requirements

- Performance/energy modelling
- Certification & validation
- Real time systems

# Summary

- BTV can help in dealing with challenges in hardware, software and requirements
- BTV itself offers many interesting & important research challenges

# Comments?