virtutech

Virtutech Simics

Jakob Engblom, PhD
Business Development Manager
Virtutech
jakob@virtutech.com

Our Technology

Full system simulation

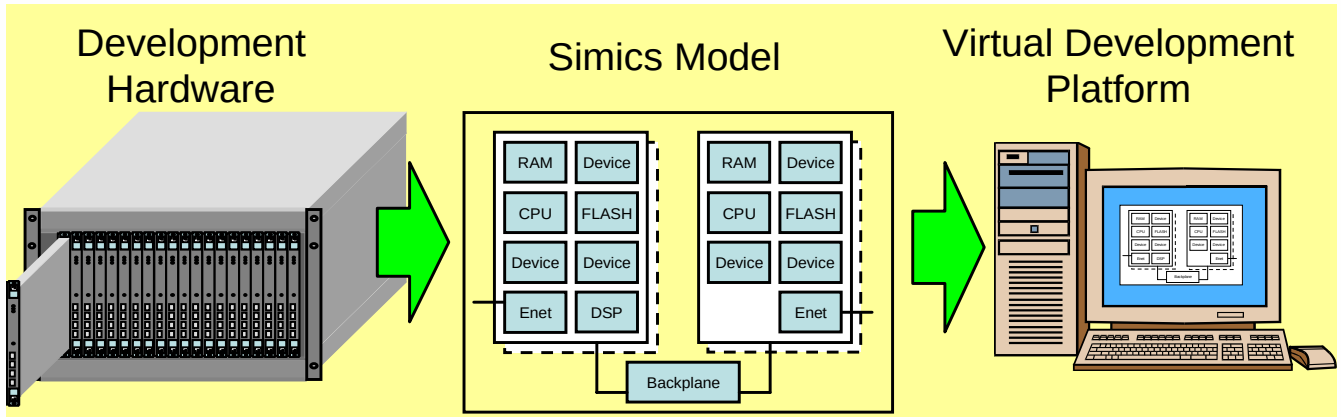
- Networking, backplanes
- System-level from the beginning

Runs complete software stack

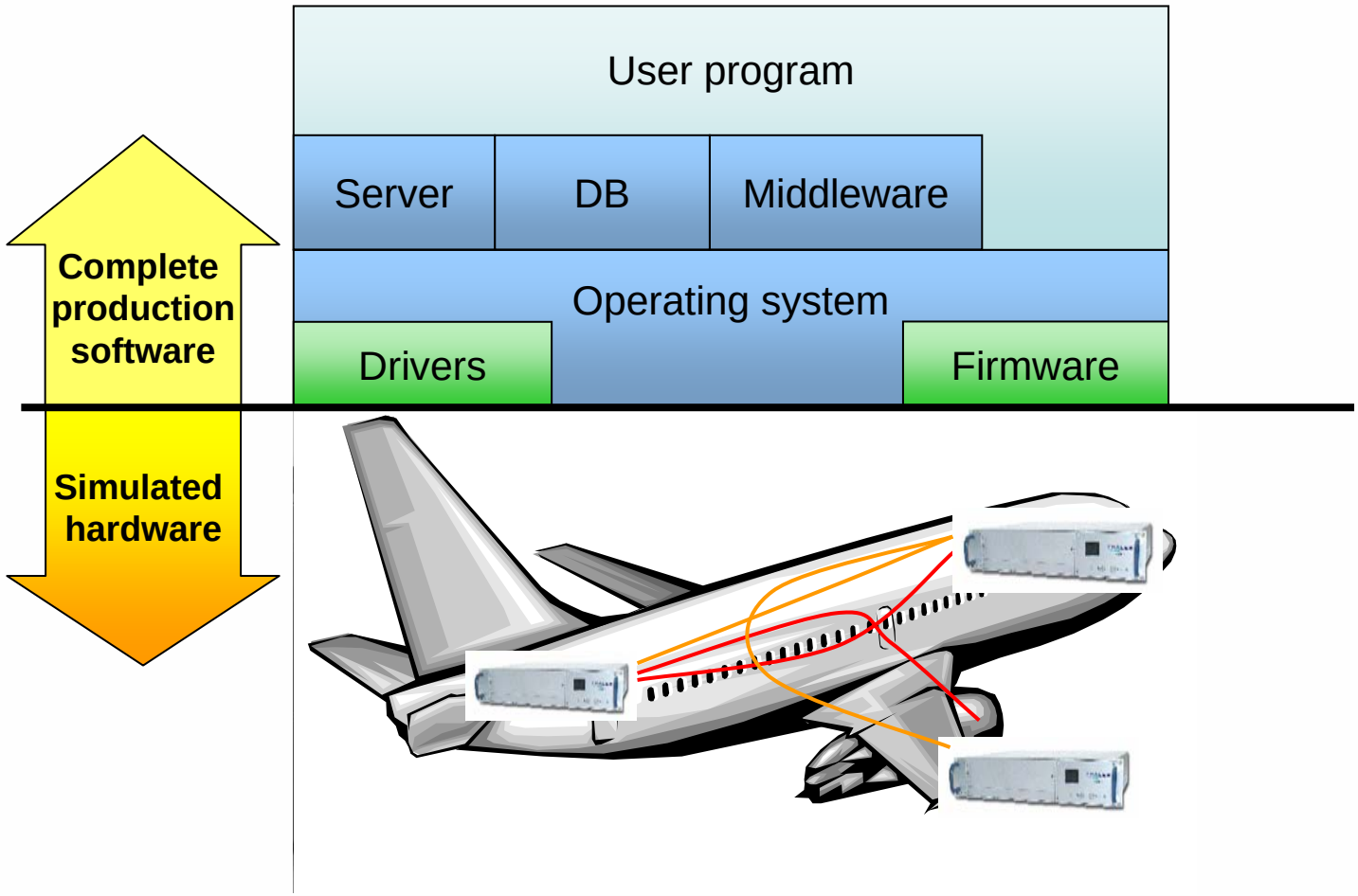
- Firmware, device drivers, OS, etc…

Very high performance

- Typically 100s of MIPS
- Multiple GIPS top benchmark



Our Technology: Full-System Simulation

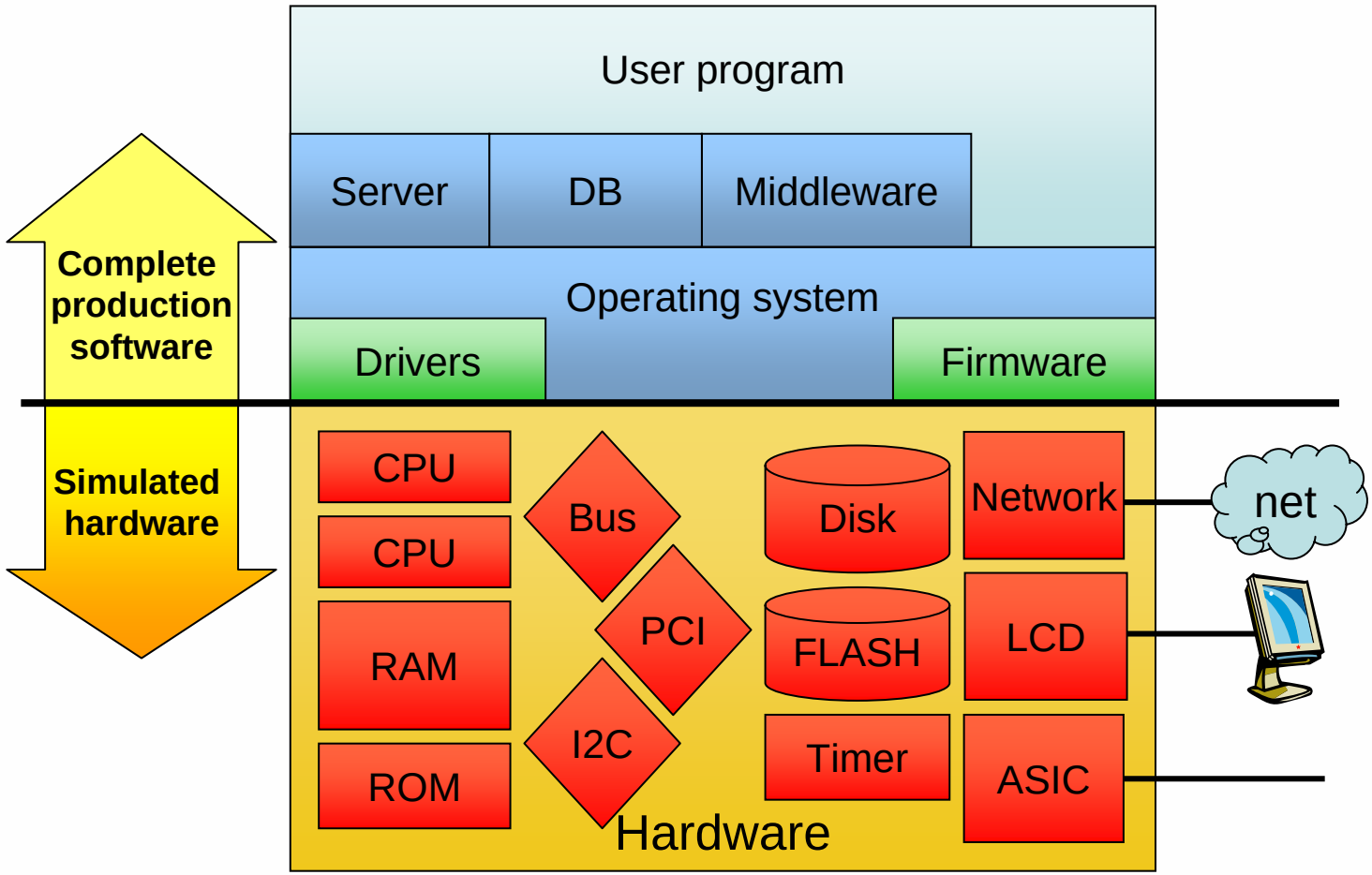


Our Technology: Full-System Simulation

Identical build tools chain

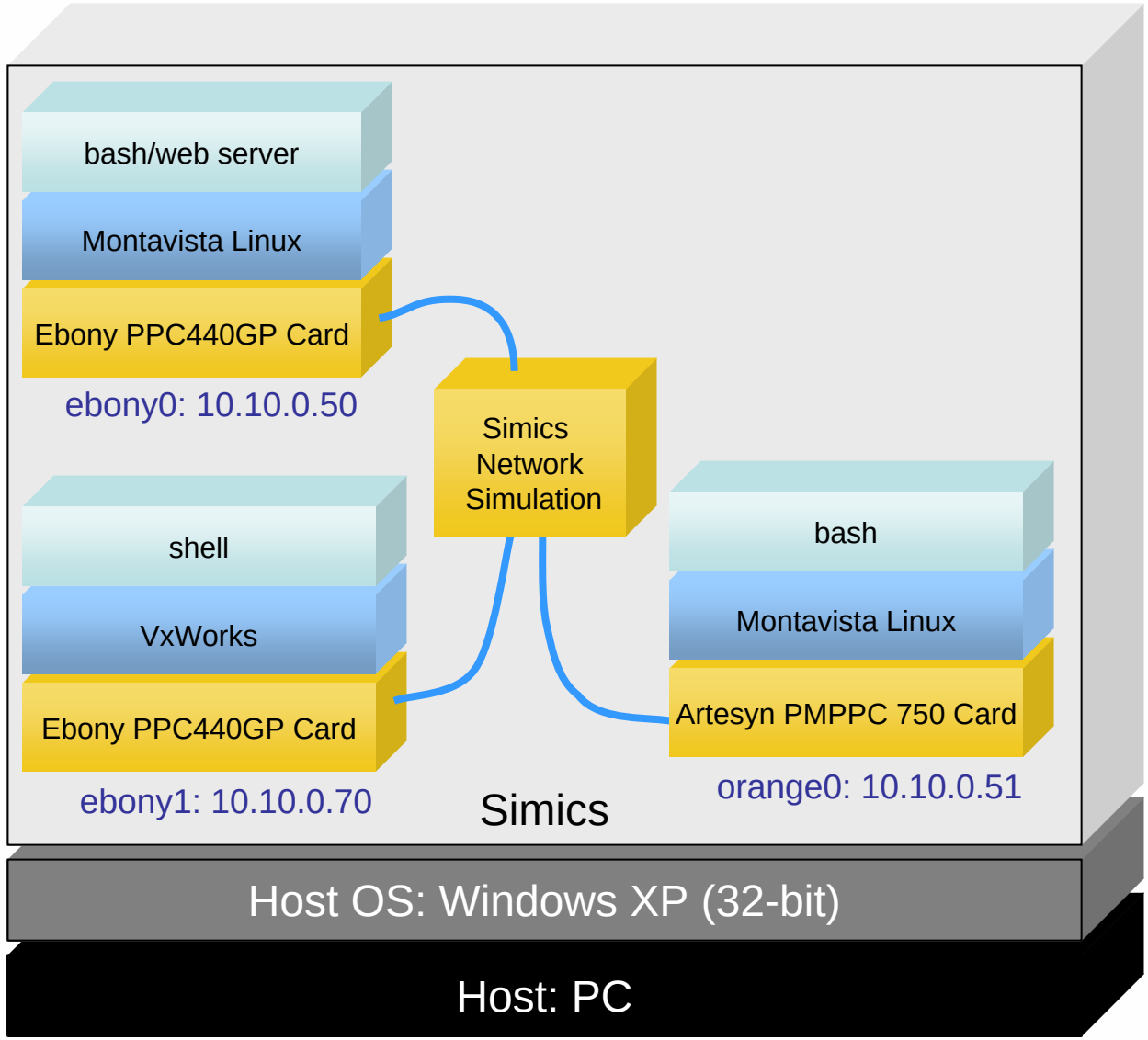
The software can't tell the difference

Runs binaries from real target

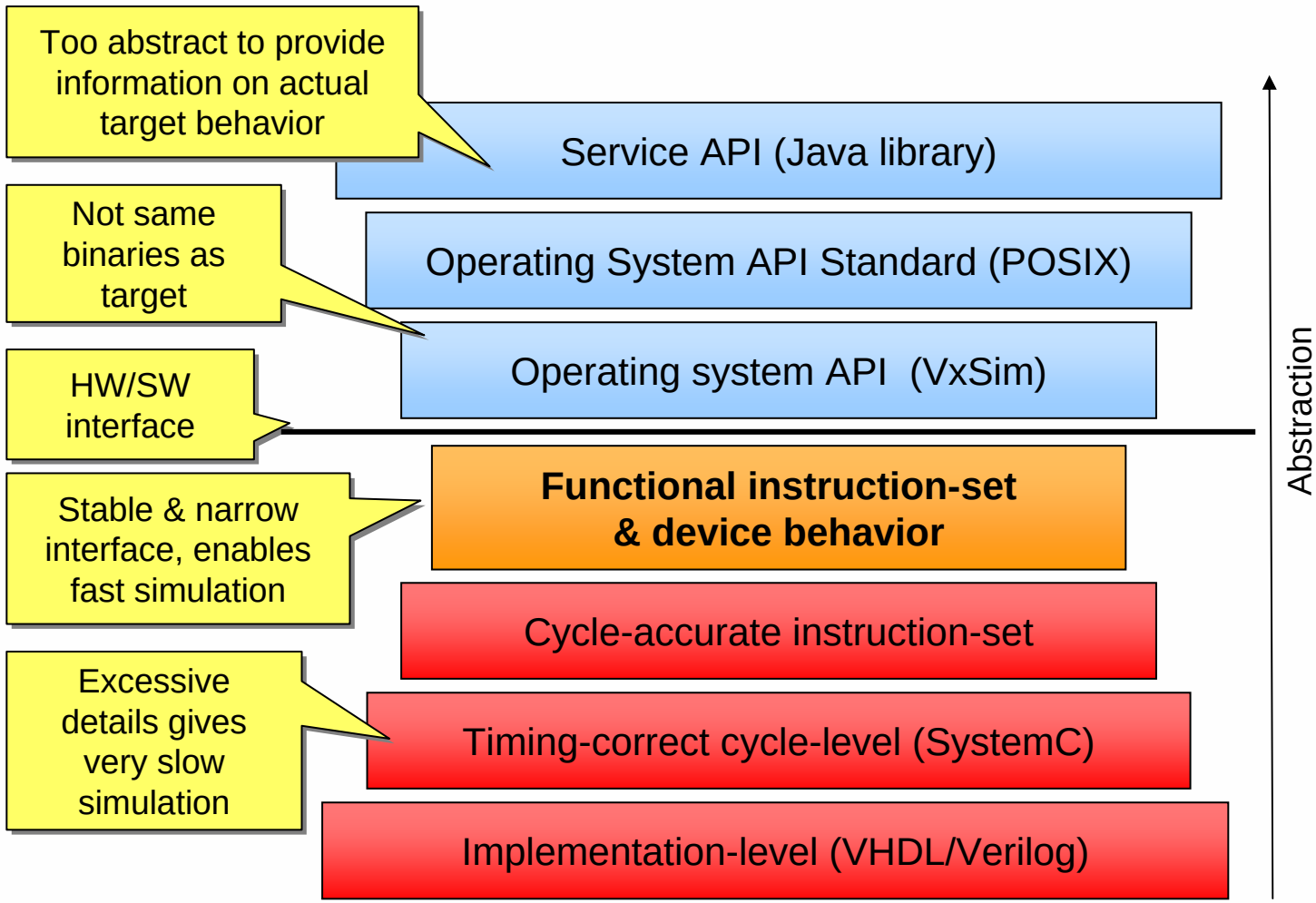



virtutech

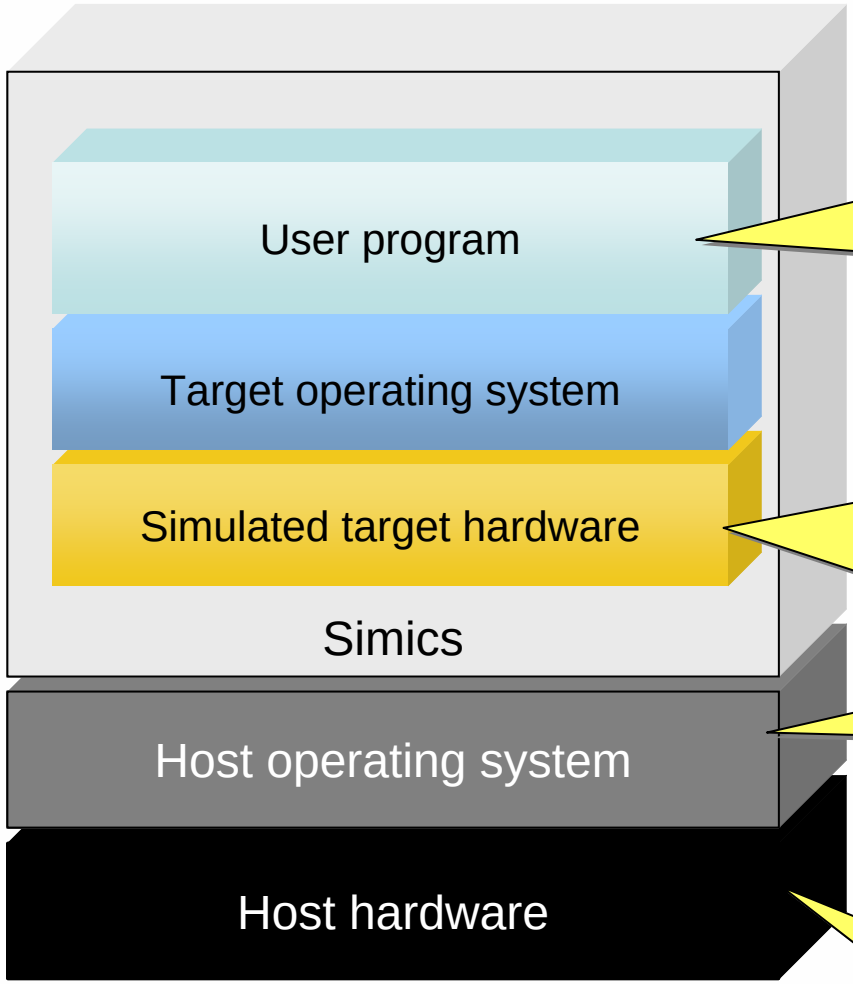
What You Just Saw



Simulation Levels



Complete Virtualization



All software: arbitrary & unmodified. Same as on a real system.

Arbitrary; currently supports Alpha, AMD64, ARM, IA64, MIPS, PowerPC, Sparc, x86, TI C6400, msp430

Linux, Solaris, Windows

PC (32- & 64-bit) or Sparc

Simics Modeling Level: Processor

- Instruction-set simulation (ISS)
- Complete and correct processor functionality
  - All instructions semantics bit-correct vs real machine
  - Supervisor-mode & user-mode
  - Runs the complete target instruction set
    - Including Altivec, SSE, 3dNow, VIS, etc. extensions
  - All accessible values represented
    - User-level registers
    - Supervisor-level registers
    - Model-specific registers, ASIs, debug register, etc.
- Memory-management unit
- Timing abstracted
  - Add details if required

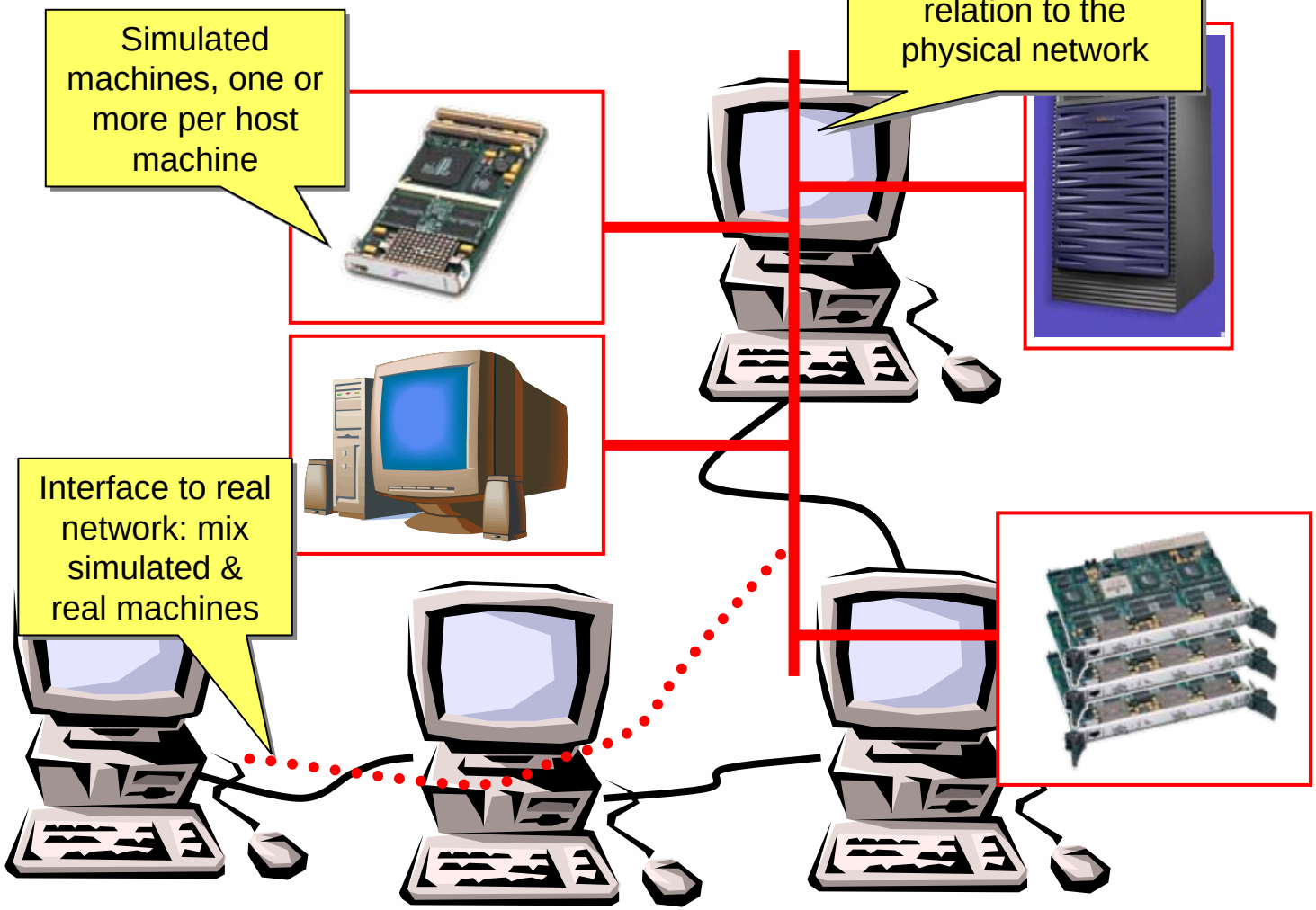
Simics Modeling Level: Devices

- Hardware modeled as a **set of devices**
  - Memory map of machine (as seen by processor)
  - At the programming register level
- Model the program-visible behavior
  - Configuration registers
  - Control register
  - Data transmitted & received
- Transaction-level modeling
  - Reads, writes, DMA transfers, network packets
- ASICs & FPGAs
  - Model programming interface behavior
  - Not detailed implementation
- Detailed timing can be added if required

Simics Modeling: No Arbitrary Limits

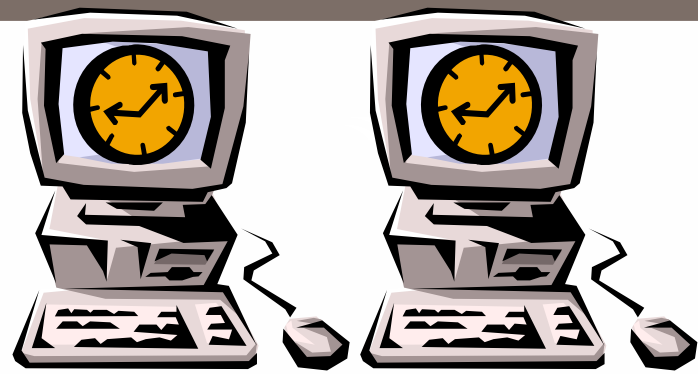
- Boards/machines:
  - Single processor
  - Multiprocessor
  - Shared memory, local memories
- Backplane/interconnect:
  - Network (ATM, Ethernet, FibreChannel, ...)
  - Shared memory
- System level:
  - Multiple boards and machines
  - Heterogeneous processors, boards, machines
- Scalability:
  - Always allows 64-bit memory space
  - Simulation can be distributed

Network Simulation with Simic



Real network of physical machines

Simics Network Timing



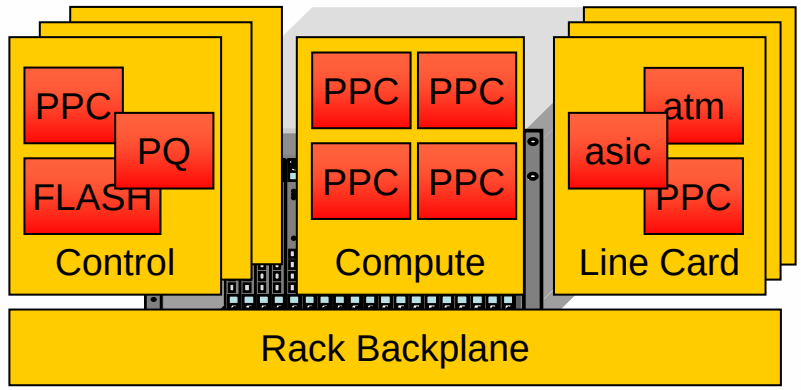
- Globally synchronized timing
  - Across processors
  - Across machines
  - Across the network
  - One processor stops → all stop
- Correct relative speed
  - 500 MHz processor will execute 10 times more instructions than a 50 MHz processor, in the same time
- Virtualized & controlled time
  - Insulated from external time
  - Time does not advance when simulation is stopped
  - Slower or faster than real world time
  - Global across the simulated network

virtutech

Some Example Systems

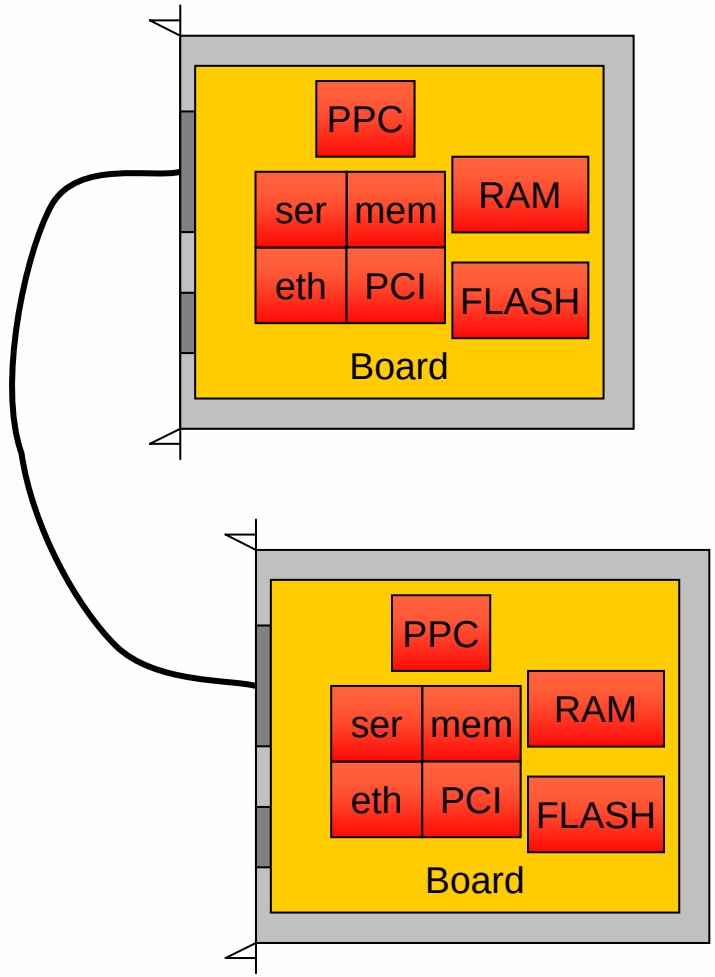
Example Model Built with Simics: Telecom

- Telecomms Switch
  - ATM Backplane
  - 10+ different card types
    - Control cards
    - Compute cards
    - Line cards
    - Timer units
    - Multimedia cards
  - 20+ cards in a rack
  - Multiple processor types
    - PowerQUICC II
    - PowerPC 750
    - PowerPC 405
    - TI C64 DSP
  - Multiple network types
    - ATM, Ethernet, Serial



Example Simics Model: Single Board

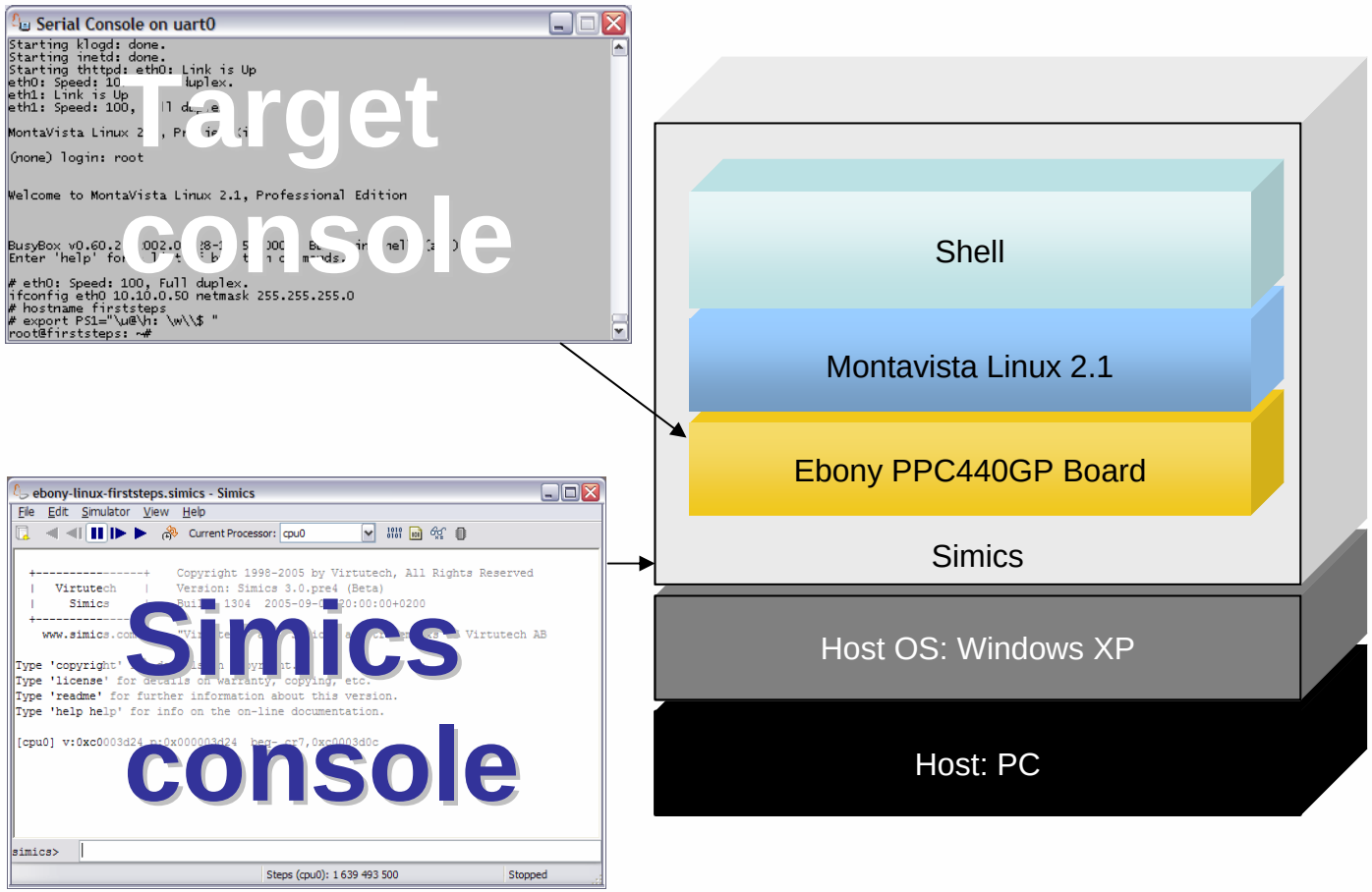
- Aerospace board
  - Single processor
    - PowerPC 750gx
  - Integrated system unit
    - Memory controller
    - Ethernet network
    - Serial ports
    - PCI
  - On-board memory
    - FLASH
    - RAM
  - PCI/X connections
  - VxWorks, in-house OS
  - Multiple cards networked



virtutech

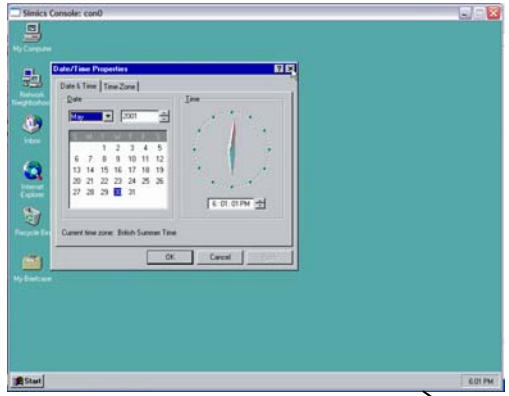
Some more Quick Demos

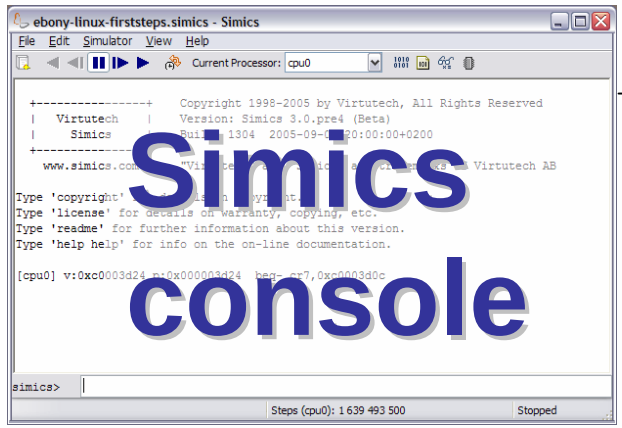
Booting a Machine



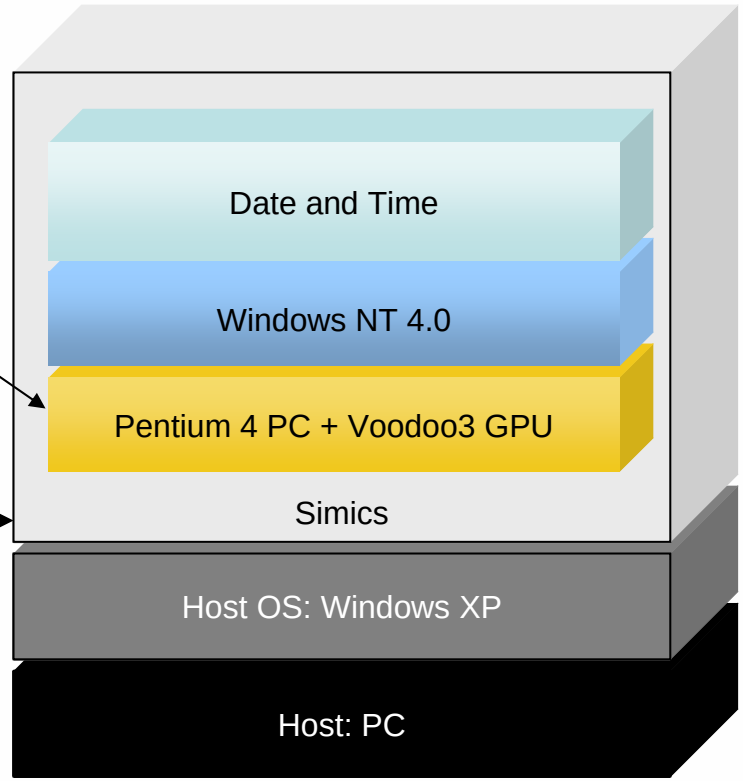

virtutech

Opening a Checkpoint & Controlling Time





Simics console



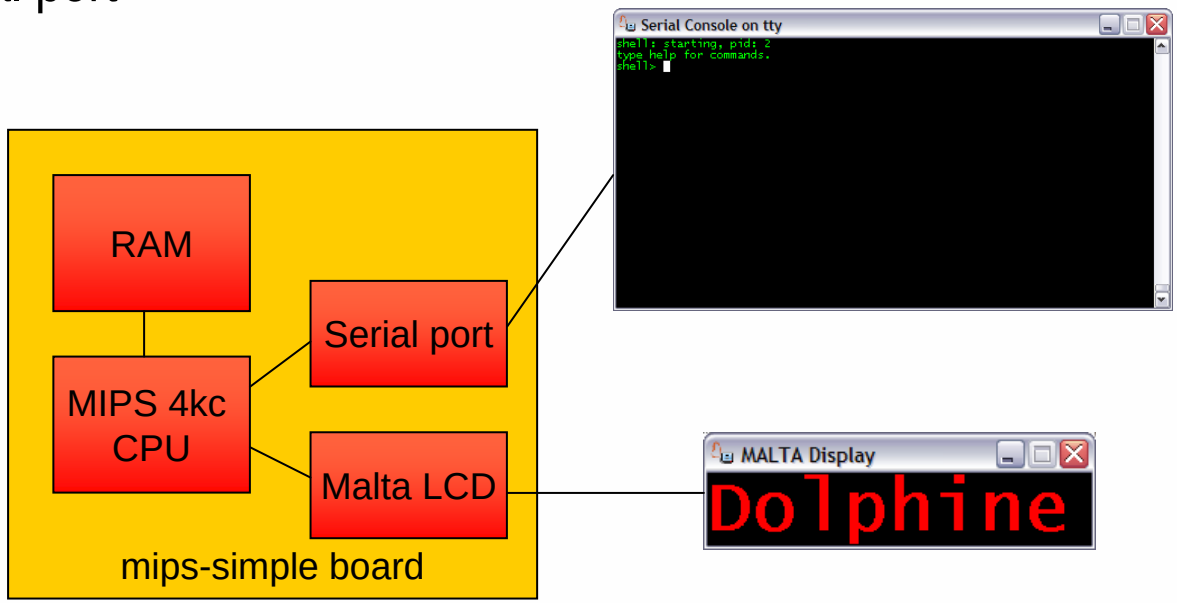

virtutech

virtutech

Simics in the CompSys Course

Virtual MIPS Computer

- Processor
- Memory
- Serial port
- LCD



- Simpler than most real-world boards = easy to program
- Runs on Sparc workstations & PCs

Hardware of the Simulated Machine

- Counter
  - Built into the MIPS processor
  - Use for periodic interrupts
- Serial port
  - Use for text input and output
  - Interrupt-driven use mandatory
- Malta LCD
  - Output-only, use for fun features
- Memory
  - Used to store your code
  - 8 MB default, should be plenty

virtutech

Helpful Simics Features

Handy Features of Simulation

- Checkpointing
  - Store current state; pick up and continue later
  - Position workload once, use many times
  - Distribute a system state to multiple developers
- Determinism
  - Same initial state gives same execution
  - Repeat the same execution any number of times
  - Investigate a problem time after time
  - Add instrumentation and reexecute

Handy Features of Simulation

- Visibility (insight without intrusion)
  - All state can be observed
    - And used to set breakpoints
  - All events can be traced and logged
    - Including interrupts, status register changes, ...
  - Input and output visible
    - Log activity, trace accesses, break on IO
- Controllability
  - Any part of machine or state can be changed
  - Fault injection

Handy Features of Simulation

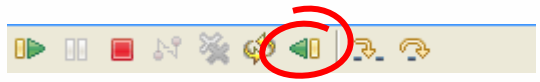
- Virtual time
  - Time is completely virtual
  - Machine does not run away
- Backwards debugging
  - Roll back execution to previous state
  - Step backwards in time
  - Reverse breakpoints
  - Set bookmarks in time

SIMICS HINDSIGHT

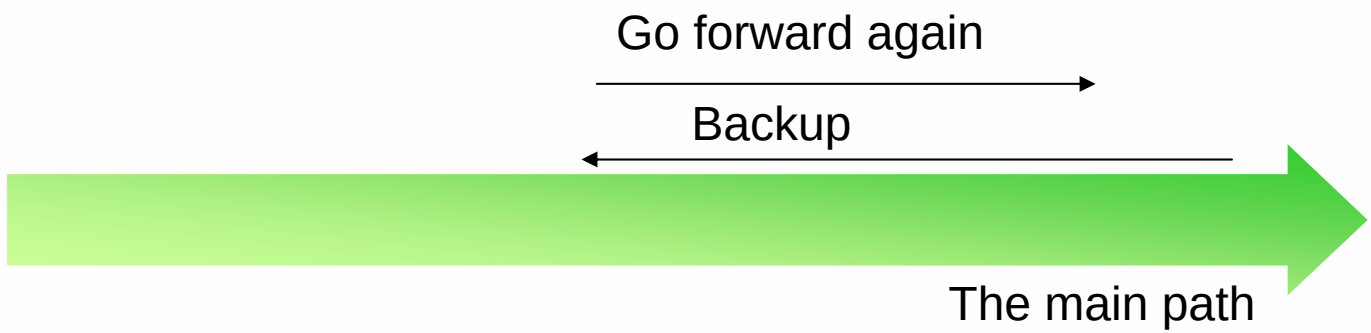
Ultimate Debug Tool – Simics Hindsight

- Going forwards: any debugger

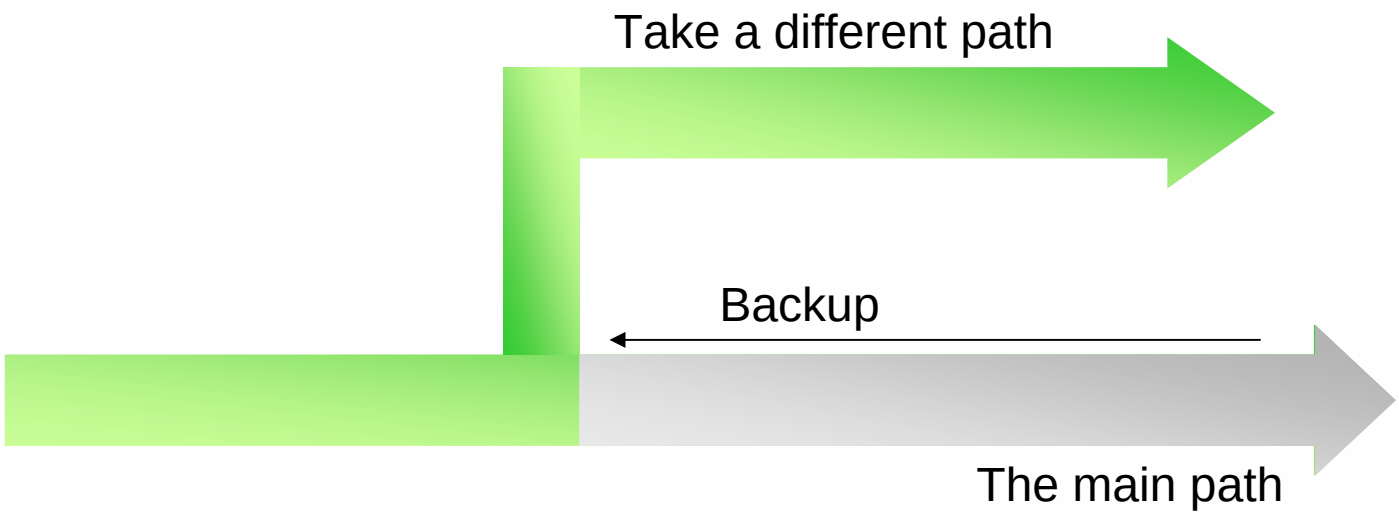
- Back up and find out what happened: Hindsight



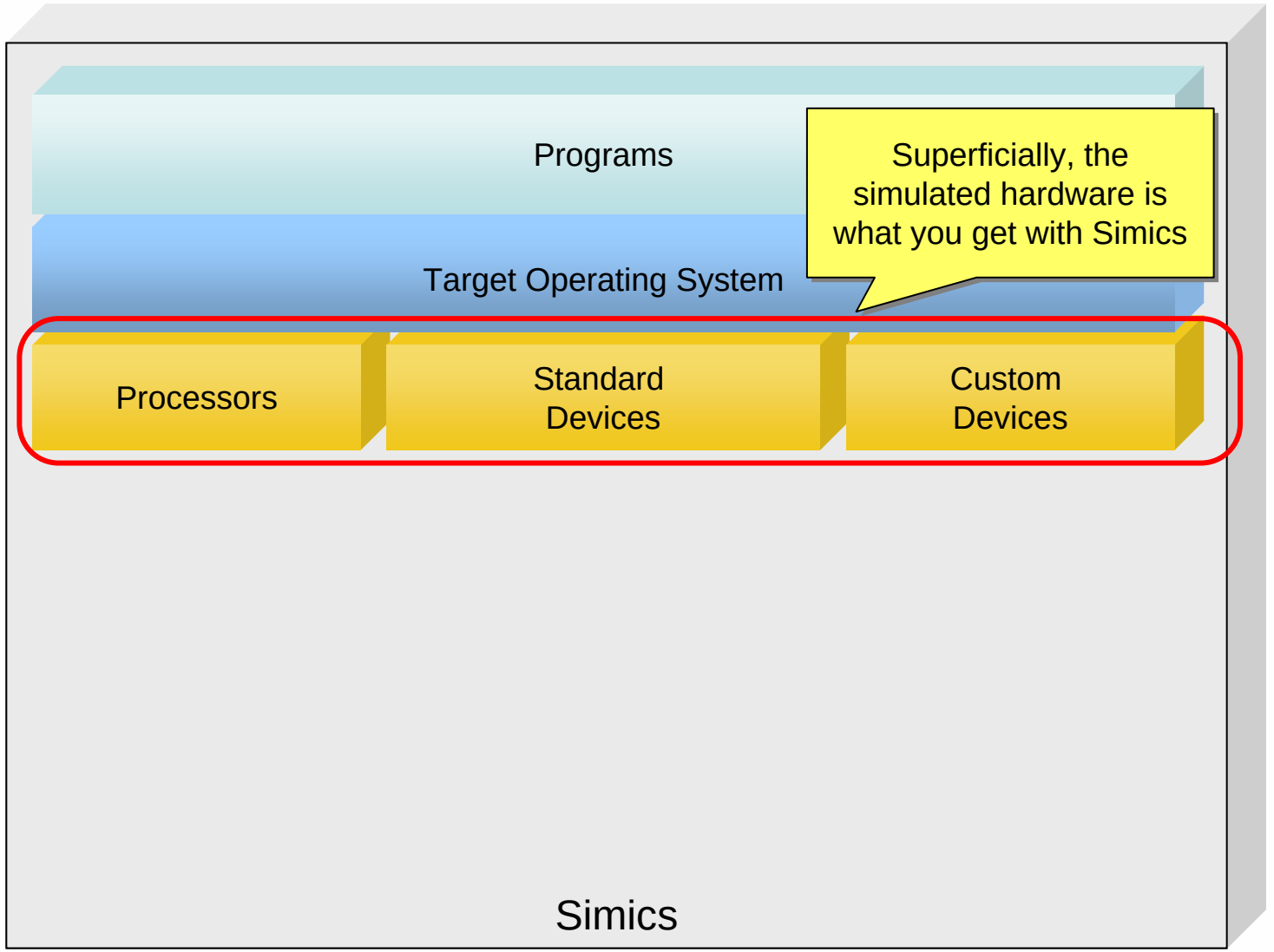
Backup... and go forward again



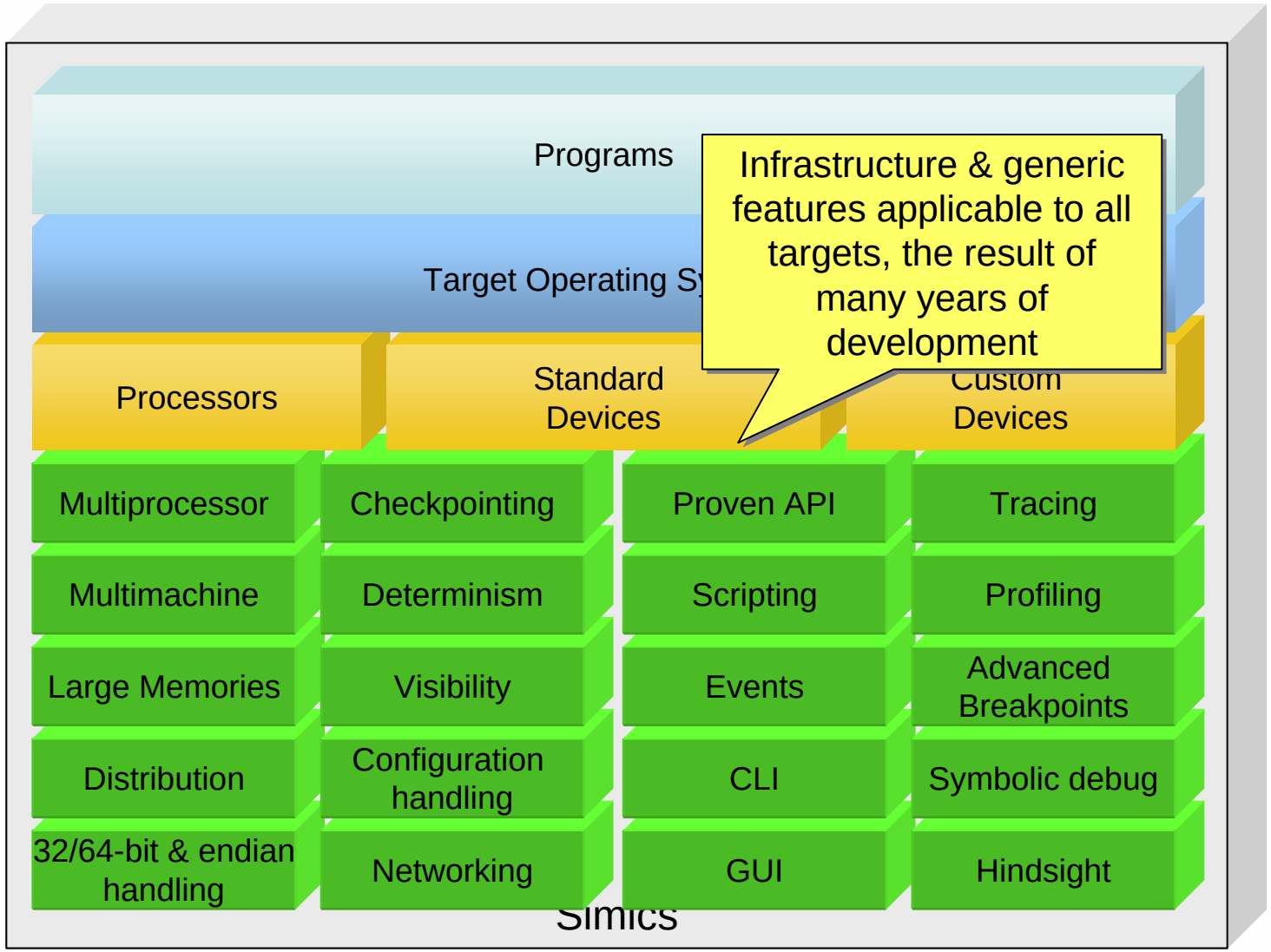
Backup... and try a different path



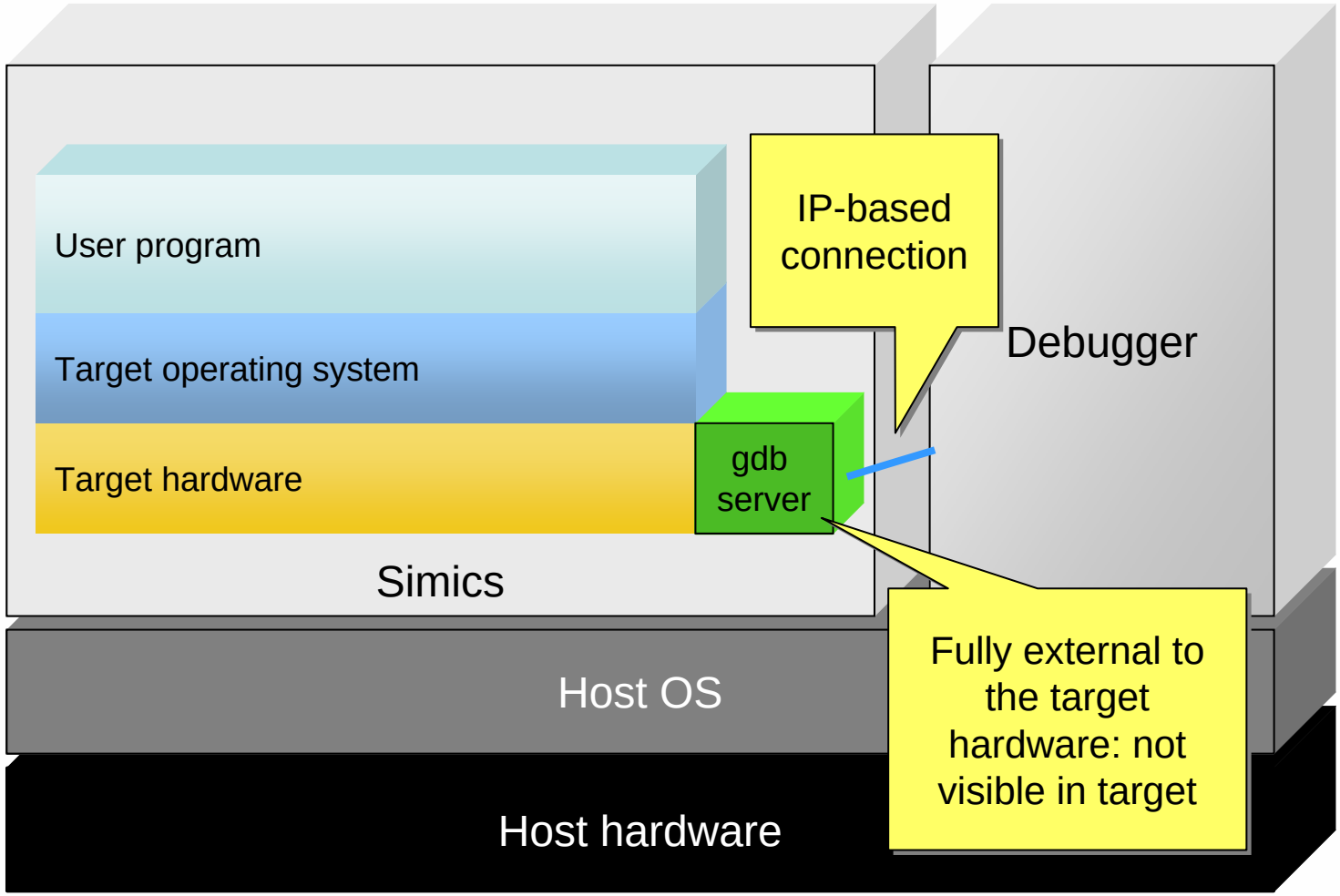
Summary: Simics Features not only Simulation



Summary: Simics Features not only Simulation



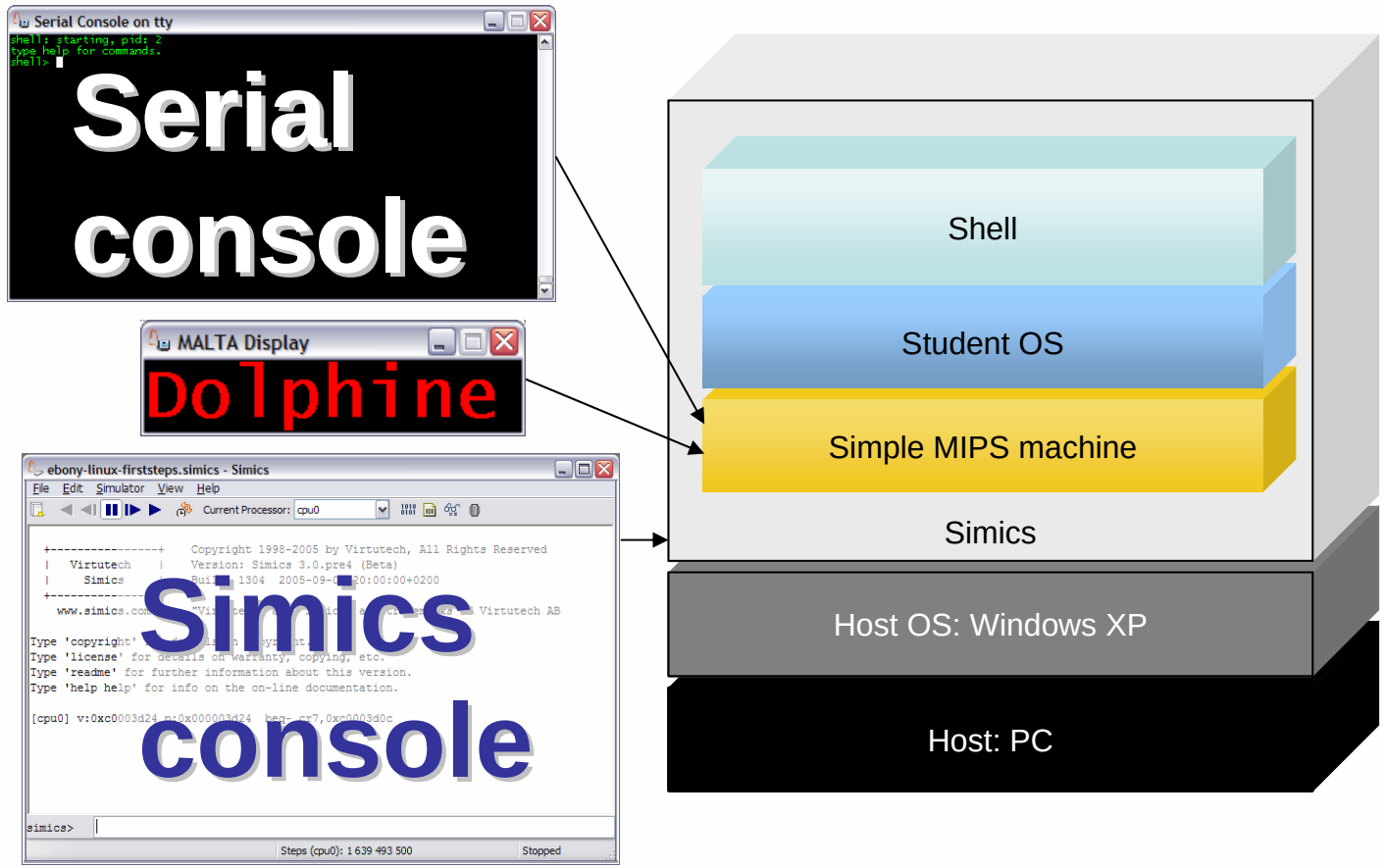
Debug Server with Simics



virtutech

Demo of Last Year's OS

Booting a Machine




virtutech

Features Used

- Serial port activity logging
- LCD activity logging
- IO trace, break
- Exception trace, break
- Control register trace, break
- Hindsight
- Starting & stopping
- Console scripting, programmed input

virtutech

Getting Simics

Simics installation at Uppsala

- Used in Erik Hagersten's research group
- Runs on all machines at MIC

- Current: Simics 1.6, 2.0, 2.2 installed

- Simics 3.0 will be installed in time for the course
  - This is what you will be using
  - Includes Hindsight backwards debugger
  - Awaiting final release from Virtutech

Simics Academic Licensing

- Simics is offered to academia for free
- Academia gets the full Simics tool, all targets

- Personal licenses
  - Free, renewable yearly
  - Node-locked to a single computer
  - Obtain via www.simics.net
- Site licenses
  - Free
  - 10000 licenses baseline, floating
  - Much cheaper than buying specialized hardware labs

virtutech

Thank You!