SIMICS – Overview and usability in DESEREC

Karl Mayer
IABG mbH

September 26th, 2006

DESEREC

Dependability and Security by Enhanced Reconfigurability

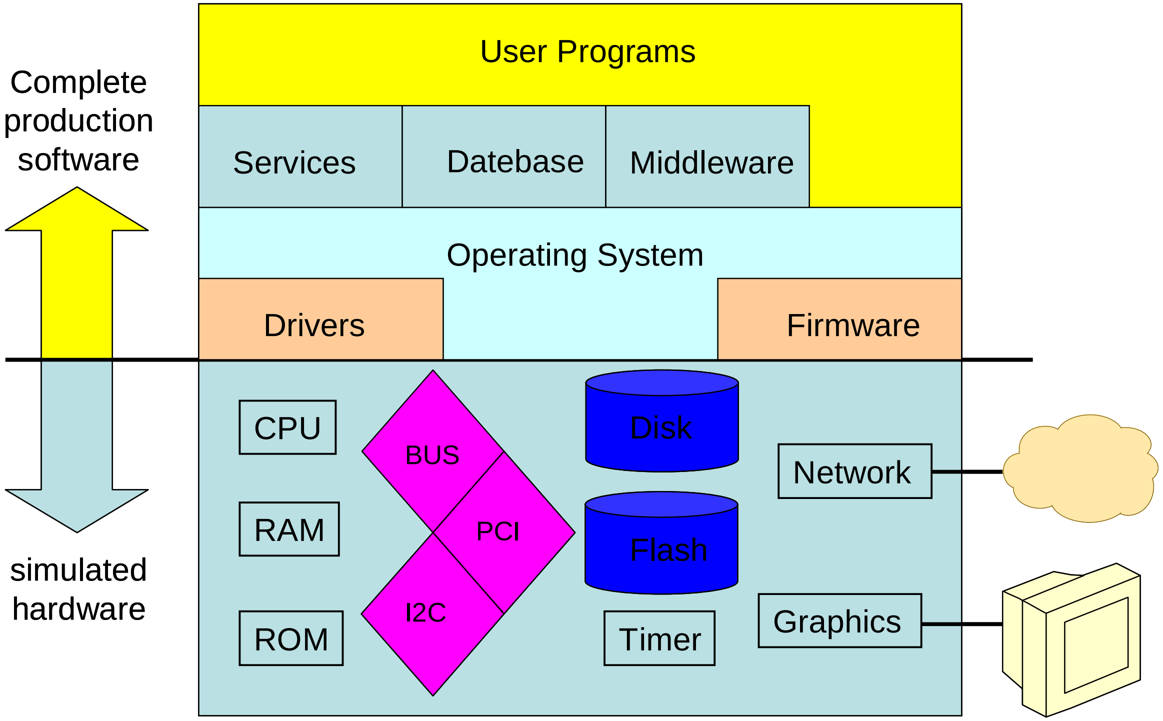
Information Society Technologies

Agenda

- What is Simics?
- How to perform Simics simulations?
- What are the requirements?
- What are the most interesting features of Simics?
- How can Simics be used in DESEREC?
- Short Demo (remote VNC connectivity?)
- Questions? (please ask immediately)

What is Simics?

- Simics has been developed by Virtutech (www.virtutech.com)
- Simics is a "full-system simulator"
  - *Simics* simulates the hardware of a system at such a level of detail that complete software stacks from real systems can run on the hardware without any modification



What is Simics?

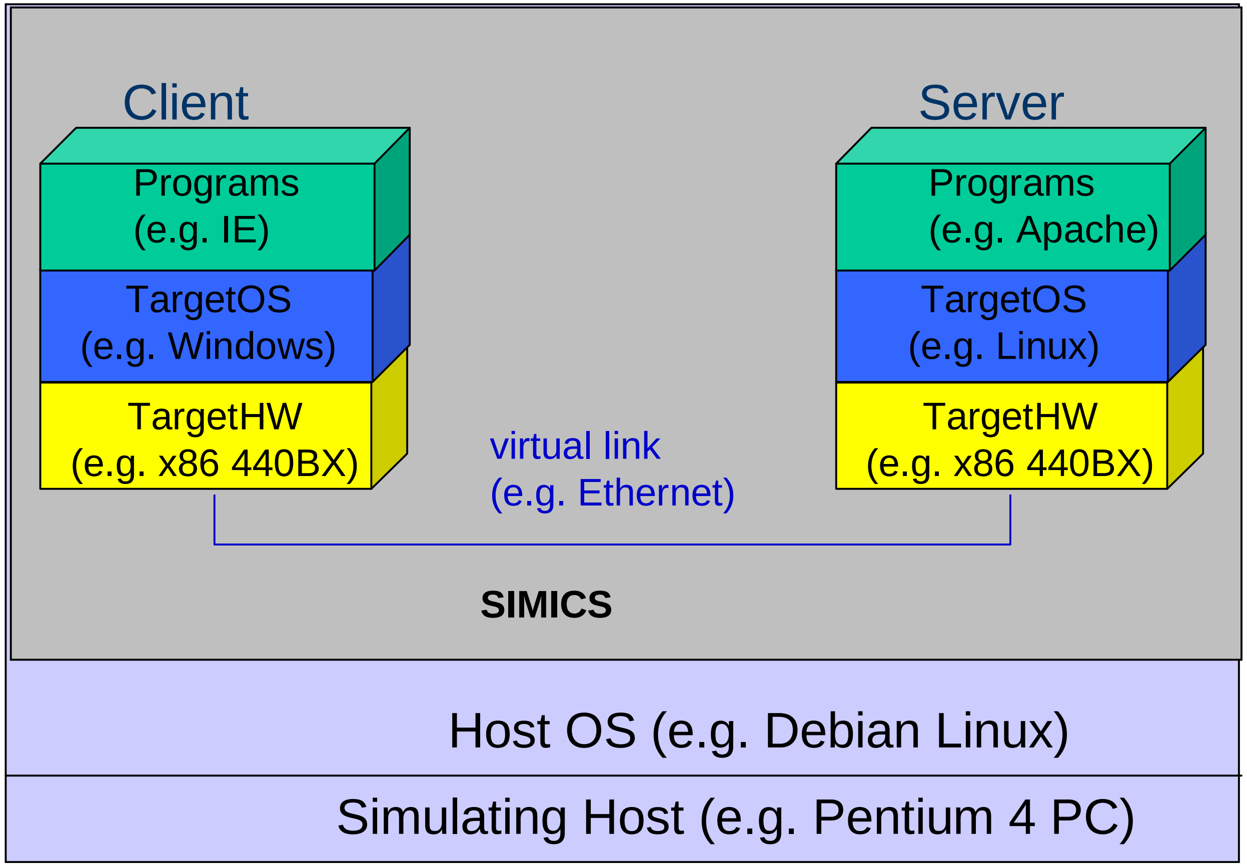
Benefits of a "*full-system simulator*"

- n This enables the user,
  - 4 without possessing the hardware,
  - 4 and without modifying the software
- n to
  - 4 test, debug, and improve software
  - 4 make performance measurements
  - 4 validate the overall system
  - 4 change hardware quickly (e.g. increase power or memory)
  - 4 get scalability information

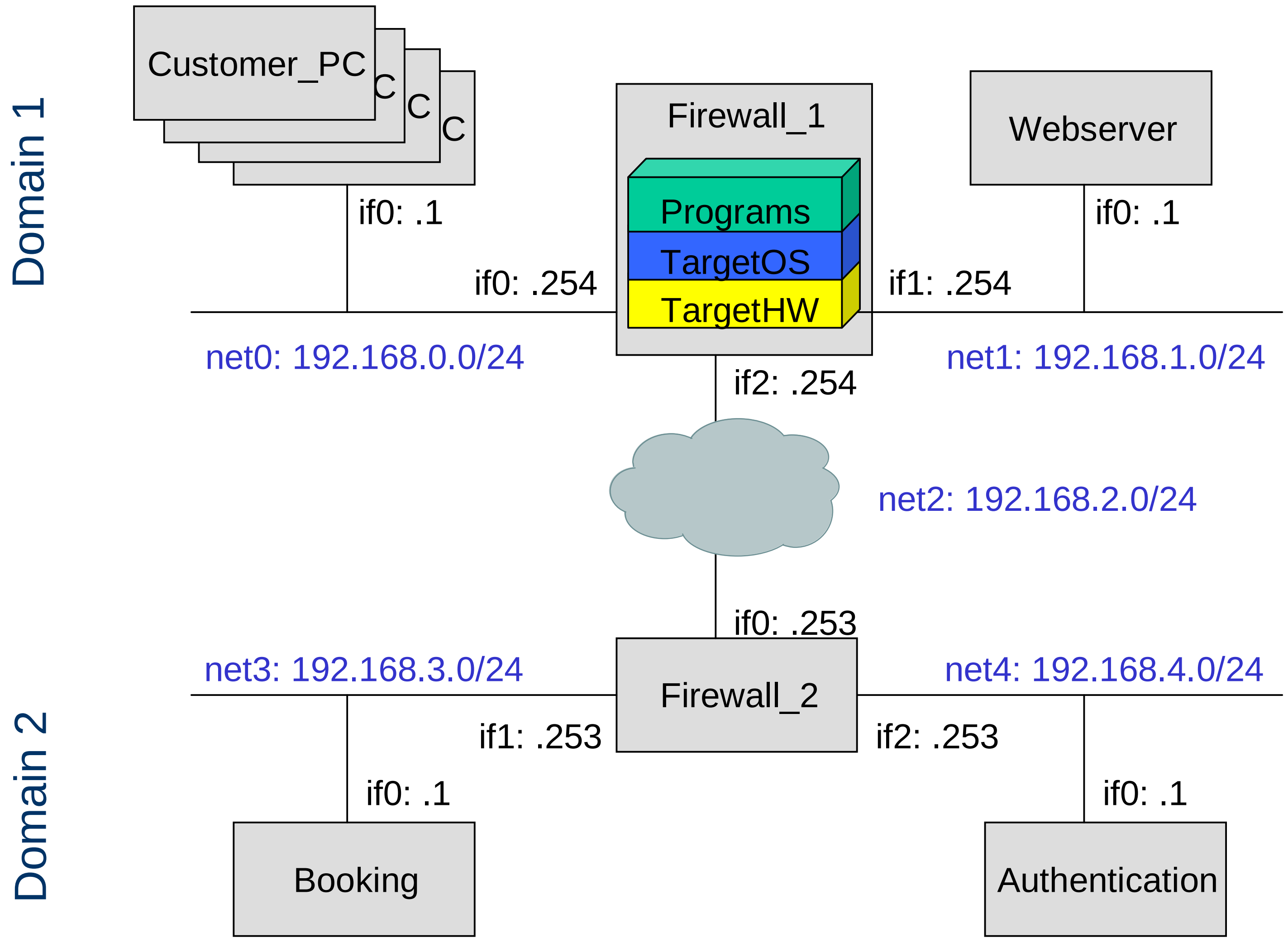
How to perform Simics simulations?

- Setup your Simics environment
  - Buy host systems (e.g. powerful PCs running Linux)
  - Install Simics on each host system
- Plan your simulation
  - Identify hardware, software, and network types that represents your simulated architecture
  - Identify/create your network configuration (e.g. IP addresses, routes, etc.)
  - Plan the test to be performed (e.g. penetration tests)
- Create your Simics simulation
  - Create your own hardware component models (or reuse existing component models)
  - Connect your components together to create system models (or reuse existing system models)
  - Create/download/obtain your software running on each system
  - Create your Simics configuration file
- Run the simulation
  - Perform your tests

How to perform Simics simulations?



How to perform Simics simulations?



How to perform Simics simulations?

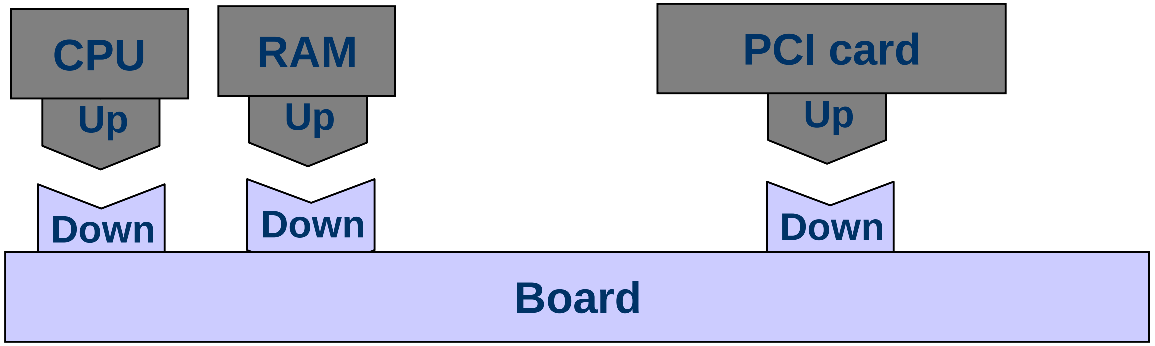
- *Some available target (system) models:*
  - SunFire:
    - simulates the Sun Enterprise 3000–6500 server series from Sun Microsystems
    - runs Solaris or Linux
    - the processor modelled is UltraSPARC II
  - Ebony:
    - models a PPC-based Ebony card with a PPC440GP 32-bits processor
    - it boots Linux 2.4 and VxWorks
  - *x86 440BX:*
    - simulates various x86 compatible processors, ranging from 486 to Pentium 4 and AMD64 processors
    - it is capable of booting several Linux versions, Windows NT4.0, 2000 and XP
    - it includes standard PC devices, such as graphic devices, north and south bridges, floppy and hard disks

How to perform Simics simulations?

- *Some available network links:*
  - *Ethernet:*
    - simulates an ideal Ethernet link without collisions
    - performs delivery of complete frames sent from one device to any other device connected to the link
    - can be viewed as Ethernet hub or switch to which several devices can be connected
    - latency is adjustable
  - Serial link
    - forwards packets between two serial link interfaces (p2p connection)
    - can be seen as cable
  - Wireless links???
    - there are plans for the future

How to perform Simics simulations?

- *How to create your own component models?*
  - *Device Modelling Language (DML):*
    - C-like programming language for writing device models for Simics
  - *Issue:*
    - *Really low level (definition of registers, etc.)*
    - *Documentation of real components often not available (e.g. Cisco)*
    - *Software (e.g. driver) has to run on top of the model*
- *How to connect your components together?*
  - *Connectors help in creating systems*
  - *Connector directions: Up, Down, Any, Hot plug*



How to perform Simics simulations?

- *How to customize ready-to-run configurations?*
  - *For some targets there are already ready-to-run configuration files including target model (e.g. x86 440BX) and software (e.g. Linux)*
  - *Change parameters of system configurations:*

    ```
    $freq_mhz = 100
    $memory_megs = 128
    $host_name = "ebony0"
    ```

  - *Change/extend Simics scripts:*
    - *creates a simulation with two ebony boards with different parameters*

    ```
    $freq_mhz = 100
    $memory_megs = 128
    $host_name = "ebony0"
    set-component-prefix "ebony0_"
    run-command-file "ebony-linux-common.simics"

    $freq_mhz = 200
    $memory_megs = 256
    $host_name = "ebony1"
    set-component-prefix "ebony1_"
    run-command-file "ebony-linux-common.simics"
    ```

  - *Change components and there attributes*

What are the requirements?

- *Simulation host(s)*
  - *32-bit x86 architecture with Linux (e.g. Red Hat 7.3 or Debian) or Windows (e.g. Windows 200 or newer)*
  - *64-bit x86 architecture with Linux*
  - *64-bit SPARC with Solaris*
  - *with at least 512MB RAM (the more the better) and some GB free disk space*

- *Simics Models for the various components*
  - *processors*
  - *memory*
  - *interfaces*
  - *network links*
  - *etc.*

- *Software that runs on top of the emulated hardware*

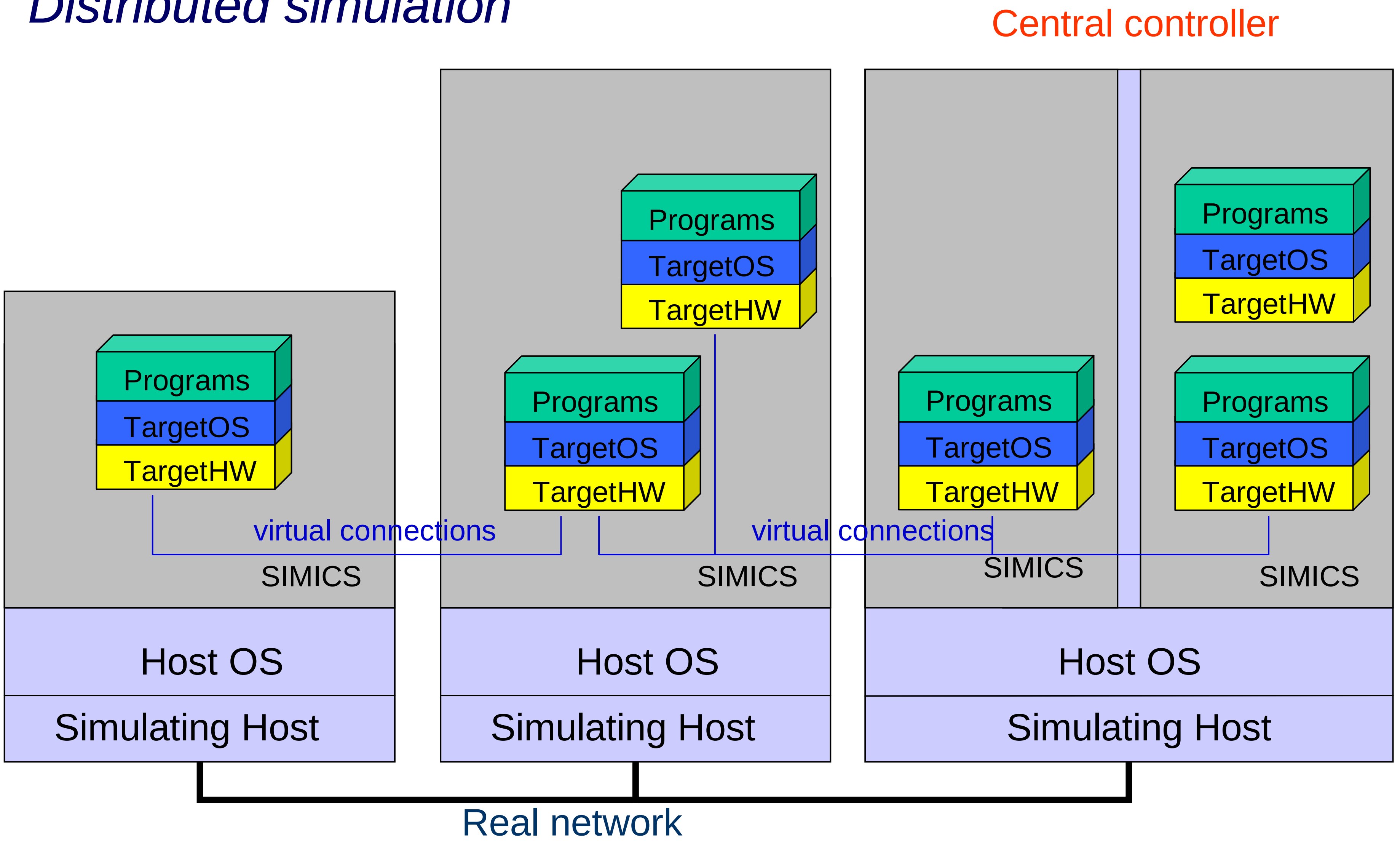
What are the most interesting features of Simics?

Connection with real networks

- *simulation progresses in real time mode*
  - *possibly skipping of events in order to keep real-time speed*
- *connection types:*
  - *Port forwarding:*
    - *the simulation host forwards preconfigured ports between simulation and real network*
    - this is limited to TCP and UDP traffic
  - *Ethernet bridging:*
    - *Ethernet frames are forwarded from the real Ethernet interface of the simulating host to the virtual Ethernet interface in the simulation and vice versa*
  - *IP routing:*
    - *the simulation host represents an IPv4 router between the simulated Ethernet and the real Ethernet*
  - *Host connection:*
    - *The simulation host is connected as a host to the simulated Ethernet*
    - *Simulated targets cannot access other real systems (unless the simulation hosts provides router functionality)*

What are the most interesting features of Simics?

Distributed simulation



What are the most interesting features of Simics?

Checkpointing

- *Simics allows to store all states of a simulation in a so called checkpoint*
- *Simulation can be started from a certain checkpoint*
- *For example, a checkpoint can be stored after all target systems have booted (booting takes a lot of simulation time) and several simulations (with equal simulation setup) can be started from this point*

Hindsight

- *For debugging of software/system errors, Simics can run back in time*

What are the limits of Simics?

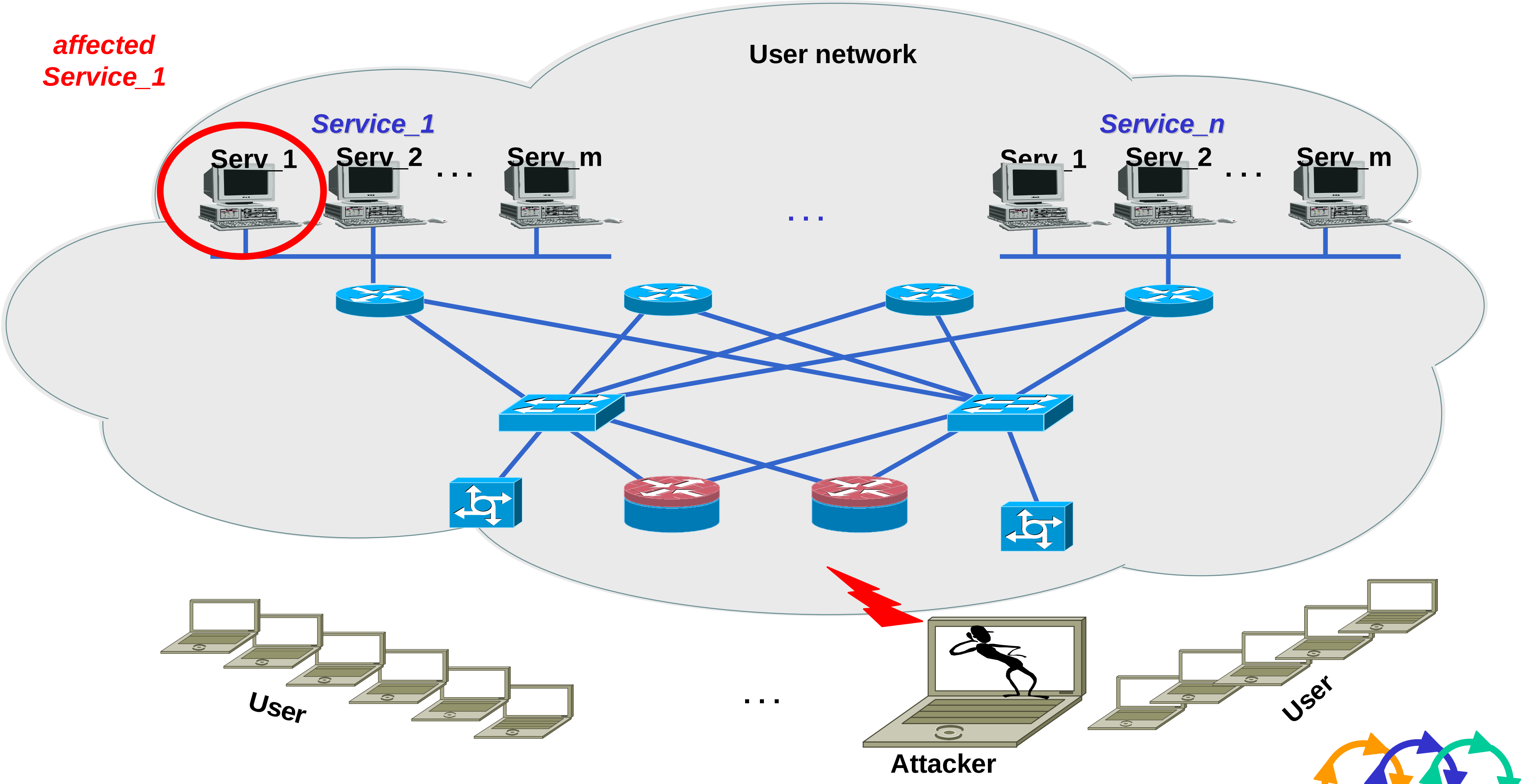
- *Availability of components:*
  - *A lot of different platforms have been modelled so far; however, not all*
  - *Creating new models is time consuming*
  - *For vendor specific components no insights may be given that are required for modelling*
  - *Using an available model with similar features and functions as substitution for a non-available one is feasible and appropriate in many cases, e.g. using Linux PC router instead of a Cisco router*
- *Simics accuracy*
  - *Although Simics models hardware very detailed, it is not the real hardware*
  - *Simics simulation should be calibrated*
  - *Regarding performance, Simics simulation give not exact numbers but a magnitude/rough numbers*

What are the limits of Simics?

- *Simics' scalability depends on:*
  - *what is the CPU power of one simulation target in relation to the CPU power of the host system*
  - *what is the duty cycle of the simulated CPU*
  - *what is the RAM size of the simulated target compared to the RAM size of the host system*
  - *how many targets are simulated on one host*
  - *the time one is able to wait for a simulation result (in case of virtual time mode)*

How can Simics be used in DESEREC?

- *Assessment of implications of vulnerabilities in a save environment*
  - *Administrator can test vulnerability of outdated software*
  - *Administrator can test software against new threat*



How can Simics be used in DESEREC?

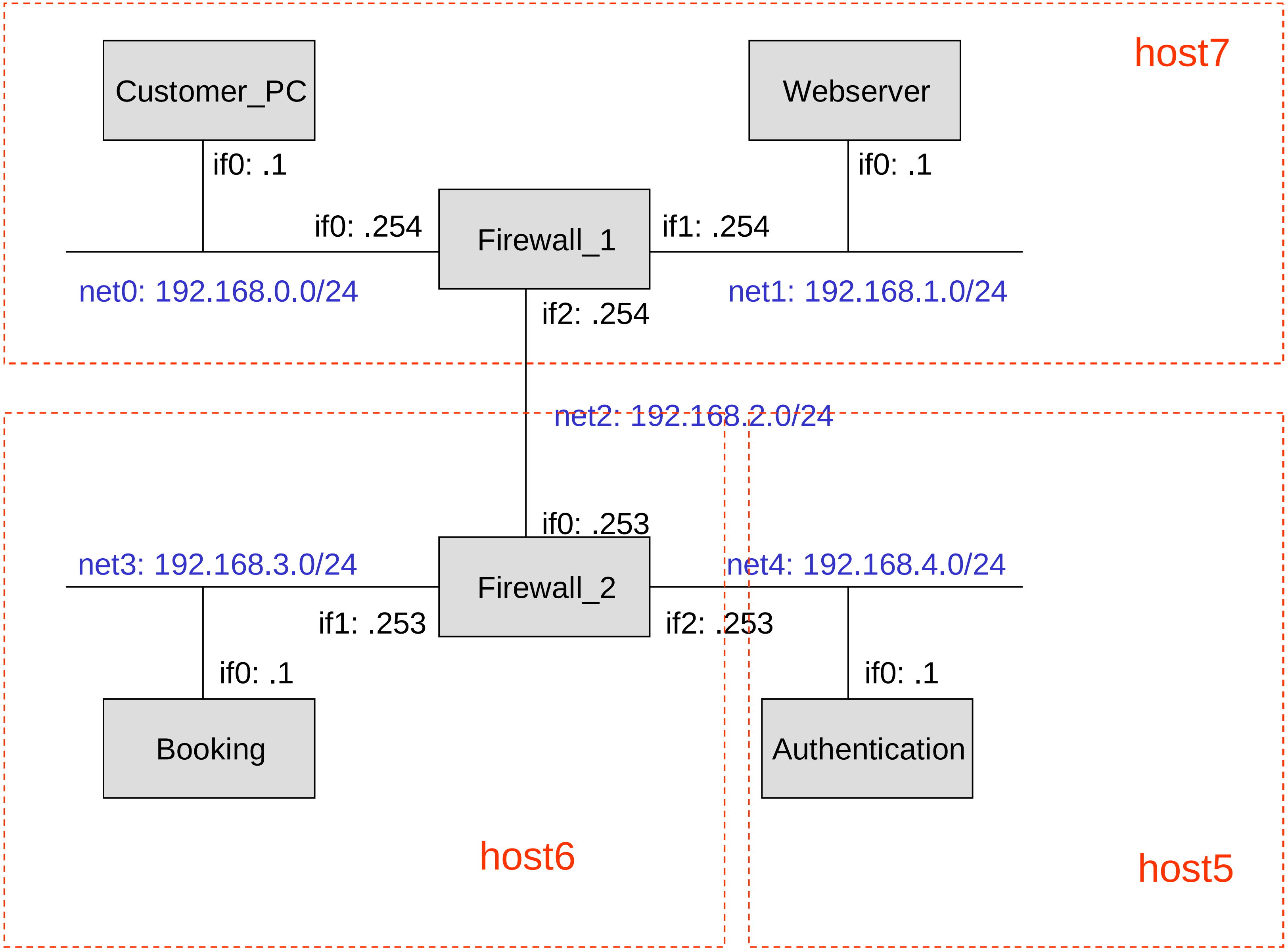
- *Pre-assessment of reconfiguration effects in a save environment*
  - *Perform reconfiguration and identify the best way to do it*
  - *Perform attacks during the reconfiguration process and evaluate threat level*
  - *Only the relevant systems needs to be part of the simulation (e.g. firewall, server, attacker)*
  - *Simulations can be stopped, frozen, and repeated*
  - *Simics allows to go back in time and check registers etc.*

- *Pre-assessment of performance implications of reconfiguration in a save environment*
  - *Assess performance of the reconfiguration process (e.g. duration)*
  - *Assess performance of affected services during reconfiguration (throughput, delay, number of simultaneous connected clients, etc.)*
  - *Only the relevant systems needs to be part of the simulation*
  - *Simulations can be stopped, frozen, and repeated*

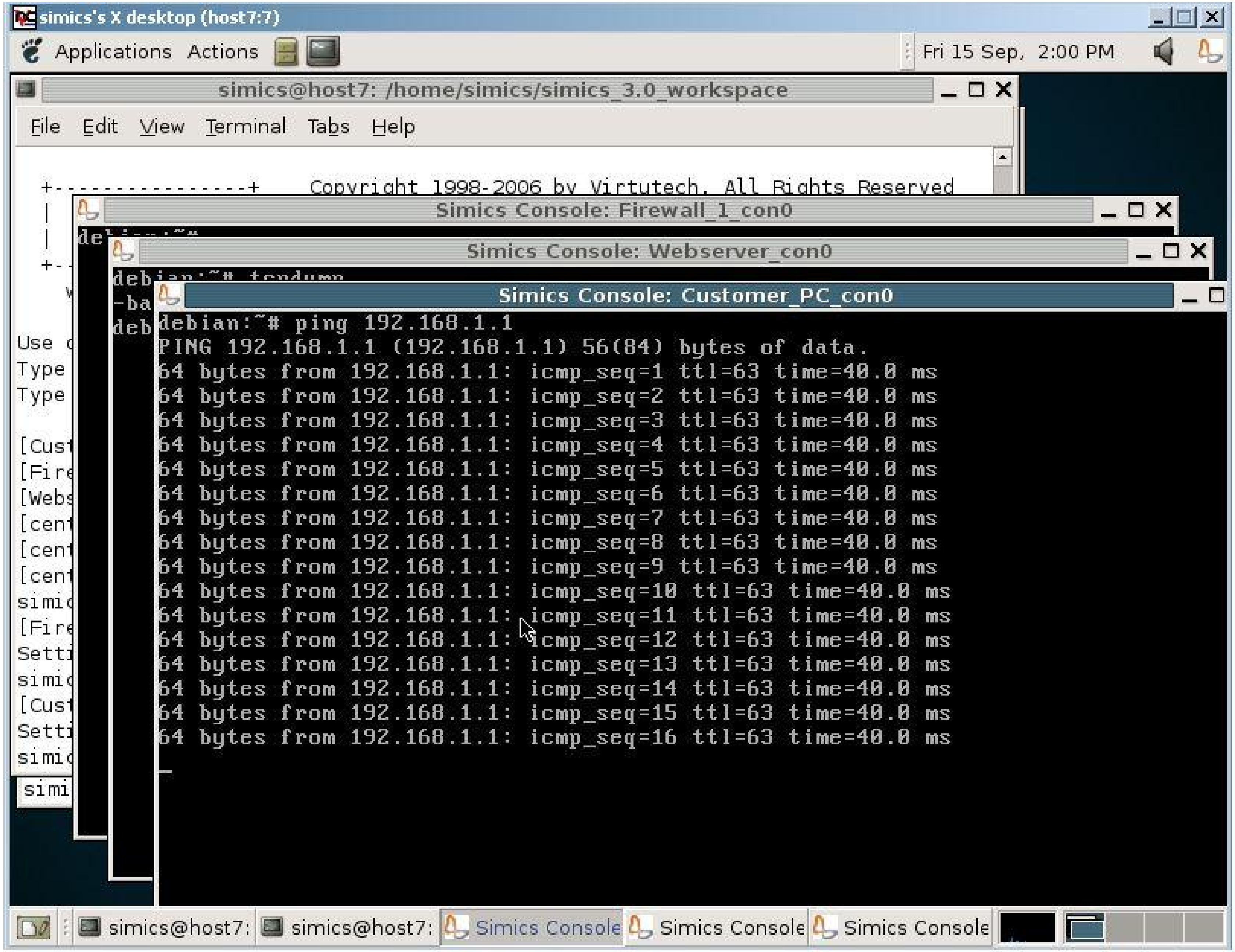
How can Simics be used in DESEREC?

- *Relation of Simics to the DESEREC architecture*
  - *Simics is a simulation tool that will contribute to the generation of Detection & Reaction scenarios*
- *Steps to be performed:*
  - *Identification of critical path in an information system*
  - *Create a Simics simulation of the information system including all* ***hardware*** *and* ***software*** *of the* ***critical path*** *including:*
    - *operating systems*
    - *network services (e.g. firewalls, IPsec gateways, etc.),*
    - *applications (server and client programs, etc.),*
    - *performance tools (iperf, mgen, monitoring tools, etc.),*
    - *attack tools (nmap, nessus, webspy, aldebaran, etc.),*
    - *configuration files (e.g. access lists, firewall rules, IPsec configurations, etc.)*
    - *user behaviour (time driven scripts that perform certain actions at certain times)*
    - *components and network failure models (e.g. shutting down interfaces at certain times)*
  - *Produce and assess system configurations, identify and assess vulnerabilities, and identify performance bottlenecks*

Demo


host7
Customer_PC
Webserver
if0: .1
if0: .1
if0: .254
Firewall_1
if1: .254
net0: 192.168.0.0/24
net1: 192.168.1.0/24
if2: .254
net2: 192.168.2.0/24
if0: .253
net3: 192.168.3.0/24
Firewall_2
net4: 192.168.4.0/24
if1: .253
if2: .253
if0: .1
if0: .1
Booking
Authentication
host6
host5

Demo


simics's X desktop (host7:7)
Applications Actions
Fri 15 Sep, 2:00 PM
simics@host7: /home/simics/simics_3.0_workspace
File Edit View Terminal Tabs Help
Simics Console: Firewall_1_con0
Simics Console: Webserver_con0
Simics Console: Customer_PC_con0
simics@host7:
simics@host7:
Simics Console
Simics Console
Simics Console

```
Copyright 1998-2006 by Virtutech, All Rights Reserved
```

```
debian:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=11 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=12 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=13 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=63 time=40.0 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=63 time=40.0 ms
_
```

Questions

Questions?

Contact

Karl Mayer

Technical consultant
IABG mbH, Munich
+49 -89 -6088 -2066
mayer@iabg.de
www.iabg.de