Linux and Microkernels

Hermann Härtig
Linux Infotag
TU Dresden
Oktober 2004

1992

Andy Tanenbaum ./. Linus Torvalds

“The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O.”

“Microkernels have won.”

Linus Torvalds on Microkernels as fashion

“In fact, this made me think that the **microkernel approach was essentially a dishonest approach** aimed at receiving more dollars for research. I don't necessarily think these researchers were knowingly dishonest. Perhaps they were simply stupid. Or deluded. I mean this in a very real sense. The dishonesty comes from the intense pressure in the research community at that time to pursue the microkernel topic. In a computer science research lab, you were studying microkernels or you weren't studying kernels at all. So everyone was pressured into this dishonesty, even the people designing Windows NT. **While the NT team knew the final result wouldn't approach a microkernel, they knew they had to pay lip service to the idea.**”

... and he had a valid point here !

Linux Torvalds on MACH ...

... as basis for Apple OS X

"Frankly, I think it's a piece of crap,"

What is a (real) microkernel-based system ?

"The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, and possibly the I/O."

"Microkernels have won."

Andy Tanenbaum, 1992

"The alternative is a microkernel-based system, in which most of the OS runs as separate processes, mostly outside the kernel. They communicate by message passing. The kernel's job is to handle the message passing, interrupt handling, low-level process management, **but no I/O drivers**."

"Microkernels will win."

Hermann Härtig, 2004

An example of a real microkernel: L4 family of kernels (Jochen Liedtke)

The microkernel provides:

- address spaces
- threads
- communication
- interrupts and page faults as messages

An example implementation: L4/Fiasco (TU Dresden)

ca 15 KLOC

mature

fast

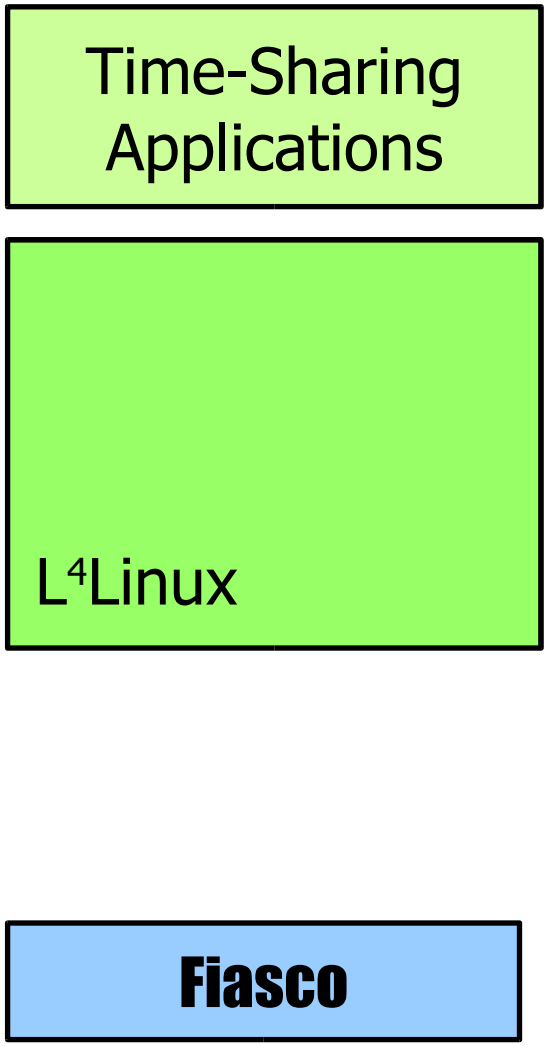
download it from

os.inf.tu-dresden.de
(GPL)

Another implementation:
L4/Pistacchio: Karlsruhe and Sydney

Dresden

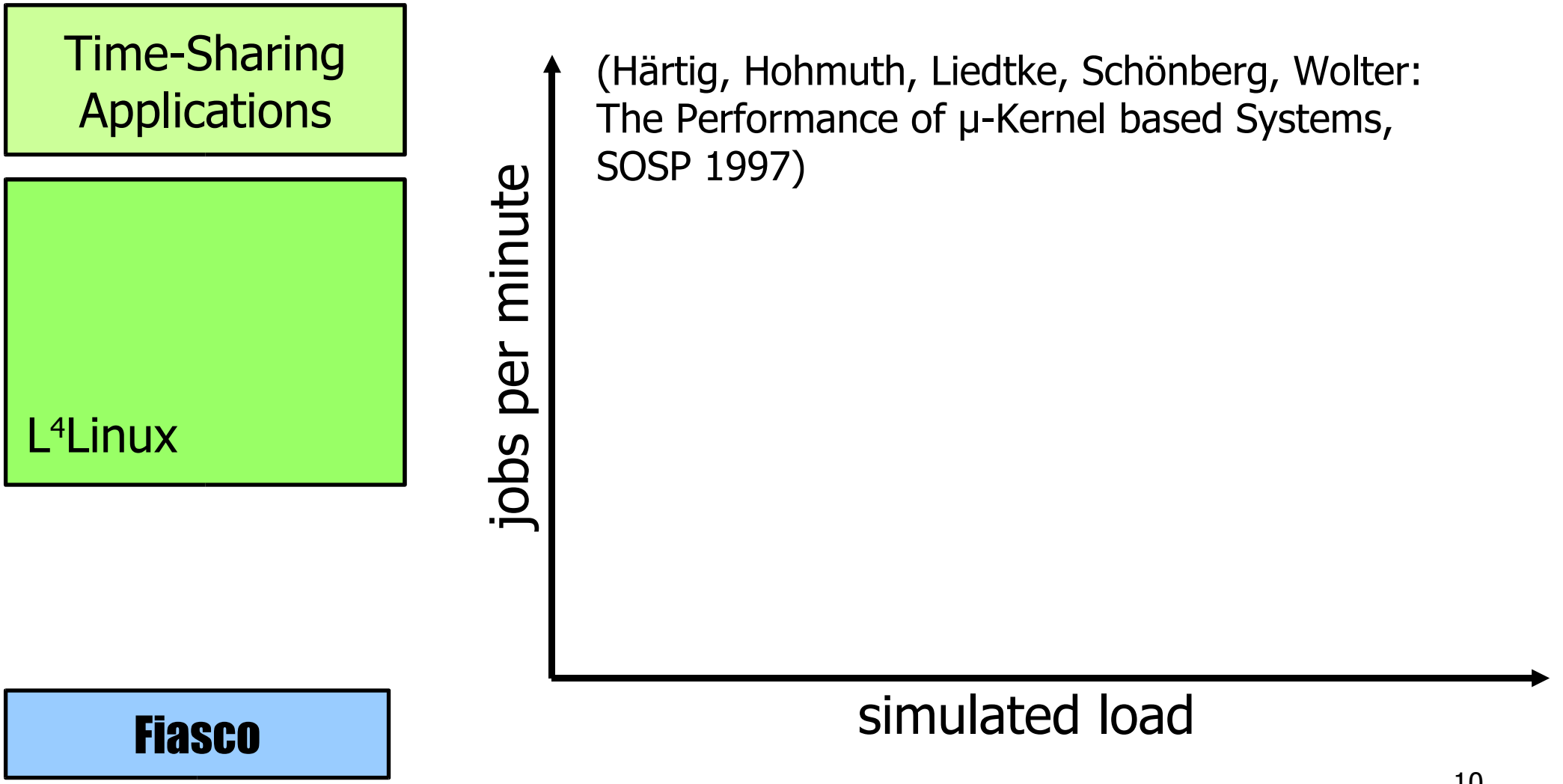
L4Linux: the first L4 application (TU Dresden)



download it from

os.inf.tu-dresden.de
(GPL)

L4Linux: the first L4 application (TU Dresden)

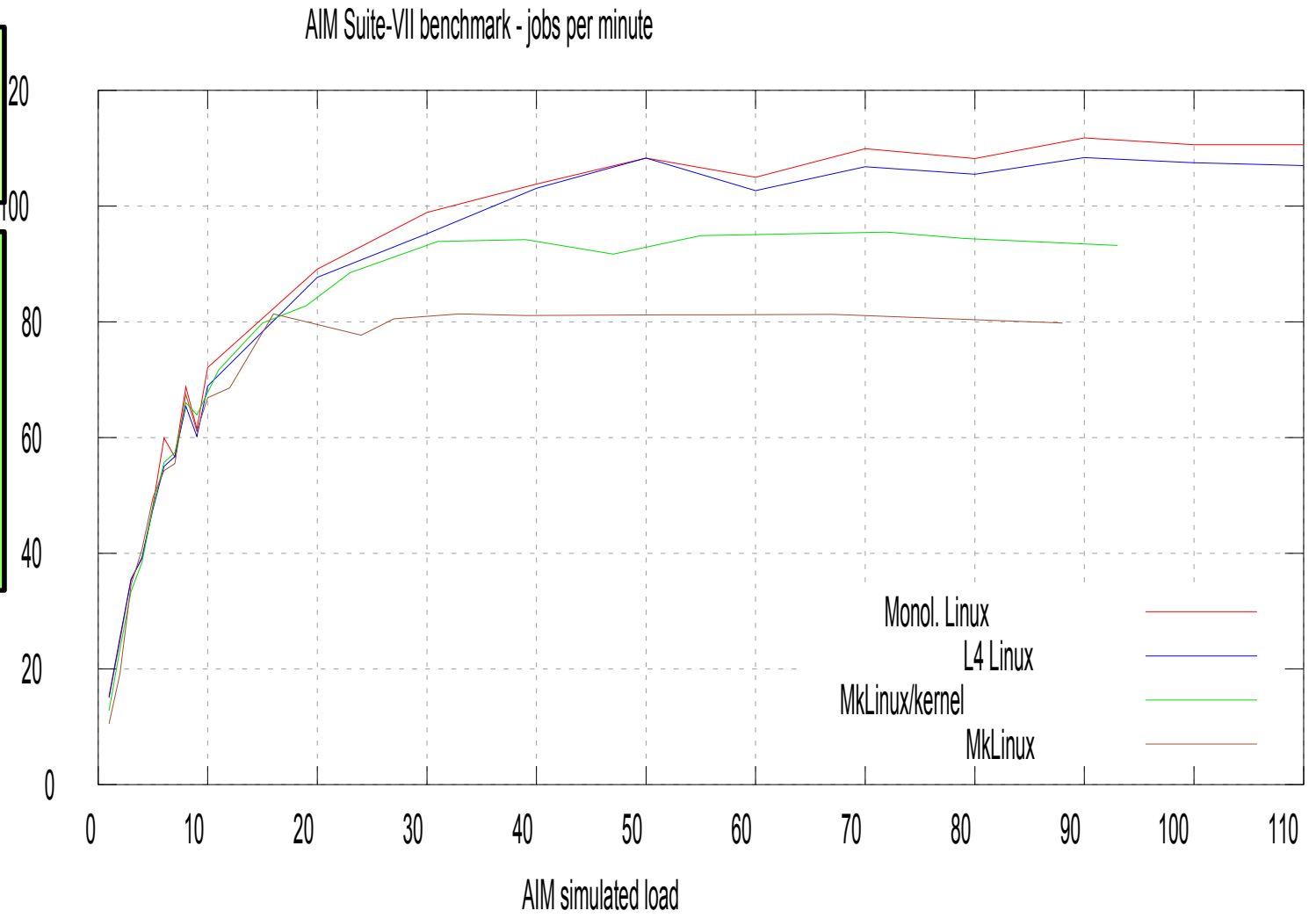


L⁴Linux: the first L4 application (TU Dresden)

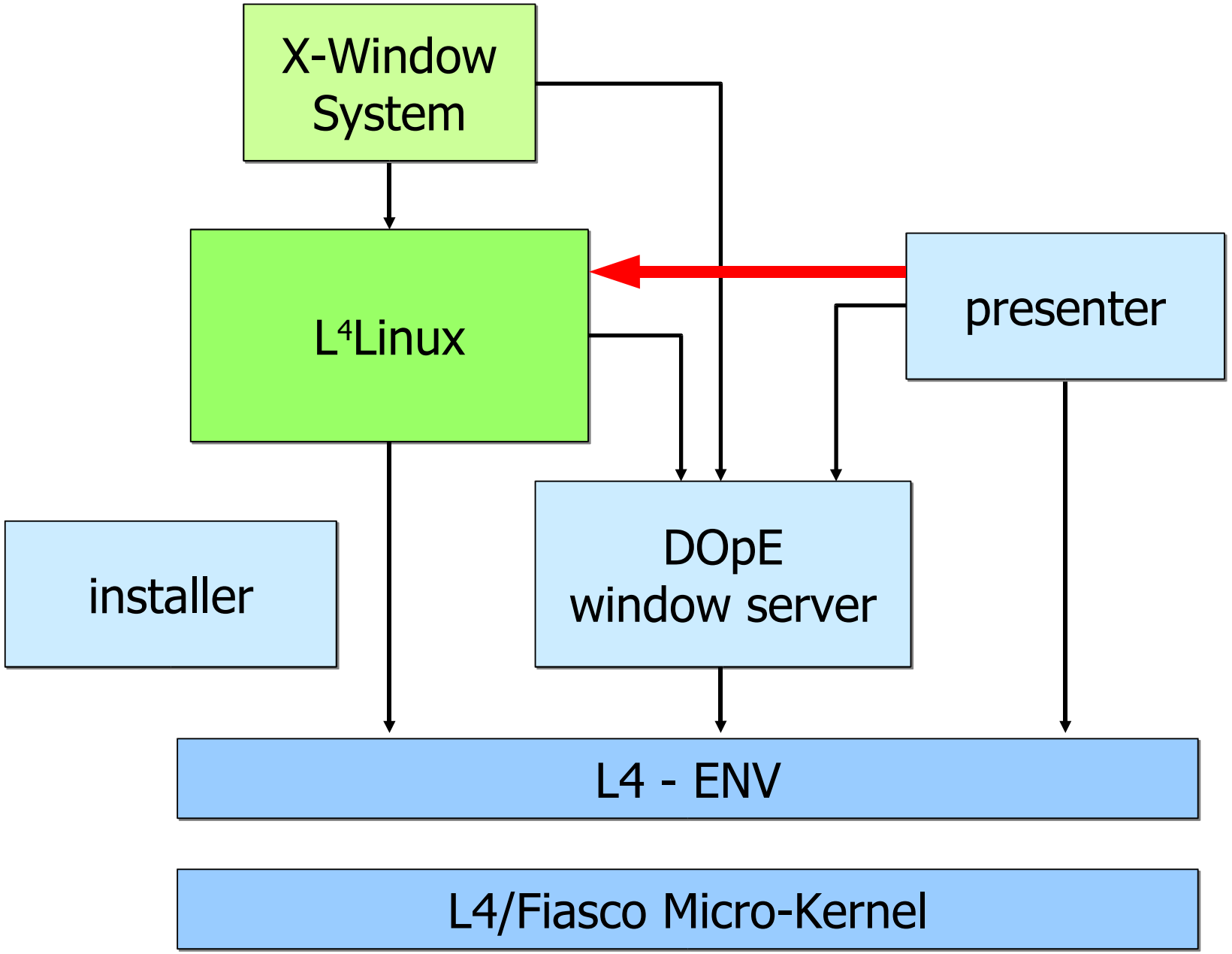
Time-Sharing Applications

L⁴Linux

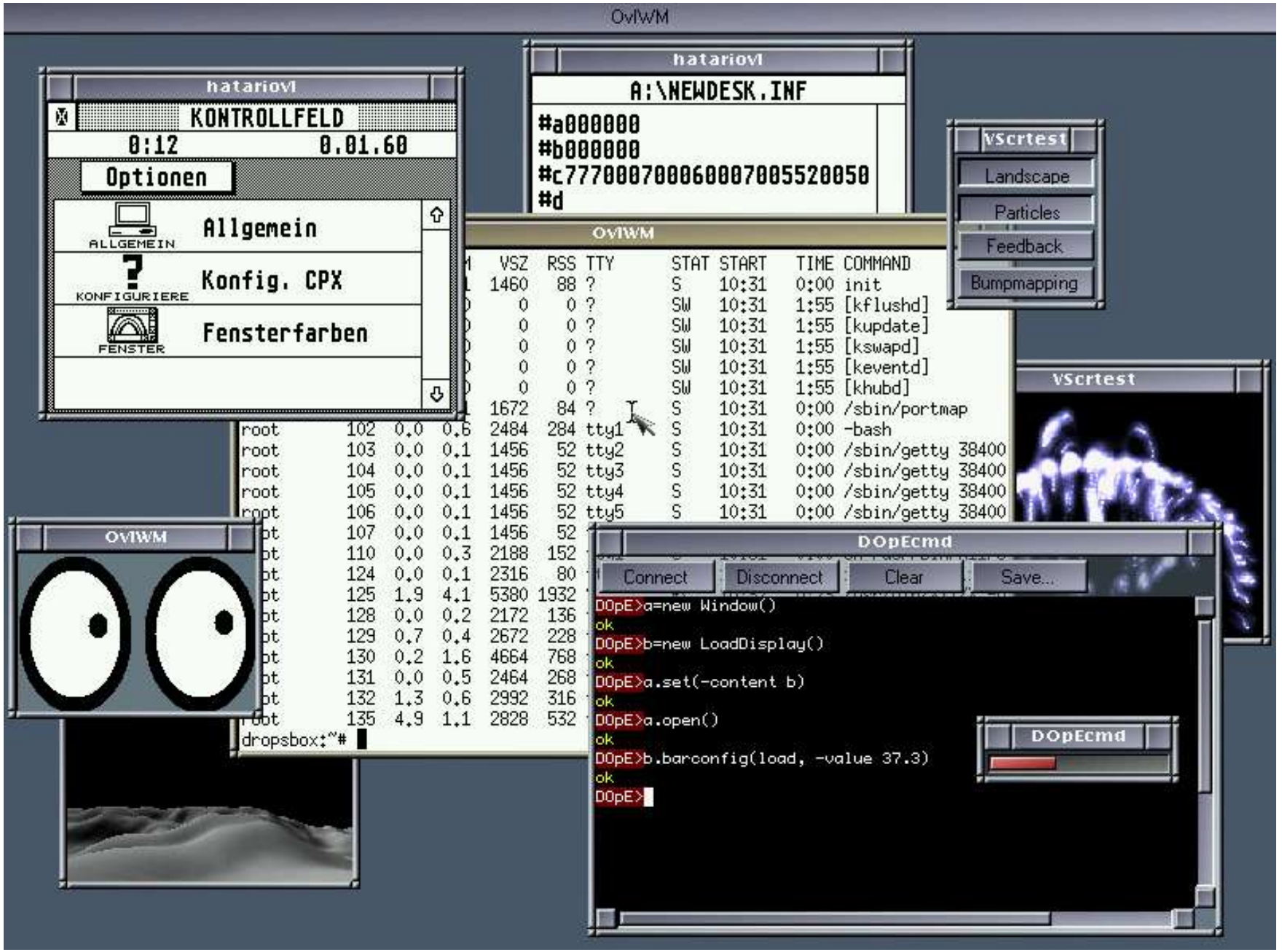
Fiasco



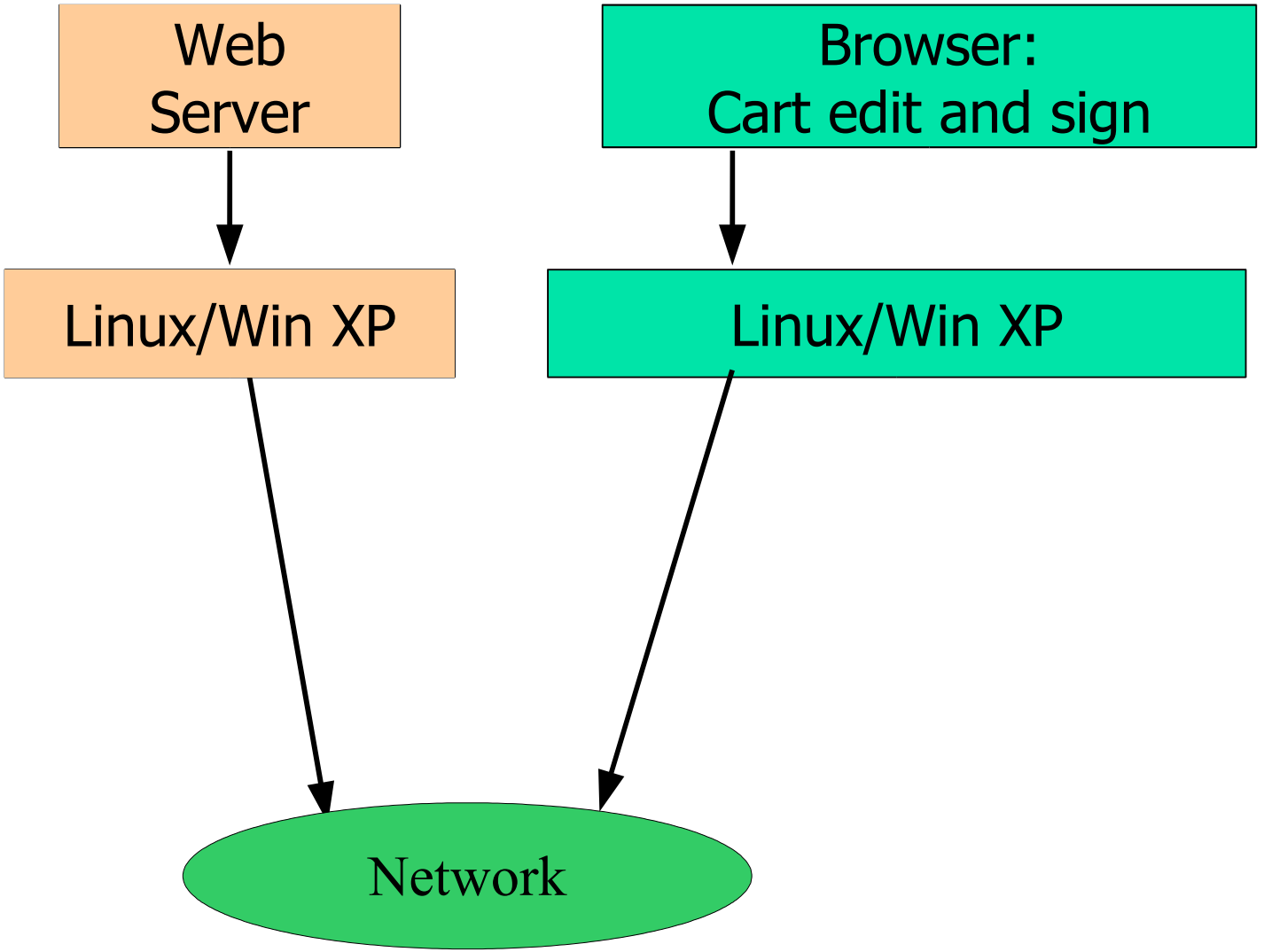
But, where is
the benefit ... ?


X-Window System
L4Linux
presenter
installer
DOpE window server
L4 - ENV
L4/Fiasco Micro-Kernel

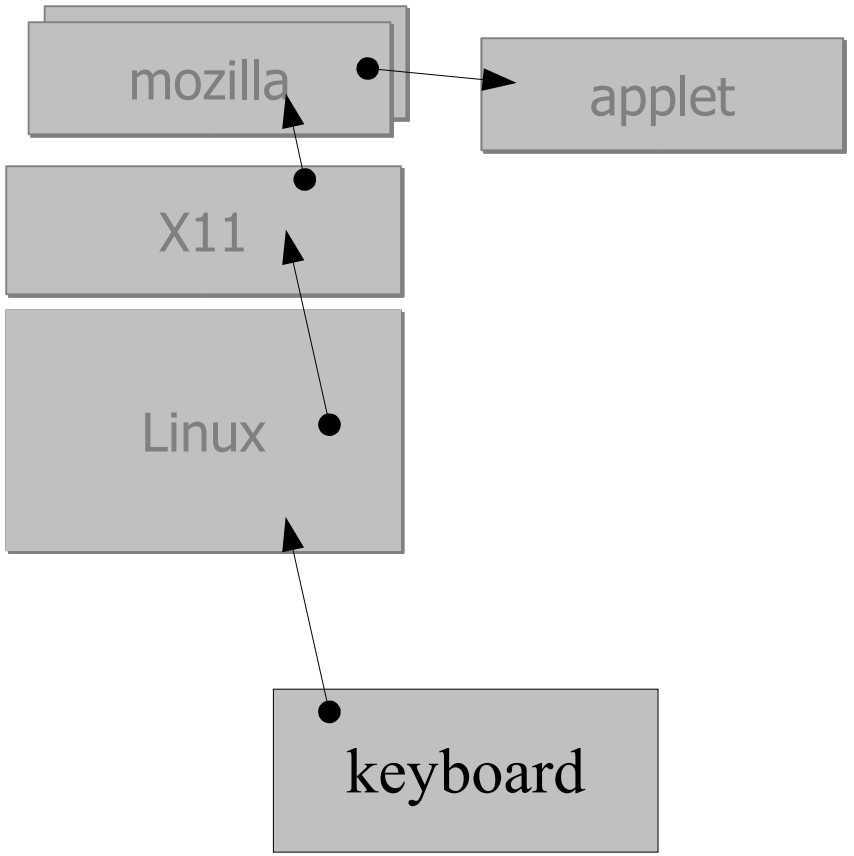
Screen Shot


OvIWM
hatariovl
KONTROLLFELD
0:12
0.01.60
Optionen
ALLGEMEIN
Allgemein
KONFIGURIERE
Konfig. CPX
FENSTER
Fensterfarben
hatariovl
A:\NEWDESK.INF
#a000000
#b000000
#c777000700060007005520050
#d
VScrtest
Landscape
Particles
Feedback
Bumpmapping
OvIWM
VSZ RSS TTY STAT START TIME COMMAND
1460 88 ? S 10:31 0:00 init
0 0 ? SW 10:31 1:55 [kflushd]
0 0 ? SW 10:31 1:55 [kupdate]
0 0 ? SW 10:31 1:55 [kswapd]
0 0 ? SW 10:31 1:55 [keventd]
0 0 ? SW 10:31 1:55 [khubd]
1672 84 ? S 10:31 0:00 /sbin/portmap
root 102 0.0 0.6 2484 284 tty1 S 10:31 0:00 -bash
root 103 0.0 0.1 1456 52 tty2 S 10:31 0:00 /sbin/getty 38400
root 104 0.0 0.1 1456 52 tty3 S 10:31 0:00 /sbin/getty 38400
root 105 0.0 0.1 1456 52 tty4 S 10:31 0:00 /sbin/getty 38400
root 106 0.0 0.1 1456 52 tty5 S 10:31 0:00 /sbin/getty 38400
dropsbox:~#
VScrtest
OvIWM
DOpEcmd
Connect
Disconnect
Clear
Save...
DOpE>a=new Window()
ok
DOpE>b=new LoadDisplay()
ok
DOpE>a.set(-content b)
ok
DOpE>a.open()
ok
DOpE>b.barconfig(load, -value 37.3)
ok
DOpE>
DOpEcmd

Internet Transaction



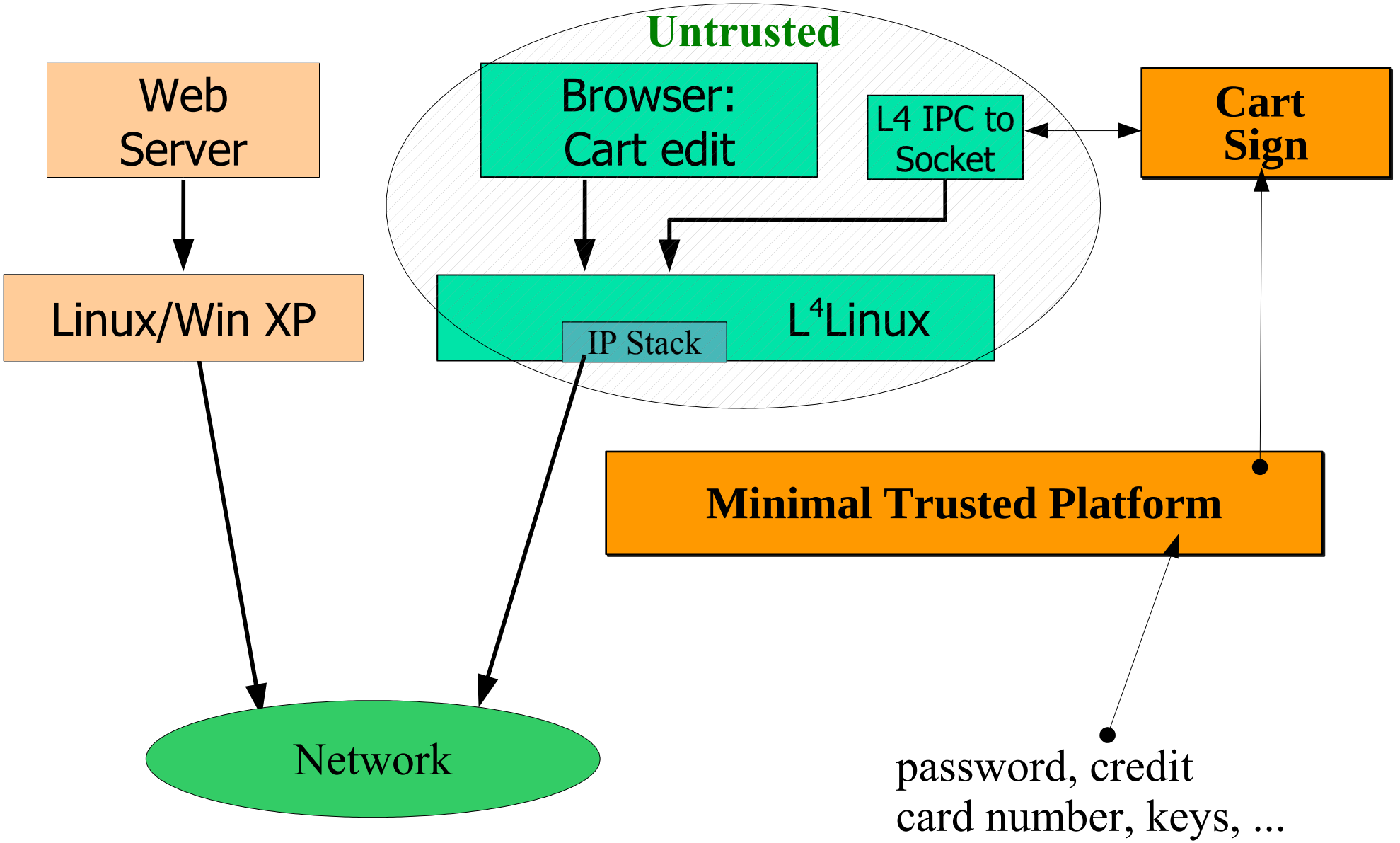
Your password(s), credit card number, ...



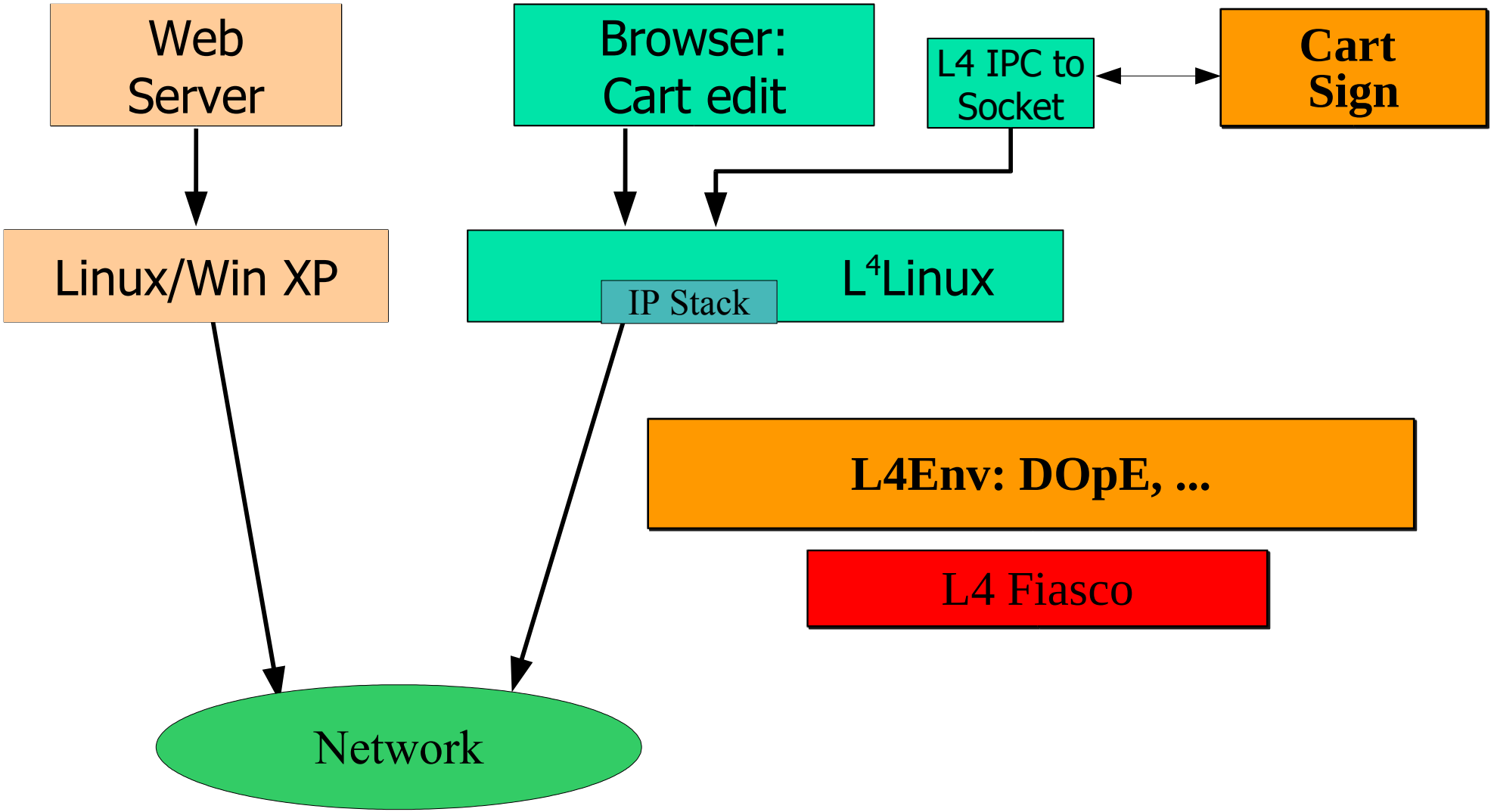
see:

Understanding Data Lifetime
via Whole System Simulation
Jim Chow, Ben Pfaff, Tal Garfinkel, Kevin Christopher, and Mendel Rosenblum, Stanford University
Usenix Security 04

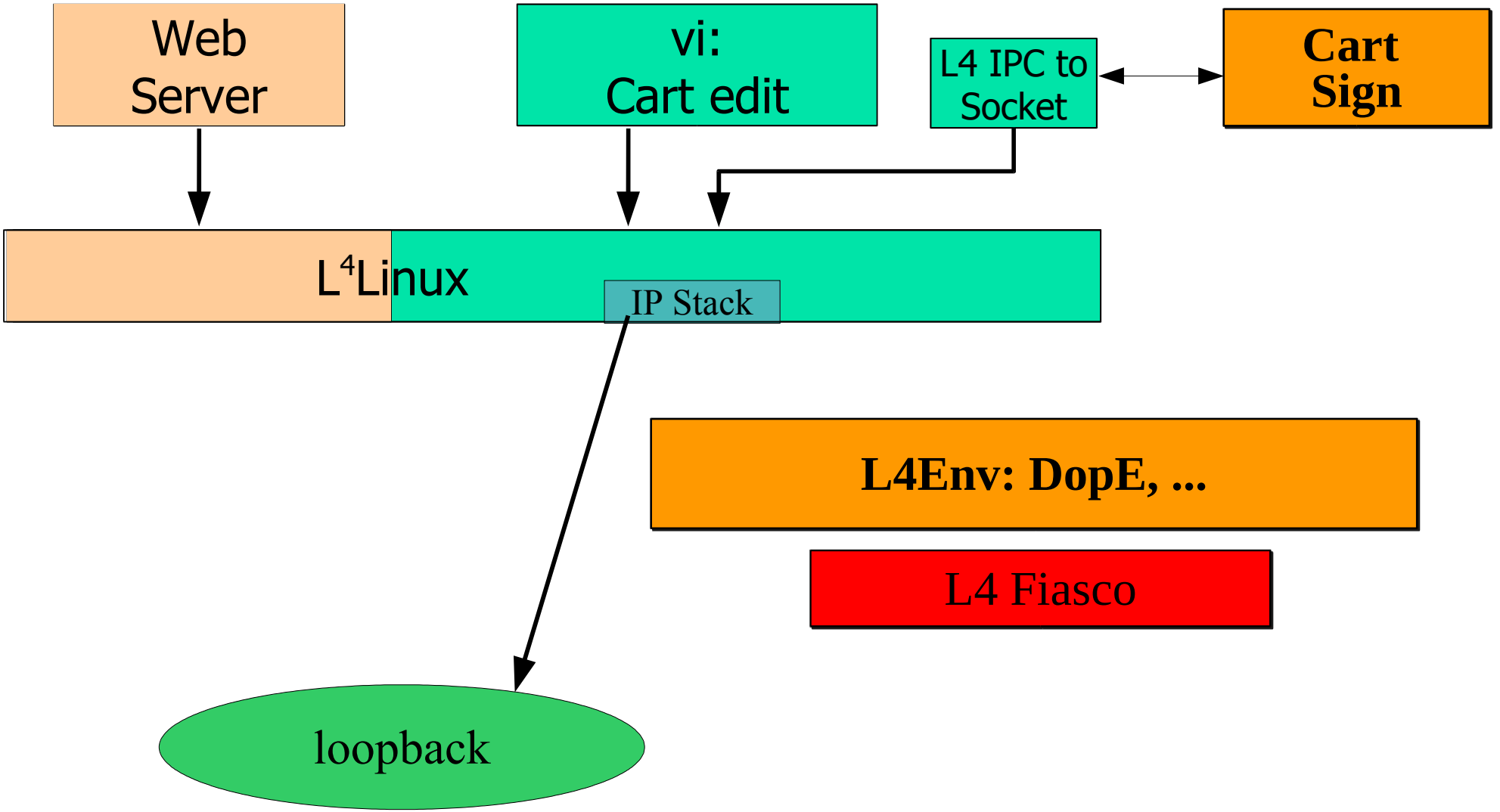
Split Transaction



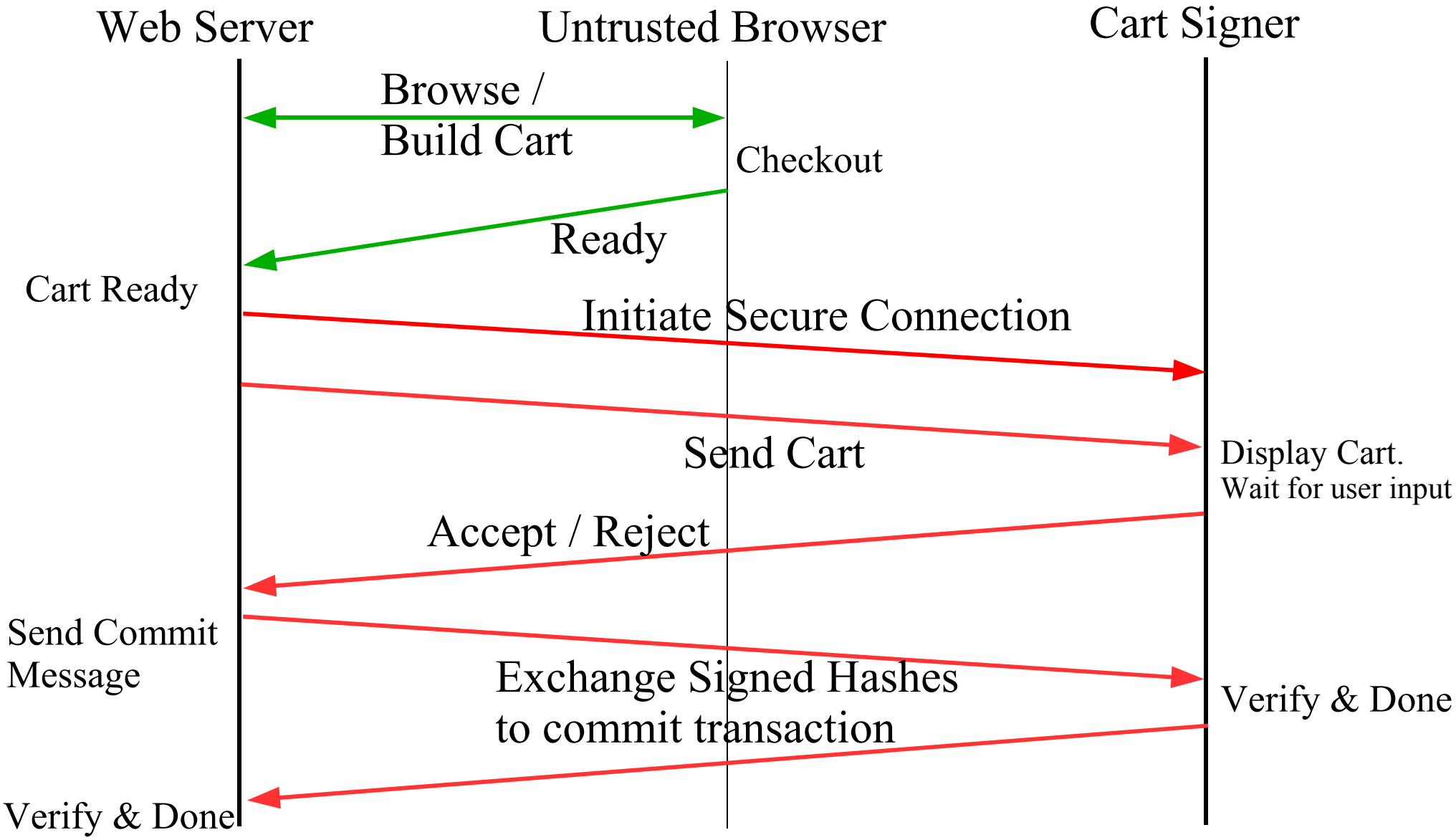
Split Transaction Demo



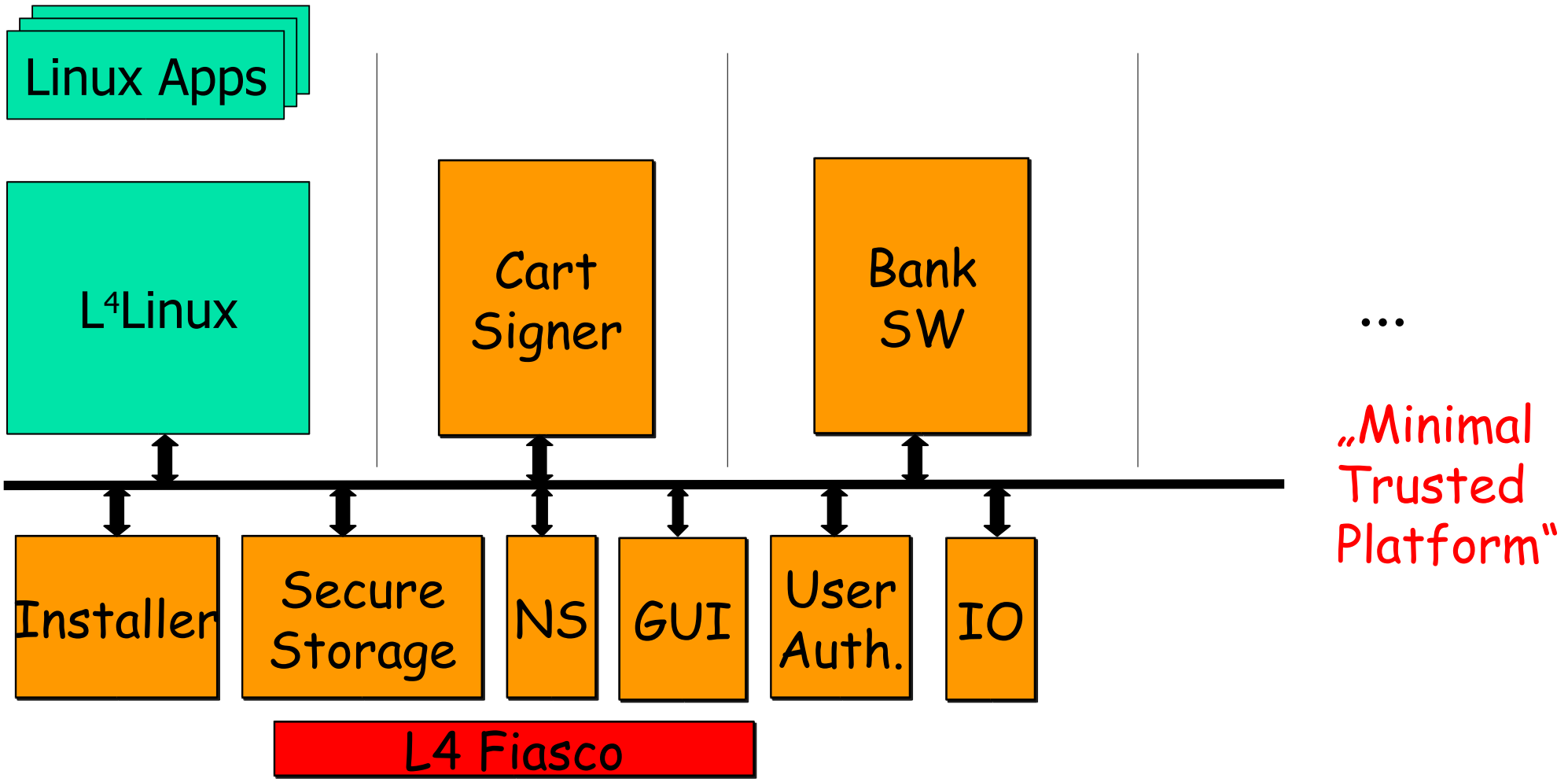
Split Transaction Demo



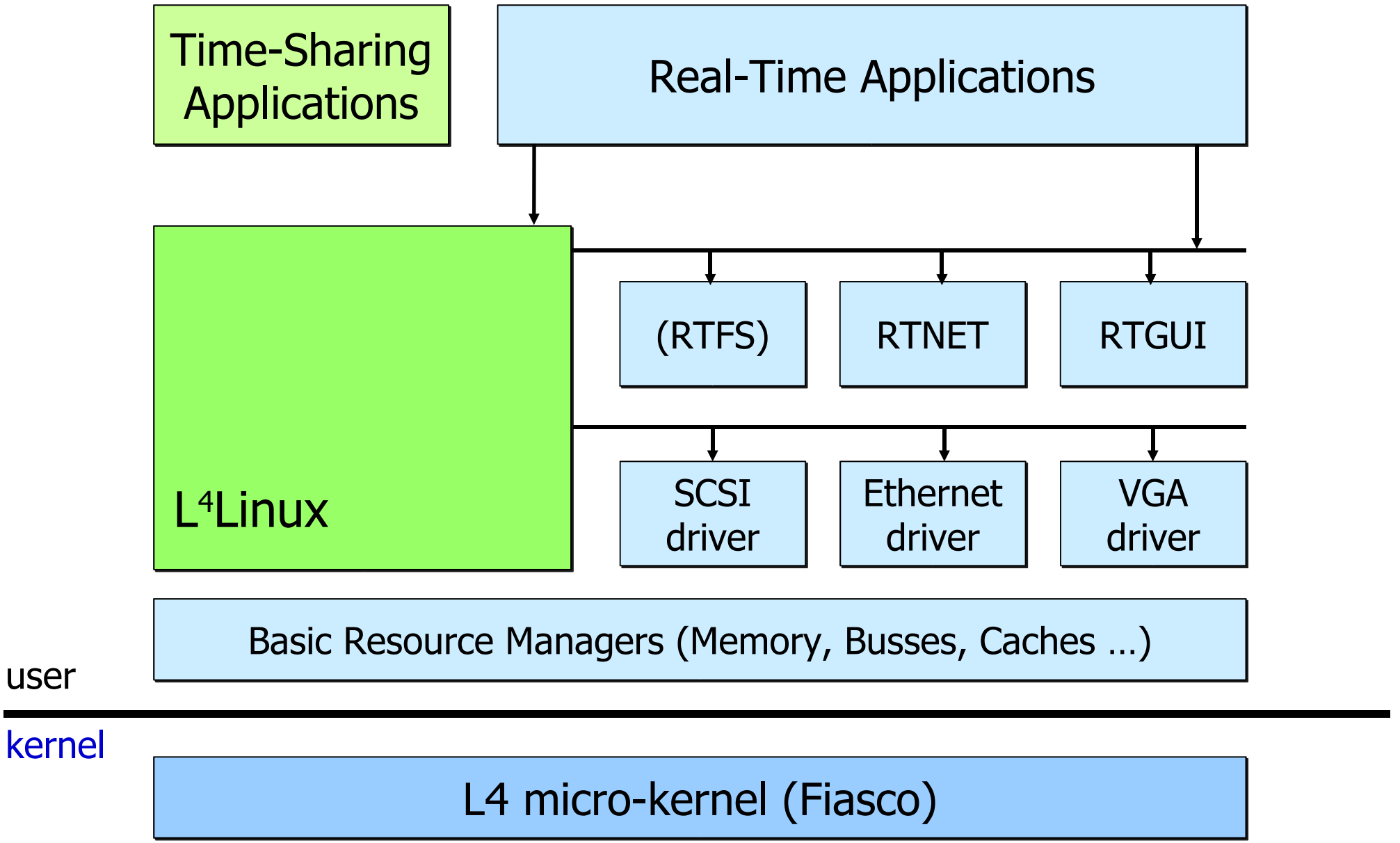
Message Sequence



NIZZA


Linux Apps
L4Linux
Cart Signer
Bank SW
...
„Minimal Trusted Platform“
Installer
Secure Storage
NS
GUI
User Auth.
IO
L4 Fiasco

DResden Real-Time OS



Linux and Microkernels, the future ?

An open source alternative to Microsoft NGSCB ?

Danke!
Fragen ???

Various

- Split Transactions have been designed and implemented by Lenin Singaravelu of Georgia Tech during his internship at TU Dresden paper forthcoming
- citations from various sources on the internet
-