Unikernels, Multikernels, Virtual Machine-based Kernels

http://d3s.mff.cuni.cz/aosy

CHARLES UNIVERSITY
Faculty of Mathematics and Physics

Department of Distributed and Dependable Systems

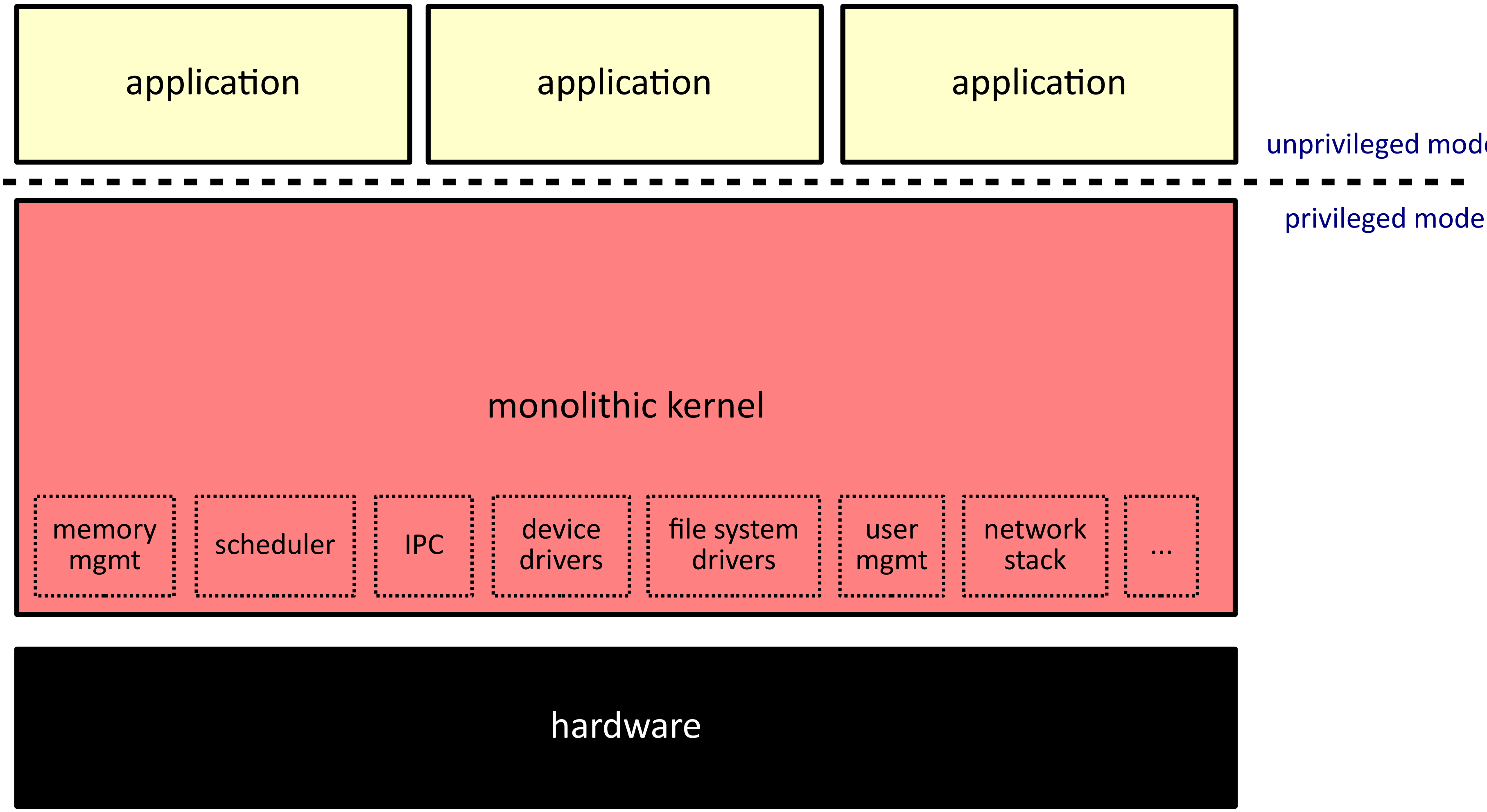
Martin Děcký

decky@d3s.mff.cuni.cz

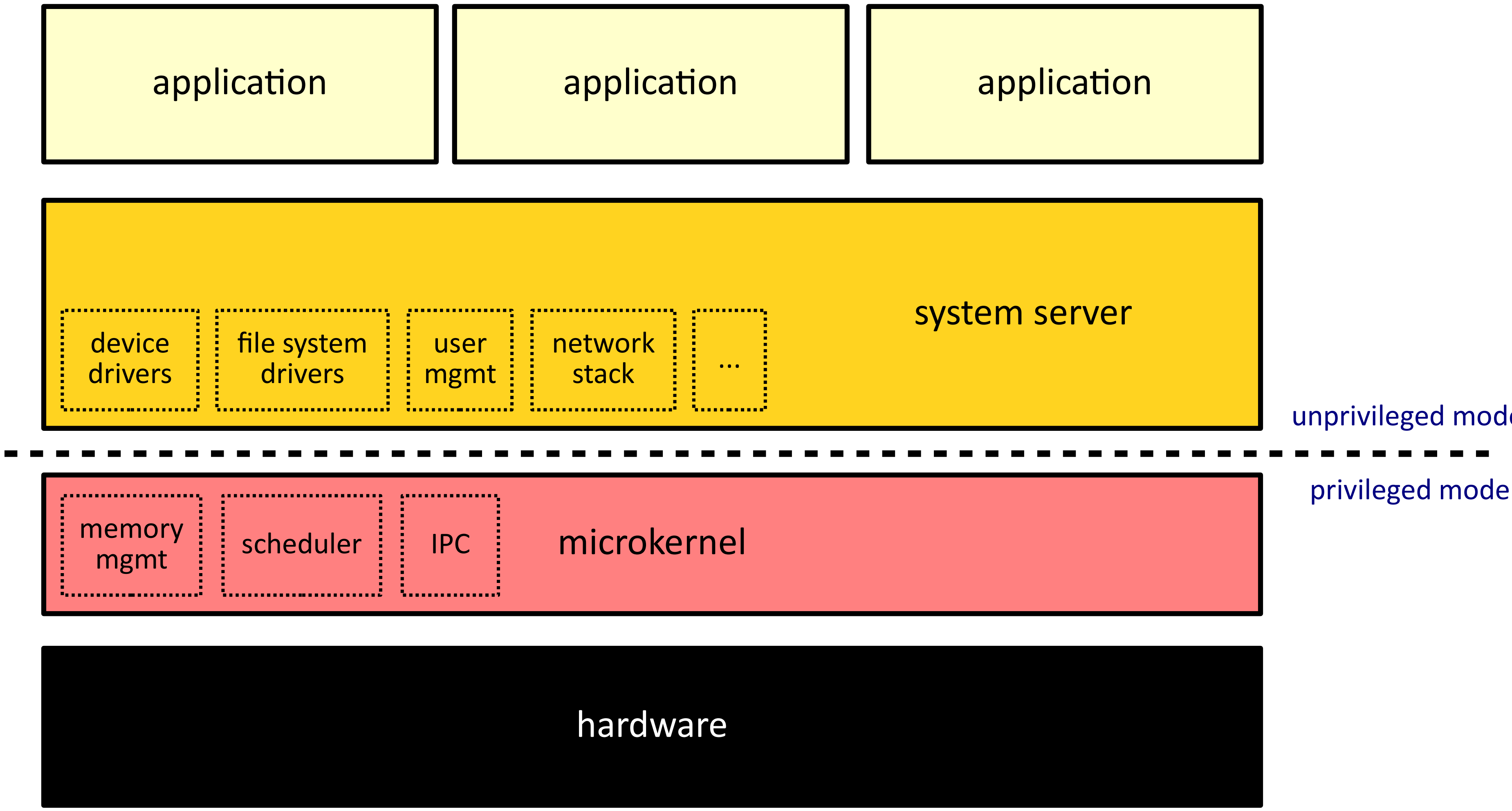
Recall: Common OS Taxonomy

- **Special-purpose operating systems**
  - Real-time operating systems
  - Hypervisors (type 1)
  - ...
- **General-purpose operating systems**
  - Monolithic kernel
  - Single-server microkernel
  - Multiserver microkernel
  - Hybrid kernel (?)

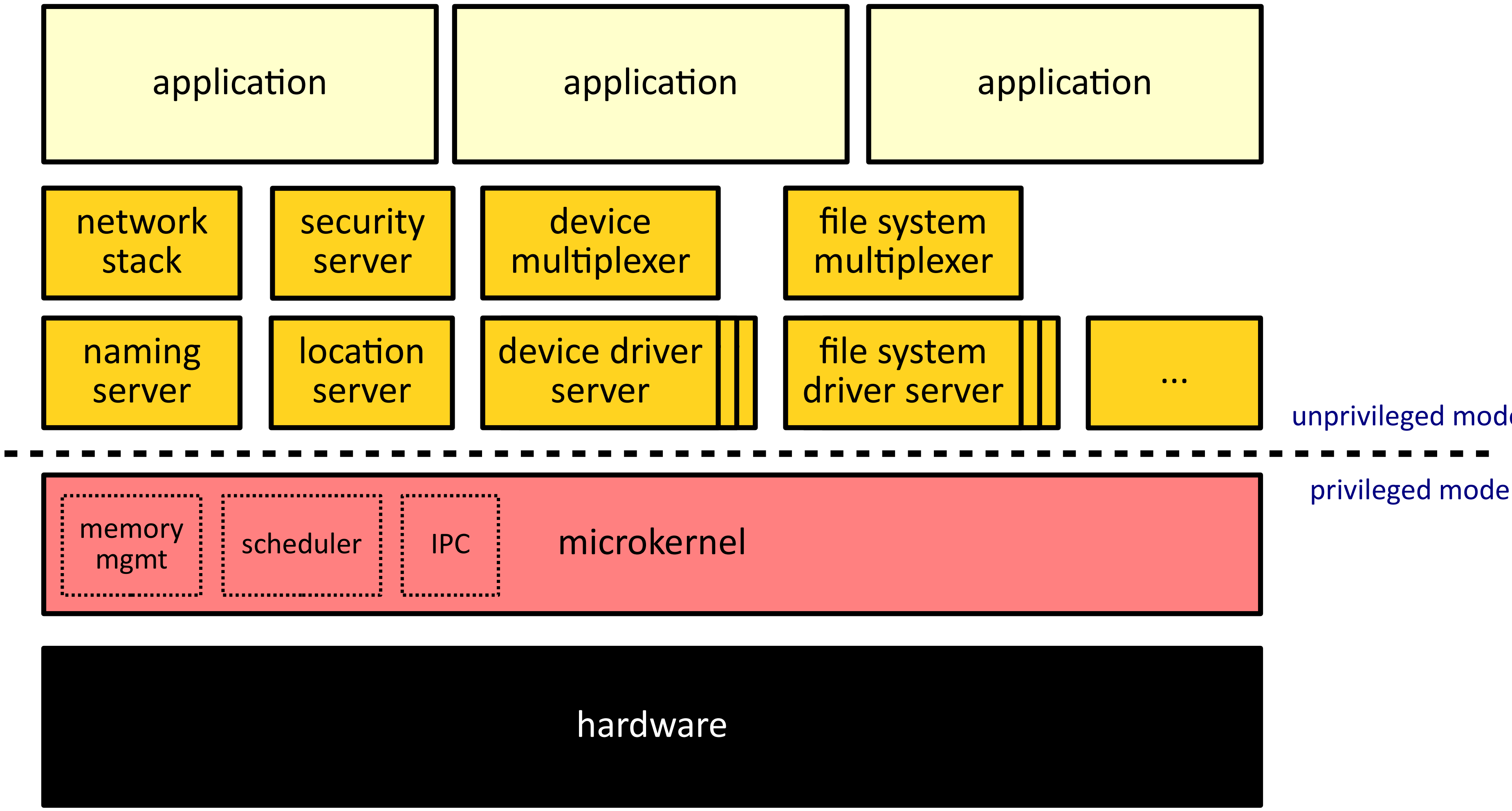
Monolithic Kernel



Single-server Microkernel



Multiserver Microkernel



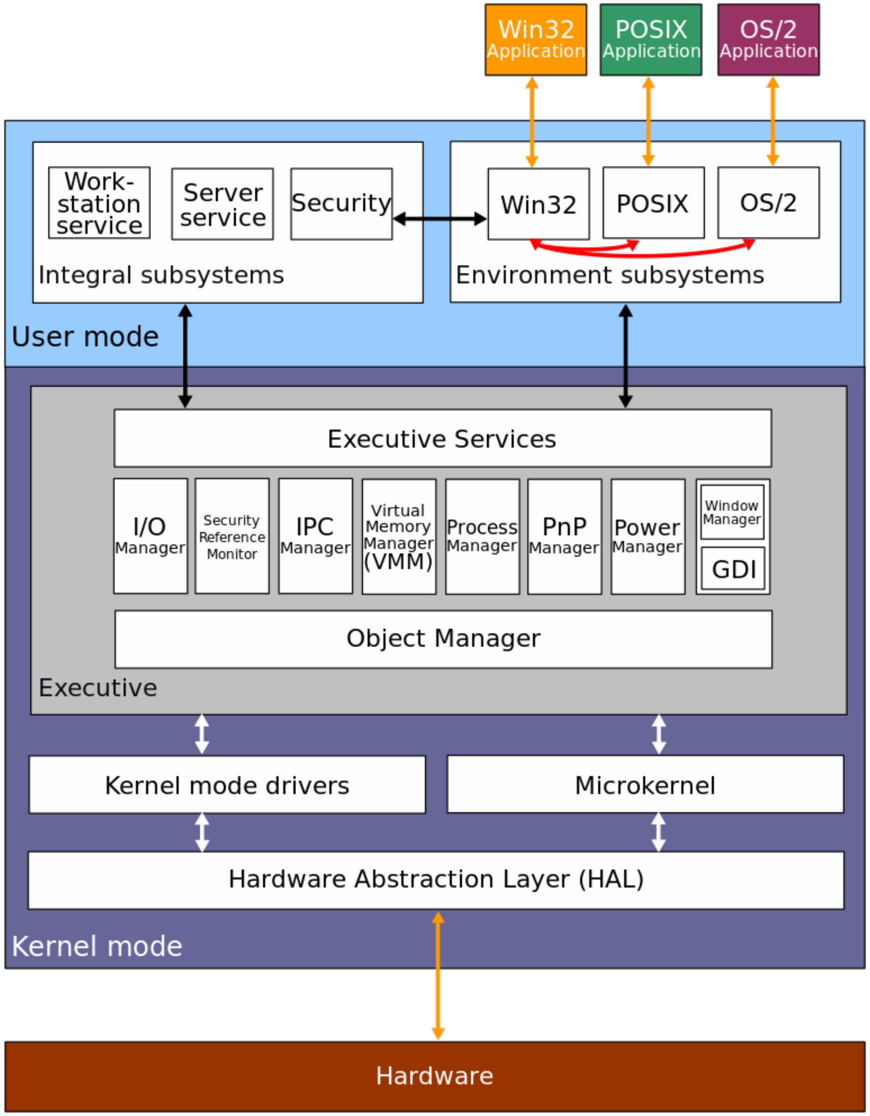
Examples

- **Monolithic kernel**
  - Linux, Solaris (UTS), Windows, FreeBSD, NetBSD, OpenBSD, OpenVMS, MS-DOS, RISC OS
- **Single-server microkernel**
  - CMU Mach, AmigaOS, NeXTSTEP
- **Multiserver microkernel**
  - HelenOS, MINIX 3, Genode, GNU/Hurd, Redox, RefOS (seL4)

Hybrid Kernel

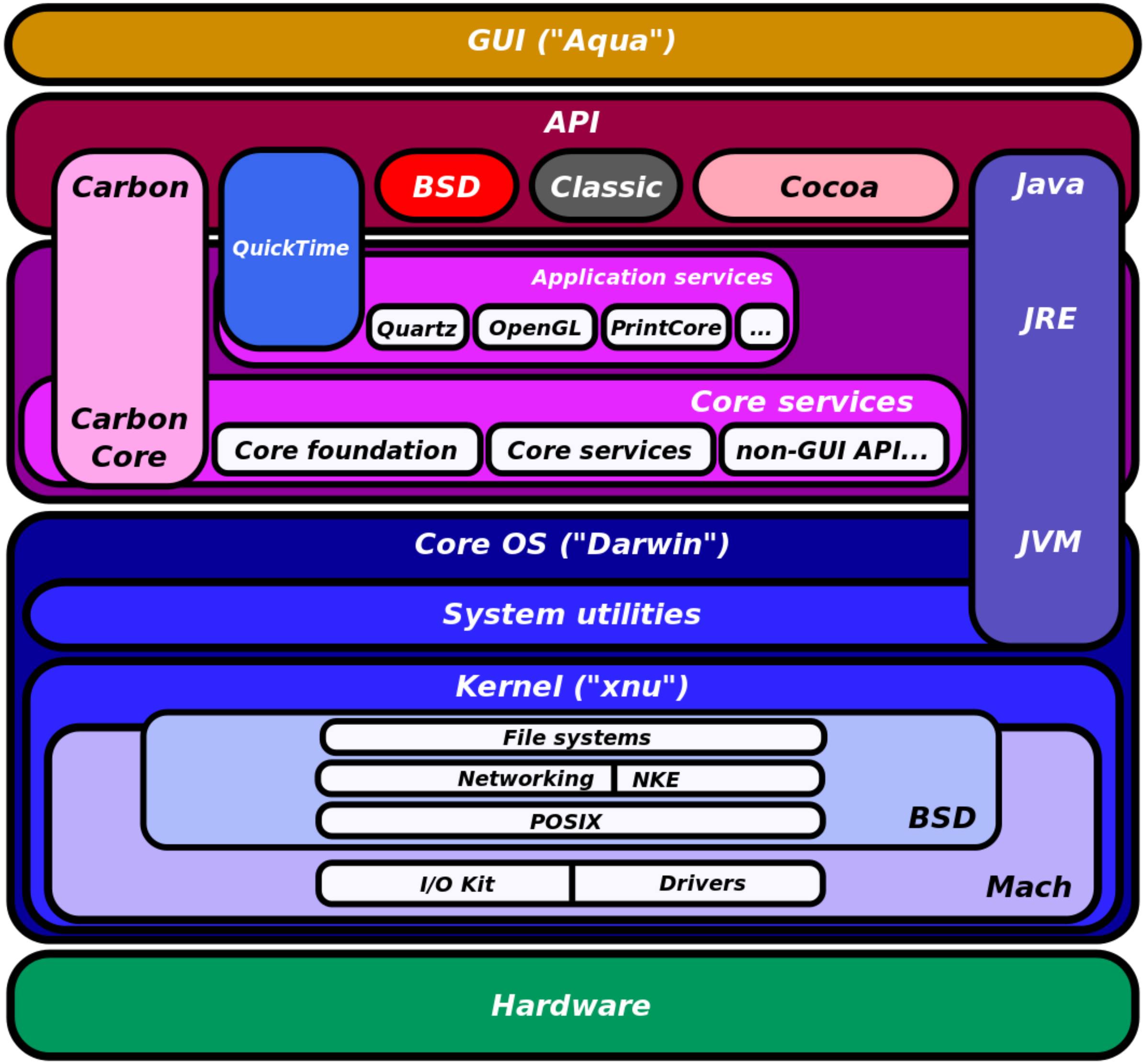
- **No universally accepted definition**
  - "Architecture that somewhat resembles a microkernel, but is not a pure microkernel."
  - Windows NT
    - Internal architecture and communication abstractions inspired by CMU Mach (message passing, ports)
    - Originally user space device drivers
      - Later moved into the kernel, more recently some driver classes in user space again
  - macOS (iOS, etc.)
    - The core xnu component actually contains the CMU Mach 3.0 microkernel
    - Native and ported (BSD) system functionality in user space
      - Gradual erosion towards kernel drivers
  - BeOS (Haiku), Syllable, DragonFly BSD, Plan 9, NetWare, eComStation

Windows NT


Win32
Application
POSIX
Application
OS/2
Application
Work-
station
service
Server
service
Security
Win32
POSIX
OS/2
Integral subsystems
Environment subsystems
User mode
Executive Services
I/O
Manager
Security
Reference
Monitor
IPC
Manager
Virtual
Memory
Manager
(VMM)
Process
Manager
PnP
Manager
Power
Manager
Window
Manager
GDI
Object Manager
Executive
Kernel mode drivers
Microkernel
Hardware Abstraction Layer (HAL)
Kernel mode
Hardware

[1]

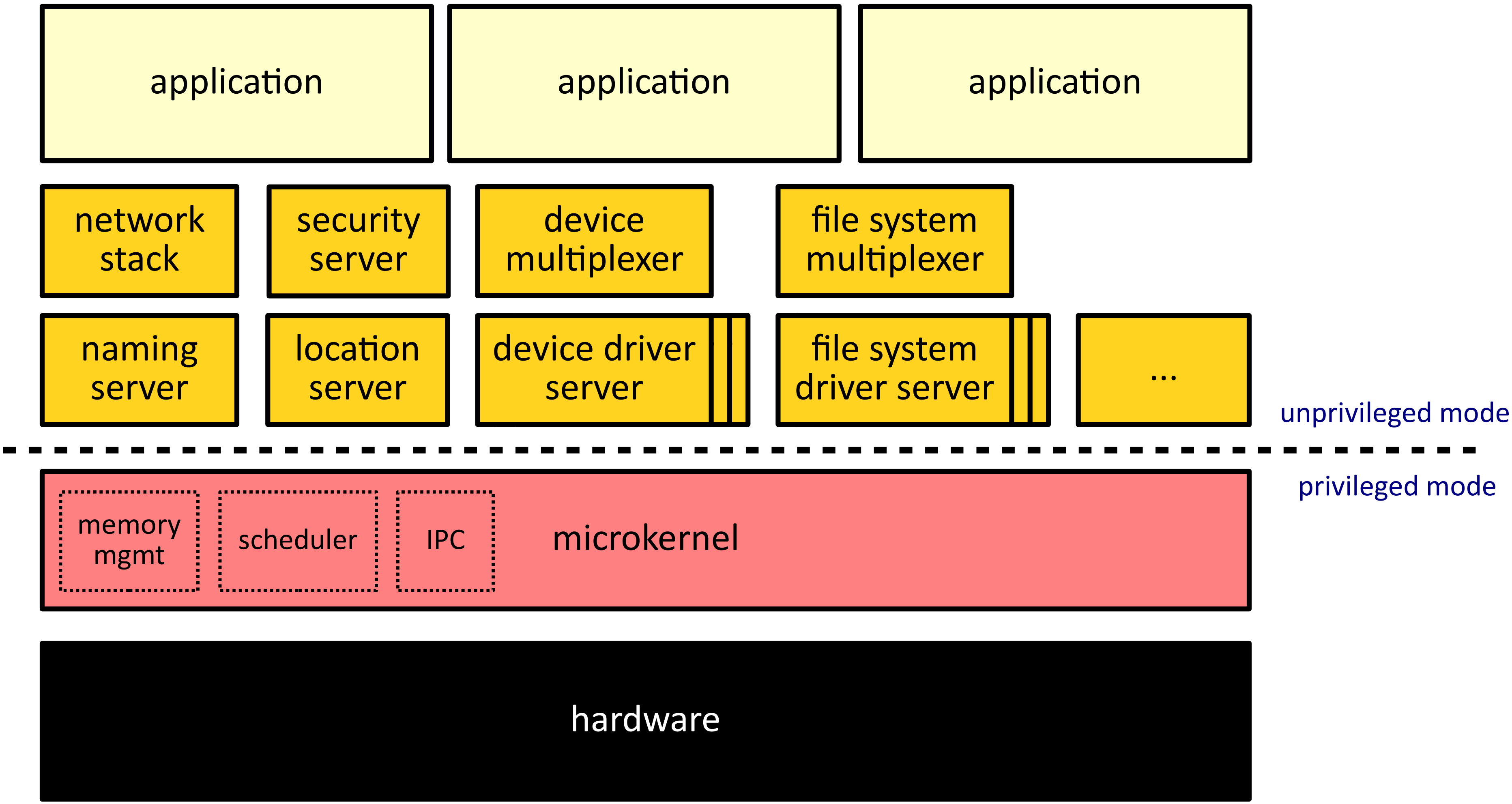
macOS



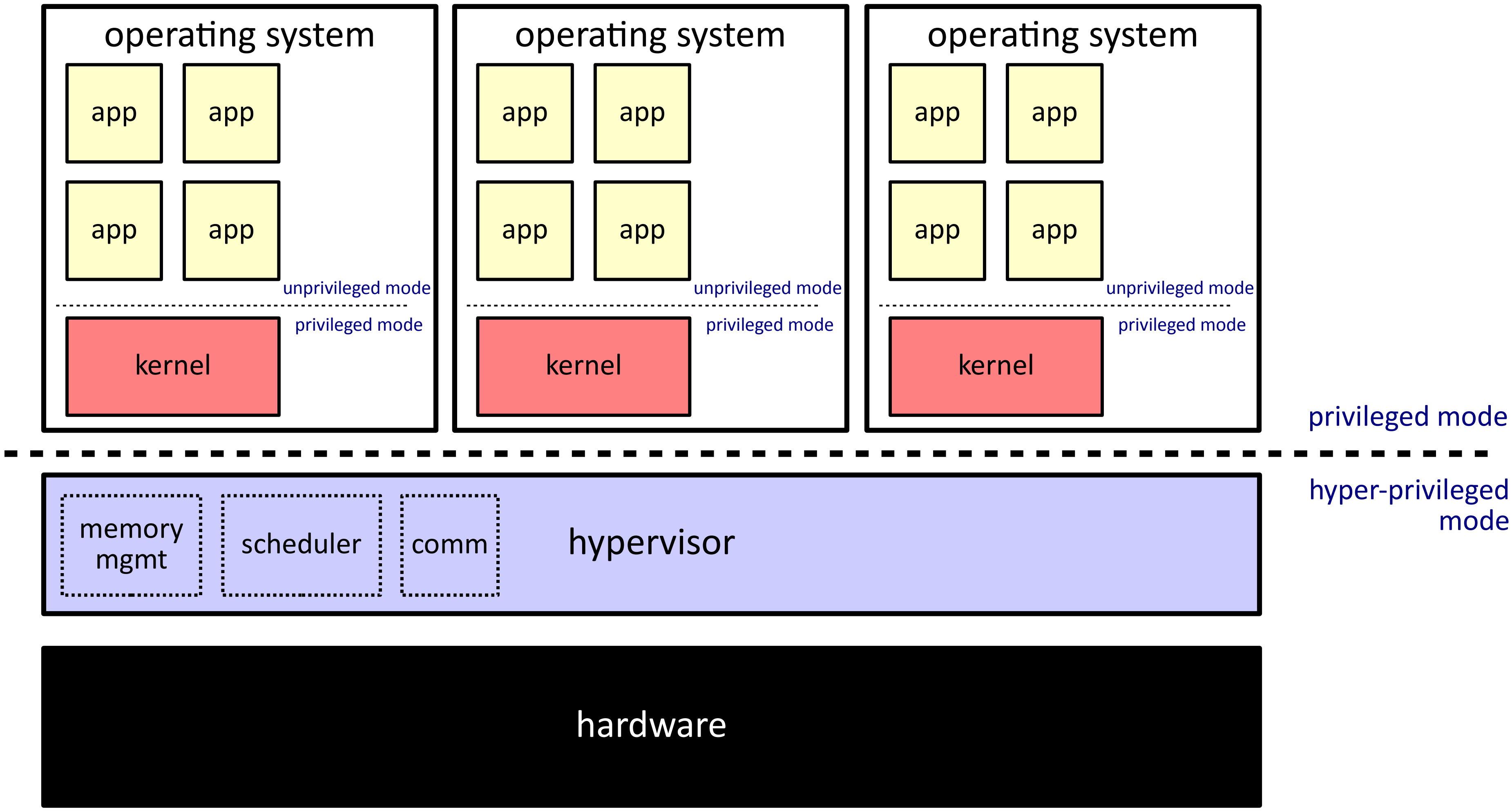
GUI ("Aqua")
API
Carbon
QuickTime
BSD
Classic
Cocoa
Java
Application services
Quartz
OpenGL
PrintCore
...
JRE
Carbon
Core
Core services
Core foundation
Core services
non-GUI API...
Core OS ("Darwin")
JVM
System utilities
Kernel ("xnu")
File systems
Networking
NKE
POSIX
BSD
I/O Kit
Drivers
Mach
Hardware

[2]

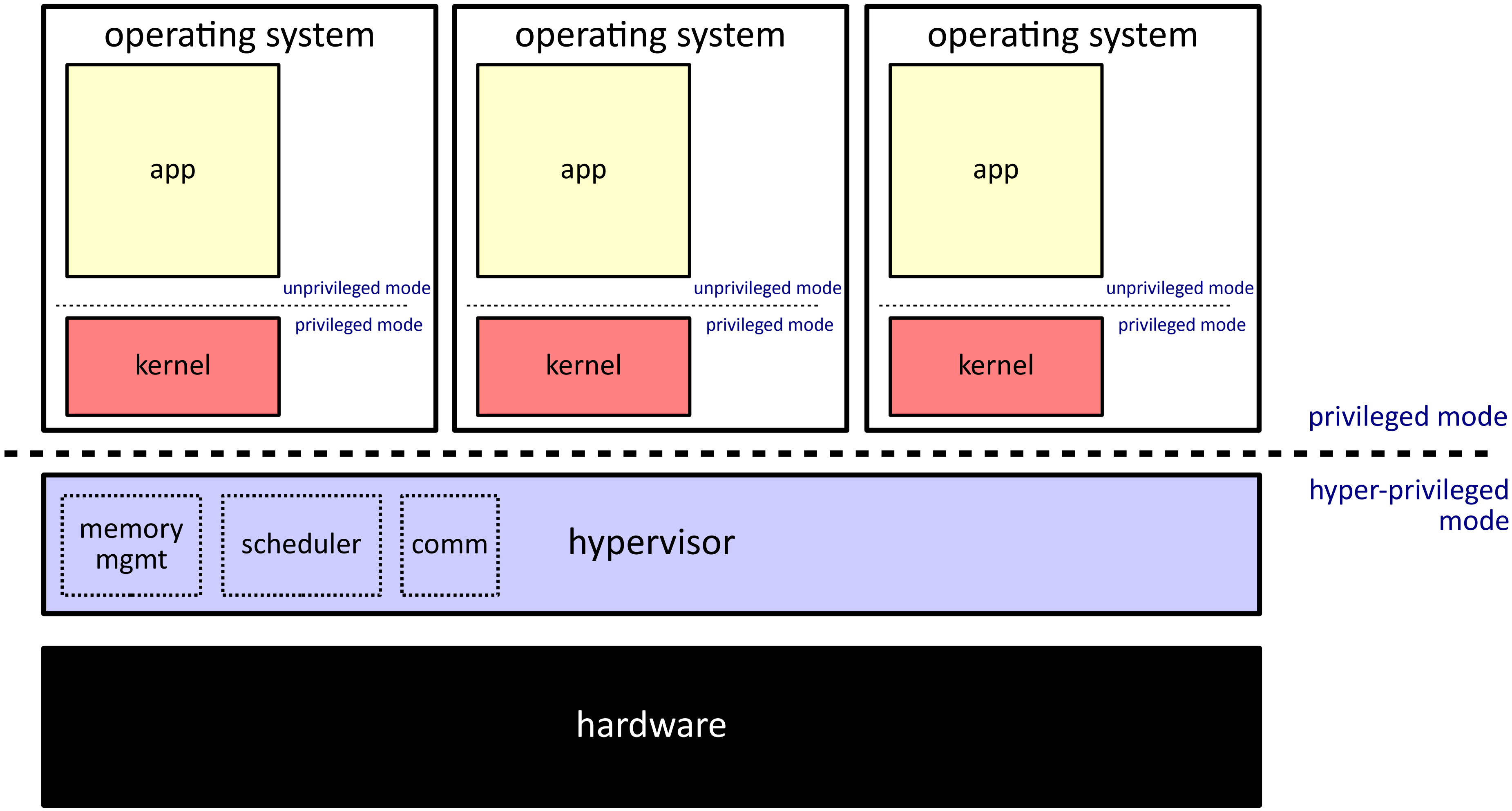
Multiserver Microkernel (reprise)



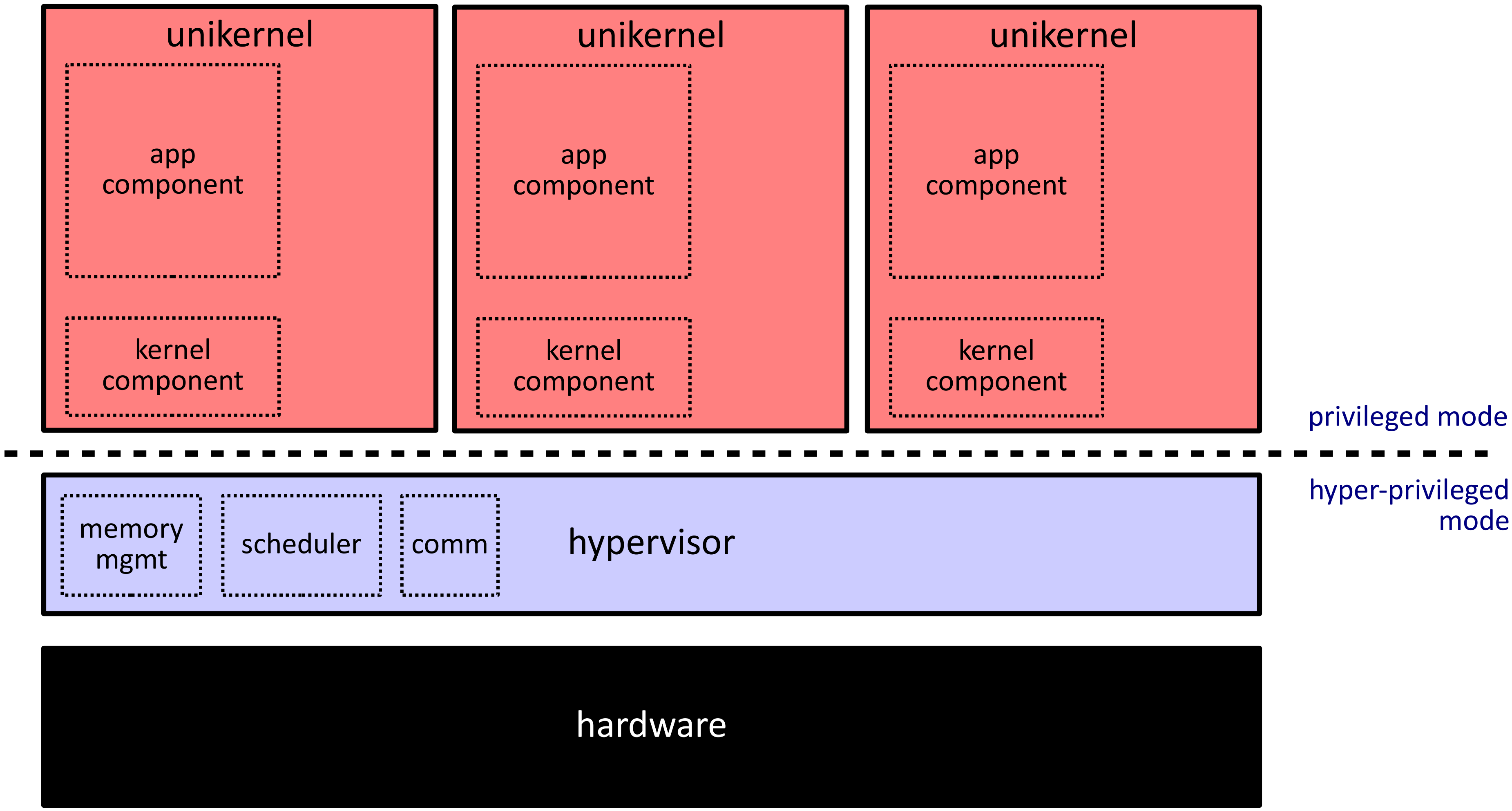
Hypervisor (Type 1)



Common Cloud Deployment



Unikernel


unikernel
app component
kernel component
unikernel
app component
kernel component
unikernel
app component
kernel component
privileged mode
hyper-privileged mode
memory mgmt
scheduler
comm
hypervisor
hardware

Unikernel (2)

- **Library operating system**
  - Payload (application) merged with the kernel
    - Kernel component acts as a library providing access to the hardware, threading, file systems, etc.
      - Only necessary functionality
    - Mostly static (single image), but there are dynamic variants
    - Code runs in privileged mode and single address space
      - No mode switches, address space switches
      - Syscalls can be replaced by function calls
      - Isolation/security provided by the underlying hypervisor

Department of Distributed and Dependable Systems

Unikernel (3)

- **Exokernel**
  - MIT since 1994
  - Goal: End-to-end principle
    - Limiting the number of abstractions (compared to monolithic kernels)
    - Limiting the communication complexity (compared to microkernels)
  - Co-existence with a regular kernel
  - ExOS (MIT)
  - Nemesis (University of Cambridge, University of Glasgow, Swedish Institute of Computer Science, Citrix)
    - Multimedia applications

Department of Distributed and Dependable Systems

Unikernel (4)

- **Rumprun**
  - POSIX compliant, BSD-compatible run-time environment
    - Original concept: NetBSD anykernel
      - Possibility to compile NetBSD drivers and subsystems either as traditional kernel components or as standalone user space libraries (rump kernels)
      - Rump kernels communicate with the host kernel using syscall interface
  - Replacing the syscall interface of rump kernels with a hypercall interface to the hypervisor
  - “Bare metal” execution also possible
- **Drawbridge**
  - Win32-compatible run-time environment
    - Originally a Win32 environment running in a Windows picoprocess

Unikernel (5)

```
git clone https://github.com/rumpkernel/rumprun.git
cd rumprun

git submodule init
git submodule update
```

(depending on the local C compiler, apply a small patch from https://github.com/rumpkernel/rumprun/issues/86)

```
./build-rr.sh hw                    ./build-rr.sh xen

export PATH=”$PATH:`pwd`/rumprun/bin”

x86_64-rumprun-netbsd-gcc -o module module.c
rumprun-bake hw_virtio unikernel.bin module

rumprun kvm -i unikernel.bin
```

Department of Distributed and Dependable Systems

Unikernel (6)

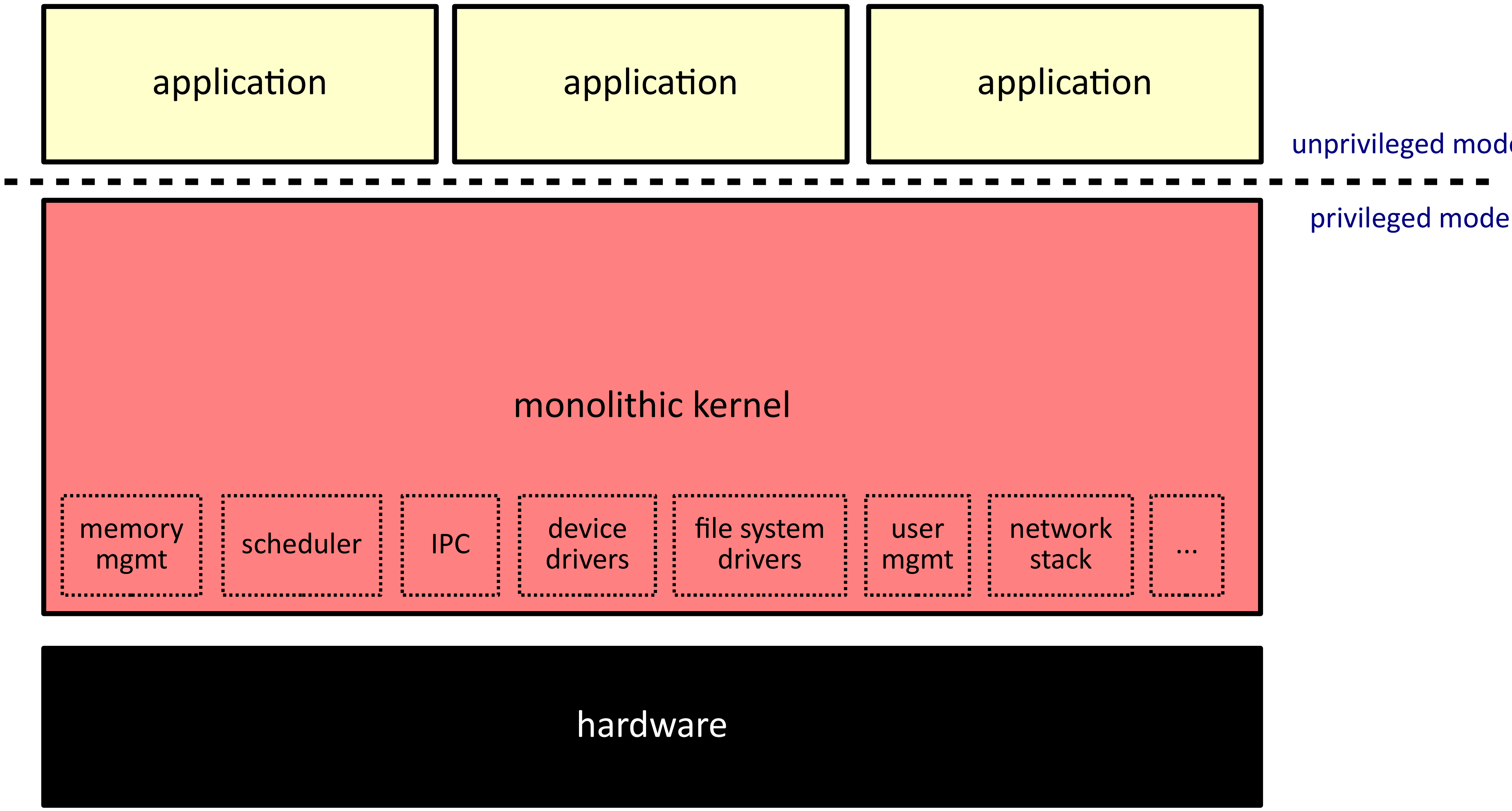
- **OSv**
  - Linux-compatible application environment
    - Not just a run-time, but a complete OS compatibility
    - Goal: Running hosted applications in unmodified run-time environments for Linux
      - E.g. Java EE application running in Tomcat for Linux
  - No notion of users, single address space, processes emulated using threads
    - Similar deployment as in Linux (shell scripting, etc.)

Department of Distributed and Dependable Systems

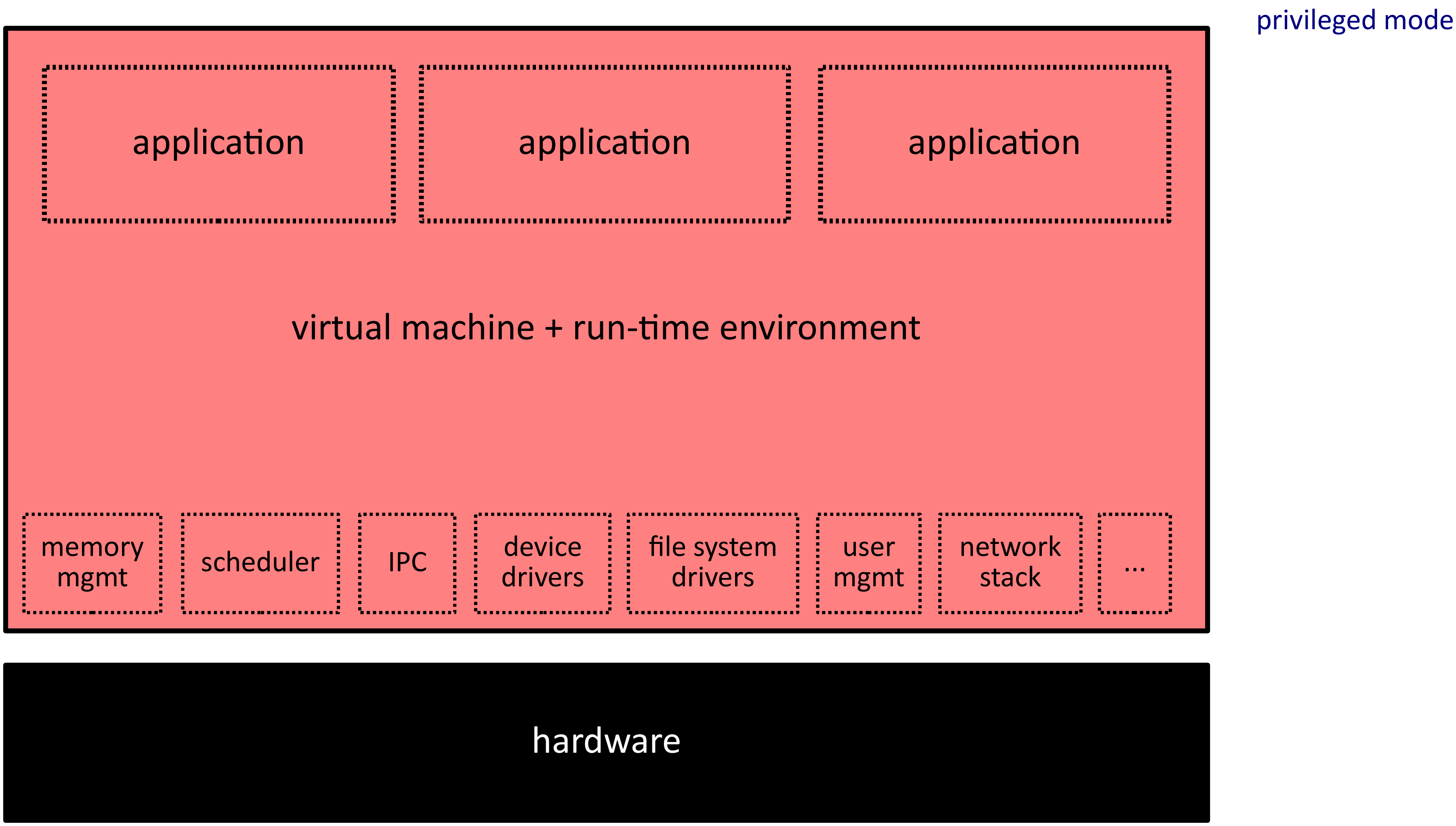
Unikernel (7)

- **Managed language run-time in kernel**
  - Clive (Go)
  - ClickOS, IncludeOS, HermitCore (C++)
  - HaLVM (Haskell)
  - LING (Erlang)
  - MirageOS (Ocaml)
  - Runtime.js (JavaScript)

Monolithic Kernel (reprise)



Virtual Machine-based Kernel



Department of Distributed and Dependable Systems

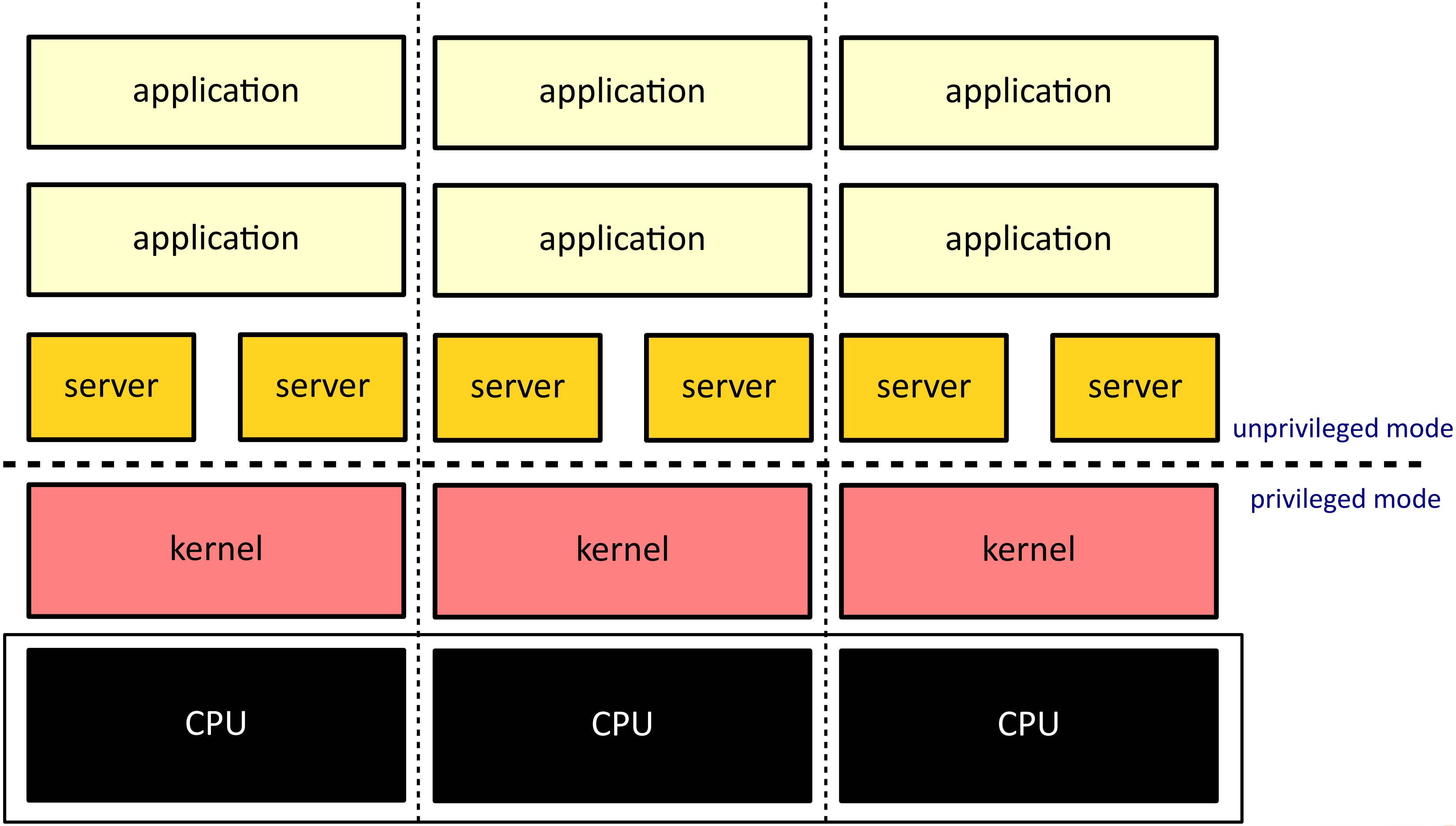
Virtual Machine-based Kernel (2)

- **Inferno**
  - Derived from Plan 9 from Bell Labs
    - “Everything is a file” paradigm
      - All global objects represented as file system paths (global namespace)
      - All operations with objects mapped to common file system operations (walk, stat, create, open, read, write, close, etc.)
      - Local object identification using file descriptors
    - Dis virtual machine and Limbo type-safe language
      - Ada-like syntax, Modula inspired modules, concurrency model based on Communicating Sequential Processes, garbage collector
    - File system operations mapped to Styx communication protocol (compatible with 9P2000)
      - Distributed computing

Virtual Machine-based Kernel (3)

- **Singularity**
  - Microsoft (2003 – 2010)
    - Sing# virtual machine
      - Based on Spec#
        - Superset of C# with Eiffel-like specification of code contracts (object invariants, preconditions, postconditions, non-nullable types)
        - Static checker (based on theorem prover)
        - Run-time checker
      - Extends Spec# with support for communication channels and low-level constructs (structures, inline assembler)
      - Bartok just-in-time compiler (CIL to x86)
  - Research prototype
    - Midori expected to be a commercial variant and future replacement of Windows NT (discontinued in 2015)
  - Several similar approaches
    (MOSA, Cosmos for C#; JNode, Phantom OS for Java)

Multikernel



Multikernel (2)

- **Barrelfish**
  - ETH Zürich, Microsoft Research
  - Side note: Mackerel hardware description language
    - Driver synthesis
  - Support for heterogeneous CPU cores
  - Common asynchronous messaging abstraction between cores/nodes
    - Explicit inter-core communication (as opposed to cache coherency)
    - Practically no shared memory and state between cores
    - Managing state replicas using distributed algorithms between cores

Department of Distributed and Dependable Systems

Barrelfish: Mackerel Language

- **Describe devices and control registers**
  - Used to generate C accessor functions
  - Also for other low level data structures

```
device HPET lsbfirst ( addr base ) "High-Precision Event Timer" {
    register gcap_id ro addr(base, 0x0) "General Capabilities and Identification" {
        rev_id               8    "Revision Identification";
        num_tim_cap          5    "Number of Timers";
        count_size_cap       1    "Counter Size";
        _                    1    mbz;
        leg_rt_cap           1    "Legacy Replacement Rout Capable";
        vendor_id_cap        16   "Vendor ID";
        counter_clk_per_cap  32   "Main Counter Tick Period";
    };
```

Department of Distributed and Dependable Systems

Multikernel (3)

- **Barrelfish**
  - Kernel acts as a “CPU driver”
    - Event-driven, single-threaded, non-preemptable
      - Processing syscalls from user space, interrupts from devices and other cores
      - User space processes communicate with the CPU driver using a dispatcher object (local user space thread scheduler)
    - ~10.000 lines of C, ~500 lines of assembler
  - Common messaging abstraction does not mean common (sub-optimal) messaging transport
    - Fast path messaging between cache-coherent cores: Sending messages in cache lines
      - Sender writes words sequentially into the cache line (interconnect cache line invalidate)
      - Receiver polls on the last word of the cache line (interconnect cache line fetch)

Barrelfish: Interface Specification

- **Describe types and messages**
  - Used to generate IPC stubs

```
interface timer "Timer service" {
     // set the one (and only) timeout value (in us) for this client
     message set_timeout(uint64 timeout);
     // add the given increment (in us) to the running timer for this client
     message add_to_timeout(uint64 increment);
     // cancel the outstanding timeout
     message cancel_timeout();
     // wakeup response when the timer is triggered
     message wakeup();
     // request for the remaining time of the currently-running timer
     message get_remaining();
     // response containing remaining time of running timer
     message remaining(uint64 time);
};
```

Department of Distributed and Dependable Systems

Q&A

References

[1] Grm wnr, Xyzzy n, https://commons.wikimedia.org/wiki/File:Windows_2000_architecture.svg
[2] Utente:Sassospicco, https://commons.wikimedia.org/wiki/File:Diagram_of_Mac_OS_X_architecture.svg