



Linux Kernel

EIM/INFM

Frank Erdrich
frank.erdrich@emtrion.de

Warum ein Betriebssystem?

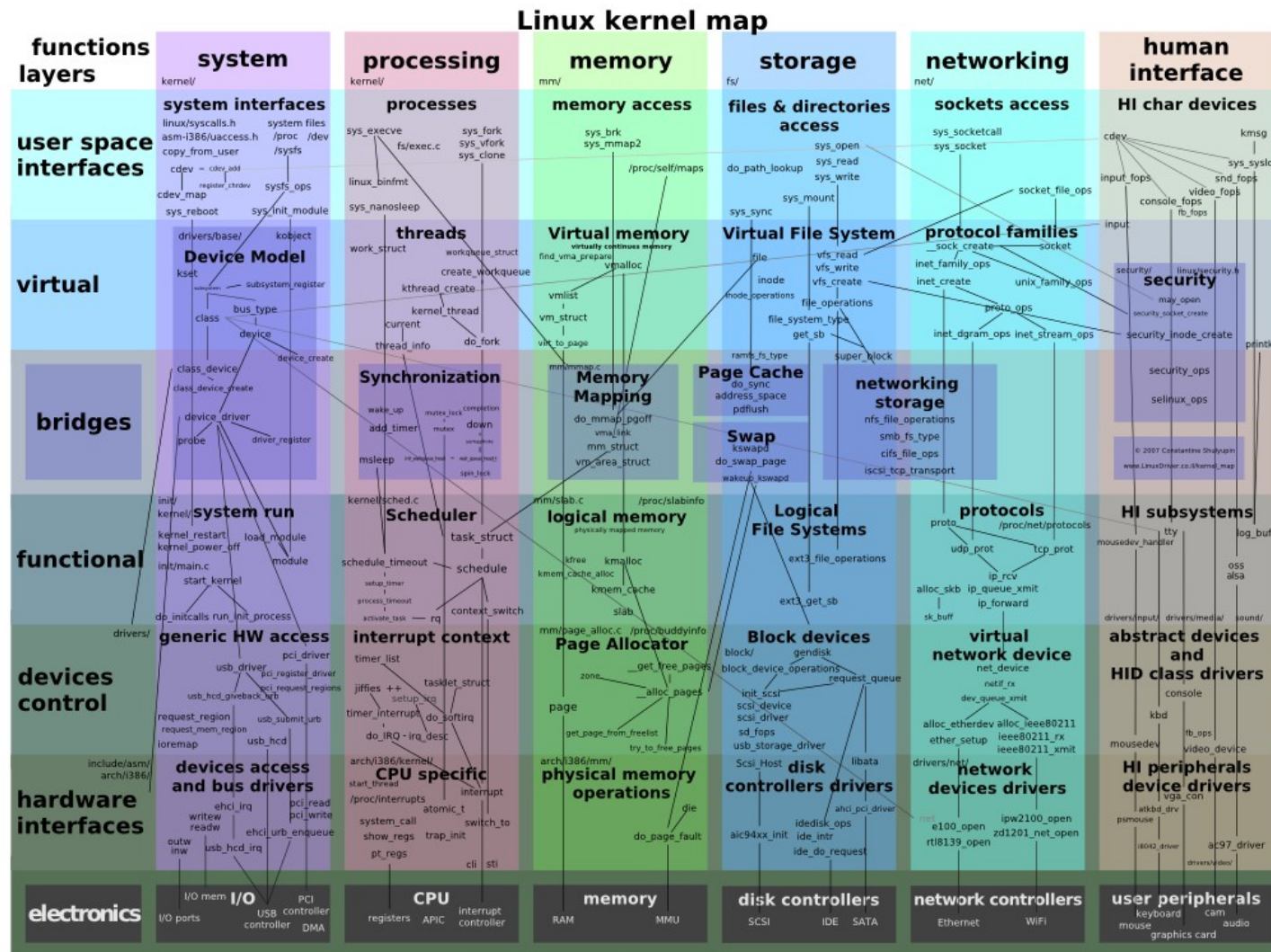


Warum ein Betriebssystem?

- Hardwareabstraktion
- Speicher und Ressourcenverwaltung
- Schutz der Applikationen gegeneinander
- Mehrere Prozesse „gleichzeitig“
- Zugriffssteuerung auf HW



Warum ein Betriebssystem?

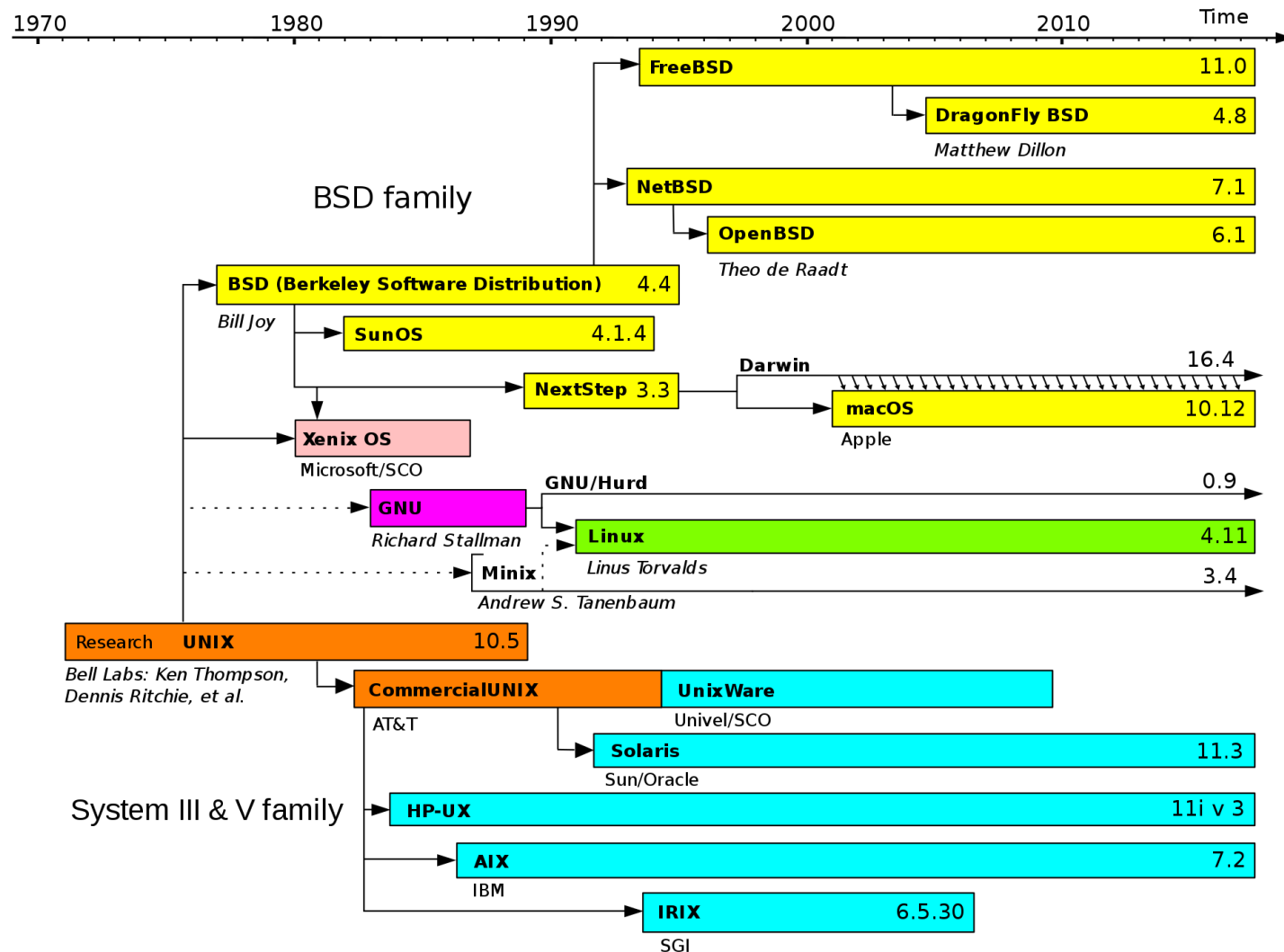


Linux Kernel

- Entwickelt von Linux Torvalds
- Open Source
- Communitygetrieben
- Unterstützt diverse Plattformen
 - X86
 - ARM
 - RISC V
 - PowerPC



History Unix/Linux



Quelle: <https://en.wikipedia.org/wiki/Linux>

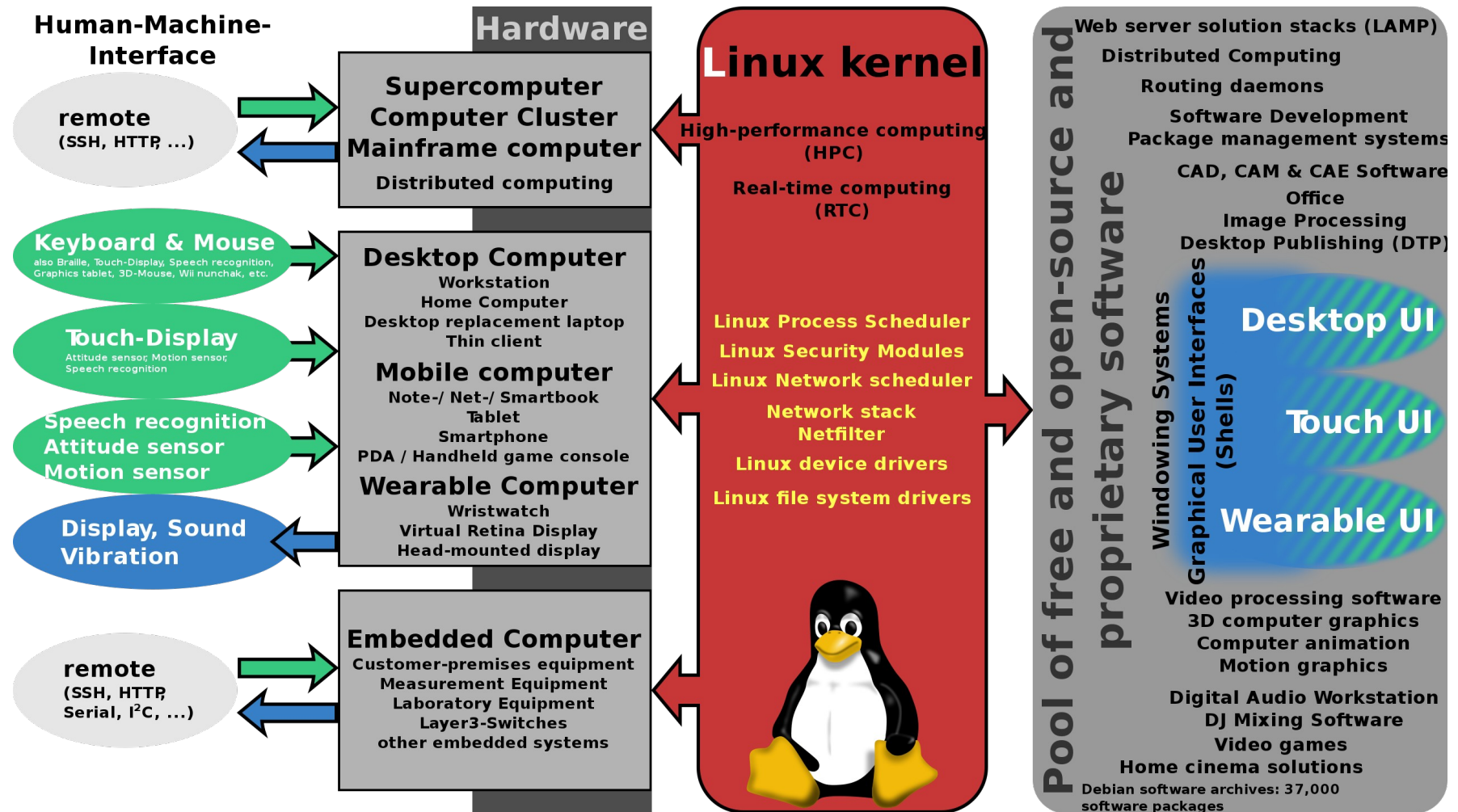


Linux Kernel in Zahlen

- Erstes Release 17. September 1991
- 25,3 Millionen Lines of Code (v4.15)
 - Windows 7 > 40 Millionen LoC
 - MacOS X > 80 Millionen LoC
- > 1600 Entwickler
- Unterstützt >20 Plattformen

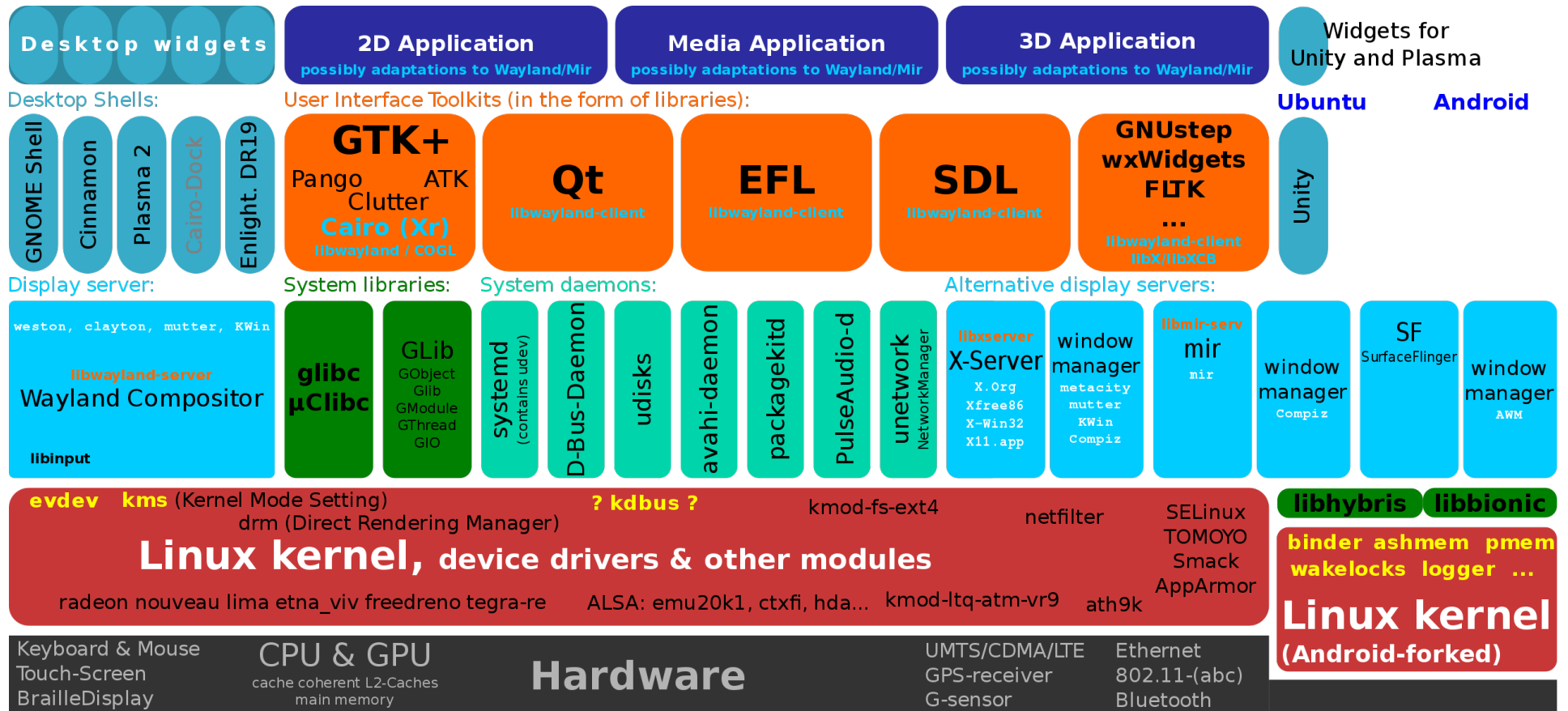


Wo zu finden?



Quelle: <https://en.wikipedia.org/wiki/Linux>

Linux Desktop



Quelle: <https://en.wikipedia.org/wiki/Linux>



Linux != Linux?

- Linux Kernel
 - Weitestgehend POSIX kompatibel
 - monolithisch
- Linux Distribution
 - Kernel + Userspace



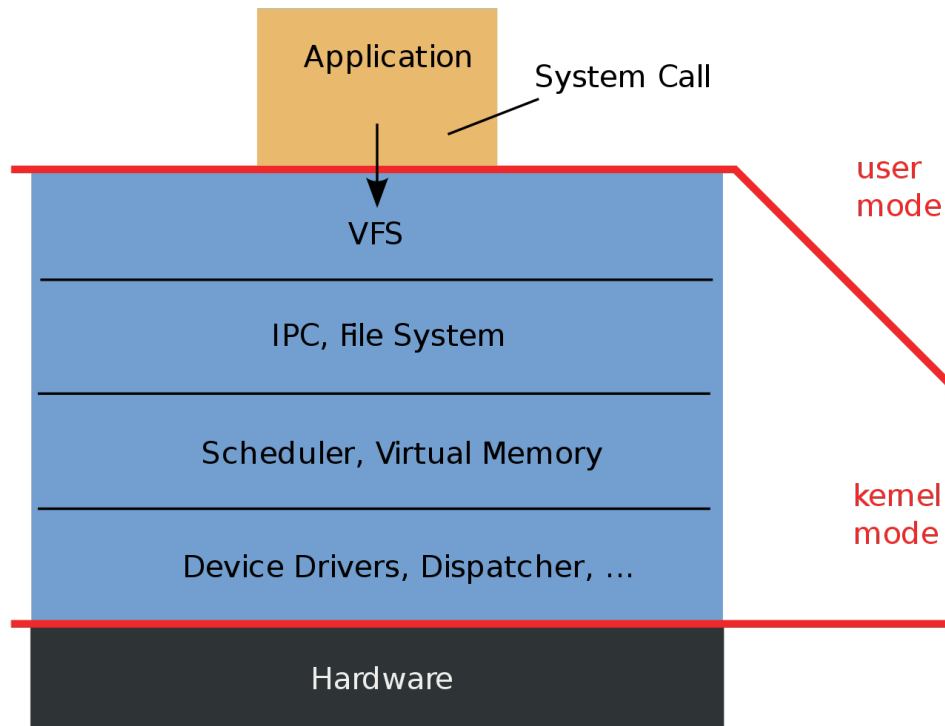
POSIX

- Portable Operating System Interface
- Für Unix entwickelte Programmierschnittstelle
- Enthält unter anderem
 - Prozesse
 - Threads
 - Synchronisation
 - Signals
 - File I/O

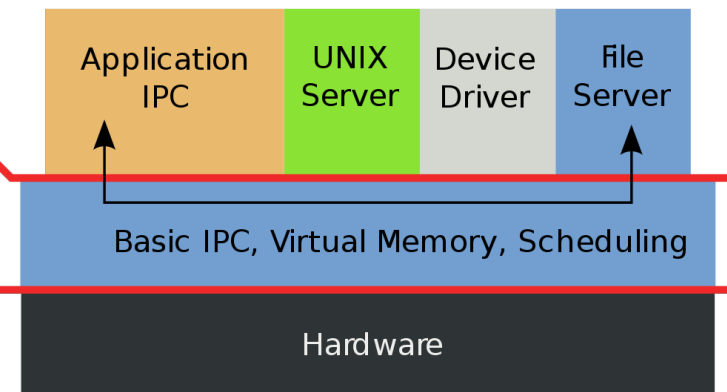


Monolith vs. Microkernel

Monolithic Kernel
based Operating System

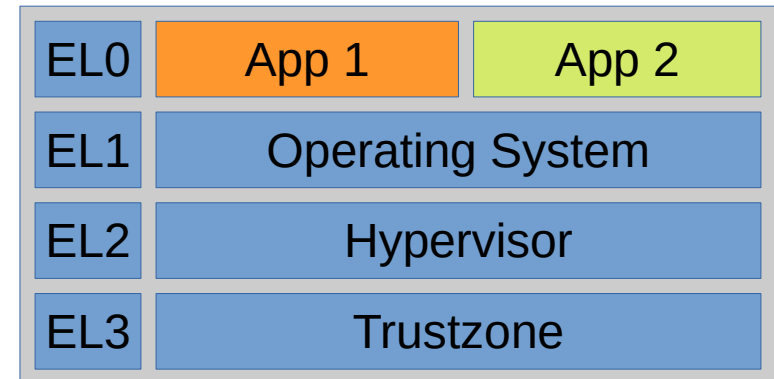


Microkernel
based Operating System



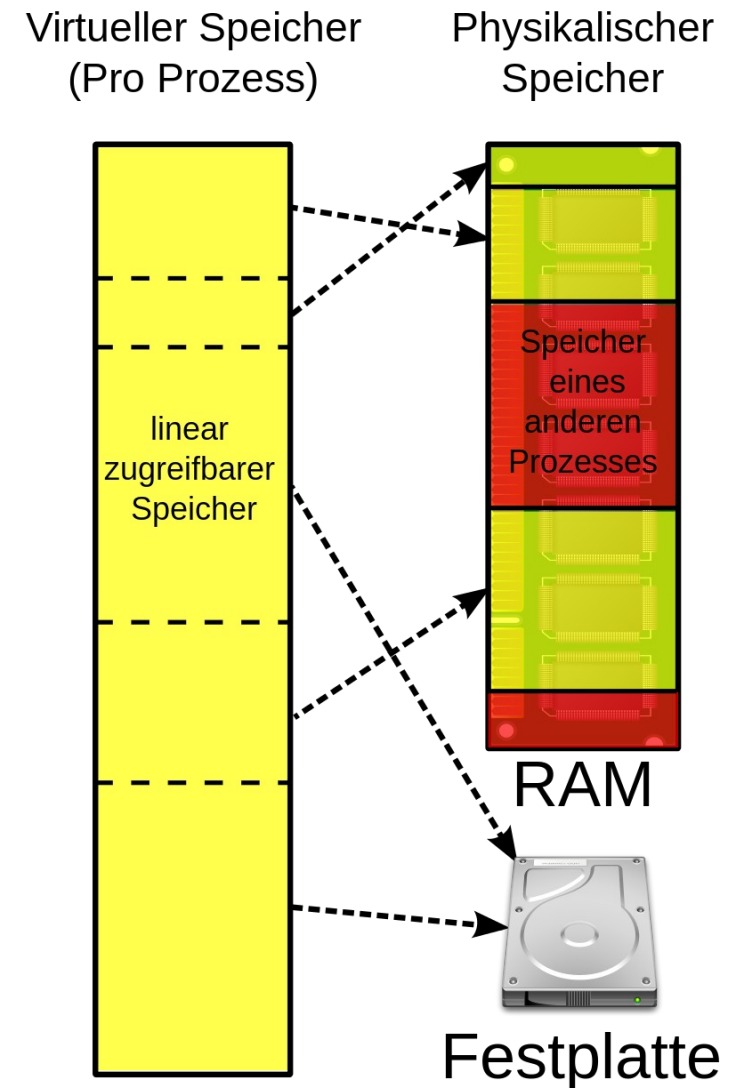
Execution Level

- Privilegiertes Ausführungslevel für OS
 - u. a. MMU-Zugriff
 - Direkte Hardwarezugriffe
 - Volle Kontrolle über CPU



Virtueller Speicher

- Linearer Speicher für jede Applikation
- Schutz der Applikationen untereinander
- Schutz des Kerns



Virtueller Speicher

- Aber auch
 - Shared Memory
 - Mapping von Hardware in den normalen Speicherbereich
 - Zugriffsrechte für Speicherbereiche

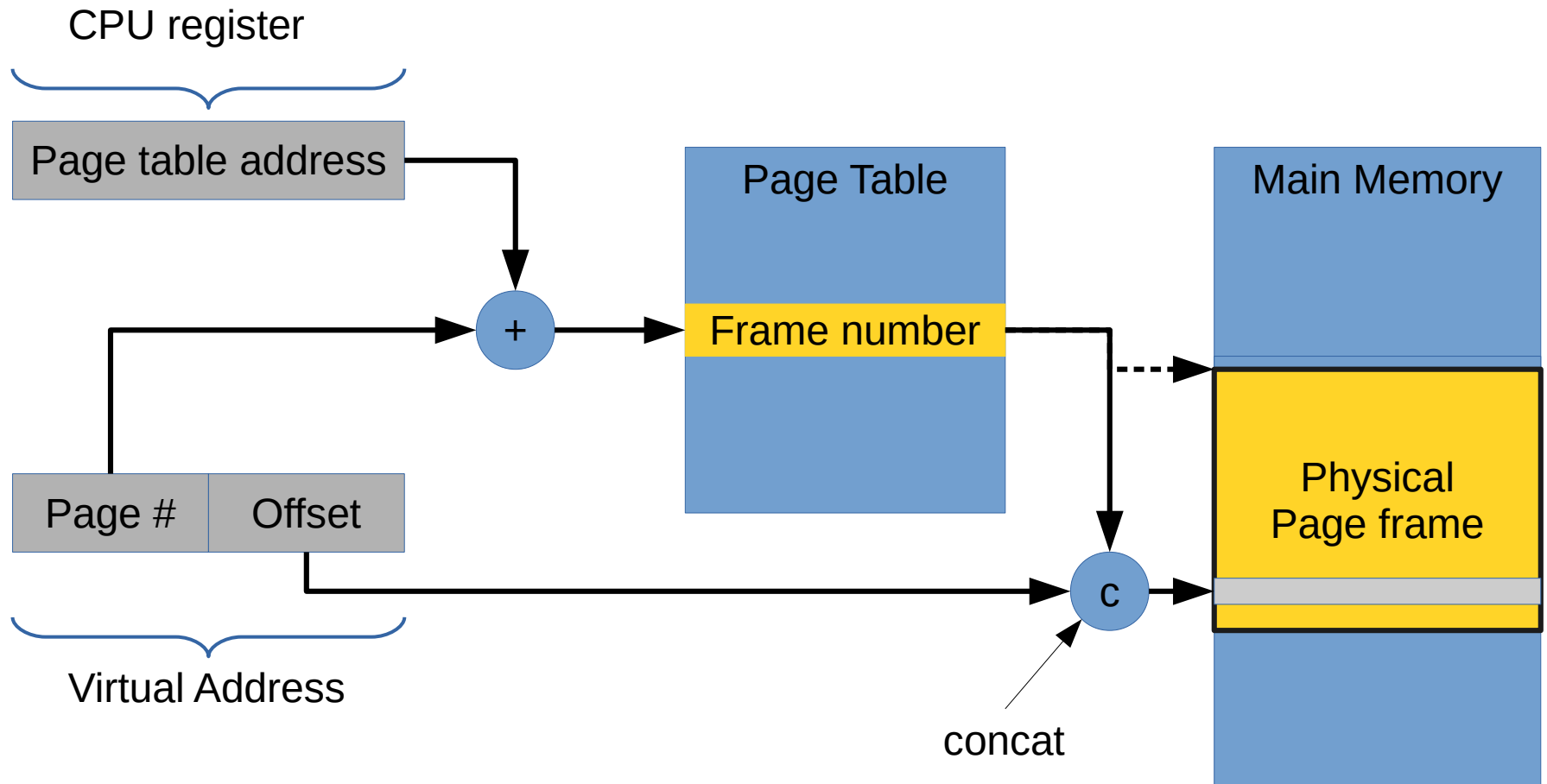


Virtueller Speicher

- Page Table
 - Übersetzung der virtuellen Adresse in physikalische Adresse
 - Evtl. Mehrstufig
- Caching mittels TLB (Translation Lookaside Buffer)
- Teil der MMU (Memory Management Unit)



Virtueller Speicher

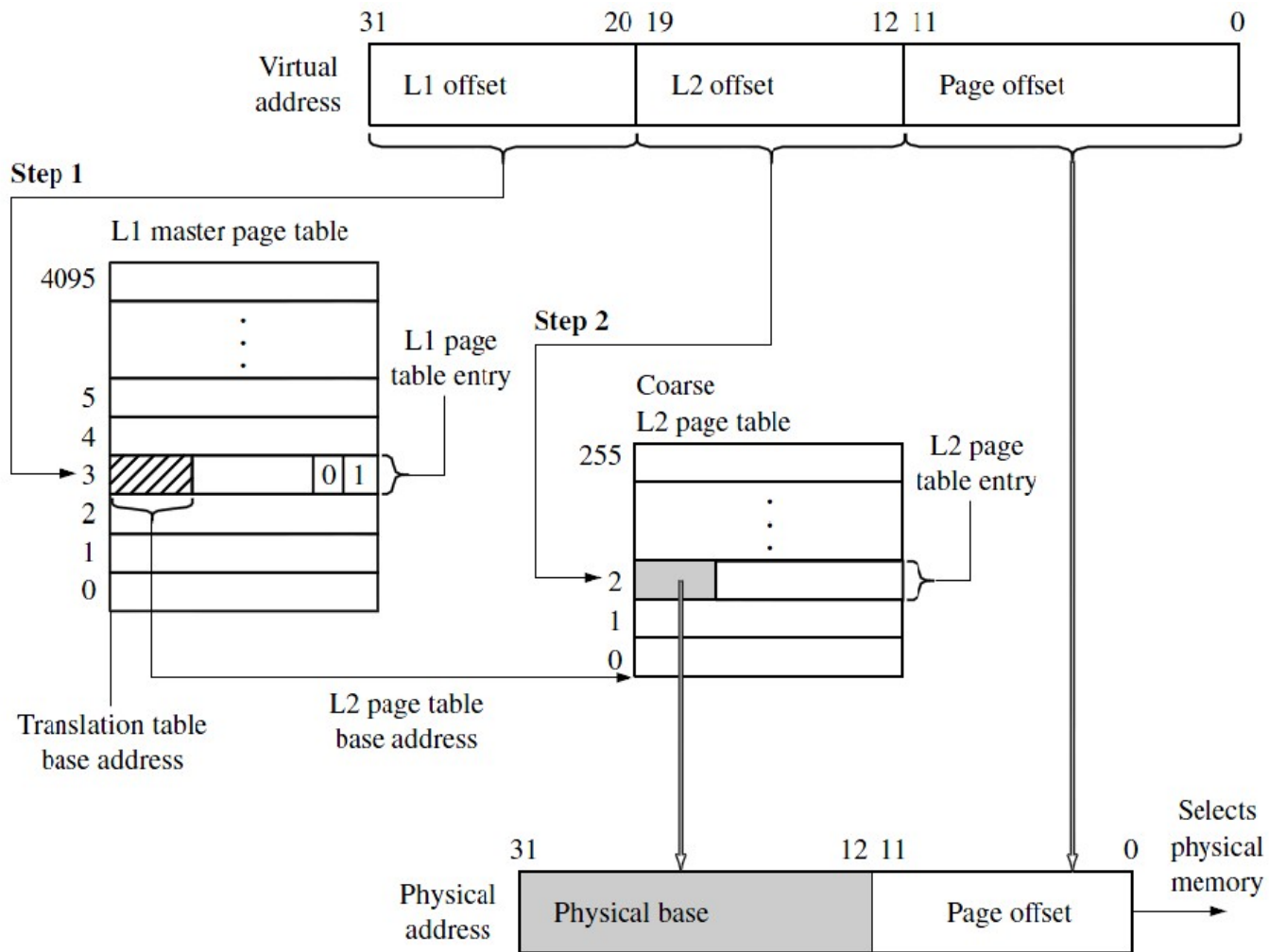


VMsAv7

- Virtual Memory System Architecture
 - Pagesize 4KB, 64KB, 1MB, 16 MB
 - 2-level Translation
 - 1st level: 4096 x 1MB
 - 2nd level: 256 x 4 KB
 - 7 Zugriffslevel
 - Unterscheidung zwischen Privileged und User
 - XN-Bit: eXecute Never



VMSAv7



VMSEv7

- Translation Table Base Register
 - TTBR0
 - Für Prozessspeicher -> eine TTB pro Prozess
 - TTBR1
 - Für Betriebssystem
- Kein Flush bei Contextwechsel notwendig
 - Identifier für globalen oder Prozessspeicher



VMSEv7

- Zusätzlich Übersetzung bei Virtualisierung
 - 2-Stage Decoding (Hyp-Mode)
 - Virtual to Intermediate Physical
 - Intermediate Physical to Physical
- Mehr dazu im Kapitel Virtualisierung
- Zusätzlich Variante mit LPAE

