

# Von DT zu ACPI

Pros und Cons zum Einsatz von x86  
vs. ARM im Embedded-Linux-Umfeld

# Struktur

- Einleitung
- Plattform-Beschreibung: ACPI und Device-Tree
- Beispiel Intel Apollo Lake
- Zusammenfassung

# Einleitung

# Trends in Hardware und Software

## Hardware

- SoCs im Embedded *und* PC-Bereich
- Wärmeentwicklung spielt immer eine Rolle
- Kosten nähern sich an

## Software-Anforderungen aus Embedded-Sicht

- Integration komplexer GUIs
- Integration in Netzwerke
- Verwendung von Standard-Protokollen und Methoden aus der Applikationsentwicklung

# Geschichte: **Embedded** vs. PC

- Diverse **ISAs**
- Kein Bootloader
- Kein Betriebssystem
- „**sofort**“ da
- „bare-metal“

# ARM®



# MIPS TECHNOLOGIES

# RISC-V

# Geschichte: Embedded vs. PC

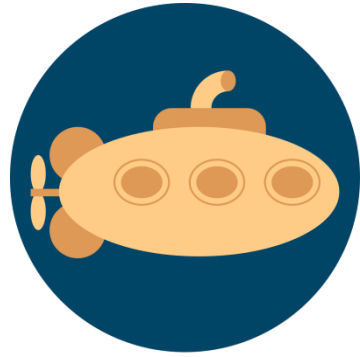
- **BIOS** („Basic Input Output System“)  
Ressourcenliste und Initialisierung,  
HW-Abstraktion (Disk, Grafik, ...)
- Relativ einheitliche Hardware
- Bootet von Disk (in **Minuten**)
- **Betriebssystem** DOS („Disk Operating System“)



# Embedded und PC wächst zusammen

## Embedded

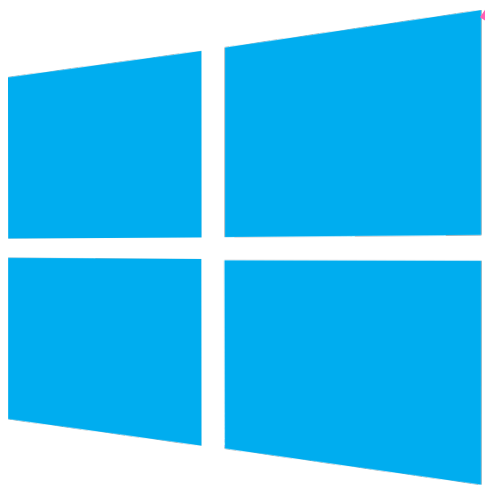
- HW-Bootloader
- 2nd-Stage Bootloader



U-Boot

## PC

- (UEFI)-BIOS
- x86 dominiert, aber auch andere



- „echte“ Betriebssysteme mit HW-Abstraktion
- Bootzeit: einige Sekunden
- Hardware-Vielfalt



Plattform-Beschreibung nötig!

# Geschichte: Plattform-Beschreibung

Embedded-Linux



- Hardware-Liste als **C-Code** im Kernel (Plattform-Code)
- d.h. HW-angepasstes Betriebssystem (Compile-Time)
- Immer noch **viel Legacy-I/O**: I<sup>2</sup>C, SPI, UART, I<sup>2</sup>S, PWM, GPIOs

2012: **Device-Tree**

PC



- Hardware-Liste und Interface im BIOS (Compile-Time)
- Zunehmend **PnP-fähige Busse**
- Power-Management früh wichtig

1996: **ACPI**

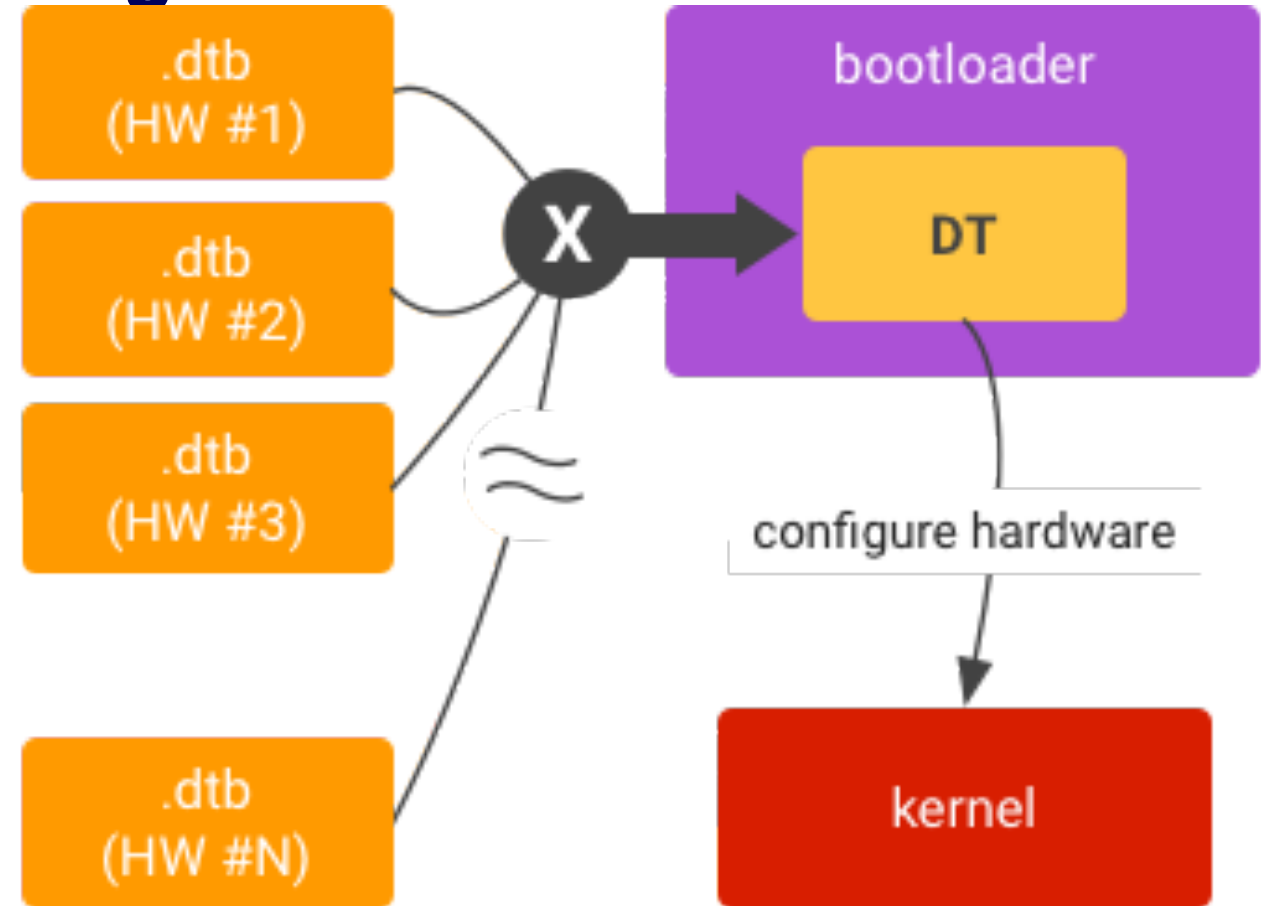
(Advanced Configuration and Power Interface)



# Plattform-Beschreibung: ACPI und Device-Tree

# Idee der Plattform-Beschreibung

- Alle **nicht-PnP-Komponenten** des Systems werden in einer Struktur mit ihren Eigenschaften **deklariert**
- Die Plattform-Beschreibung wird im **Bootloader** hinterlegt oder ermittelt und wird dem Betriebssystem beim Laden als Parameter übergeben

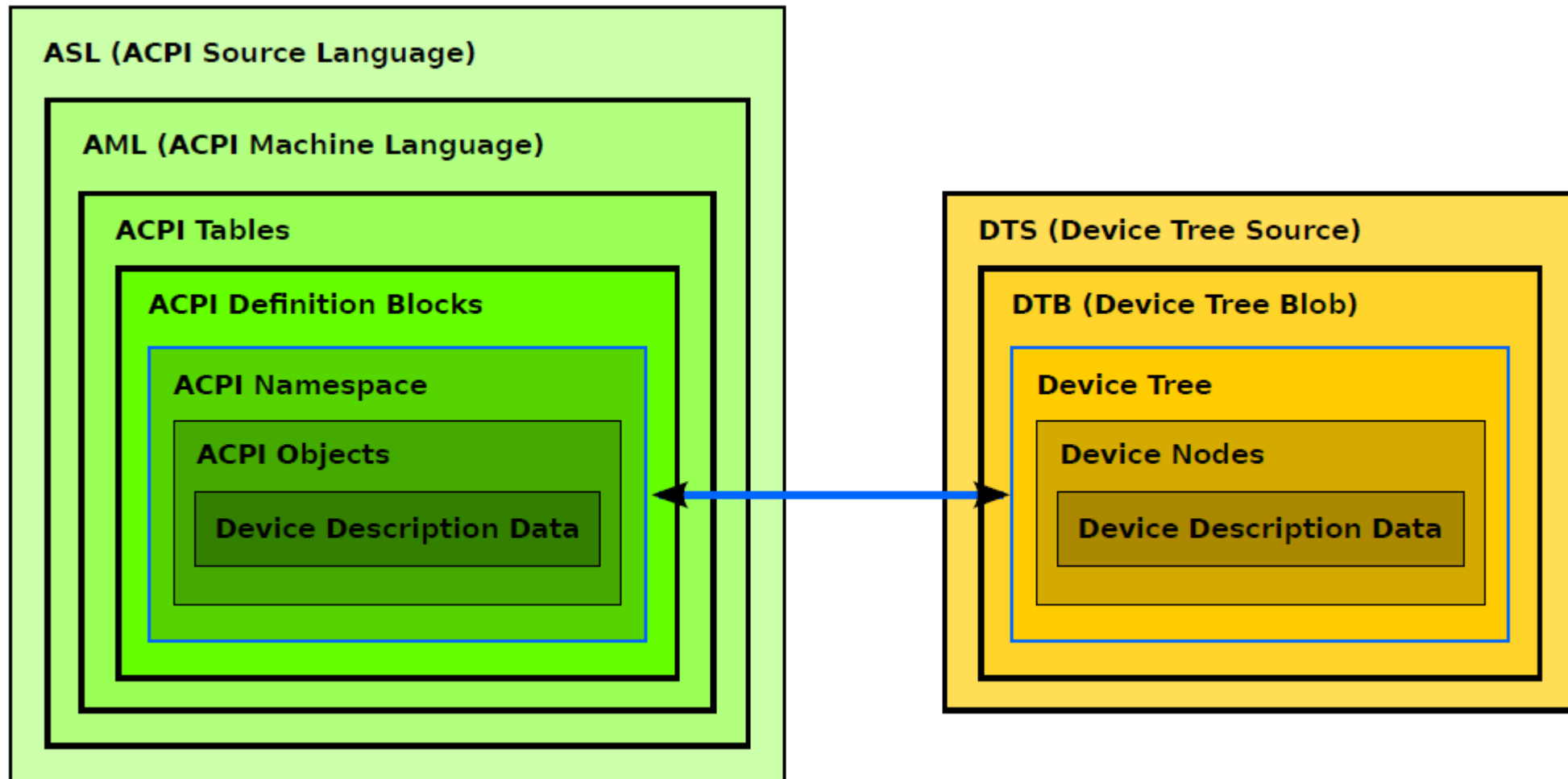


**Problem: BIOS i.d.R. proprietär!**



identify hardware and load one with .dtb

# Struktur-Vergleich ACPI und DT



# Unterschiedliche Syntax und Semantik!

## Device-Tree

```
&i2c6 {  
    status = "okay";  
    clock-frequency = <400000>;  
    tpsusbc: tpsusbc@38 {  
        compatible = "ti,tps6598x";  
        interrupt = <152>  
        reg = <0x38>; };};
```

## ACPI/ASL

```
Device (TUSB) {  
    Name (_HID, "INT3515")  
    Name (_CRS, ResourceTemplate () {  
        I2cSerialBus (0x38, 400000, ...  
            "\\_SB.PCI0.I2C6")  
        GpioInt (Edge, ActiveLow, PullUp,  
            "\\_SB.GPO0", ...) {53}}})
```

# ACPI / DT Support in Linux

- In Linux traditionell **zwei verschiedene Subsysteme/ APIs!**
- Jeder Treiber benutzt traditionell **eine** der APIs!

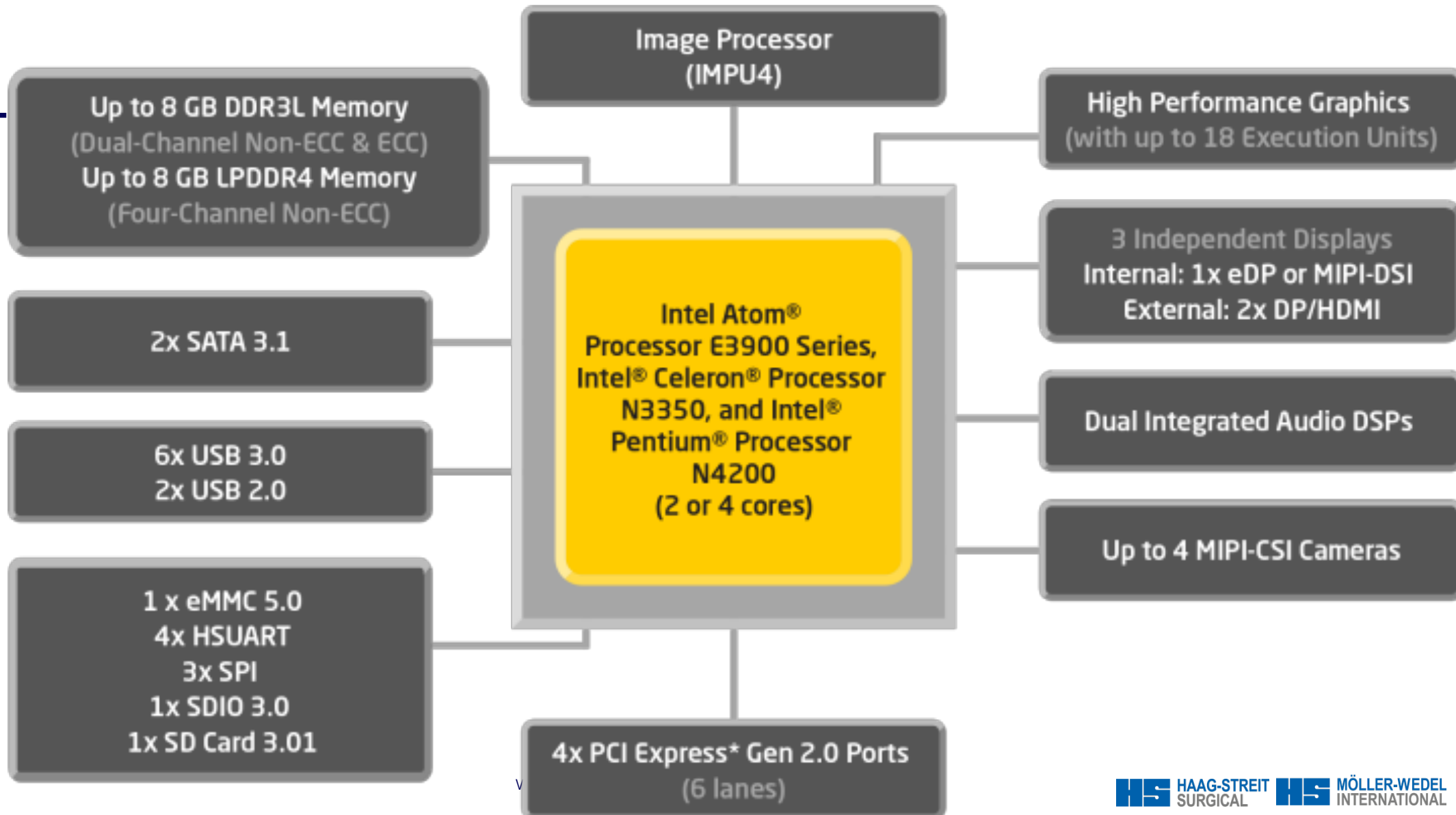
# Vereinheitlichte Firmware-Property-API

- Seit 2014: **fwnode** oder device-property API
- Seit 2014: Spezielle Hardware-ID (`_HID`) im ASL um einen Device-Tree-kompatiblen Treiber zu referenzieren: `PRP0001`
- Im DT wird ein Treiber über einen String beliebiger Länge referenziert, in ACPI über einen 64-Bit Integer
- Lösung für proprietäres BIOS: **DSDT-Overlays**, die unter Linux nachgeladen werden können und die BIOS-Tabellen ergänzen
- Wenige Treiber nutzen bereits die neue fwnode-API

# Beispiel: Intel Apollo Lake

# Apollo Lake als Embedded SoC

- 4 x UART
- 6 x I2C
- 2 x SPI
- GPIOs





# Apollo-Lake Integration

- Verfügbar als Modul in diversen Formaten von verschiedenen Herstellern
- Integriert Flash, RAM, PMC, BIOS
- Von Intel gepflegtes BSP auf Yocto-Basis
- Von Intel beigesteuerter Coreboot-Port



- Starke Grafik-Leistung
- USB und PCIe

# Probleme während der Integration

- **Legacy-Devices** durch ACPI-Overlays: Probleme bei der Rückwärtskompatibilität zu alten Entwicklungsständen
- **Sound**: Im Embedded-Bereich ist I<sup>2</sup>S üblich, funktioniert zunächst nicht
- Einige **Treiber** mussten angepasst werden
- **Lange Bootzeit** durch BIOS
- **Software-Update**: BIOS kann nur durch BIOS geflasht werden
- SoC **komplexer** als vergleichbare ARM-SoCs

# Pros and Cons

## Pro

- Starke **Grafik-Leistung**, drei DP-Displays direkt ansteuerbar
- Sehr gutes **Preis-/Leistungs-Verhältnis**
- **Toolchain einfacher**, da kein Cross-Compiling nötig

## Contra

- BIOS: **Startzeit**
- BIOS: Anpassungen nur durch Hersteller
- BIOS: **Komplexität**
- **Treiberanpassungen** momentan noch oft nötig (ACPI)

# Zusammenfassung

# Zusammenfassung

- Integration eines x86-SoM **aufwändiger** als bei einem vergleichbaren ARM-SoM
- Hauptproblem: **proprietäres BIOS**
- Mittelfristig: Slimbootloader / Coreboot
- Nebenproblem: **Linux-Support** für Embedded x86 unvollständig
- Mittelfristig: Besserer Kernel-Support



**HAAG-STREIT**  
SURGICAL



**MÖLLER-WEDEL**  
INTERNATIONAL