

Untersuchungen zur Zulassung von Software unterschiedlicher Sicherheitsklassen auf einem Prozessormodule unter dem neuartigen Betriebssystem PikeOS



Automotive Safety & Security 2008
Stuttgart, 19. – 20.11.2008

Überblick



- Vorüberlegungen
- Konzept des FMS/MMS Demonstrators
- Status des Demonstrators
- Ergebnis und weitere Nutzung

FMS: Flight Management System
MMS: Mission Management System

Vorüberlegungen

- Heutige Rechnerarchitektur ist sehr leistungsfähig und erlaubt Multiprozessanwendungen
- Vereinheitlichung von Flugzeugrechnern reduziert Kosten
- Bessere Rechnerausnutzung durch Nutzung für alle Sicherheitsklassen: Untersuchung für Kombination von FMS und MMS
 - Anforderungen
 - Architektur
 - Software Module

Vorüberlegungen

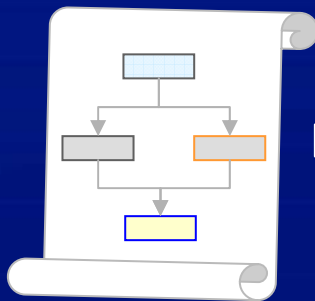


- Ziel
 - Erstellung eines Konzeptes für eine Rechnerplattform für FMS/MMS Anwendungen
 - Implementierung auf Prototypingbasis unter Nutzung bestehender Software
 - Demonstration des Konzeptes
 - Rahmen für Zulassbarkeit

Vorüberlegungen

- Prototyping / Evaluierung FMS/MMS Rechnerkonzept

Identifikation funktionale FMS/MMS-Architektur



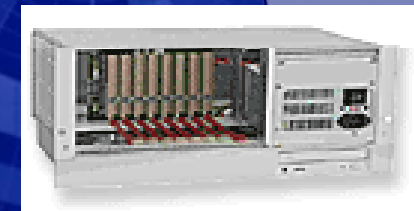
Prototyping HW/SW-Konzept



- „safety critical“ Appl
- „mission critical“ Appl



FMS/MMS Demonstrator mit Bsp. Appl.



Bsp. Appl.: Beispiel Applikation
LCC: Life Cycle Costs

Überblick



- Vorüberlegungen
- Konzept des FMS/MMS Demonstrators
- Status des Demonstrators
- Ergebnis und weitere Nutzung

FMS: Flight Management System
MMS: Mission Management System

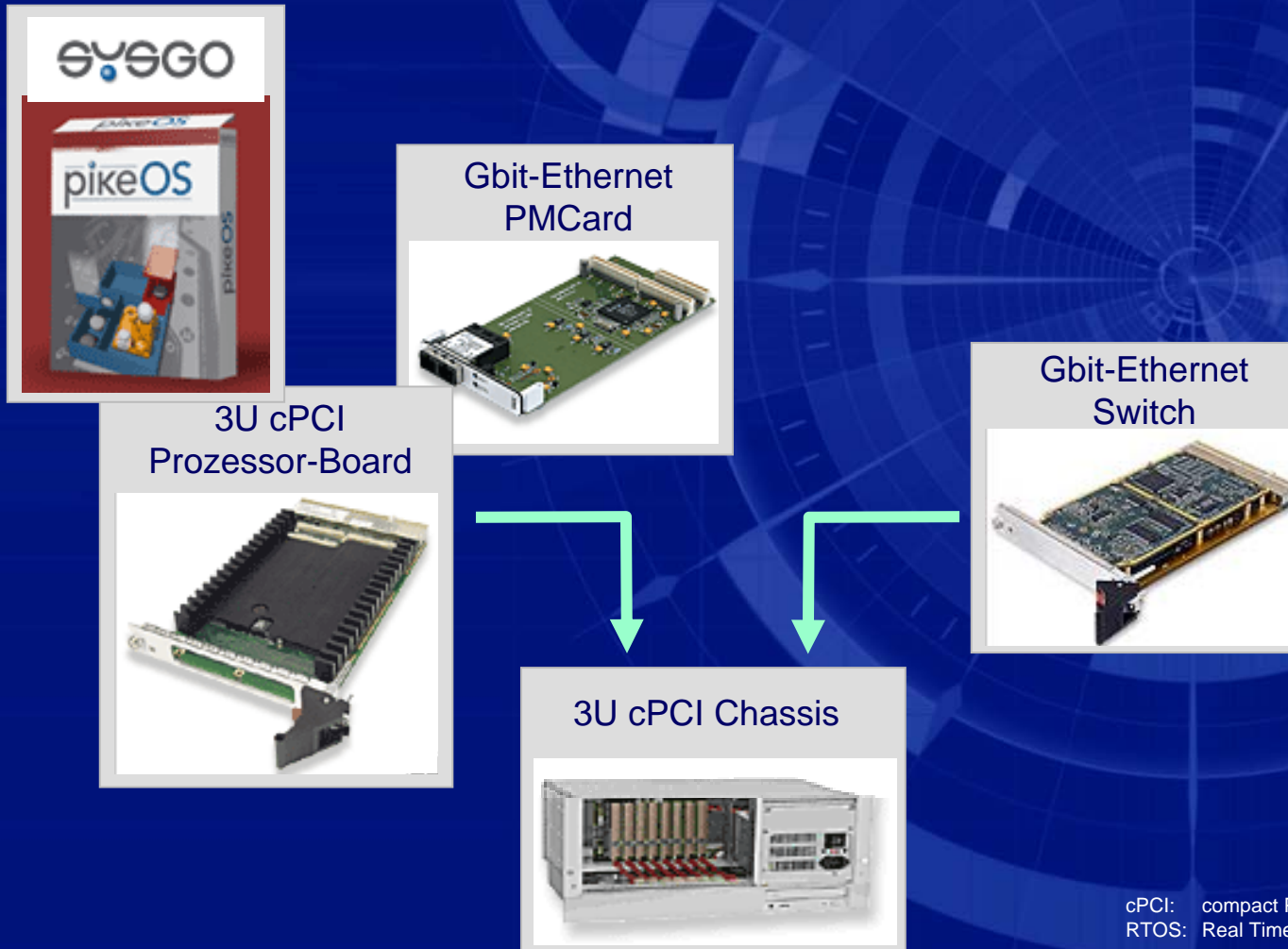
Konzept des FMS/MMS Demonstrators



- HW/SW Konzept
- Betriebssystem
- Aufbau

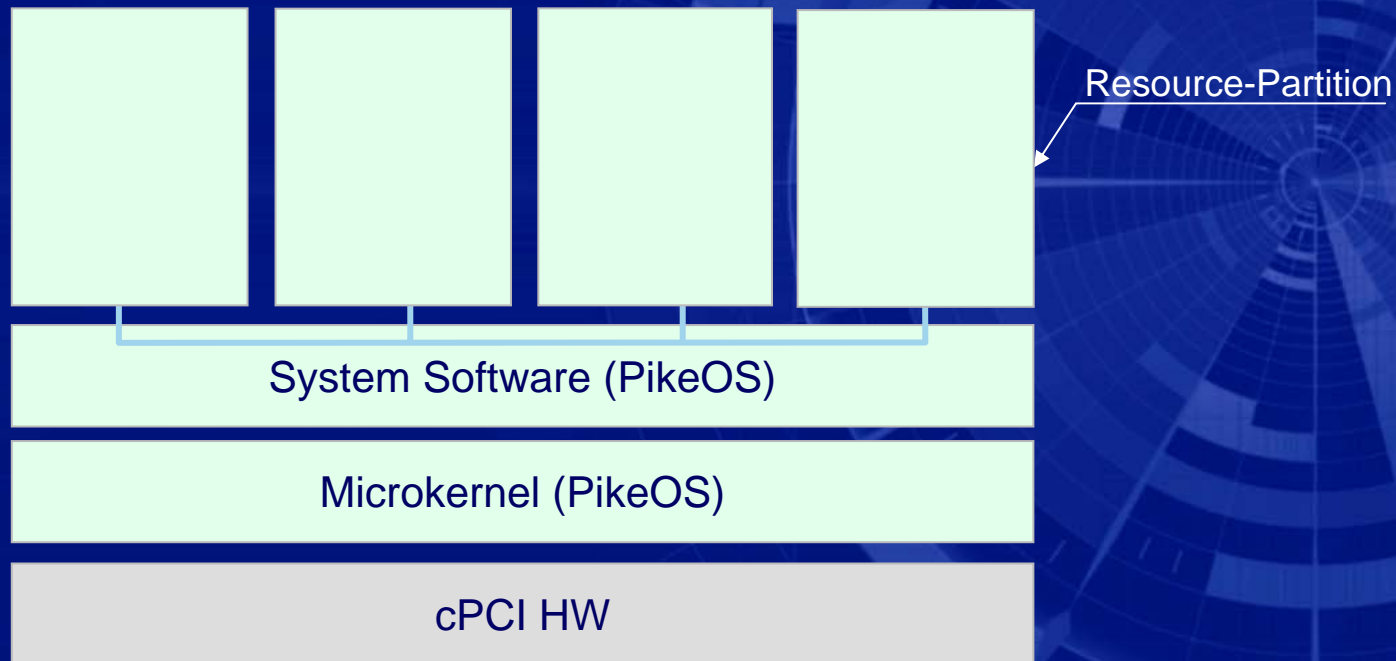
HW/SW-Konzept des Demonstrators (1)

- Europäisches RTOS



cPCI: compact Peripheral Component Interconnect
RTOS: Real Time Operating system

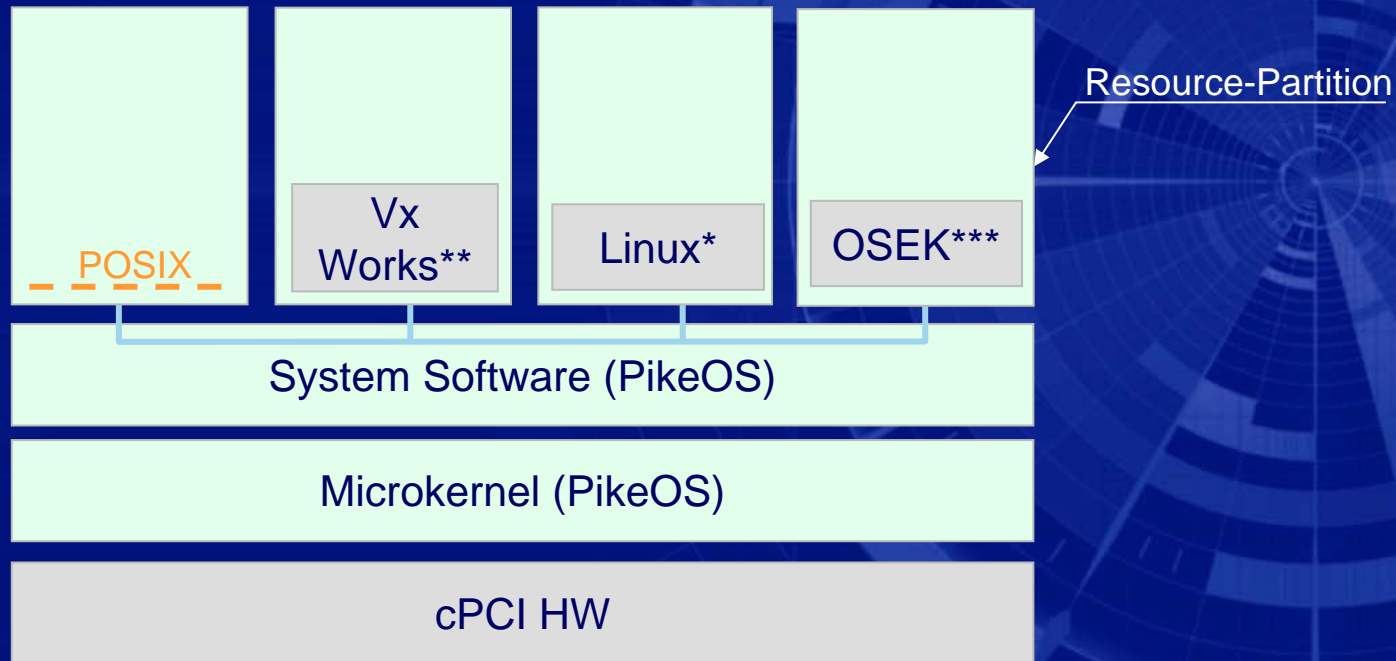
HW/SW-Konzept des Demonstrators (2)



- * LINUX SW-Modul von SYSGO für PikeOS
- ** VxWorks SW-Modul von SYSGO für PikeOS
- *** OSEK SW-Modul von SYSGO für PikeOS

OSEK: Offene Systeme und deren Schnittstellen
für die Elektronik in Kraftfahrzeugen

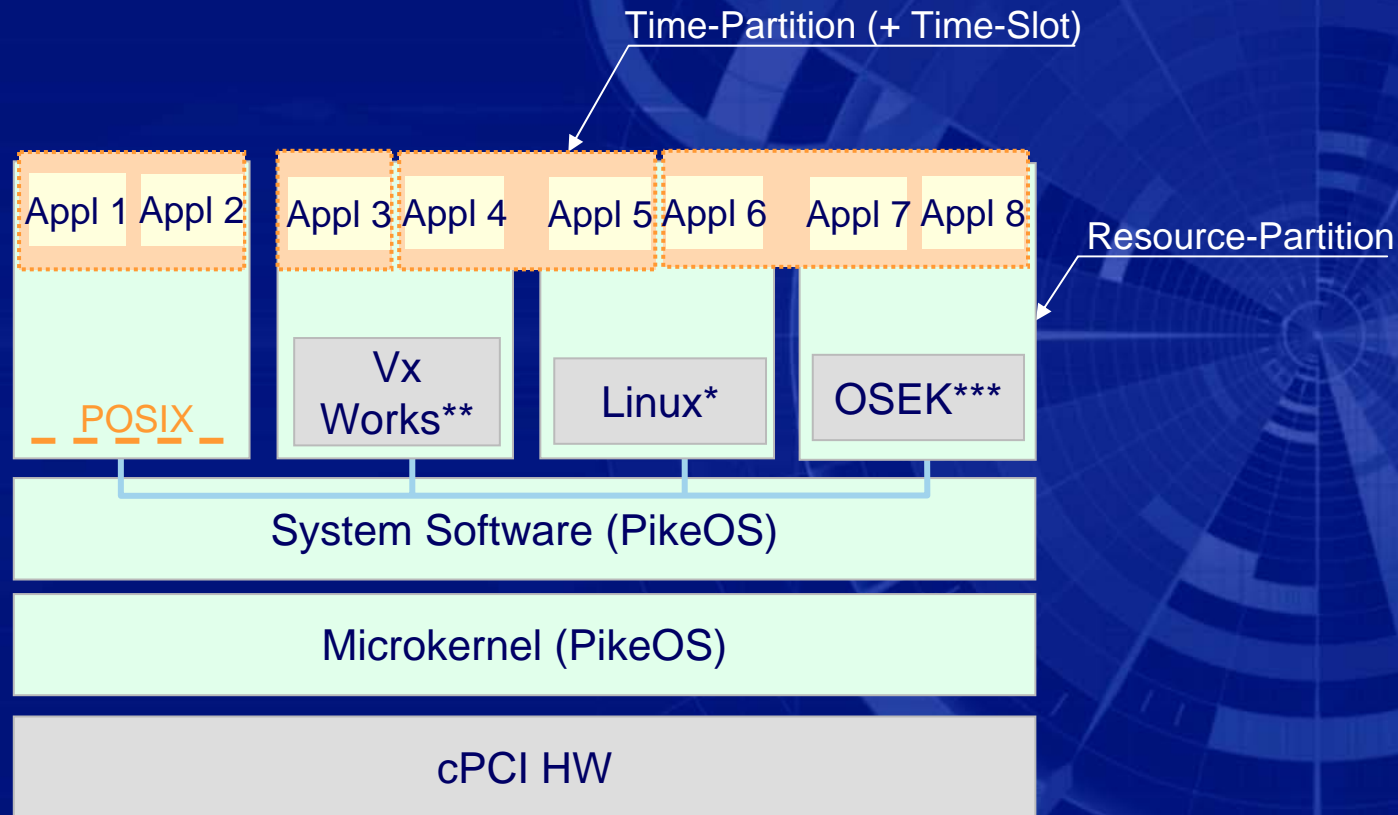
HW/SW-Konzept des Demonstrators (2)



- * LINUX SW-Modul von SYSGO für PikeOS
- ** VxWorks SW-Modul von SYSGO für PikeOS
- *** OSEK SW-Modul von SYSGO für PikeOS

OSEK: Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen

HW/SW-Konzept des Demonstrators (2)



- * LINUX SW-Modul von SYSGO für PikeOS
- ** VxWorks SW-Modul von SYSGO für PikeOS
- *** OSEK SW-Modul von SYSGO für PikeOS

OSEK: Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen

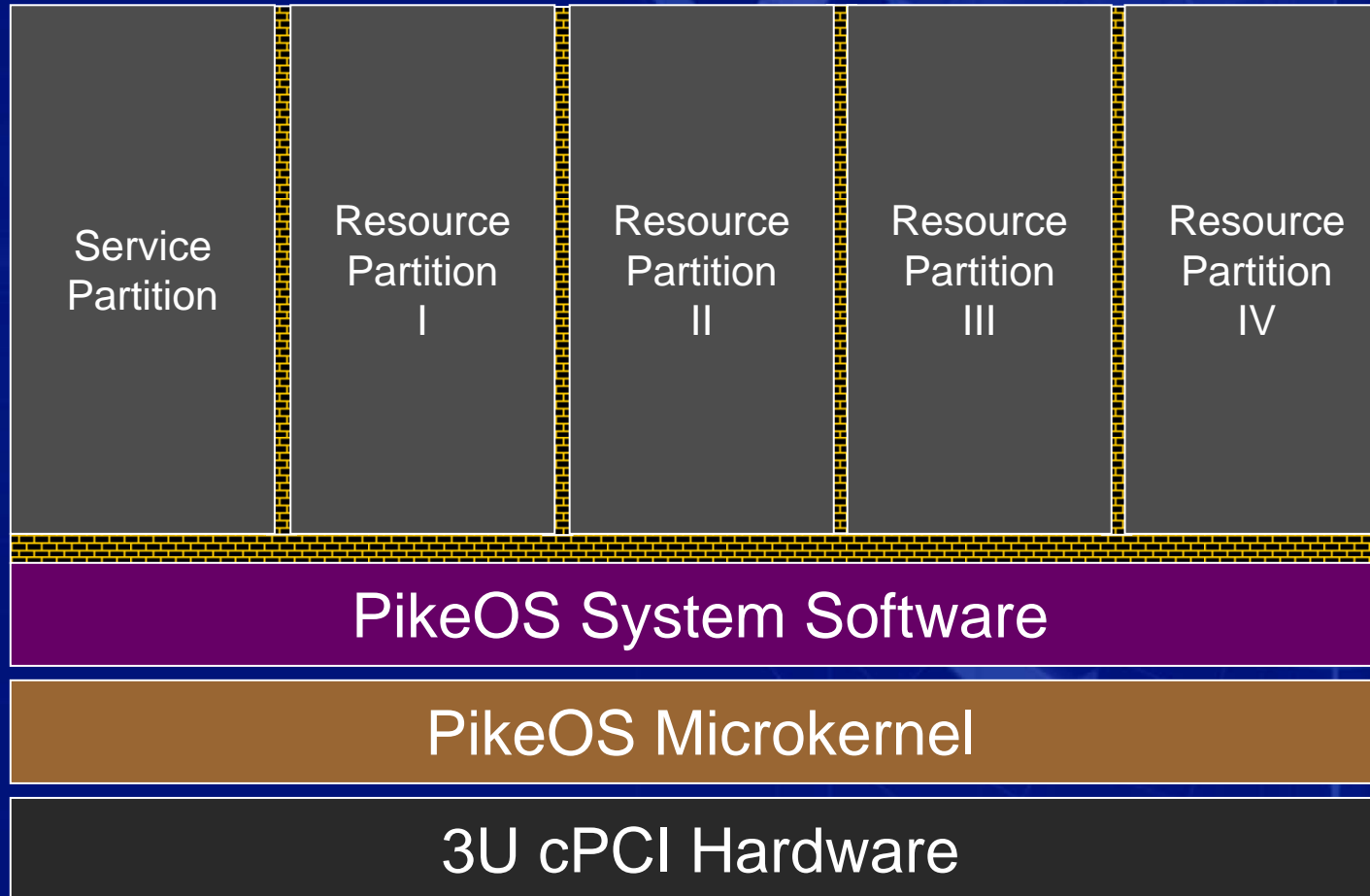
PikeOS Microkernel und System Software

PikeOS Aufbau:

- Microkernel
- Resource Partitionen
- Time Partitionen und Scheduling
- Konzept der Personalities
- Kommunikation unter PikeOS
- Fehlerbehandlung

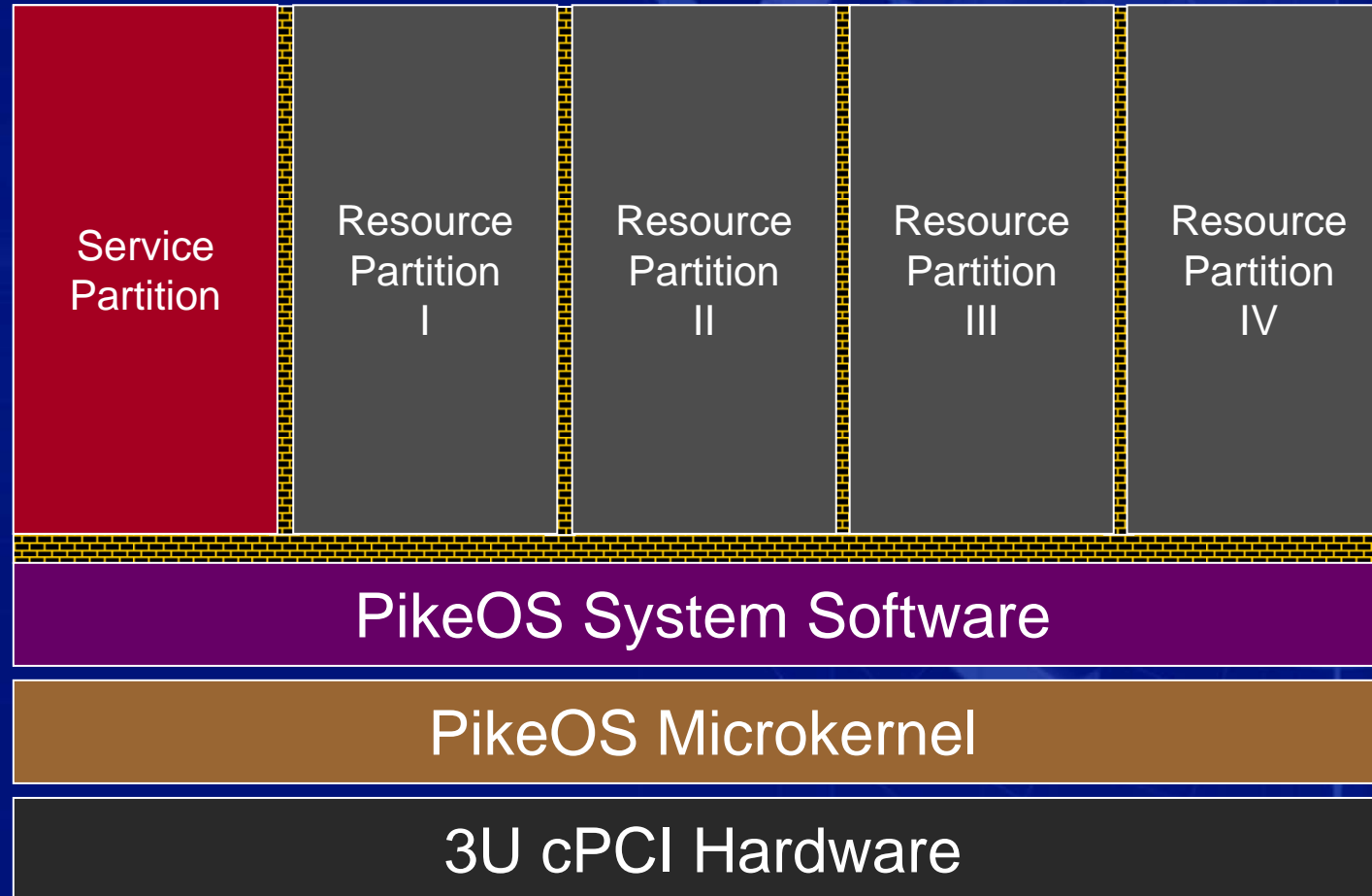


Partitionierung und Scheduling



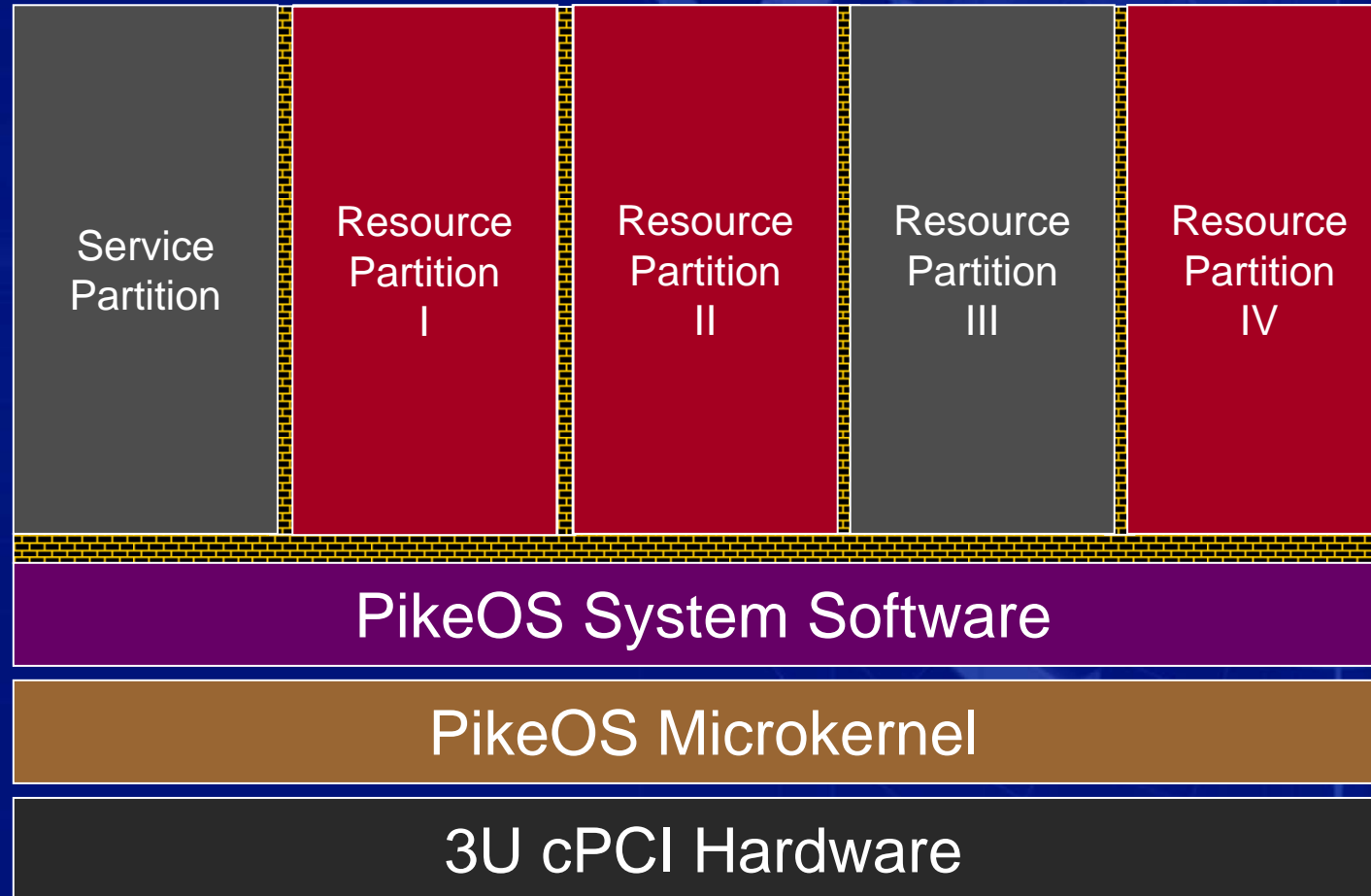
Partitionierung und Scheduling

Time Partition 0



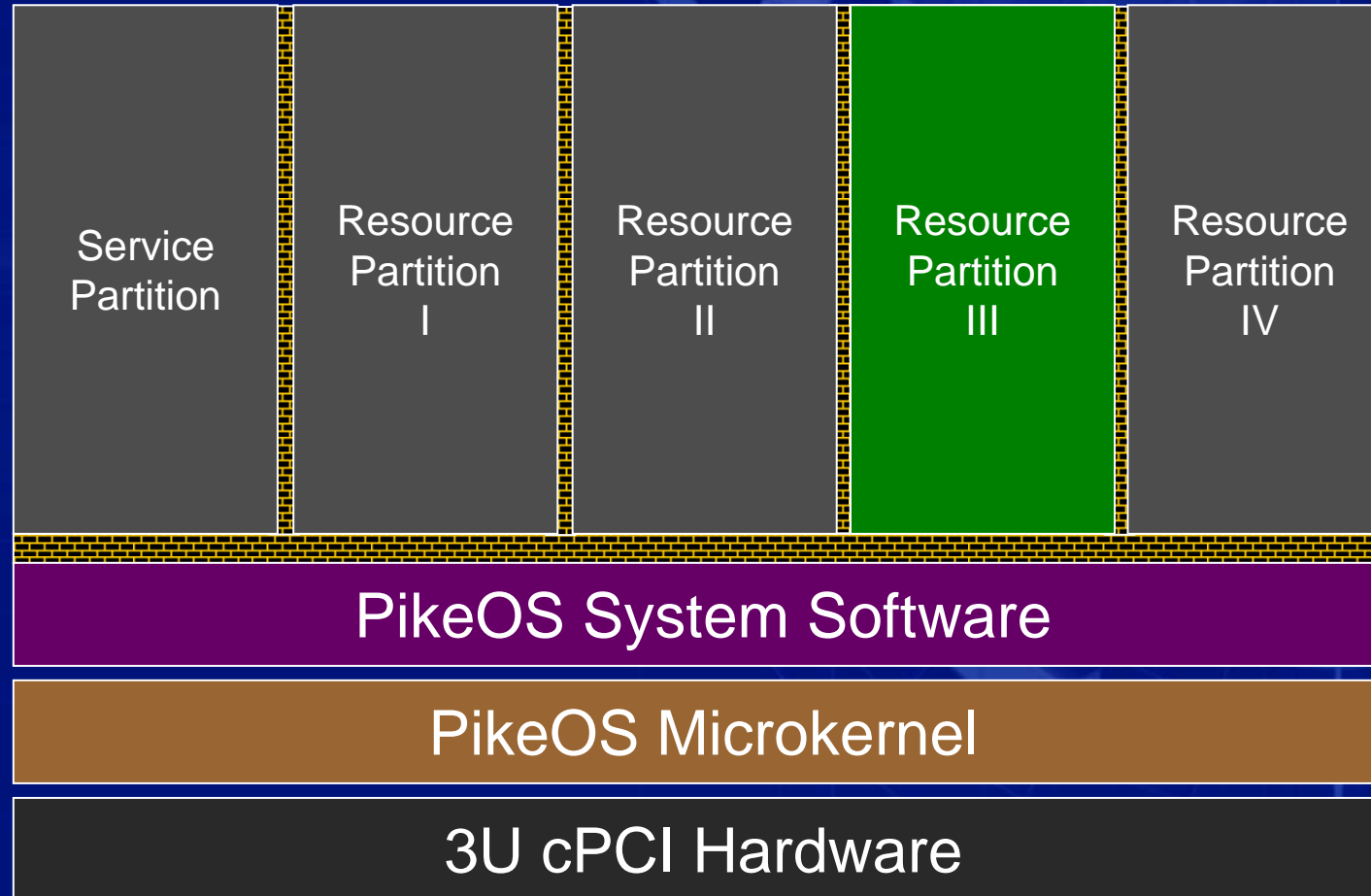
Partitionierung und Scheduling

Time Partition 1

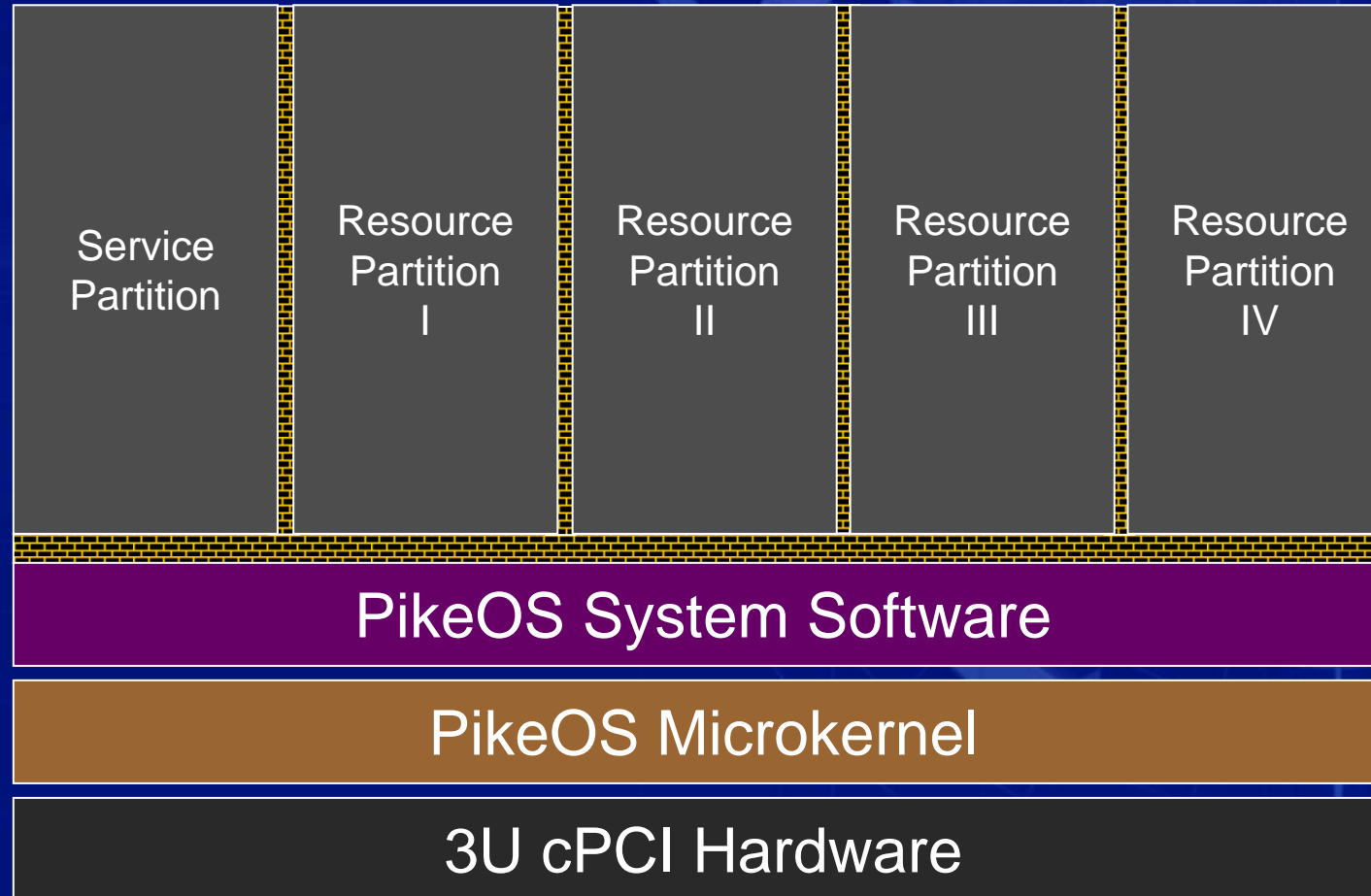


Partitionierung und Scheduling

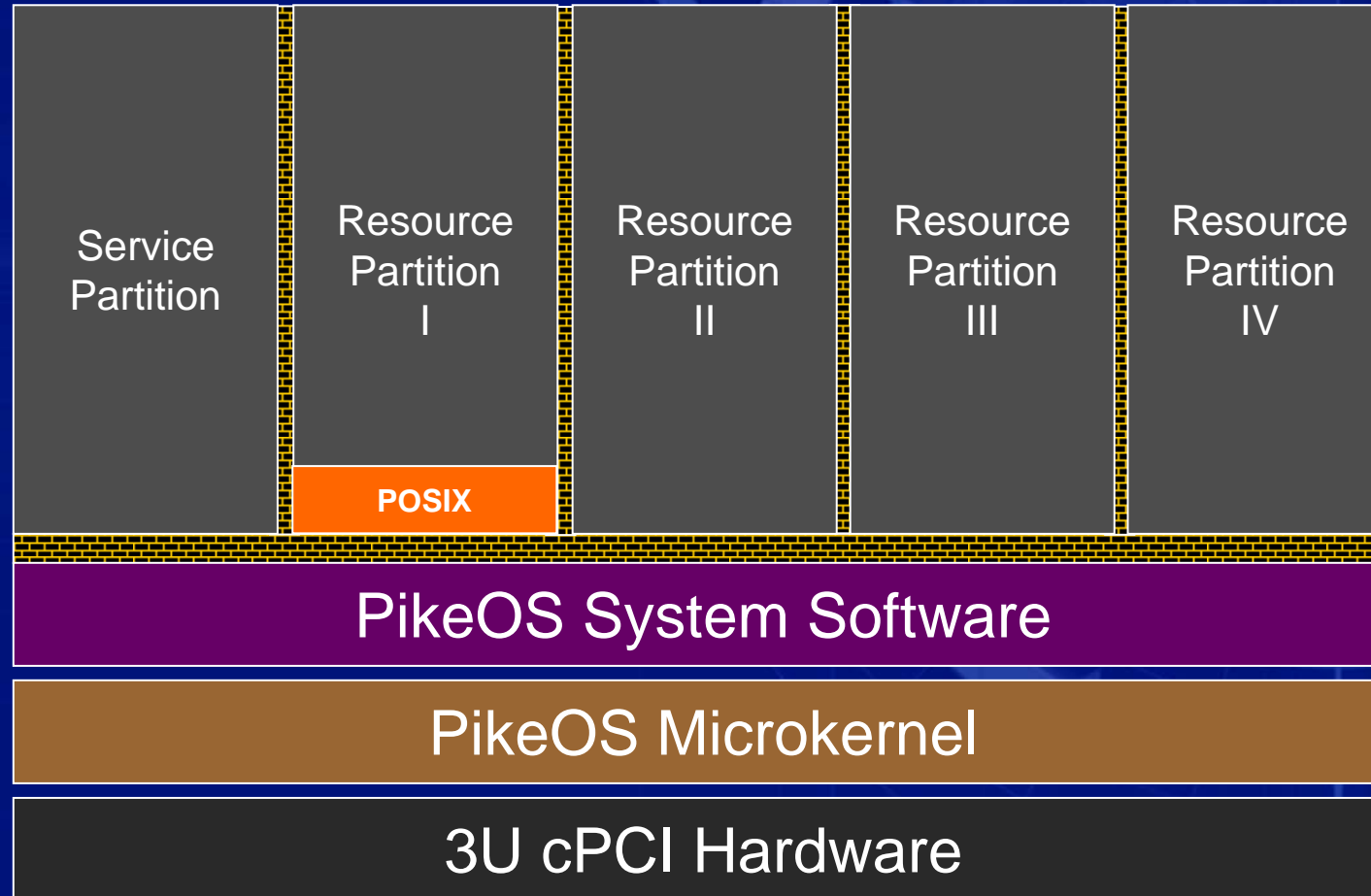
Time Partition 2



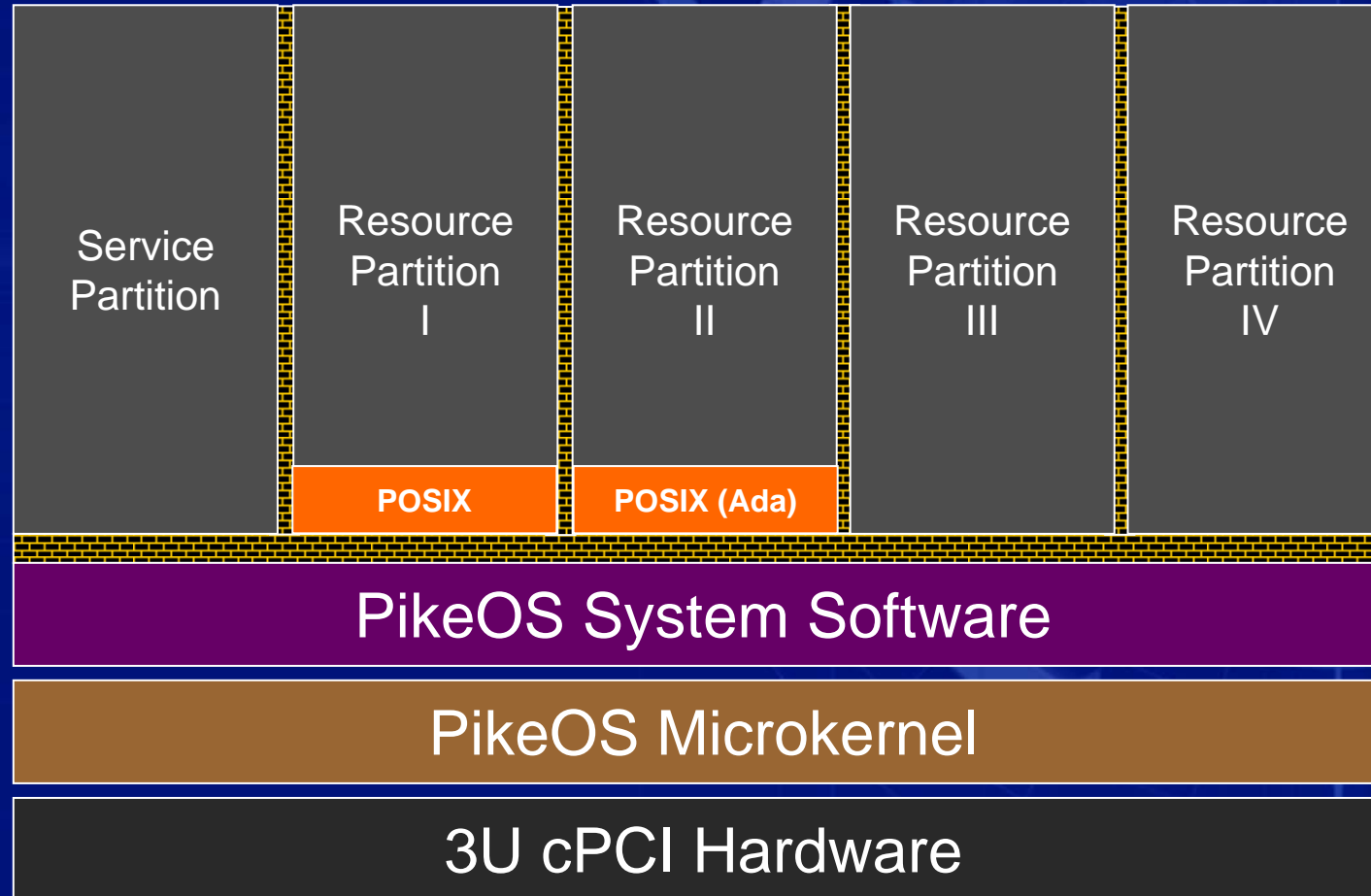
Konzept der Personalities



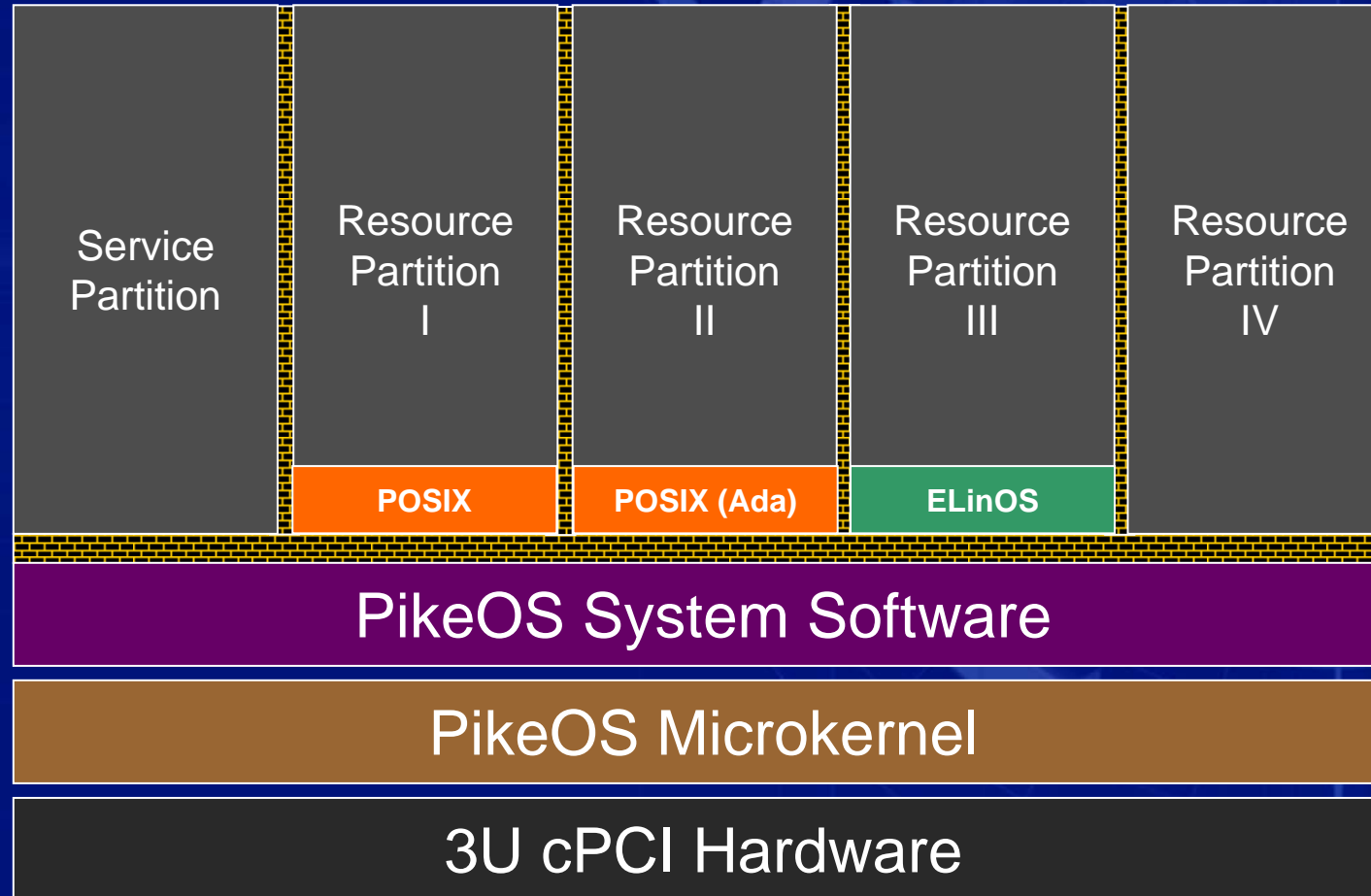
Konzept der Personalities



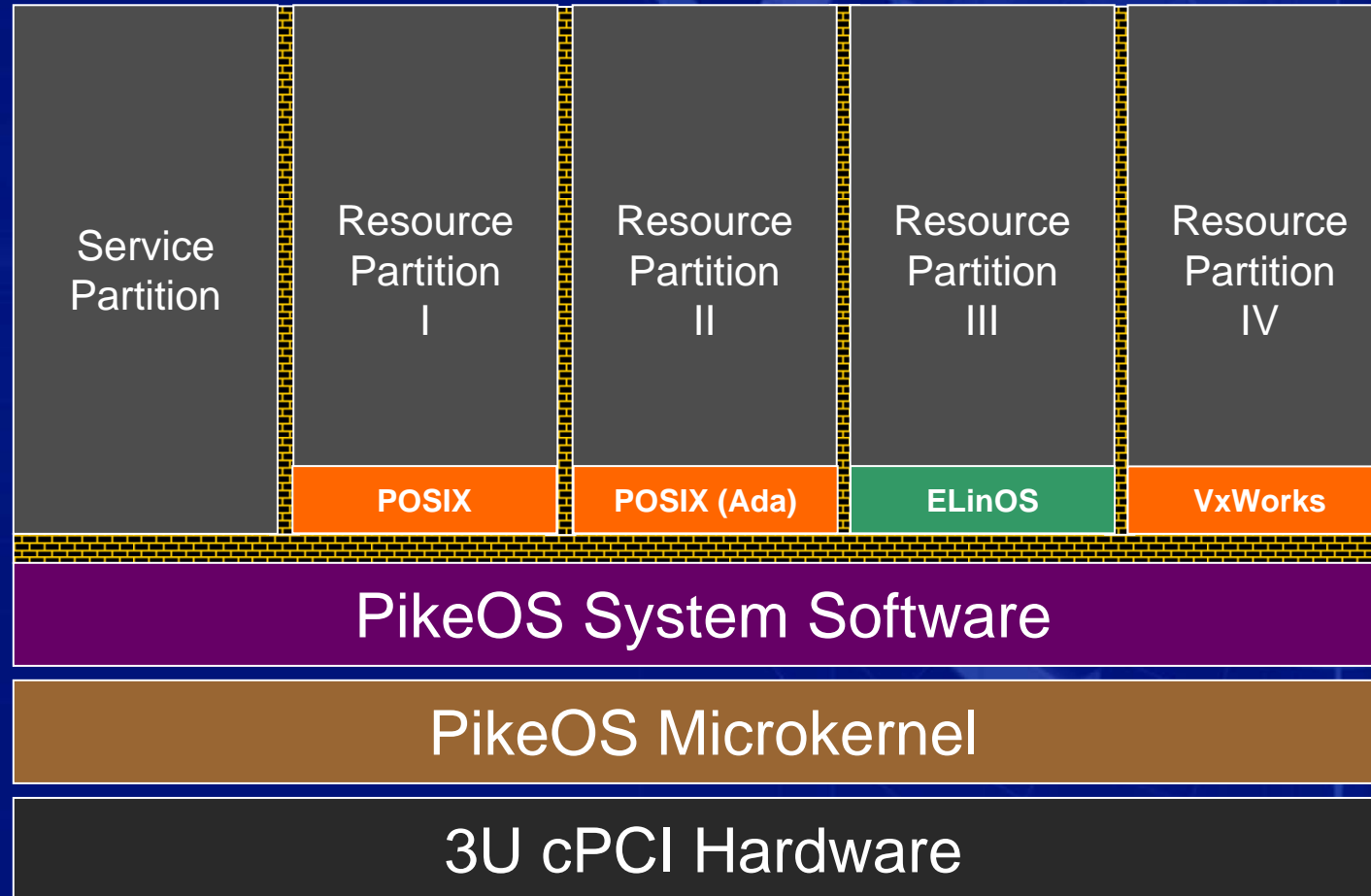
Konzept der Personalities



Konzept der Personalities



Konzept der Personalities



Kommunikation unter PikeOS

- Inter Process Communication
 - Port Kommunikation
 - Kommunikation ist synchron, unidirektional und ungepuffert
 - Kommunikation zwischen Threads verschiedener Resource Partitionen möglich
 - Event Kommunikation
 - Kommunikation ist synchron
 - Threads können Event Counter nutzen
 - Shared Memory Objects
- Shared Memory File System
- Netzwerk mit unterschiedlichen Protokollen und Interfaces (z.B. TCP/IP auf Ethernet)

Health Monitoring und Exception Handling

- Health Monitoring Klassifizierung
 - Module Error Level
 - Partition Error Level
 - Process Error Level
- Handling des Error Codes auf Applikationsebene oder entsprechend der Einträge in System, Module und Partition Health Monitor Tabelle
- Registrierung von Short und Full Exception Handler zur Verarbeitung durch Threads

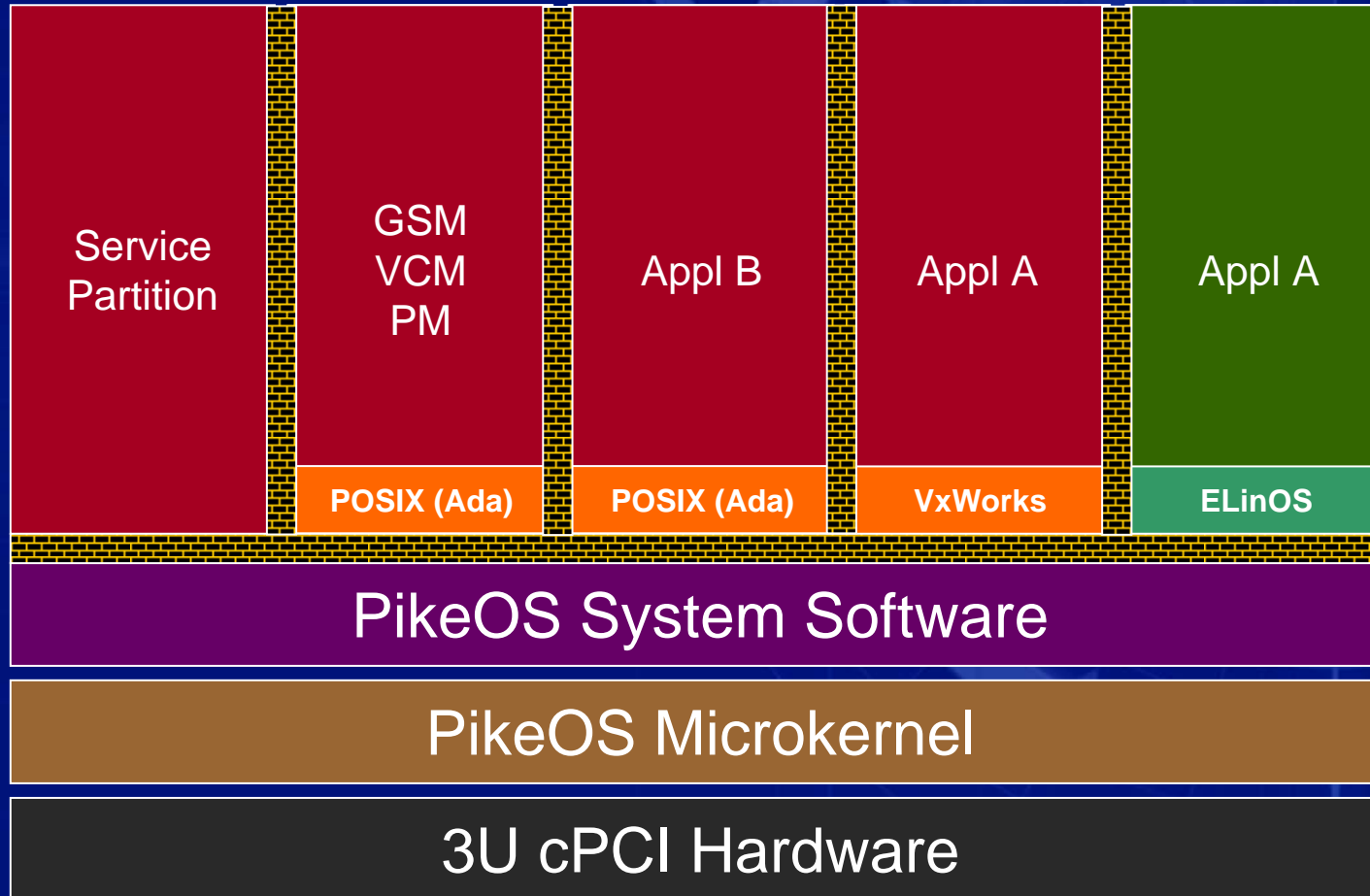
Überblick



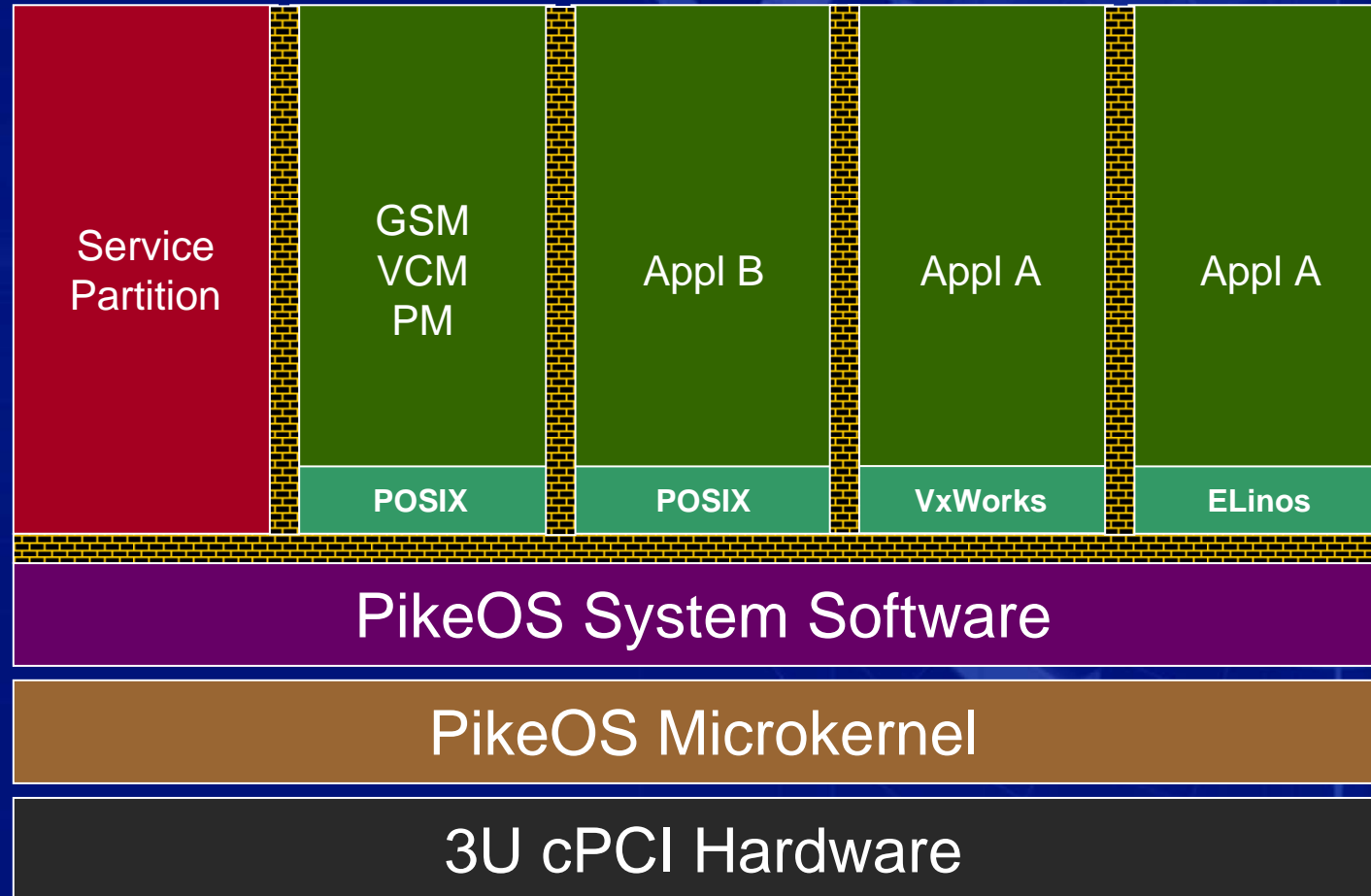
- Vorüberlegungen
- Konzept des FMS/MMS Demonstrators
- Status des Demonstrators
- Ergebnis und weitere Nutzung

FMS: Flight Management System
MMS: Mission Management System

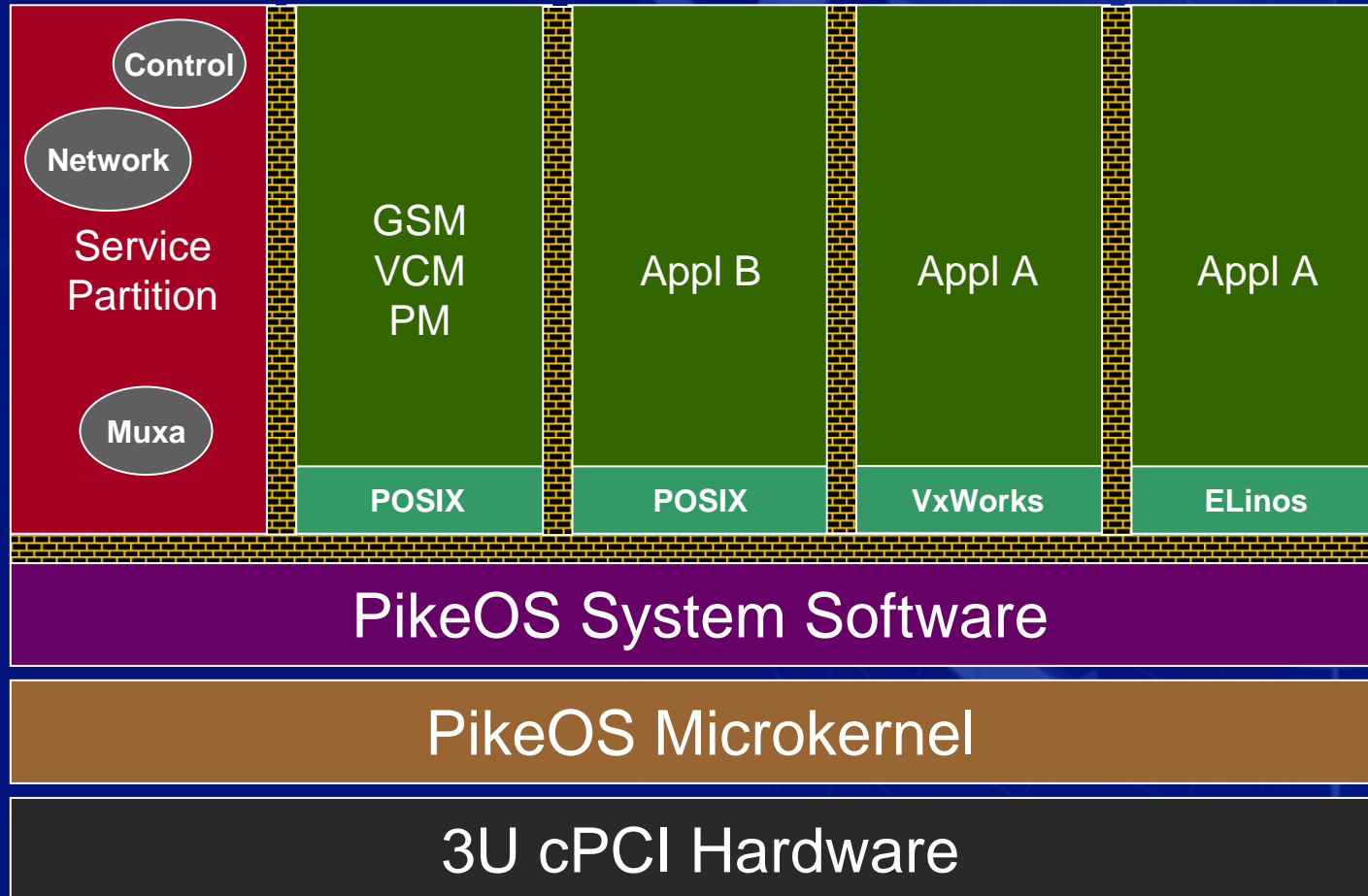
Demonstration der implementierten Funktionalitäten



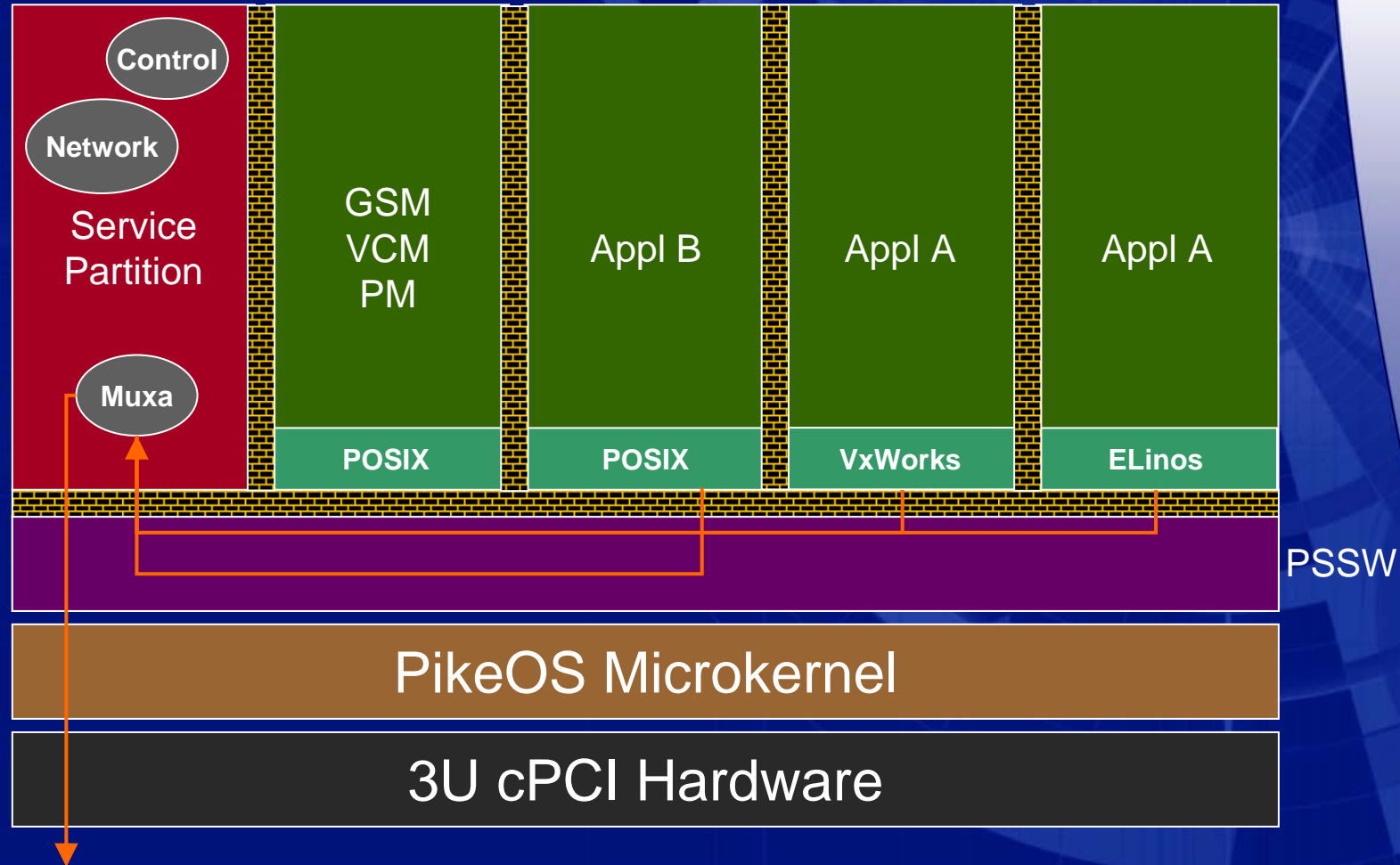
Demonstration der implementierten Funktionalitäten



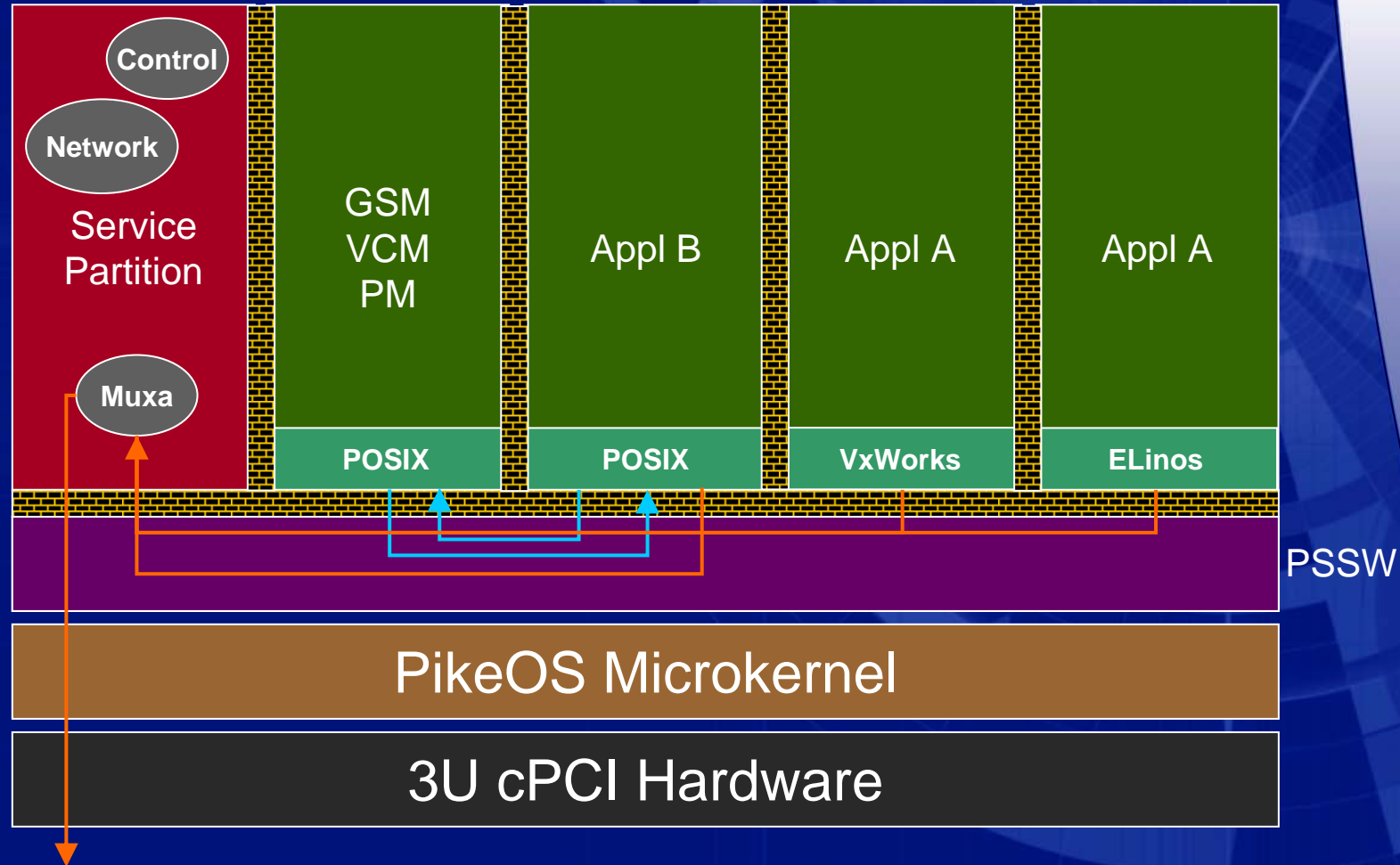
Demonstration der implementierten Funktionalitäten



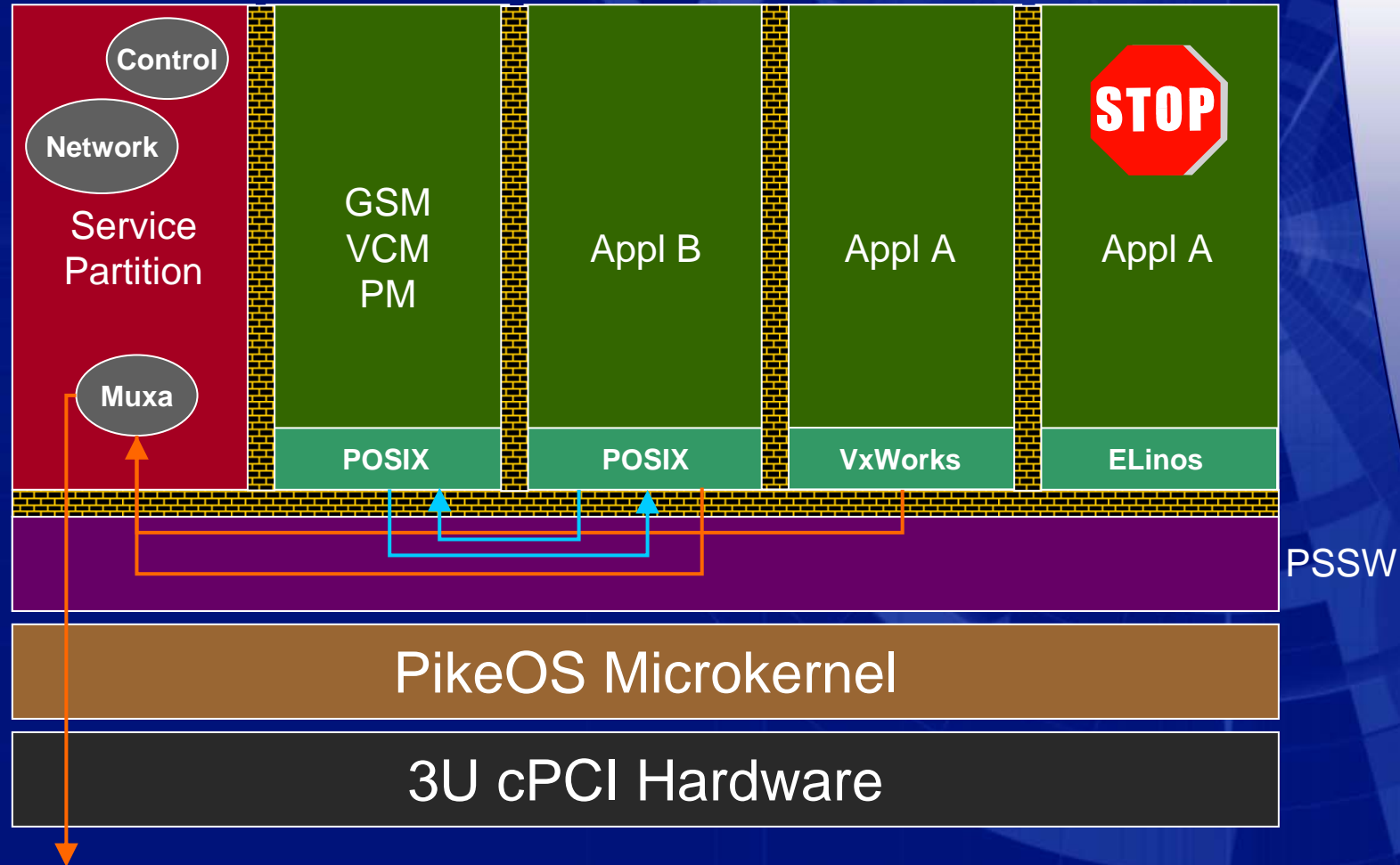
Demonstration der implementierten Funktionalitäten



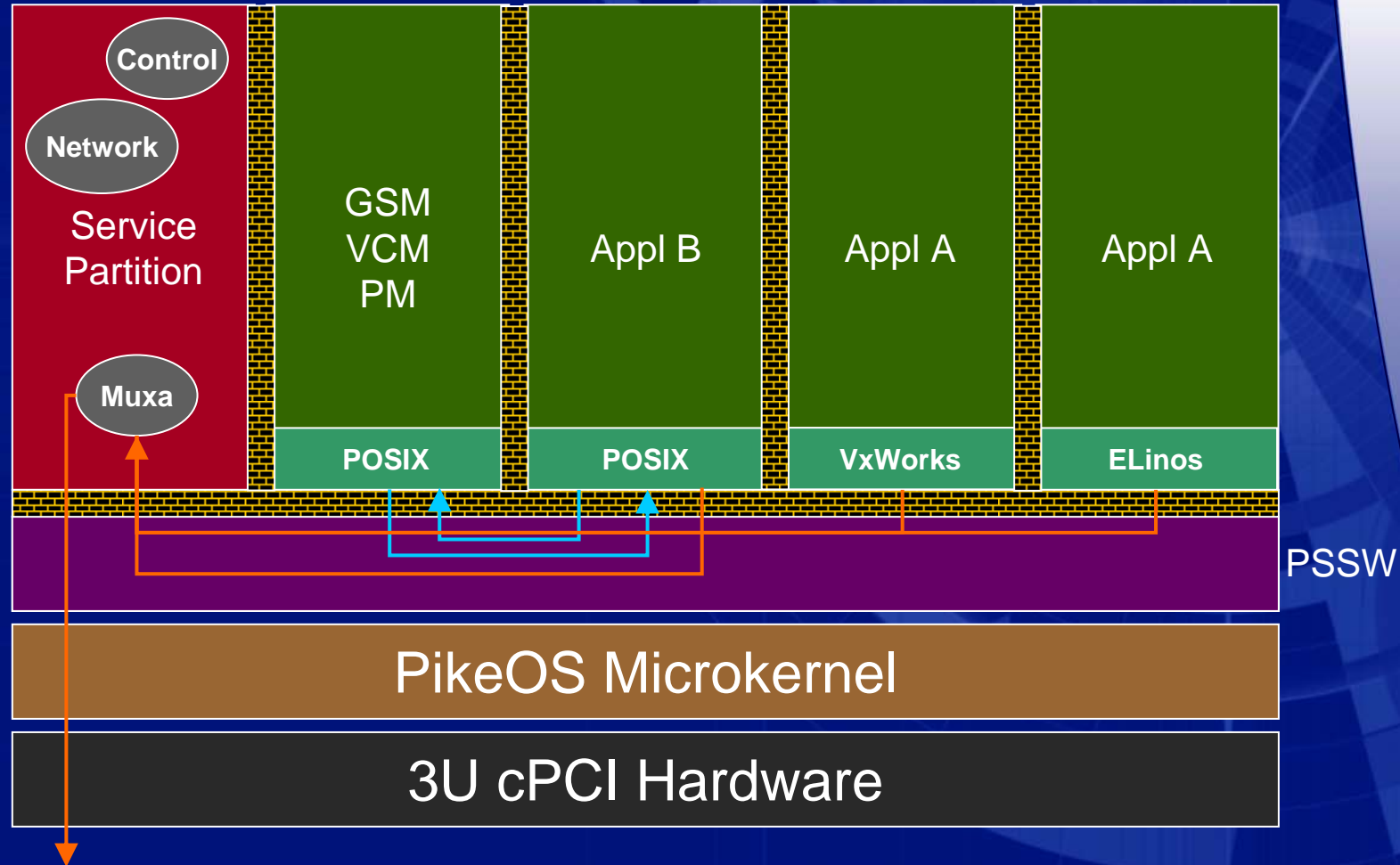
Demonstration der implementierten Funktionalitäten



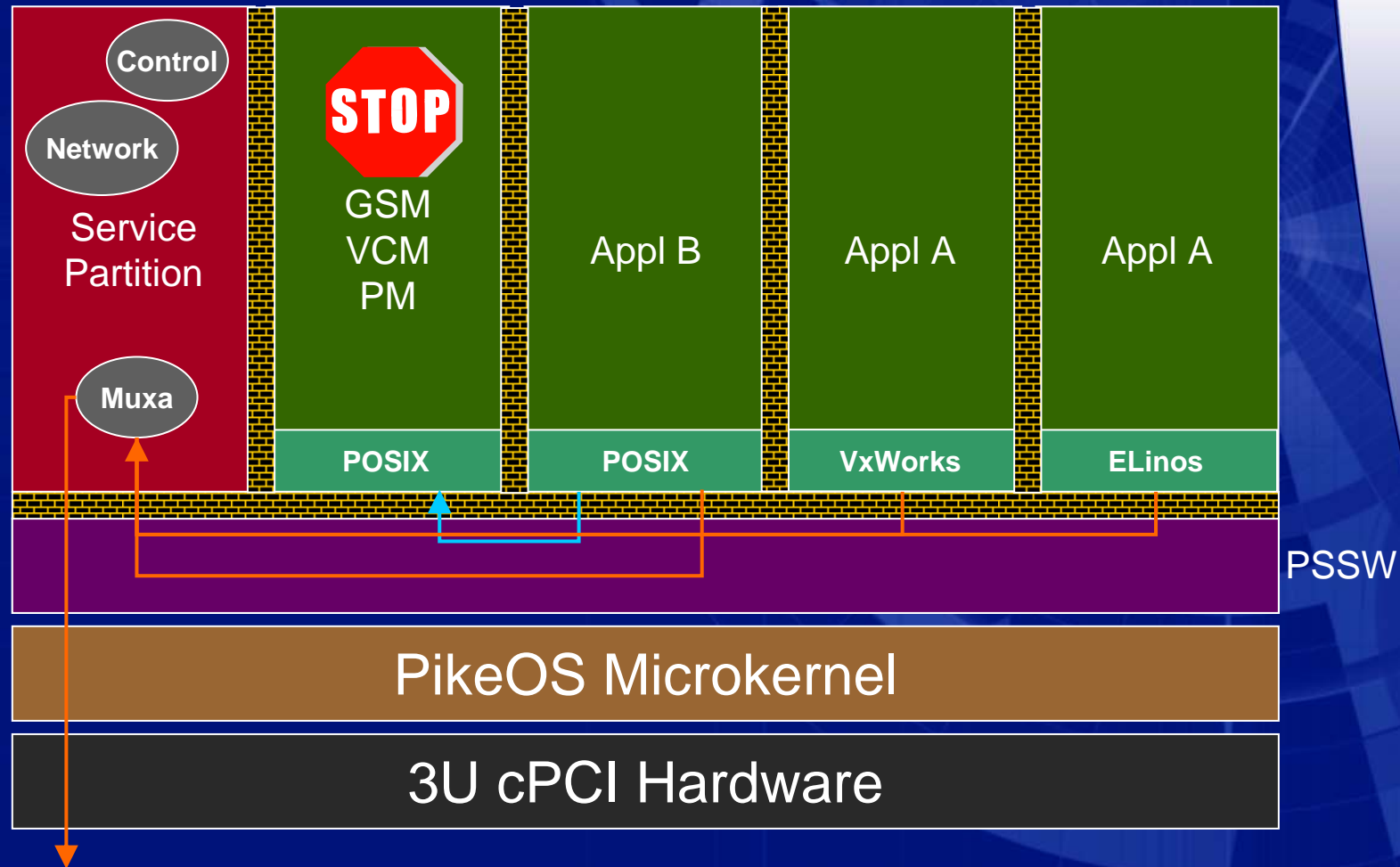
Demonstration der implementierten Funktionalitäten



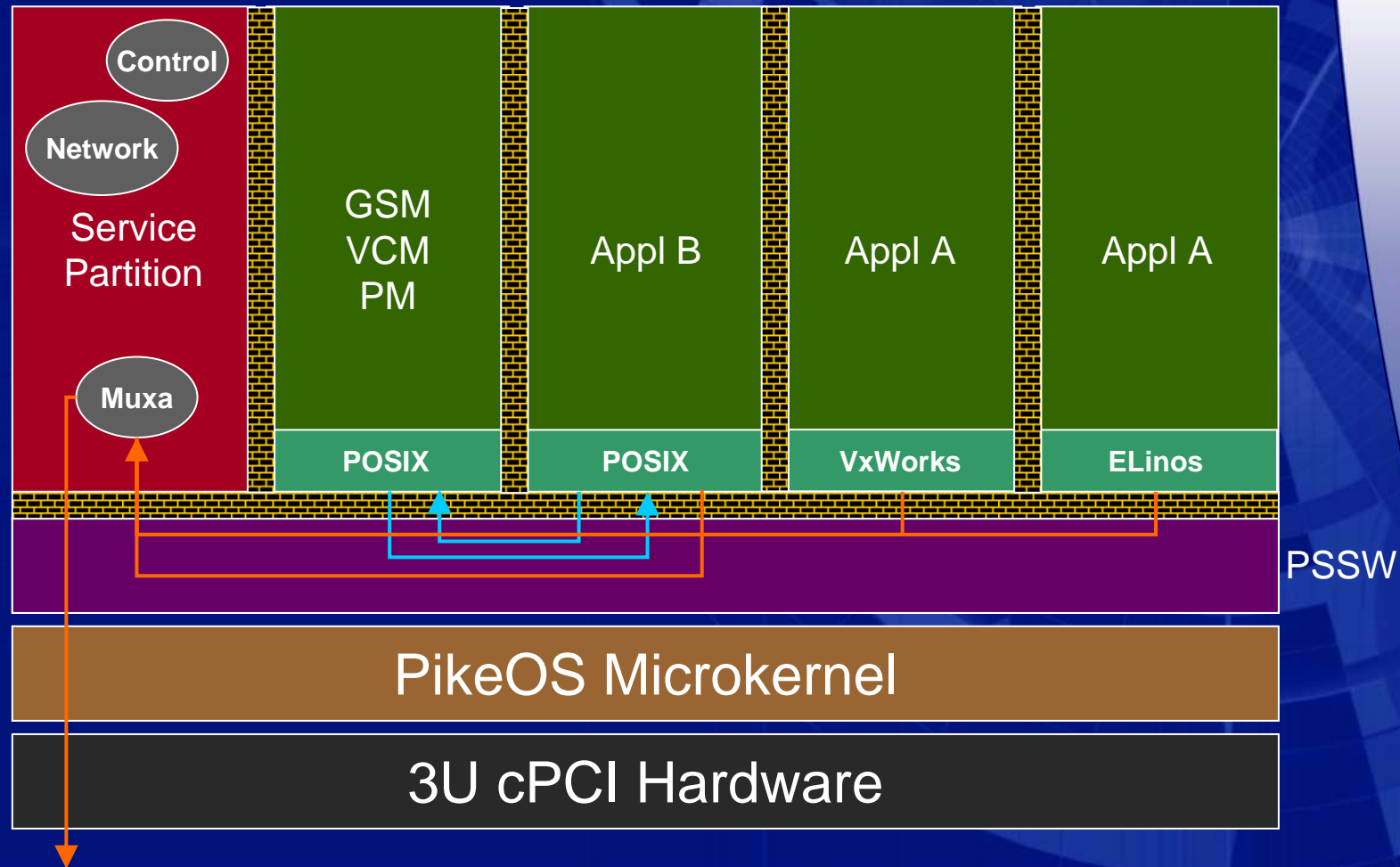
Demonstration der implementierten Funktionalitäten



Demonstration der implementierten Funktionalitäten



Demonstration der implementierten Funktionalitäten



Überblick



- Vorüberlegungen
- Konzept des FMS/MMS Demonstrators
- Status des Demonstrators
- Ergebnis und weitere Nutzung

FMS: Flight Management System
MMS: Mission Management System

Ergebnis und weitere Nutzung

- Ergebnis der Demonstration
 - Unabhängigkeit der Partitionen gezeigt
 - Korumpieren des Kernels auch durch Device Driver nicht nachweisbar
 - Portierung bestehender Software durch Personalities mit geringen Aufwand
 - Portierung des Systemmanagements und der Kommunikation nach STANAG 4626 von LynxOS
 - Portierung der Applikationen A und B von UNIX und VxWorks
 - Konfiguration Laufzeitumgebung durch Tools nur zum Teil möglich, Konfiguration per Hand notwendig und fehlerträchtig

Ergebnis und weitere Nutzung

- Weitere Nutzung EADS
 - Multiboard Demonstrator mit realer Flugsoftware auf Basis der Implementierung
 - Prototyping zur Erzeugung der Systemtabellen auf Basis bestehender Toolkette
- Weitere Arbeiten bei SYSGO
 - Vereinfachung des Microkernels
 - Verbesserung der Toolunterstützung
 - Zertifizierung nach DO178B Level A