EB corbos and the L4Re microhypervisor: Open-source automotive safety

Alexander Much, Michael Hohmuth, Adam Lackorzynski
2018-09-19, Vancouver, Linaro Connect 2018
About EB

**Technical competencies**
EB’s technical core competencies are development of automotive-grade (software) products and engineering services.

**Employees**
More than 2200 employees worldwide. Spans three continents and ten countries.

**Consistent growth**
Average growth (CAGR) > 10 %

**Global presence**
Development and business offices in Austria, China, Finland, France, Germany, India, Israel, Japan, Romania and USA.

**Continental AG**
Wholly owned, independent subsidiary of Continental AG.

**100+ million**
Over 90 million vehicles on the road and 1 billion embedded devices.
What we do

Automated driving
- Hardware and software products for development, test, visualization, and validation.
- Key software components to bring automated driving functions and systems to serial production.

Vehicle infrastructure
- AUTOSAR standard
- Single- & multi-core OS
- Functional Safety OS
- Embedded Security
- Automotive networks, e.g. Ethernet

Connected car
- Intelligent big data analytics & online diagnostics
- Scalable backend infrastructures
- Cyber security solutions plus modular add-ons by Argus
- Software updates over-the-air

User experience
- Navigation client for connected use cases
- Electronic horizon provider enabling map-based ADAS functions
- Model-based development of multimodal user interfaces
- Augmented reality solutions

Consulting services
- Consulting services for Functional Safety and Software Architectures
- Lean Software Development
- Established agile processes

Verification and validation
- End-to-end testing of complex embedded software systems
- Test concept development
- Independent verification and validation of software systems
Interesting times...

- Machine learning
- Crowd-sourced data
- System of systems
- Third party access

- Personalization
- Shortened development cycles
- Evolution after SOP
- New topics
  - new business models
Mobile on wheels or wheels on mobile?

What comes first?

We need to completely re-think the E/E architecture:

• Domain or zonal architectures

• Centralized computing units

• High-speed, reliable and dependable networking

• Connected vehicle within infrastructure eco-systems

Cloud and mobile first!

Source: https://pxhere.com/en/photo/1064249, CC0 Public Domain
Phone and cloud vs. vehicle

What needs to be „more“ secure?

Most prominent answer: „Of course, my car!“

People don’t realize:
• How many security solutions are in today’s phones
• Cloud and phones set the „state-of-the-art“
• ... not cars!

Evolution of E/E architectures

### Domain Architecture
- Signal-based communication
- System of ECUs
- Predictable communication
- Function orientated topology

### Centralized Architecture
- Central computing nodes
- Mix of signal based and service orientated communication
- Partly centralized functions
- Software upgradeability

### Zonal Architecture
- IP/Ethernet communication
- Centralized applications / functions
- Computing power for AD and AI
- Anything anywhere (sensors/actors)
- Architecture follows software / system demands
Building blocks of the next architecture

EB corbos and the L4Re microhypervisor: Open-source automotive safety

HPC = High performance controller

Horizontal deployment of functions

Vehicle API / basic services / information layer

Real time and sensor/actuator layer

Computing layer

Back-end

Every information anywhere – enables horizontal deployment of services and updating service. → But needs to be controlled for safety and security reasons
EB corbos
Safety, security and performance
EB corbos – The architecture

- New CPU-intensive (safety-relevant) functions: e.g. sensor fusion
- Novel user functions: e.g. App Store
- Reuse of existing vehicle functions from Classic AUTOSAR (SWCs)
- Secure startup, authentication
- Safety-relevant vehicle functions, monitoring of performance partitions

Performance partitions
- Adaptive AUTOSAR
- POSIX OS
- Virtual machine

Security partition
- Classic AUTOSAR
- AUTOSAR OS
- Trusted Execution Environment
- Trusted OS
- AUTOSAR Safety OS

Hypervisor
- Secure Boot
- Performance cores
- Safety cores

High-performance computer
EB corbos – The architecture (II)

Performance partitions
- App
- EB corbos AdaptiveCore
- EB corbos Linux
- POSIX RTOS
- EB corbos Hypervisor

Security partition
- App
- Trusted Execution Environment
- Trusted OS
- EB tresos Safety OS

Safety partition
- App
- EB tresos AutoCore

High-performance computer
Secure Boot
Performance cores
Safety cores

Tools
- EB tresos Studio
- Configuration
- Code generation
- EB corbos Studio
- Application development
- Integration and deployment
- Logging and debugging

Software
EB tresos
EB corbos
Services
3rd party

Hardware (SoC)
EB corbos AdaptiveCore

EB corbos and the L4Re microhypervisor: Open-source automotive safety

EB corbos AdaptiveCore

Application

Runtime for adaptive applications

Adaptive application

Services

Communication management
ara::com/rest/dds*

Diagnostic management

Network management

Persistency

Update & configuration management

Signal-2-service mapping*

Platform health management

Cryptography*

Identification & Access management*

Platform health management

Execution management

Log & Trace

Hardware acceleration*

Log & Trace

Platform health management

Execution management

Cryptography*

Persistency

Identity & Access management*

Network management

Communication management
ara::com/rest/dds*

Platform health management

Hardware acceleration*

Execution management

Log & Trace

Persistency

Malfunction management

Application

EB corbos Hypervisor

EB corbos Linux

OS

POSIX RTOS

Future content*

3rd Party

Alternatives

EB corbos

Tools

Generic

HW-depend.

Future content*

2018-09-19 | Linaro Connect 2018 | Public | © Elektrobit Automotive GmbH 2018 | All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Distributed safety management

**Vehicle functions partition**
- Container
  - Vehicle function
    - Virtual resources
- Execution manager
- Health manager
- Diagnostic manager
- Persistence manager

**Privileged partition**
- Container
  - Vehicle function
    - Virtual resources
- Virtual resources
- Health manager
- Physical resources

**Classic AUTOSAR components**
- Health control
  - WDG
  - Lockstep Safety OS

**Adaptive AUTOSAR on Linux**
- Hypervisor
  - Bootloader

**Core**
- Monitor
- Control
Platform security layers

- Control flow integrity
- Virtual address space separation
- ASLR, sanitizers, etc.

- Resource access control
- Intermediate address space separation
- 1st-stage MMU

- Scheduling domains
- Resource constraints

- Hardware resource separation
- Physical address space separation
- 2nd-stage MMU

- Processes
- Containers
- Operating systems
- Hypervisor

- Classic μC
- HSM
- Performance cores
- Performance μP
- Secure engine
- Switch

- HSM (EVITA medium)
- HIS SHE support
- Crypto accelerators
- Life cycle management
- Hardware access protection
- Crypto accelerators
- 3 core logic (Secure, Public & PKA)
- Dedicated RAM/ROM (key material)
- eFuses
- DoS prevention
- VLAN tagging
- Static ARP tables
- Monitoring ports

2018-09-19 | Linar Connect 2018 | Public | © Elektrobit Automotive GmbH 2018 | All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
**Hypervisor use-cases**

**ECU Consolidation**
Increasing capabilities of nowadays performance controllers enable suppliers to consolidate multiple in-car applications to one single device.

**Network Separation**
Growing Car-2-X connectivity requires secure separation of out-bounded connections to the in-vehicle network.

**Mixed Criticality Systems**
Virtualization brings in the key technology to build fail operational software systems with mixed safety integrity levels.

---

**Your benefit**
EB corbos Hypervisor
Based on the L4Re microhypervisor
Noteworthy L4Re features

**Isolation**

- Capabilities as references to kernel (and user-land) objects
  - Provides information hiding (local naming) and access control
  - Enables reasoning about isolation and freedom from interference
  - No capability to shared object
    → No way to communicate or interfere

- Designed to even allow preventing sharing 2nd-class kernel objects (allocators ...) and invisible architectural state (not 100 % there yet...)

**Real-time**

- Real-time per-CPU scheduler: Fixed priority round robin
  - Support for thread-group budget scheduling planned
  - WFQ (non-RT) also available
  - Cross-CPU thread / VCPU migration supported

- Short critical sections w/ IRQs off, preemption points
- Fine-granular wait-free locking
  → Excellent interrupt-response times

- No cross-CPU shared state in critical paths, no big kernel lock
  → Excellent scalability
Noteworthy L4Re features (II)

Virtualization

• Hardware-assisted virtualization
  – Untrusted (user-level) virtual-machine monitors (VMMs) for platform emulation
  • uvmm: Tiny VMM for Linux guests. Upstream ARM Linux “just works”
  • l4-kvm: Uses Qemu/KVM in a Linux guest to provide platform for Windows guests (x86 only)
• Also available: Paravirtualization with L4Linux
  – A user-mode Linux kernel running on L4Re

Microapps

• Microapps: Native L4Re applications
  – Small TCB: no dependency on any rich OS, no Dom0
  – No dependency on VMM
  – No virtualization overhead
• POSIX subset for microapps: L4Re Runtime Environment
  – Supports libc, C++ library, pthreads, etc.
  – Natural extension of kernel API with useful OS abstractions, e.g. for address-space management
Noteworthy L4Re features (III)

I/O virtualization

- Device pass-through to VMs or driver microapps
  - DMA security via IOMMU (ARM: WIP)
- Native drivers and multiplexing for various buses and devices
  - PCI, serial console, AHCI, framebuffer
- Virtual networking among VMs supported
  - Virtual Ethernet switch or p2p connection
  - Virtual socket connections
- Virtio supported

Where to get it?

- Go to www.kernkonzept.com/download.html
- Or www.l4re.org
- Early access at github.com/kernkonzept

Licensing?

- (Mostly) GPL version 2
- Commercial licenses: Dual licensing capability
  - Require CLA for contributions, essential for attracting investments needed for certification
  - Also, a customer requirement in Automotive
- Kernkonzept serves as maintainer & gatekeeper for contributions
Solutions for interesting times

- **Machine learning**
  - Crowd-sourced data

- **Personalization**
  - Real-time capable
  - Shortened development cycles

- **High-assurance security**

- **Long-term maintenance and operations**

- **Automotive safety up to ASIL-D**
  - Based on open-source and established, well-proven implementations

- **System of systems**

- **Third party access**

- **Evolution after SOP**

- **New topics**
  - new business models
Get in touch!

alexander.much@elektrobit.com
michael.hohmuth@kernkonzept.com
adam.lackorzynski@kernkonzept.com
www.elektrobit.com