Scalable and Flexible
Software Platforms for
High-Performance ECUs

Christoph Dietachmayr
Sr. Engineering Manager, Elektrobit
November 8, 2018
Agenda

New E/E Architectures and High-Performance ECUs

Non-Functional Aspects: Safety | Security | Cloud

Adaptive AUTOSAR / EB corbos:
Where Safety Meets Performance
### Major Market Trends

<table>
<thead>
<tr>
<th>Trends</th>
<th>Impact on E/E Architecture</th>
<th>Impact on SW Architecture</th>
<th>Adaptive AUTOSAR base technology for</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E-Mobility</strong></td>
<td><img src="image" alt="Shrink of powertrain reduces hw complexity" /></td>
<td>• Software complexity increase</td>
<td>• safe</td>
</tr>
<tr>
<td><strong>Automated Driving</strong></td>
<td><img src="image" alt="High data volumes" /></td>
<td>• Central car computer approach</td>
<td>• secure</td>
</tr>
<tr>
<td><strong>Mobility Service</strong></td>
<td><img src="image" alt="Car-to-X connectivity" /></td>
<td>• Fail operational systems</td>
<td>• flexible</td>
</tr>
<tr>
<td></td>
<td>![Update and Upgrade capability (over the air)]</td>
<td>• Availability</td>
<td>• up to date</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Service oriented architecture (SOA)</td>
<td>• high performant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Holistic security approach</td>
<td>central car control units.</td>
</tr>
</tbody>
</table>
Evolution of E/E Architectures

**Today**
- Signal based communication
- System of ECUs
- Predictable communication
- Function orientated topology

**Tomorrow**
- Central computing nodes
- Mix of signal based and service orientated communication
- Partly centralized functions
- Software upgradability

**Future**
- IP/Ethernet communication
- Centralized applications/functions
- Computing power for HAD and AI
- Anything anywhere (sensors/actors)
Scalable and Flexible Software Platform for High-Performance ECUs

Potential HPC Architecture – SOP 2019

Infrastructure Software (Operating System and Middleware)

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

### Performance Partitions
- **Functions**
  - Adaptive AUTOSAR
  - POSIX OS
  - Virtual machine
- **Hypervisor**
- **Secure Boot**
- **Performance cores**

### Security Partition
- **Functions**
  - Classic AUTOSAR
  - AUTOSAR OS
  - Trusted execution environment
  - Trusted OS
- **Virtual machine**

### Safety Partition
- **Functions**
  - Classic AUTOSAR
  - AUTOSAR Safety OS

**High-performance computer**

© Elektrobit (EB) 2018
Communication in a Service-Oriented Architecture
Signal Based vs. Service Based Communication

Signal based

- **Sender**
  - Just transmit (implicit availability info)
  - Doesn’t matter whose receiving
- **Receiver**
  - Just Listen
  - Doesn’t matter whose sending

Service Based

- **Sender (= Service Provider)**
  - Offer service & maintain subscriber ①
  - Send message to subscriber
- **Receiver (= Client)**
  - Find service & subscribe to service ②
  - Receive message ③
Signal Based: New Services Require Changes to Gateway

- Gateway
  - Detect crash
  - Lock while driving
  - Key
  - Open via smartphone
  - Ego data
  - Provider Speed
  - Authentication
  - Open/close door

- Services:
  - Service 1
  - Service 2
  - Service 3
  - Service 4

© Elektrobit (EB) 2018
Service Discovery Phase – Find & Match Services

- Detect crash
- Lock while driving
- Key
- Open via smartphone
- Ego data
- Provider Speed
- Authentication
- Open/close door
- Services can be distributed - flexible on various ECUs
- No dependency or changes in Gateway necessary
- Service 1
- Service 2
- Service 3
- Service 4
- ...
Additional Non-Functional Requirements Arise

- Security
- Cloud Connectivity
- Functional Safety
Normal Operation of Distributed Service

1oo2D system

Fault tolerant Ethernet

Sensors/Actuators

1 out of 2  disabled  critical  non-critical

Service 1
Service 2
Service 3
Diagnostics

Service 5
Service 6
Service 4

Service 1
Service 2
Service 3
Diagnostics
One Channel Failed

1 channel active
- Still operational or handover to driver
- Internal recovery
- Failure recovery

Fault tolerant Ethernet

Sensors/Actuators

1 out of 2
- disabled
- critical
- non-critical
Reconfiguration of Services

Requirements for reconfiguration

Req. 1: Services can be dynamically relocated

Req. 2: Sensor/actuators are redundant or accessible via network as a service

1 out of 2

disabled
critical
non-critical

Fault tolerant Ethernet

Sensors/Actuators

1oo2D system

Service 2
Service 1
Service 3
Diagnostics

Service 2
Service 1
Service 3
Diagnostics

Service 5
Service 6
Service 3
Service 1

Service 2
Service 1
Service 3
Diagnostics

© Elektrobit (EB) 2018
Safety Monitoring Concept

Vehicle Functions Partition
- Container
  - Vehicle Function
    - Virtual Resources
- Container
  - Vehicle Function
    - Virtual Resources
- Container
  - Vehicle Function
    - Virtual Resources

Privileged Partition
- Physical Resources
- Health Manager
- Adaptive AUTOSAR on Linux
- Diagnostic Manager
- PERSISTENCY MANAGER
- Virtual Resources

Classic AUTOSAR Components
- WDG
- Health Control
- Adaptive AUTOSAR on Linux
- Classic AUTOSAR
- Lockstep Safety OS

Bootloader
- Hypervisor
- Execute manager
- Health Manager
- Diagnostic Manager
- PERSISTENCY MANAGER
- Virtual Resources

Core
- Safety Core
- Safety Core

Security Layer Concept

- Processes
- Containers
- Operating Systems
- Hypervisor

Control Flow Integrity
Virtual Address Space

Resource Access Control
Intermediate Address Space Separation (1st-Stage MMU)

- Scheduling Domains
- Resource Constraints

Control Flow Integrity
Hardware Resource Separation
Physical Address Space Separation
2nd-Stage MMU

- Classic µC
- HSM
- Performance Cores
- Performance µP
- Secure Engine
- Switch

Hardware

- HSM (EVITA medium)
- HIS SHE support
- Crypto Accelerators
- Crypto Accelerators
  3 Core Logic (Secure, Public & PKA)
  Dedicated RAM/ROM (key material)
  eFuses
- DoS prevention
- VLAN Tagging
- Static ARP tables
- Monitoring Ports
Cloud Connectivity: Use-Case Remote Update

Architectural Principles:
• Central external connection
• Distribution of updates across multiple ECUs

Supporting Features:
• Coordinated A/B Update across ECUs
• Application containerization
• Secure communication
• Layered security architecture

Company infrastructure and IT systems to be included
Adaptive AUTOSAR – EB corbos
Where Safety Meets Performance

EB corbos AdaptiveCore
AUTOSAR Adaptive Platform for safe and secure applications

EB corbos Linux
Container based Linux

EB corbos Studio
Integrated development environment for EB corbos products

EB corbos Hypervisor
Micro-Kernel based type1 hypervisor to run multiple (different) operating systems on one single CPU
EB’s High-Performance Computer (HPC) offering...

One-Stop Shop Software Provider for Next Generation ECUs

Performance Partitions:
- EB corbos AdaptiveCore
- EB corbos AdaptiveCore
- EB corbos Linux
- EB corbos Hypervisor
- EB corbos Linux
- EB corbos AdaptiveCore

Security Partition:
- EB tresos AutoCore
- Trusted Execution Environment
- Trusted OS

Safety Partition:
- EB tresos AutoCore
- EB tresos Safety OS

Tools:
- EB tresos Studio
- Configuration
- Code Generation
- EB corbos Studio
- Application Development
- Integration & Deployment
- Logging & Debugging

Software (SoC):
- Hardware
- Performance Cores
- Secure Boot
- Safety Cores
- EB tresos Studio
- EB corbos Studio
- Configuration
- Code Generation
- Application Development
- Integration & Deployment
- Logging & Debugging
EB corbos Studio

Integrated Development Environment for EB corbos Products

- **Application Development**
  Develop applications according the AUTOSAR Adaptive platform.

- **Integration**
  Integrate software components with EB corbos AdaptiveCore and other products from EB corbos product portfolio.
EB corbos Studio – Features

Application development

Create and modify application design
- Specify the application for the AUTOSAR Adaptive platform
- Add and modify required and provided services
- Use system data model usually provided by the OEM

Implement and build the application
- Implement the application in C or C++

Run, Debug and Test the application
- Run and debug the application on the PC in a virtual machine running EB corbos Linux, or directly on the target
EB corbos Studio – Features

Integration

Integrate applications
- Define process and startup parameters for the applications
- Define dependencies to other applications

ECU integration
- Create configuration files for EB corbos AdaptiveCore
- Configure network connectivity
- Define ECU and vehicle states
Conclusion

- Define Performance controllers build on top of a software system architecture on the basis of AUTOSAR

- Adaptive AUTOSAR alone is not the solution for all performance controllers

- Classic and Adaptive AUTOSAR form a foundation for complex automotive software systems

- System properties must be ensured thru system architecture, particularly for Safety / Security / Reliability
Thank you for your attention!

Questions?

Christoph Dietachmayr
Sr. Engineering Manager, Elektrobit
Christoph.Dietachmayr@elektrobit.com