Developing a mixed-critical AUTOSAR Adaptive ECU with Safety & Security by Design

Dr. Sergey Tverdyshev, SYSGO AG
Dr. Timo Kerstan, Vector Informatik GmbH
Content

- Introduction
- MILS Concept
- AUTOSAR Adaptive Introduction
- Integrating AUTOSAR Adaptive Platform
- Assurance
About SYSGO

• Founded in 1992
• Leading European Provider for Embedded Operating Systems
• Offices in Germany (Mainz, Ulm, Rostock, Hamburg), France (Paris, Lyon), Czech Republic (Prague) and UK
  • 80% Engineers have safety and security certification competences
  • Distributors around the world
• Core competences
  • operating systems and HW-SW platform integration
  • safety & security cert. for avionics, automotive, railway, space, medical, IoT
• Independent entity of the THALES Group since 2012
SYSGOVector Joint Venture

- A joint venture between Vector and SYSGO has been founded at the end of 2017
  - Create a consistent and uniform AUTOSAR Adaptive Platform
- This JV enables the closest possible cooperation to smoothen the solution regarding technical aspects and business model.
- Vector is taking over sales and distribution for the common solution.
- Both companies’ sw-tools will merge into a single customer facing IDE
- Further certification expertise will be delivered by SYSGO as an industry leading expert in certification processes
- Product training on application suite and integration suite
About the speaker

• Dr. Sergey Tverdyshev  
  Director Research and Technology at SYSGO

• Relevant highlights
  • PhD with the topic  
    “Formal verification of gate-level Computer Systems”
  • 11+ years experience in domain of  
    development and analysis of operating systems and hypervisors
  • 7+ years experience in safety certification
  • 6+ years experience in security certification
  • 30+ publications in domain RTOS, hypervisors,  
    security, and formal methods
  • Founder of the MILS.community
Typical E&E Architecture of a modern vehicle

- E&E is driven by automated driving, Car-2-X, dynamic deployment of customer applications
- ECUs consolidated into a smaller number of ECUs in a service oriented architectures
- Trend goes to having two types of ECUs:
  - I/O control units, e.g. actuators, sensors
  - Performance or **domain controls units**
    - e.g. for data fusion, trajectory computation, intra-vehicle and external communications
- Needed high amount of computing power, fail-operational functionality, high assurance while maintaining security via update over the air and security by design.
Example for Heterogeneous Hardware

- **Renesas R-Car H3**
  - CPU cores
    - Quad Cortex-A57 cores
    - Quad Cortex-A53 cores
    - Dual lock-step Cortex-R7 cores
  - Various IO devices

- **Xilinx Zynq MPSoC+**
  - CPU cores
    - Quad Cortex-A53 cores
    - Dual Cortex-R5 cores
  - FPGA
  - Various IO devices

- **Challenge**
  - execute applications of different criticalities and ASIL while designing and preserving segregation between apps
Quick Example Automotive:
Secure Android-based Head-Unit + Payment Services
Safety and Security for Mixed-Critical Systems

Challenge: Resource sharing

- **Resources**
  - CPUs
  - Memory, IO memory
  - Files, drivers, devices, buses

- **Safety**
  - Integrity, availability
  - Isolation, application errors, fail safe

- **Security**
  - Integrity, availability, confidentiality
  - Possible side channels via shared resources
  - Resources and API are attack surface

Challenge: Time sharing

- **Time**
  - CPU cycles
  - Time effects of accessing shared resources, e.g. buses

- **Safety**
  - Availability, deterministic behavior, meeting deadlines
  - Right balance between time- and event-triggered tasks

- **Security**
  - Availability, confidentiality
  - Possible timing side channels via shared resources, e.g. caches, busses
  - Time is the attack surface

**Needed Solution**

**Resource Partitioning**

**Needed Solution**

**Time Partitioning**
Content

• Introduction
• MILS Concept
• Adaptive AUTOSAR Introduction
• Adaptive AUTOSAR Platform
• Assurance
Brief MILS History: 1975 - 2018

1975
- Saltzer and Schroeder “The Protection of Information in Computer Systems”

1981
- Rushby “Design and Verification of Secure Systems”

1992-2005
- IMA concepts, safety focus

2005
- Hardware virtualisation Intel VT-x, AMD-V

2007
- US Separation Kernel Protection Profile

2007-...
- US-based studies
- The OpenGroup standardisation
- Hardware was not in the focus

2012
- SKPP sun-setted by US government: HW complexity, monolithic eval, politics

2012
- EURO-MILS
- MILS Architecture
- Protection Profile for Separation Kernel
- High Assurance methods
- Compositional cert
- D-MILS
- Research on configuration and analysis of distributed MILS

2014
- Founding of the MILS Community

2015
- 1st annual MILS Workshop

2016
- CITADEL: adaptive MILS
- Preservation of assurance

2017
- certMILS: compositionality certification of system based on MILS
- IEC 62443 and Common Criteria

2018
- Joining CCUF
- Founding CCUF SK Technical Community/Working Group
MILS.COMMUNITY
selected members and research partners
MILS is a high-assurance security architecture that supports the coexistence of untrusted and trusted components, based on verifiable separation mechanisms and controlled information flow.

https://de.wikipedia.org/wiki/Multiple_Independent_Levels_of_Security
MILS Architectural Approach

- **Low-criticality Partition**
- **Medium-criticality Partition**
- **High-criticality Partition**

**Application plane**

**Refinement**

**MILS induced abstraction**

**Resource plane**

- **MILS Architecture**
- **MILS Platform (Separation Kernel)**
- **Hardware (CPUs, memory, and devices)**

Network

Actuator
MILS Architecture

AVAILBLE ON
https://mils.community/
System Development: from design to run-time

Ease of Integration
- Personalities create VMs à la carte for modularized functionality
- Integrated toolchains for gapless development (e.g. Matlab, SIMULINK, SCADE)

Security
- Often, designs already contain some separation of duties
- MILS concept is a structured approach for secure partitioning and controlled information flow

Safety
- Partitioning ensures safety, e.g. fault isolation, non-interference
- Successfully implemented & deployed as IMA concept in the avionics sector, e.g. Airbus A380, A400 and automotive


© SYSGO GMBH · PUBLIC
Content

- Introduction
- MILS Concept
- **Adaptive AUTOSAR Introduction**
- Adaptive AUTOSAR Platform
- Assurance
Why AUTOSAR Adaptive?

Drivers and technical needs

• **Main drivers**
  • Highly automated driving
  • Car-2-x applications
  • Vehicle in the cloud
  • Increased connectivity

• **Technical needs**
  • New communication technology and concepts
  • Application separation
  • Partial update of services (multiple versions of a service must be supported)
  • Portability and interoperability
  • Safety and security needs
  • High-End CPUs
  • Support of HW acceleration
Why Autosar Adaptive?

Close the Gap between Autosar Classic and Linux

<table>
<thead>
<tr>
<th>Real time Requirements</th>
<th>High, in the range of micro-sec</th>
<th>Mid, in the range of milli-sec</th>
<th>Low, in the range of sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Criticality</td>
<td>High, up to ASIL-D</td>
<td>High, at least ASIL-B</td>
<td>Low, QM</td>
</tr>
<tr>
<td>Computing power</td>
<td>Low, ~ 1000 DMIPs</td>
<td>High, &gt; 20,000 DMIPs</td>
<td>High, ~ 10,000 DMIPs</td>
</tr>
</tbody>
</table>
What is Autosar Adaptive?

Architecture

Each application runs in its own protected address space.

Access to platform functionality via libraries.

Communication via implementation specific inter process communication (IPC).
What is Autosar Adaptive?
Communication inside and between ECUs

Dynamically established communication path(s)
Content

• Introduction
• MILS Concept
• Adaptive AUTOSAR Introduction
• Adaptive AUTOSAR Platform
• Assurance
• **Adaptive MICRO SAR** is using the separation kernel architecture provided by PikeOS.

• **Types of Separation:**
  - Resource partitioning: Container for Application execution. Statically assigned resources guarantee availability of memory and I/O when needed
  - Time partitioning: tasks assigned to partitions are executed in a time partition frame to give you predictable real time performance

• Separation is base for “Freedom from Interference” as required by the ISO 26262
Integrating Adaptive Platform

Overview PikeOS Partition Architecture

Adaptive Platform
- DM
- LOG
- PERS
- COM
- EM

Adaptive Application Container
- App1
- App2
- App n

User Defined Partition
- Linux App
- Linux App
- Linux

PikeOS System Software

PikeOS Separation Kernel

Adaptive IPC
Integrating Adaptive Platform

Inter Process Communication

Adaptive Platform
- DM
- LOG
- PERS
- COM
- EM

Adaptive Application Container
- App1
- App2
- App n

User Defined Partition
- Linux App
- Linux App
- Linux

PikeOS System Software

PikeOS Separation Kernel

Adaptive IPC

ETH
Integrating Adaptive Platform

Dynamic Loading Executables
Integrating Adaptive Platform

Perspective of an Application Developer

- **Application**
  - Multi-threaded
  - Execution states
  - Manifest contains platform related information (recovery action, dependencies to services or libraries)
  - Instance configuration contains application specific static information (variant, options, …)

- **Interfaces**
  - ara::com for communication with adaptive services (basic services and user applications)
  - PSE51 is the usable OS API subset
  - The Adaptive AUTOSAR Foundation clusters (Execution Management, Persistency, etc.) are available via direct APIs

```
App1
POSIX Process

Manifest

INIT:

RUN:
Thread
Thread
Thread

SHUTDOWN:

PSE51
ara::com
Direct API

C++ Stdlib
Adaptive AUTOSAR Services
Adaptive AUTOSAR Foundation

POSIX OS
```
Content

• Introduction
• MILS Concept
• Adaptive AUTOSAR Introduction
• Adaptive AUTOSAR Platform
• Assurance
Assurance

- Assurance for Integrated Adaptive Platform shall be compositional, i.e. it shall be a sum of
  - Assurance for the platform
  - Assurance for individual applications
  - Assurance for integrating architecture of the ECU

- You don’t want to develop assurance of integrated systems as a whole including it all components, i.e.
  - by all means avoid monolithical assurance architecture ==> it is expensive

- This sum is the desired focus of system integrator
  - for functional and assurance architectures
Assurance

- We applied MILS approach to achieve compositional assurance

- The MILS approach enables us build up assurance case from
  - PikeOS Separation Kernel is certified and used as the MILS base
  - Adaptive platform is assured as a user-applications with special privileges
  - Each application is tested and/or certified independently of anything and only for the level needed for the application
  - Assurance of the integrating architecture is derived based on non-interference (i.e. separation) properties of the MILS base
Summary

• We have presented how to design a mixed-critical AUTOSAR Adaptive Platform ECU
• We showed critical and enabling components for mixed-criticality
  • i.e. MILS concept and separation kernel
• We explained how an assurance case is built for a mixed-critical for ECU based on AUTOSAR Adaptive Platform

• Visit SYSGO:
  
  **Main Booth: Hall 4-534**

  or in the

  **Safety & Security Area: Hall 4A-303**
Acknowledgment

• This work has been partially funded by the European Union Horizon 2020 research and innovation programme under grant agreement No 731456.

• This work has been partially funded by the German Ministry of Education and Research within the project BaSyS 4.0
Thank you for your attention!

More information on www.sysgo.com