Return of Experience: “Deploying virtualization in the vehicle”
Genivi AMM, Munich, May 2019
F. Walkembach, VP Marketing & Alliances
Company · Brief Overview

Founded 1991
Since 2012 part of the Thales Group

Local Facilities in Germany, France, and Czech Republic

Global Distribution and Support Network including Europe and the Pacific Rim

PikeOS® Certified RTOS and Hypervisor supporting many embedded VMs and Guest OS

ELinOS® Late 90’s SYSGO pioneered the use of Linux in the Embedded Market

Alliances JV with Vector on AUTOSAR adaptive
Agenda

• Customer use case
  • Top level requirements
  • Pain points
  • Why using a hypervisor

• Technical work packages
  • Guests
  • Hw-virt
  • Boot process
  • Selection of hardware

• Technical constraints
  • Boot process
  • BSP creation with Serial and Ethernet driver
  • Training-, Support services

• Potential issues
Use cases for a V2X solution

- Emergency Vehicle Warning
- Intersection Movement Assist
- Stationary Vehicle Warning
- Enhanced Traffic Jam Assist
Top level requirements for a V2X box

- Real time data processing
- Respecting safety considerations with early warning
- Data security & privacy
- Vehicle bus interface
- Network IEEE 802.11p / ITS G5
- GNSS
- Always localized
- Generic ECU requirements
- Generic quality requirements
- Safety certification levels

Acc. To researchgate.net
Unknown certification details at start of project

• End customer specification not precise at the start of the project
  • Unknown certification elements
  • Not yet agreed SW architecture
• But, a flexible approach in using a hypervisor
  • Elements that require ASIL certification can be added later in the project
  • The customer initially planned using PikeOS native and POSIX but
  • added new Linux partition later
  • Via Linux the need for HW-virt was introduced
  • Adding safety requirements enforce the change of only ONE partition, that go through the certification process
Why did customer prefer using a hypervisor?

- Realtime capabilities (Mix of lower and higher RT capabilities)
  - Configurable Time Schemes, WCET guaranties, handling different levels of RT
- Strong separation
- Certifiable at a later stage of the project
- Small footprint
- Flexible enhancement of other guests
- Embracing OpenSource
- Use of legacy code in VM
- Secure Boot, secure Update
- Reduction of time to market
  - Building a prototype in Linux guest for V2X quickly
  - Critical blocks can be moved to PikeOS native or POSIX
  - Limited effort for creation of a POC and
  - preparing the final architecture in parallel
Top work packages

• **Creation of BSP**
  • Adjust RAM physical addresses
  • Adapt GPIO settings
  • Integrate interrupt controller
  • Integrate timer
  • Support HW virtualization
  • Support PAE
  • Testing within SYSGO test framework

• **Development of drivers**
  • Serial and Ethernet

• **Configure and set up partitions**
  • Based on use cases
  • Boot time optimisation
Security approach with certified crypto

- Security by separation with Independent Common Criteria Assessment
- Enables SW architecture design with dedicated partitions allocated to sensitive functions
  - Controlled information flow
  - Privilege allocation per partition
  - Enable OTA for single component (when vulnerabilities are discovered)
- Enables Secure Sharing of Resources
  - Secure storage, Crypto Algorithm
- Multi-level Containerisation
  - Strong hypervisor based separation
  - Optional docker containers in a partition
- Mono-directional communication flow
Introduction

Traditional approach

- **Limited access control capabilities**
- **No STRONG separation between Applications**
- **Large code base, impossible to audit**
- **No separation between drivers**

- **User (user mode)**
- **Kernel (supervisor mode)**
- **SoC / Custom Hardware**
- **Monolithic Linux kernel**
- **Target Boot Loader**
  - CPU1
  - Serial
  - CAN
  - Hardware
  - Ethernet
  - HSM
  - Storage
Introduction

Security by separation

- User level Drivers
- Strong Static Separation
- Shared Drivers
- Mixed Guest OS (POSIX, Linux, Native)
- Small code base, ISO 26262 & CC Certified
- Kernel (supervisor mode)
- PikeOS Safe and Secure Separation Kernel and Hypervisor
- Statically Defined communication Channels
- SoC / Custom Hardware

Target Boot Loader

CPU, Serial, CAN, Hardware, Ethernet, HSM, Storage
Selected Architecture for V2X

Advantage:
- Real time
- Flexible
- Security with certified crypto algorithms
- Managed boot process
Potential issues

• **Performances (Latency)**
  • The more partitions are created the safer the system will be
  • Each partition adds latency in the data flow

• **Resource sharing design**
  • Using an hypervisor adds flexibility and enables late design decisions within an exception
  • Resource sharing control is defined by dedicated driver
  • Specification need to be ready at a very early stage

• **SOC Core allocation**
  • Time Partitioning Allocation
  • Interferences will exist
Thank you for your attention!

More information on www.sysgo.com