Safe Rendering: Maximum cost saving approach

09.05.2019 Arwed Richert
Introduction

Luxoft Automotive?
All about SW
3,000+ employees
40+ clients (>40% OEMs)
139 MUSD revenue (FY 2018)

Arwed Richert?
Started 1997
Leading R&D teams HMI Platform & Tools / Safety
ISO 26262 – automotive Safety Standard – ensuring drivers life

Goal: increase quality / reduce failure rate SW & HW

- Beyond normal ASPICE processes
- Adds methodologies, like
  - Static code analysis
  - MISRA C
  - 100 % test coverage

Standard includes risk classification scheme (ASIL-Levels):
- ASIL A (lowest integrity requirements)
- ASIL D (highest ones)

Telltale use case = ASIL B
Safe Rendering as cost driver

Typical Safe Rendering System architecture (IVI + IC) – with Safety partition

Royalties for:
• Safe Renderer
• QNX Graphics for Safety
• QNX OS for Safety
• QNX Hypervisor

Royalties for Safe components higher due to more dev efforts
## Major challenges

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>Rendering Synchronisation</th>
<th>Reusability</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Safe Renderer (SR) itself</td>
<td>• Variants (e.g. day / night)</td>
<td>• Multiple HMI FWs</td>
</tr>
<tr>
<td>• Safe BSP (graphics drivers)</td>
<td>• Animations</td>
<td>• Multiple OS</td>
</tr>
<tr>
<td>• Safe RTOS</td>
<td>• 3D</td>
<td>• Multiple MCUs</td>
</tr>
<tr>
<td>• Safe Hypervisor</td>
<td></td>
<td>• Specific Requirements</td>
</tr>
</tbody>
</table>

**Problem:**
- Royalties of Safety certified components
- 2 HMI instances rendering on the same screen
- Platform Dependency
## Possibilities to overcome

### Cost Driver

<table>
<thead>
<tr>
<th>Royalties for</th>
<th>Possible solution</th>
<th>Result / Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Safe Renderer (SR) itself</td>
<td>• Use a royalty free one (e.g. LSR(^1))</td>
<td>• All additional royalties for Safe SW eliminated</td>
</tr>
<tr>
<td>• Safe BSP (graphics drivers)</td>
<td>• Don’t use GPU driver</td>
<td>• Depends on platform (e.g. dedicated core)</td>
</tr>
<tr>
<td>• Safe RTOS</td>
<td>• Dedicated Core (Baremetal) &amp; Open Source OS</td>
<td>• Shifts part of OS to SR (e.g. MMU / GPU config)</td>
</tr>
<tr>
<td>• Safe Hypervisor</td>
<td>• As above &amp; Open Source Hypervisor</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) LSR = Luxoft Safe Renderer
Possibilities to overcome

Suggested System Architecture: Maximum achievable Royalty cost elimination – Safety on a dedicated core

Cost reductions by:
- Open source Hypervisor
- Linux
- No QNX Graphics for Safety
- Open source Luxoft Safe Renderer

→ Maximum achievable royalty savings realized

Alternatives:
- Safe AUTOSAR on R7
- Containerized System
## Possibilities to overcome

### Rendering Synchronisation & Reusability

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible solution</th>
<th>Result / Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 HMI engines rendering on 1 screen (QM HMI + Safe Renderer)</td>
<td>Let QM HMI render also Safety critical content</td>
<td>3D / animated telltales get possible</td>
</tr>
<tr>
<td>Multiple OS / Multiple MCUs</td>
<td>SR need to consist of a core library and a platform adaptation layer (e.g. SEooC)</td>
<td>Style switching issues solved</td>
</tr>
<tr>
<td>Multiple HMI Frameworks</td>
<td>SR should not be coupled to a specific toolchain</td>
<td>Dependency on platform eliminated</td>
</tr>
<tr>
<td>Specific Requirements</td>
<td>SR should be flexible</td>
<td>Increased Reusability</td>
</tr>
</tbody>
</table>
Luxoft capabilities

**Engineering Services**
- Teams of Safety Certified engineers available
  - Safety Managers
  - SW Engineers
  - Testers
  - ...
- Porting of (any) Safe Renderer
- Complete Safety Development

**Coworking / Consulting**
- System Architecture
- Safe Communication Path
- Safety Requirements
- Safety Management

**Luxoft Safe Renderer IP**
- Open Source
- Royalty free
- Works with any main QM HMI
- Maximum cost saving approach (dedicated core) or traditional mode
- Advanced functionality (Animations, ...)

www.luxoft.com