Software Defined AGL with SOAFEE
The Changing Automotive Development and Deployment Paradigm

- Centralised Automotive Compute Architecture
- Transition towards Software Defined Systems
- Ability to monetise after sale
Problem Statement
A Perfect Storm of Complexity

Software Complexity

- Transition towards ‘Software Defined’
  - More capabilities expressed in software
  - Volume of code in production vehicles going through exponential growth
- Software tied to ‘BSP’ through non-standard interfaces and characteristics
- Software is not portable
  - Requires huge amount of re-integration
  - Some OEM’s prefer to start again!
- Unable to amortise software cost over multiple production runs

Hardware Complexity

- Heterogeneous SoC’s becoming increasingly complex
  - CPU/MCU/GPU/NPU/…
- Multiple compute islands within the system that have different execution characteristics
- Difficult to the software and developers to make efficient use of all hardware available
- Software becomes tied to a specific hardware architecture and system topography
Researching a solution

Where has the problem been seen before?

How did the industry address the problem in their domain?

Can we embrace the same approach in our domain?
Cloud-native ++

- Functional Safety
- Realtime
- Heterogeneous Compute
- Accelerator aware
- IO aware
- ...

Arm
Scalable Open Architecture for Embedded Edge
Introducing SOAFEE

Scalable Open Architecture For Embedded Edge

An **industry initiative** to extend cloud-native software experience to automotive workloads, incorporating a **Special Interest Group (SIG)**

A **software architecture** which enables cloud technologies to be combined with automotive functional safety and real-time requirements

An **open-source reference software implementation**, for seeding open-source / commercial ecosystem.

Enabled by SystemReady, PSA, ...
Use-case: Radar proximity warning
Continental use-case

- Distance data from radar
- Static connection to sensor
- IP based communication with access control to inputs
- Audible warning control
Orchestrator Actor view of Continental use-case

Pub/Sub Topic
Message direction (pub or sub)
Configuration: consumed by orchestrator to configure TSN network through CNI
Payload: Contract between Publisher and Subscriber

Orchestrator knows how many publishers are on a ‘topic’ so can ensure bandwidth availability and configure TSN subsystem appropriately through CNI.

**Proximity Warning (Front Left)**
- radar: location: frontLeft
- distance: 2.4m

**Proximity Warning (Front Right)**
- radar: location: frontRight
- distance: 10m

Audio Topic
- frequency: 100Hz
- payload: 200 bytes
- deliveryLatency: 10us

Requirements
- Process incoming message within 100us
- Generate output to topic within 200us of message ingress

Payload
- consumed by orchestrator to configure TSN network through CNI
- Contract between Publisher and Subscriber
- Orchestrator knows how many publishers are on a ‘topic’ so can ensure bandwidth availability and configure TSN subsystem appropriately through CNI.

Audio Mixer
SOAFEE Cloud Native Architecture Vision

Framework for enabling mixed critical workload across cloud and vehicle

Mixed Criticality Aware Orchestrator

- QM/ASIL-B Service
- HAL
- OS

Standard Based Virtual Compute Environment & Tools

SOAFEE Framework

Cloud-Native Service deployment

- QM/ASIL-B Service
- HAL
- OS

- Container runtime
- Hypervisor

- Firmware
- High Compute CPU

- asil-D Service
- RTOS

- ASIL-D Service

= container
= monolithic
= optional
arm

SOAFEE Ecosystem
Members

- A framework defined through collaboration
- Representation from across the industry

More being added every week... Details at https://soafee.io/about/members/
Workshops and Whitepapers

Accelerating Software-Defined Vehicles through Cloud-To-Vehicle Edge Environmental Parity

In this white paper we look at what it means to apply cloud-native approach to automotive system development, focusing particularly on achieving environmental parity between cloud to edge execution environments. The paper explores the impact this approach has in accelerating the time to market of software-defined vehicles and the role that SOAFEE plays as key technology enabler.

Read the white paper

More details available at https://soafee.io/blog/
Software Defined AGL enabled by SOAFEE
Two routes forward

AGL as a SOAFEE compliant workload

- AGL as a containerised workload
  - Makes use of SOAFEE defined portability
  - All Open Standards based (VirtIO etc)
  - Aiming for binary compatibility
- Deployable to any SOAFEE compliant distro
  - Allows OEM’s to select middleware independent of workloads
  - Makes AGL more attractive to OEM’s
- Eases route to adoption by OEM’s
  - Not single sourcing for any component in system delivery
Two routes forward

AGL as a SOAFEE compliant middleware

+ SOAFEE is a software architecture expressed through standards
  - AGL can implement to these standards
  - Pull in the SOAFEE vision of mixed critical, real-time and safety
+ Ease deployment of third party workloads into the AGL ecosystem
  - For example Autonomous/ADAS functions through OpenAD Kit
  - Enable technology enabled by the SOAFEE ecosystem to deploy to AGL
SOAFAEE Blueprints with Project Apollo

Moonshot soon to be renamed – possibly Apollo
SOAFEE Blueprint

Container
Workload A

Container
Workload B

SOAFEE
(EWAOL or SOAFEE Partner Implementation)

Standards Compliant Firmware
(SystemReady)

Arm - based Hardware Platform

Solution SW Elements

Application Specific Workload(s)

Base SW Stack
• Cloud Native
• Containerized Workloads
• Orchestration
• Tooling
• Validated on Standard Firmware

Compute Platform
• An opportunity to showcase new platforms
Project Apollo (f.k.a. Moonshot)

+- DevOps in the Cloud
  - Native software development
  - Software integration and testing
  - Automated software testing
  - Enabled by existing standards, e.g., VirtIO, etc.

+- Deployment at the Edge
  - Container orchestration
  - Real-time workloads
  - Safety-critical workloads
  - Use de-facto standards such as Kubernetes
SOAFEE Integration Lab

- Provide reference implementation feature development (part of the reference implementation WG)
- Provide needed infrastructure to regression test the ref. implementation and ensure a stable platform that current and future Blueprints can use
- Validate Blueprint workloads against targeted HW platforms
- Prove the DevOps Workflow by leveraging the Blueprints as test and validation scenarios