Building Bridges with a common Data Middleware

Status of the PoC project after the Porto AMM:

Integration of an Onboard Sync Server into the vehicle

Christian Mühlbauer, Data Expert Team BMW AG
Data Centric Architecture. Motivation & Goals.

- **Model and technology agnostic**
- **Common architecture for data transfer and handling**
- **Common data domain models to interchange data**
- **On- & Off-board share common data architecture**
- **Efficient data transfer**

**Everywhere & Anytime**
- Increasing the availability of data
  - Data is available, even if single ECU’s are shutdown

**Cost efficient**
- Reduction of costs
  - Shorter feedback loops, faster reaction to regulations
  - Time-to-market for data-centric use cases

**Simplicity**
- Simplify interaction with vehicle on & off-board
- Data centric use cases on & off-board
Typical E/E architecture styles in automotive

Multi-bus gateway architecture

Domain controller architecture

Zonal architecture with centralized compute

Sensor/actuator

ECU

Gateway

Functional domain controller

Centralized generic compute

Zonal controller

How can a data-centric architectural approach based on a zonal architecture be implemented?

https://www.eetasia.com/identifying-e-e-architecture-requirements-for-autonomous-vehicle-development/
The basis: Data-Centric Vehicle APIs for Software-Defined Vehicle

Touchpoints

STEP 1
Data-Centric Vehicle API as Enabler

STEP 2
Simplify here…

STEP 3
…to benefit here!

PoC Focus

Services

Apple CarPlay
AOSP / AAOS
Containers
Cloud

POSIX
sPOSIX
OSEK
January 23: Our starting point

Key questions for a End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data ...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge ...
- ... can be shared with others?
January 23: Best way to find answers? PoC!

Minimum PoC goals:

- Amount of automotive requirements which can be covered by 3rd party middleware
- Scalable VSS data model applicability tested in end to end middleware
- Different data domains (different „trees“) can share same end to end infrastructure for sync, access, permissions …
- Standardized data models like VSS(o) can serve as a base for AI / Knowledge generation

Optimum:

- Open source Reference Implementation for 3rd parties to align with Automotive needs
Key questions for an End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share the same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge...
- ... can be shared with others?
Key questions for a End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data ...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge ...
- ... can be shared with others?
Today: Porto PoC extended by Onboard - Sync Server

Key questions for a End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data ...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge ...
- ... can be shared with others?
Today: Porto PoC extended by Onboard - Sync Server

Key questions for an End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share the same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data ...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge ...
- ... can be shared with others?
Key questions for a End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data...
  - ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge...
- ... can be shared with others?
Challenges we faced

General / Legal / Technical Challenges:
- Different touchpoints mean different SDKs (e.g. different functional scope, maturity)
- Different touchpoints mean different architectures (e.g. x86, ARM)
- Handling time-shifted actuator changes vs. sensor value
- In order to meet all country-specific legal requirements regarding data protection, a technical solution is needed that can fulfill all requirements
- Data model changes cause adjustments in many different places. Is there „perfect“ data model governance supported by a toolchain?
- Ensuring compatibility after breaking changes (programmatically not easy)
- Long data point names but also deep nesting can lead to problems (UUIDs?)
- Subscriptions on field level sometimes not easy to handle, e.g. for complex objects
- Local cache compatibility when changing the sync endpoint is challenging
- What degree of modularization can be achieved to remain as technology agnostic as possible (later Playground discussion!)

Key questions for a End-to-End Data Architecture:
- ⚠️ How data can be shared between all touchpoints?
- ✅ How different domains of data share same tech stack
- ✅ How should a bidirectional sync work?
- ⚠️ Who is responsible for conflict management?
- ⚠️ Who takes care about permissions, roles, rights and privacy?
- ⚠️ How the data model can be updated and synced?
- - How subscriptions can be handled?
- - How to handle historized and time series data ...
- - ....on different touchpoints?
- - ⚠️ How to handle multiple sync endpoints?
- - How to handle unidirectional data streams?
- - How (new) knowledge ...
- - ... can be shared with others?
Did we achieve already some goals? Yes!

Minimum PoC goals:

✅ Amount of automotive requirements which can be covered by 3rd party middleware

✅ Scalable VSS data model applicability tested in end to end middleware.

✅ Different data domains (different „trees“) can share same end to end infrastructure for sync, access, permissions …

⛔️ Standardized data models like VSS(o) can serve as a base for AI / Knowledge generation

Optimum:

✅ Open source Reference Implementation for 3rd parties to align with Automotive needs

⏳ Performance tests (scaling, vertical and horizontal)
Key questions for a End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data ...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How (new) knowledge ...
- ... can be shared with others?
Outlook: Knowledge Layer

https://wiki.covesa.global/pages/viewpage.action?pageId=71074417
Closing remarks

End-to-end data architecture is individual task.

BUT

• Collaboration helps to understand challenges, find solutions, and avoid redundancies
• Sustainability of infrastructure requires collaborative effort
• Pool knowledge and resources for a robust foundation
• Innovative approaches create sturdy, long-term data architecture
• Transparency and active contribution encouraged for success
Thank you for your attention!
Where we started

Key questions for an End-to-End Data Architecture:

- How data can be shared between all touchpoints?
- How different domains of data share same tech stack?
- How should a bidirectional sync work?
- Who is responsible for conflict management?
- Who takes care about permissions, roles, rights and privacy?
- How the data model can be updated and synced?
- How subscriptions can be handled?
- How to handle historized and time series data...
- ...on different touchpoints?
- How to handle multiple sync endpoints?
- How to handle unidirectional data streams?
- How new knowledge...
- ...can be shared with others?