

Design Pattern / Data Architecture documentation proposal

Purpose

The author ([Stephen Lawrence](#)) intends for this proposal to facilitate the Data Architecture team in particular and the wider Data Expert Group to find a shared view on documenting Data Patterns and Data Architecture. Contribution is therefore encouraged by commenting inline, contacting the author or in the Data Architecture calls.

This page describes a proposal to move towards joint work on documenting Data Patterns and Architecture after the Spring 2023 AMM.

Following a summary of where we stand, a proposal is made for a possible way forward. The author does not intend the proposal to be a fully formed work plan, but a skeleton roadmap on which the Data Architecture group and the Data Expert Group can iterate on to form one. An important principal is open discussion on what issues need to be tackled down, what is needed to form common understanding and sense of purpose.

- [Current situation \(summary\)](#)
- [Current situation \(description\)](#)
 - [Building for the next phase](#)
- [Proposal overview \(summary\)](#)
- [Proposal overview \(description\)](#)
 - [Document content](#)
 - [Document creation and publishing](#)

Current situation (summary)

- Clear industry need for Design Pattern and Data Arch discussions at an abstract level
 - This needs shared vocabulary and understanding of common design components and patterns
 - Required to build data exchange understanding between people from different areas
 - Industry demand for guides on how to use Covesa technology
- There is a mix of Covesa generated source material to draw on
 - Contemporary work from 2023 1H Workshop, AMM workshops and presentations
 - Variety of ad-hoc work from individuals, e.g. the authors design outlines in [vss-otaku](#)
 - Historical work from CVII Tech Stack and CCS projects
- Now need to move forward on joint work
 - This requires a shared view of goals. This proposal is intended to help facilitate that.

Current situation (description)

Covesa has a range of source material from which to use.

Contemporary work is occurring in the Covesa Data Expert Group including the Data Architecture pillar of the Group. For example the [results from the first workshop](#) included a high level scope to focus on zonal ECUs and above for the functional integration and data architecture. In addition high level deployment scenarios were sketched for in focus use cases including IVI, Smart device, Car2Cloud and Cloud2Cloud.

These sketches now need further work to build out the details. For example Smart Device will involve SmartDevice2Vehicle and SmartDevice2Cloud.

There are also various presentations from meetings and the All Member Meeting (AMM) conferences. For example this input on [Data Centric Architectures](#).

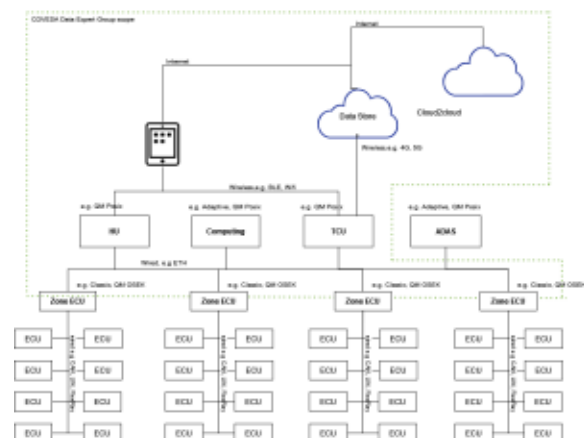
Related work is also occurring in the various Covesa projects focused on specific areas such as EV Charging, AOSP and Commercial Vehicles.

There is also individual work intended as contribution to the group work such as the authors work in his [vss-otaku](#) project, which includes discussion of combining [VSS data stores and data servers](#).

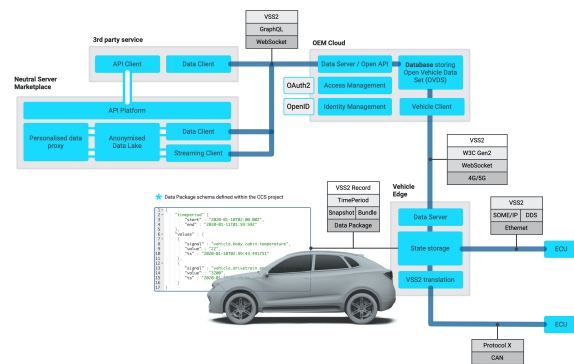
There is *historical* work from projects such as Genivi [Cloud & Connected Services \(CCS\)](#) project which investigated the requirements of open data capture in-vehicle and its connection to back-end systems in the cloud and resulted in a high level architecture concept.

Overview of Logical Architecture

Focus on functional integration and data architecture

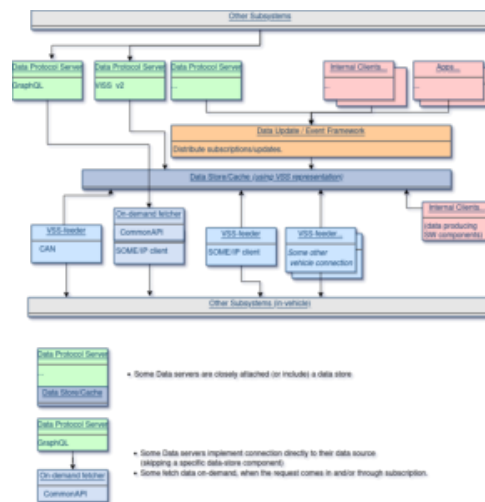


[blocked URL](#)



Source: CCS Reference Architecture - Work Breakdown Structure

CCS was followed by the [Common Vehicle Interface Initiative \(CVII\) Tech Stack](#) project which investigated the software required to transfer and process the standard VSS data model. For that a mutual understanding of terms and common components were required. The latter resulted in the in-vehicle architecture diagram on the right, which is intended to show the likely in-vehicle components and how they interact, rather than specify a specific monolithic architecture. A set of related terminology was also defined.



Source: CVII Tech Stack Partial System Diagram

Building for the next phase

As the number of adopters of Covesa technology, including VSS, has risen greatly we have seen increasing demand for descriptions of what it is intended for and how to use it. This helps both newcomers understand the technology and supports discussion within the community. At the same time the massive increase in the size of in-vehicle software and the introduction of technologies from outside of automotive has resulted in a need to quickly build understanding between people from a more diverse group of areas.

To be quickly productive this needs a shared vocabulary and understanding of common design components and patterns described in abstract terms. A solution is discussed in more detail in the proposal. To move forward on creating that *within the community* requires a *shared* view of the goals. This proposal is intended to help facilitate that.

Proposal overview (summary)

- A foundational concept is that there is a need for an approach to architecture that describes requirements and components in abstract logical terms.
 - This encourages collaboration by avoiding product comparisons (selecting "winners") and system architecture.
 - Provides a shared vocabulary for discussion and illustration.
 - Provides flexibility in how such components are arranged.
 - Should be the first approach to describing scenarios in Covesa.
 - PoCs for example would first describe the logical design before describing specific (product) technologies if needed.
- A lightweight outline or scope (e.g. what, why, how) for the content document for each topic should be created to shape its creation and explain its purpose.
 - This content of the outline should reflect the purpose. For example a Design Pattern is typically abstract and simplified to help understanding, whilst a Scenario may be more detailed and specific.
- Publishing method should be an early consideration as it will affect how content is created and discovered. It is suggested that the Hugo website framework used for the VSS documentation is a candidate.
- Whilst the proposal originates in the Data Architecture pillar of the Data Expert Group and is intended to move that project forward, there is of course a bigger picture for its use by the Covesa community.

Proposal overview (description)

Experience shows that to move the *community* forward it is helpful to discuss requirements and components in abstract logical terms. A shared vocabulary and reuse of common assets, such as patterns and diagrams, emphasis what is common, whilst avoiding the trap of focusing on what separates people in focusing on product comparisons (selecting "winners") and specific system architectures. There can also be collaboration, whilst retaining the flexibility individuals need in how components are arranged and their internal functionality. Such an approach ought to be a natural default for describing scenarios in Covesa. Repeating them socialises them within industry. PoCs could first describe the logical design before moving to specific (product) technologies if needed, e.g. 'A VISS Data Server was connected to a VSS Data Store. In our specific PoC X and Y were used.'

Document content

Patterns / building blocks:

- Data Server, Data Store
 - *Note: Focus on the makeup of these components and the connections between them. A building block in wider arch.*
 - Variants:
 - *Note: Rather than possibly confuse by trying to explain different configurations in a single document instead outline common variant patterns separately*
 - Mobile App, in-vehicle app
 - *Note: Store likely close to app process.*
 - Domain Data Service
 - *Note: Possibly data store centric, e.g. DB, with remote clients*
- Integration
 - Variants: synchronization, data flow bi/uni-directional, conflict resolution, latency, volume, frequency
- Scenarios
 - *Note: Focus likely more on viewpoints showing data flow over assemblies of components, either at high level (e.g. Data EG Deployment Scenario diagrams) or more of a focused snapshot of part of it (e.g. flow between Service, Vehicle API, Gateway and Autosar)*
 - Snapshots focusing on specific areas:
 - e.g. Vehicle API, Service + gateway and RT/ASIL southbound
 - Knowledge pyramid
 - *Note: Focus on connection between layers of pyramid. Specifically the flow of information up and down the knowledge pyramid between data, information and knowledge. Related is connection between pyramids.*
 - Vehicle API
 - Data EG Deployment scenarios:
 - *Note: From Workshop Miro and as shown in Porto AMM*
 - IVI, Smart device,
 - Car2Cloud,
 - Cloud2Cloud,
 - Data EG Touchpoints:
 - *Note: Scenarios related to the famous Covesa Scope Hedgehog touchpoints:*
 - Charging
 - Mobile Device
 - Cloud
 - Autosar
 - AI
 - Hosted OS

Document creation and publishing

Ultimately it is the *content* that is most important. The proposal below is one pragmatic approach to publishing using technology already known in Covesa, with a goal of publishing sooner, rather than later, whilst leaving room for evolution. The author is fully open to other approaches that reach that goal.

The [VSS documentation website](#) has been useful for a number of reasons. It brings the documentation together in one place, whilst supporting navigation and rich enough formatting. Being able to refer to a single URL within the site, from which other content can be found is useful for both business and technical discussions. It is therefore proposed we do the same for **publishing** the design documentation that is the topic of this proposal.

For ease of implementation it is suggested we use the same publishing technology used for the VSS documentation, namely [Hugo](#) with a similar format template. The requirements to build a static website to test documentation locally are straight forward and the automated build and deployment of Hugo websites is already supported within the Covesa CI. A useful attribute of Hugo is the split between content and publishing. The content can be defined in markdown for example and then consumed by the Hugo framework. This gives flexibility in both choosing tooling for the content creation step and possibly moving to a different website framework in the future if required, i.e. there is little Hugo lock-in.

Navigation should be a feature of the design. It is proposed that a hierarchical classification of topic types are listed in a panel on left hand side, e.g. Patterns, Architectures, Touchpoints, HowTo's etc. Selecting a document displays the content in the main window to the right. A lightweight document type classification would be useful to differentiate between generally agreed approaches and more experimental work. That would help avoid confusion from external consumers who are not involved in the upstream work.

If alternatives are proposed then it is suggested that work on the *content* starts whilst discussion of publishing technology takes place.

It is proposed that some guidelines for document **creation** would be useful. For example, diagrams should be in an open flexible format. To encourage collaboration it is important that others can extend and change diagrams. [drawio](#) is suggested as an option. It has good template support and free desktop and web editors are available, without restrictions, that support simultaneous editing.

It is helpful if draft text is developed in a format that encourages collaboration such as allowing inline comments and ease of editing. For example a document could be collectively created in Confluence before converting to .md markdown.