

Vehicle to Cloud Project

AGL Vehicle to Cloud Communications Project

- [AGL Vehicle to Cloud Communications Project](#)
 - [Project Definition](#)
 - [Scope](#)
 - [General Architecture](#)
 - [Use Cases](#)
 - [Project Timelines](#)

Project Definition

The intention of this project is to produce a prescriptive specification for the production, consumption and orchestration of messages between connected vehicle devices and the cloud using MQTT and Protocol Buffers, inclusive of recommendations for vehicle identity, security and system architecture. The message definitions in this project should align to message definitions in the `kuksa.val` gRPC project.

Scope

Documentation of recommended basic practices for MQTT V5 based vehicle to cloud communication patterns. Each of the items below can be worked on independently, with the exception of creating the reference applications and simulation, which is dependent upon at least one set of messages being finalized. HVAC will take priority.

Message format and orchestration for the most common vehicle telemetry and command based use patterns. [See Documentation Here](#)

- ☒ Create Protocol Buffer Messages in .proto files for use case driven messaging with request/response pattern. (i.e HVAC)
- ☒ Recommend MQTT Topic patterns for each message set
- ☒ Create Sequence diagrams showing the orchestration of message exchanges including timeouts, error and failure handling

Gradle Build configuration for building both documentation and Java (and other?) stubs from protocol buffer files.

The project currently uses Gradle to build message stubs in Java and C++ and generate documentation.

- ☒ Optimize the Gradle Build
- ☒ Improve the documentation template and document generation

Example message implementations for reference

- ☐ Implement references showing how to build, publish and receive MQTT messages
- ☐ Implement references showing how to take gRPC from embedded system and transfer to MQTT

Implement a message testing and simulation application/capability

- ☐ Take reference data and protocol messages and implement a vehicle messaging simulator for testing purposes. This can be local or remote and should validate message schema and pub/sub orchestration

Expose API through AGL for generating a salted hash identity for vehicle from the VIN number, using same keys as the used in creating the operational certificate.

This potentially pre-exists from another project

- ☐ Create a service within AGL for generating a salted hash, or otherwise secure identity for the vehicle and sharing with backend databases

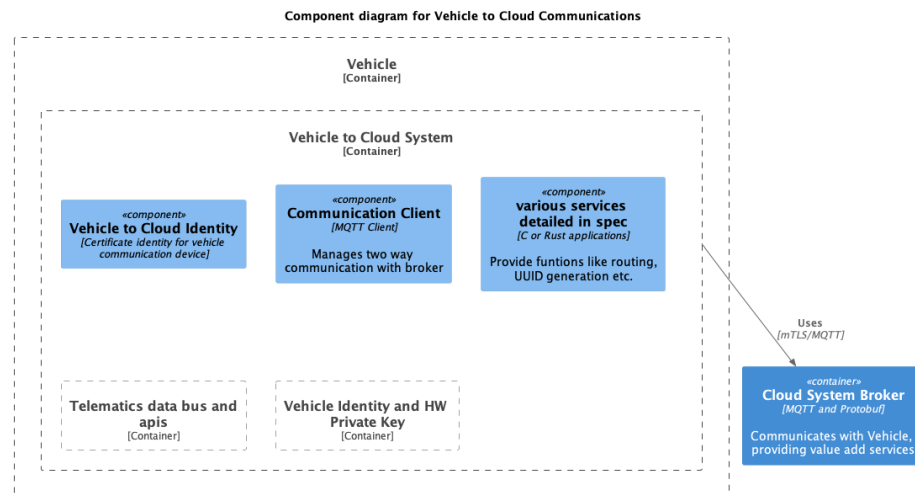
Embedding services for protobuf message serialization and deserialization and UUID generation in Linux build.

- ☐ Distribute the proper libraries for protocol buffers within the AGL build
- ☐ Also include vehicle applications/services for connectivity and communication, such as Paho, AWS IoT SDK
- ☐ Include reference embedded applications/services for some of the message patterns.

Contractor Scope

Ask	Description	Net Result	Dependency	Estimated Effort (including testing integration)
Vehicle Identity Service	Programmatic and secure method of producing a vehicle identity that doesn't expose VIN in plain text	A service or library built into AGL that enables the generation of a secure identity for the vehicle, acting as a proxy for the VIN. Ideally this would be a hash+SALT of the VIN itself and provide both security and privacy compliance.	None	0 - It seems this is already being produced by other workstreams
UUID Service for Messages	A simple UUID service for producing non-repeating message IDs for use in publishing mqtt messages	A service or library built into AGL that produces unique id's for messages that are sent via the V2C project	None	Including testing and integration this should be 40 to 80 person hours
Embedded Protobuf Library	Include protobuf libraries for C, C++ and java in Linux build	This likely already exists, but would be beneficial if Protobuf Libraries are embedded and distributed with AGL builds		~80 hours
Telecommunications Client Reference	A client that receives message requests from application services and creates and publishes messages to the broker. Also receives messages on behalf of applications from the cloud broker and routes them to the applications.	The net result of this is a reference client made in C/C++ or Rust that can be leveraged when testing or developing with the V2C project. The client should handle secure connectivity and the building and publishing of final messages on the behalf of embedded applications	Needs at least one full set of messages and documentation completed (June 23)	~200 hours
Reference Cloud Services	Implementations in AWS IoT and other cloud services showing the orchestration of messages from the cloud side	The net result of this would be a minimum connected vehicle platform that handles the cloud side of publish and subscribe for messages using the V2C specification	Needs at least one full set of messages and documentation completed (June 23)	~100 hours.

General Architecture



Use Cases

- Vehicle and Vehicle Device Provisioning
- MQTT Communication Lifecycle Events, Monitoring and Best Practices
- Client Initiation and Connection
- General Messages
- Basic Telemetry
- Remote Commands (HVAC)
- OTA Orchestration and Content Downloads
- Remote Diagnostics
- Application Defined Messages [Key Value Pairs]
- Simulation and Testing

Project Timelines - Flag - Incorporate Task Level Details

- ☒ 01/20/23 - Project Description and Details Updated in Confluence
- ☒ 01/31/23 - Initial Repository in AGL hosted Repository
- ☒ 03/01/23 - Completed draft of main contents to a "ready for invitation to collaborate" state
- ☒ 03/01/23 - Ready for collaboration beyond working group
- ☐ TBA - Initial release with contributions ready

Calls for Help - Flag - Need Task Level Details

- Need reviews of content in Repo as produced and help adjusting the diagrams as needed
- Help producing reference implementation in C, RUST, PYTHON etc.
- Help creating a quality testing and simulation framework.
- Help creating security tests to validate security of vehicle identity and vehicle to cloud communications scripts
- Need to create the vehicle identity API for AGL and understand the mechanism for translating back to VIN in the cloud system.
- Help aligning the specification message objects to VSS

What work needs to be completed prior to contractor engagement, what work can be completed in tandem with contractor work/dependent on contract work, and what work needs to be completed after contract work?