uServices

A step towards standardized vehicle interfaces for SDV

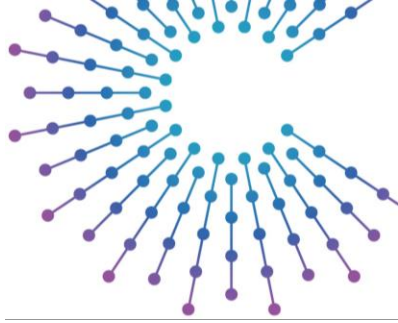
Halim Ragab – halim.ragab@gm.com

October 11 2023

COVESA

Accelerating the future of connected vehicles

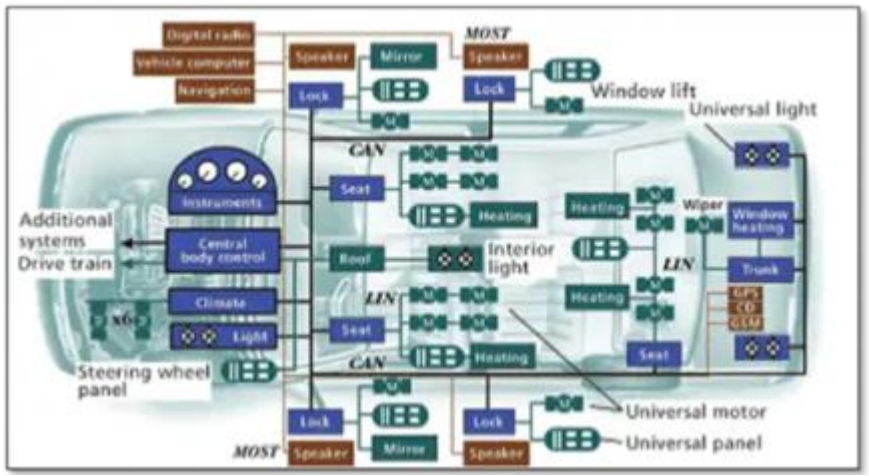
Agenda



- Introduction
- Why uServices?
- What are uServices?
- Example
  - Anatomy of uService - Hello World– Service and RPCs
  - Anatomy of uService - Hello World – Topics
  - Anatomy of uService – Properties
- Field Options
- Alignment with VSS
- Other COVESA projects .. Current status
- What is next

COVESA
Accelerating the future of connected vehicles

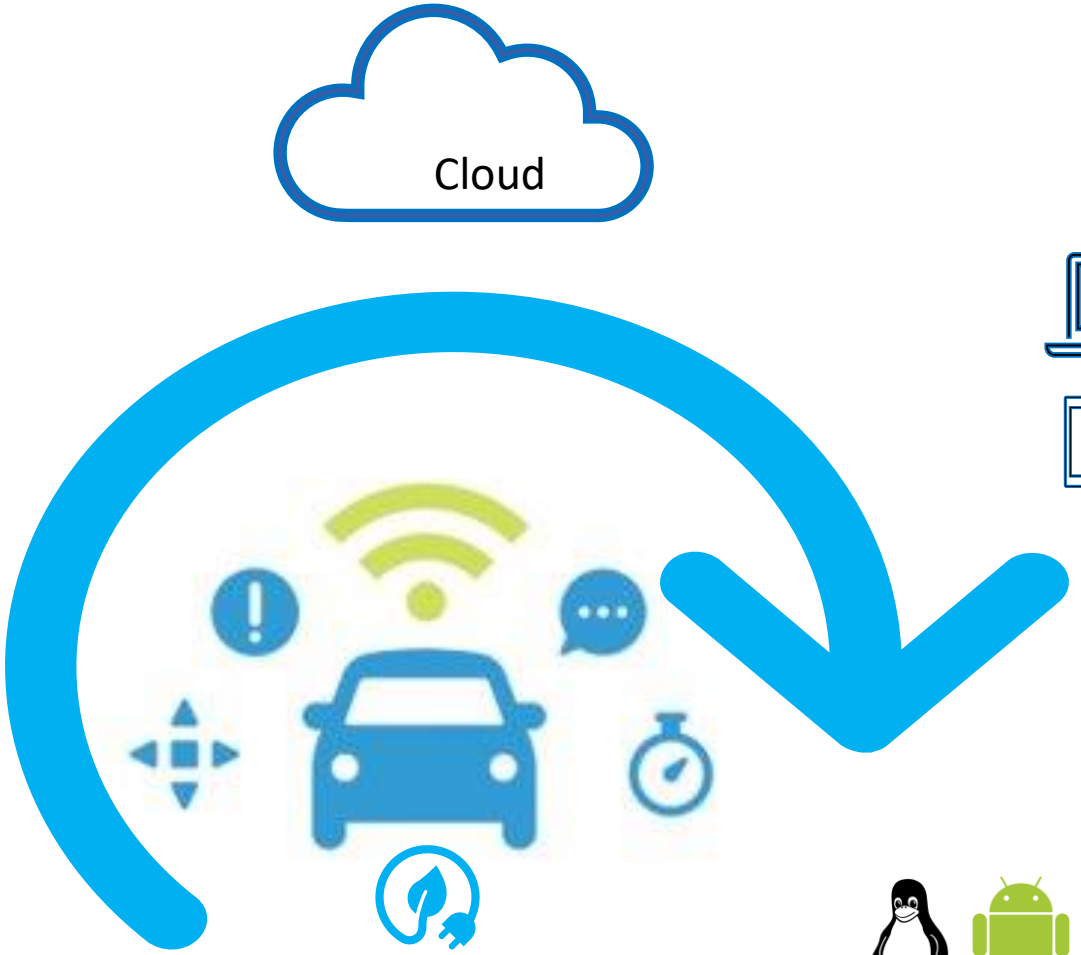
Introduction

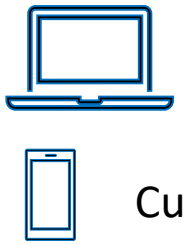


COVESA
Accelerating the future of connected vehicles

More interaction – features everywhere

Internet of Things
V2X, V2G, Home, Edge





Customer devices

AUTOSAR

COVESA
Accelerating the future of connected vehicles

Introducing uServices

The SDV Future: Challenges and Opportunities

9:00 AM - 9:30 AM Wed

Salons E, F, G, H

Keynote Plenary Thought Leadership

Speakers

Dan Nicholson

Vice President of Strategic Technology Initiatives

General Motors

The Automotive Industry is well on its way to a Software Defined Vehicle future, facing challenges and opportunities to ensure value to customer and company alike. Dan Nicholson, General Motors Vice President of Strategic Technology Initiatives, will provide perspective on these challenges and the ways the industry can address them.

uServices

Open Source Vehicle Services

We are contributing uServices as open source in COVESA

uServices Goals

Communicating through uProtocol with vehicle features	Standard API to abstract vehicle services	Enhancing collaboration	Enabling developer access for applications

general motors

15

COVESA
Accelerating the future of connected vehicles

Why uServices?

SOA

Discovery

Consistent Developer Mental Model - Everywhere

Open Source

Open Source

Build an Automotive Ecosystem that is open, active, with fully engaged community of developers and automotive suppliers

For Why services are needed, please refer to previous COVESA's AMM presentation on VSC. (e.g. https://wiki.covesa.global/download/attachments/32079873/Day1_Erik_1545_Introduction_to_VSC.pptx?version=1&modificationDate=1666625325766&api=v2)

https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec

COVESA
Accelerating the future of connected vehicles

What are uServices?

- Catalog of vehicle interfaces for standardized access to vehicle data
- Same definition across all platforms (in-vehicle software, Cloud, and mobile)
- Support for various interaction patterns (Publish/Subscribe, Client/Server, Notifications)
- Supports configurations for the service, topic or method
- Uses protobuf with custom options for the definition
- Compatibility and interoperability
  - **uProtocol**, AUTOSAR SOME/IP, VSS

uProtocol
CONNECTING AUTOMOTIVE APPS AND SERVICES, EVERYWHERE

protobuf
Protocol Buffers

Vehicle Signal Specification

COVESA
Accelerating the future of connected vehicles

Anatomy of uService – Hello World – service + RPCs

Service name

Service version

Service id

Methods (RPCs)

Method_id

```
1  // Hello World service
2  //
3  service HelloWorld {
4
5    // Service Metadata - Name, version, id, RPC methods
6    option (name) = "example.hello_world";
7    option (version_major) = 1;
8    option (version_minor) = 0;
9    option (id) = 999;
10
11   // Say Hello method
12   // The method URI is:
13   //  up:/example.hello_world/1/rpc.SayHello
14   rpc SayHello(HelloRequest) returns (HelloResponse) {
15     option (method_id) = 1;
16   }
17 }
```


COVESA
Accelerating the future of connected vehicles

Anatomy of uService – Hello World – Topics/Events

A message representing a Topic (Event)

Instances of the topic

Example **uProtocol** URIs for the topics

```
// Timer message
// This message is published as payload of the topics below:
//  up:/example.hello_world/1/one_second#Timer
//  up:/example.hello_world/1/one_minute#Timer
message Timer {

  // Time
  google.type.TimeOfDay time = 1;

  enum Resources {
    one_second = 0;
    one_minute = 1;
  }
}
```

Meta-data that defines a topic

```
message TimerOptions {
  option (base_topic_id) = 1;
  Timer.Resources resource_name = 1 [ (resource_name_mask) = "*" ];
}
```


COVESA
Accelerating the future of connected vehicles

Anatomy of uService – Properties (Configurations)

Custom extensions defining the configurations of the service or a topic

```
service BodyCabinclimate {
  // Service meta-data option definitions - Name, version, id, rpc method
  option (name) = "body.cabin_climate";
  option (version_major) = 1;
  option (version_minor) = 2;
  option (id) = 5;
  option (number_of_row_1_zones) = 2;
  option (number_of_row_2_zones) = 0;
  option (number_of_row_3_zones) = 0;
```

```
message ZonesOptions {
  option (base_topic_id) = 10;
  option (temperature_setpoint_min) = 16;
  option (temperature_setpoint_max) = 31;
  option (number_of_blower_levels_row_1) = 8;
  option (number_of_blower_levels_row_2) = 8;
  option (is_auto_available) = true;
  option (blowers_available_row_1) = 1;
  option (blowers_available_row_2) = 0;
  option (blowers_available_row_3) = 0;
  option (airdistribution_available_row_1) = 1;
  option (airdistribution_available_row_2) = 0;
  option (airdistribution_available_row_3) = 0;
```


COVESA
Accelerating the future of connected vehicles

Field options

- Units
- Min/Max value
- Default value
- Resolution
- Read only/Write only
- Authentication
- Permissions
- .. etc

```
// Temperature in degrees Celsius.
float temperature = 2 [
  (unit) = CELSIUS,
  (max_value) = 50,
  (min_value) = 0,
  (default_value) = 22
];

// Fan speed setting auto state.
AutomaticMode fan_auto_state = 3 [ (readonly) = true ];

// Fan setting in percent. 0 = Off, 100 = Maximum air is coming through the
// vents.
int32 fan_speed = 4 [ (unit) = PERCENT, (max_value) = 100, (min_value) = 0 ];
```

Alignment with VSS

- uServices aligns with VSS names in general, and it introduced another extension to have references to VSS hierarchy when possible

cabin_climate service topics

```
/ Individual Zone statuses and requests.
//
message Zone {

  // reference to VSS branch
  option (vss_ref) = "Vehicle.Cabin.HVAC.Station";

  // The zone resource. It is returned in an event message and set by the
  // calling applications in sifferent update methods.
  Resource id = 1;

  // Temperature in degrees Celsius.
  float temperature = 2 [
    (unit) = CELSIUS,
    (max_value) = 50,
    (min_value) = 0,
    (default_value) = 22
  ];

  // Fan speed setting auto state.
  AutomaticMode fan_auto_state = 3 [ (read only) = true ];

  // Fan setting in percent. 0 = Off, 100 = Maximum air is coming through the
  // vents.
  int32 fan_speed = 4 [ (unit) = PERCENT, (max_value) = 100, (min_value) = 0 ];

  // Air distribution setting auto state.
  AirDistribution air_distribution = 5;

  // True = Zone power is on.
  bool is_power_on = 6;
```

SingleHVACStation.vspec

```
#
# A single HVAC station in the vehicle.
#

FanSpeed:
  datatype: uint8
  type: actuator
  min: 0
  max: 100
  unit: percent
  description: Fan Speed, 0 = off. 100 = max

Temperature:
  datatype: float
  type: actuator
  unit: celsius
  description: Temperature

AirDistribution:
  datatype: string
  type: actuator
  allowed: ['UP', 'MIDDLE', 'DOWN']
  description: Direction of airstream
```


COVESA
Accelerating the future of connected vehicles

Today's status

- uServices Project repo https://github.com/COVESA/uservices
- Documentation is in the repo
- Current Services:
  - example.hello_world
  - body.cabin_climate
  - body.horn
  - body.mirrors
  - chassis
  - chassis.braking
  - chassis.suspension
  - propulsion.engine
  - Propoulsion.transmission
  - vehicle.exterior
  - vehicle

  - More will be added as they get created

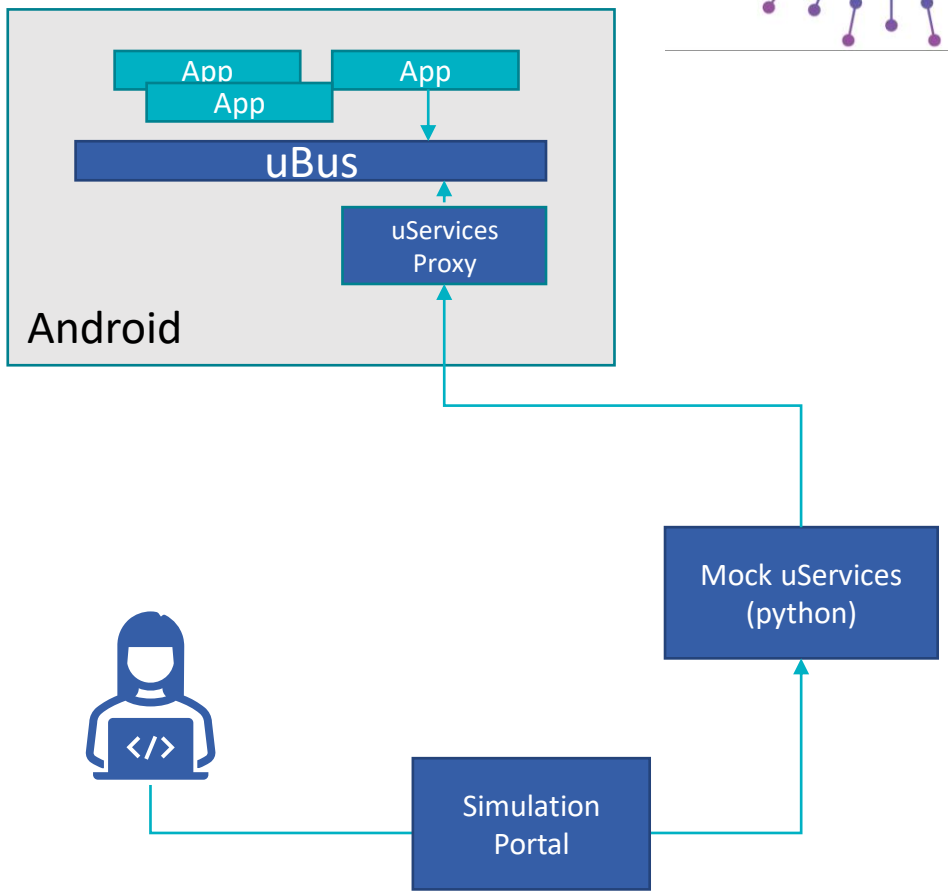
Capabilities

- Vehicle Capabilities
  - Cabin HVAC:
    - Get On/Off State of HVAC system
    - Control On/Off State of HVAC system
    - Get On/Off State of Air Conditioning System
    - Control On/Off State of Air Conditioning System
    - Get On/Off State of Air Recirculation
    - Control On/Off State of Air Recirculation
    - Get On/Off State of Front Defroster
    - Control On/Off State of Front Defroster
    - Get On/Off State of Rear Defroster
    - Control On/Off State of Rear Defroster
    - Get Estimated Cabin Air Temperature
    - Get Temperature per zone (HVAC Station)
    - Control Temperature per zone (HVAC Station)
    - Get Fan Speed per zone (HVAC Station)
    - Control Fan Speed per zone (HVAC Station)
    - Get Air Distribution per zone (HVAC Station)
    - Control Air Distribution per zone (HVAC Station)
  - Cabin Seating
    - Get Seat Position per seat
    - Control Seat Position per seat
    - Get Seat Heating Mode (Heat, Vent, Cool) and Level per seat
    - Control Seat Heating Mode (Heat, Vent, Cool) and Level per seat
    - Get Seat Occupancy Status
    - Get Seat Belt Status
  - ADAS Perception:
    - Publish List of Potentially Moving Objects
    - Publish Road Objects
    - Publish Static Objects

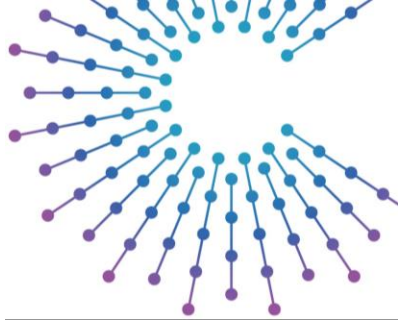
What is next

- Mock Implementation in python
- Android uProtocol proxy
- BDD Files describing behavior of the uServices

Given ..
When ..
And ..
Then ..



Other COVESA projects



- Common Vehicle Interface Working Group is Active
- Many activities are going on right now.
  - VehicleAPI
  - IFEX
  - VISS
  - OpenAPI/Async API
  - uServices → GM contribution towards standardized vehicle interfaces
  - DDS

COVESA
Accelerating the future of connected vehicles

Join the discussion

All Member Meeting – Fall 2023 - Common Vehicle Interfaces Working Session

1:00 PM - 2:45 PM Thu. October 12th 2023

Weekly Meeting - Common Vehicle Interfaces (Interface Pillar Data Expert Group)

11:00 AM-12:00 PM Eastern Time

COVESA
Accelerating the future of connected vehicles

COVESA

Accelerating the future of connected vehicles