uProtocol
Connecting Automotive Apps and Services Everywhere!

Steven Hartley
Technical Fellow, SDV Architecture
steven.hartley@gm.com
Agenda

- Problem Statement (Why?)
- What is uProtocol?
- How?
  - Building Blocks
  - Three Layers
- Next Steps
Problem Statement (Why?)

The Past

Present
What is uProtocol (Guiding Principles)

- SOA 2.0
- Resiliency
- Discoverable
- Consistent (simple) Developers
  Mental Model - Everywhere
How: The Building Blocks

**URIs to identify and address Things..**

```
up://<USERINFO@><UDEVICE>.<UDOMAIN>:<PORT>/<UE>/<UE_VERSION>/<RESOURCE|rpc.METHOD>?<QUERY>#MESSAGE
```
Software Entities

Software Entity (uE)

Apps & Services
- Custom uEs
- Standardized uEs

Extends

Legend
- Core Functionality
- Communication backbone

uProtocol

Core uP uEs
- uTwin
- uDiscovery
- uSubscription

Dispatchers
- uBus
- uStreamer
- DPR

Application Layer (uP-L3)

Communication Layer (uP-L2)
Putting it all together
Next Steps

● Lots more to share!

To be continued...
THANK YOU!
Backup
# What is uProtocol?

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>uDevice</td>
<td>Independent execution environment that has platform software components to implement uProtocol (ex. ECUs, Android-OS, Linux). Typically, uEs can communicate with each other within a uDevice using Inter-process communication (IPC) protocols</td>
</tr>
<tr>
<td>uDomain</td>
<td>Collection (group) of uDevices using DNS nomenclature (ex. vehicle, Cloud, etc…)</td>
</tr>
<tr>
<td>uE</td>
<td>Software Entity that talks uProtocol</td>
</tr>
<tr>
<td>uApp</td>
<td>uE that performs the role of consumer</td>
</tr>
<tr>
<td>uService</td>
<td>uE that performs the role of producer</td>
</tr>
<tr>
<td>Resources</td>
<td>Something that can be manipulated/controlled/exposed by a service (ex. Door, window, camera, etc..)</td>
</tr>
<tr>
<td>Topics</td>
<td>Subject that a producer produces to and a subscriber subscribes to per the publisher-subscriber design pattern</td>
</tr>
</tbody>
</table>
## Purpose

<table>
<thead>
<tr>
<th>Where did it come from?</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the destination?</td>
<td>sink</td>
</tr>
<tr>
<td>What type of event?</td>
<td>type</td>
</tr>
<tr>
<td>What is the unique identifier?</td>
<td>id</td>
</tr>
<tr>
<td>What format is the data?</td>
<td>datacontenttype</td>
</tr>
<tr>
<td>What schema is the data?</td>
<td>dataschema</td>
</tr>
<tr>
<td>What is the event data</td>
<td>data</td>
</tr>
</tbody>
</table>

### Transport Protocol

**CloudEvent**

**Metadata (attributes)**

**Data**
Architecture Patterns

Observer Pattern

Subject

Fire Event

Subscribe

Observer

Publish-Subscribe Pattern

Publisher

Publish topic (Event)

Subscribe

Fire Event

Topic / Event

Subscriber

Server RPCs
Application Layer (uP-L3)

Overview

- Business logic definition layer
- Interface (methods, topics, messages) declared in proto files
- Architecture Patterns for communication:

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPC</td>
<td>Unary Request &amp; response between uEs</td>
</tr>
<tr>
<td>Publication</td>
<td>Publisher produces events for many consumers/subscribers</td>
</tr>
<tr>
<td>Notification</td>
<td>Fire &amp; forget event from one uE to the next (1:1 not 1:n relationship)</td>
</tr>
</tbody>
</table>

- Application layer messages placed in CE payload (data)

uProtocol Application Layer uEs

- Set of core services that MUST be supported in all uDevices (ex. uSubscription & uDiscovery)
uSubscription

- **Purpose**
  - To allow local subscribers to subscribe to locally produced topics
    - Both the subscribes and producers are on the same device
  - To allow subscribers to subscribe to remotely produced topics (and vice versa)
    - Producers and subscribers are not in the same device
  - To allow producers the ability to create/delete topics

- **Additional Functionality**
  - Manage distributed subscriptions states
  - Notification for observers when subscription state changes
  - Management of topic lifecycle
  - Advanced subscription attributes
uDiscovery

- **Purpose**
  - Provide a distributed database of deployed (static) information about uThings (uDevices, uDomains, uEs, etc...)
  - CRUD operations for uEs to get/set data in the database

- **Taxonomy**
  - Schema, nodes, classifications of uThings
  - Node
    - A globally addressable uThings and contains a list properties as well as 0-n child Nodes
  - Properties
    - Key-value pair of information about said Node (can be of any various scalar and non-scalar types)
uP-L2 Purpose

- To define events types and their use cases
- To describe event attribute details (use and purpose)
- Dispatching/routing of events (using event attributes)

<table>
<thead>
<tr>
<th>Type</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publish</td>
<td>Publish generic message</td>
</tr>
<tr>
<td>File</td>
<td>Publish file transfer event. Files are transferred automatically by publishing this message type</td>
</tr>
<tr>
<td>Request</td>
<td>RPC Request message</td>
</tr>
<tr>
<td>Response</td>
<td>RPC Response Message</td>
</tr>
</tbody>
</table>
Communication Layer (uP-L2)

**uBus**
Intra-uDevice uE-2-uE Event Dispatcher

**uStreamer**
Inter-uDevice Event Dispatcher
Transport/Session Layer (uP-L1)

Purpose

- To define “how” to implement bidirectional point-2-point communication between uEs over existing Internet, automotive, and OEM proprietary standards
- Connection management & establishment
- CE Formats (how CEs are encoded in the transport protocol)

Sample Transports

<table>
<thead>
<tr>
<th>Transports</th>
<th>Protocol</th>
<th>CE Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android</td>
<td>Binder</td>
<td>Protobuf</td>
</tr>
<tr>
<td>Vehicle ←→ Cloud</td>
<td>MQTT</td>
<td>JSON</td>
</tr>
<tr>
<td>Within Cloud</td>
<td>HTTP/2</td>
<td></td>
</tr>
<tr>
<td>Inter-Device (In-vehicle)</td>
<td>SOME/IP</td>
<td>SOME/IP</td>
</tr>
</tbody>
</table>