



# uProtocol

## Connecting Automotive Apps and Services Everywhere!

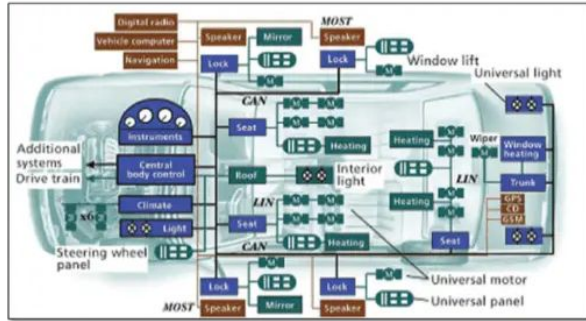


Steven Hartley  
Technical Fellow, SDV Architecture  
[steven.hartley@gm.com](mailto:steven.hartley@gm.com)

# Agenda

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

# Problem Statement (Why?)

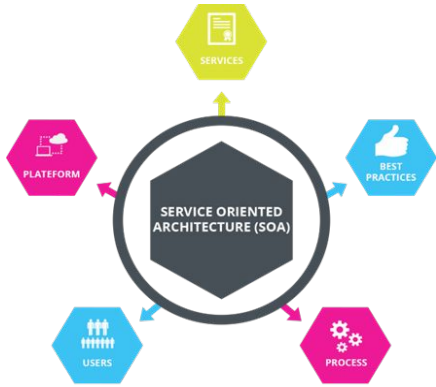


The Past



Present

# What is uProtocol (Guiding Principles)



**SOA 2.0**



**Resiliency**



**Discoverable**



**Consistent (simple) Developers  
Mental Model - Everywhere**

# How: The Building Blocks



***URLs to identify and address Things..***

scheme

authority

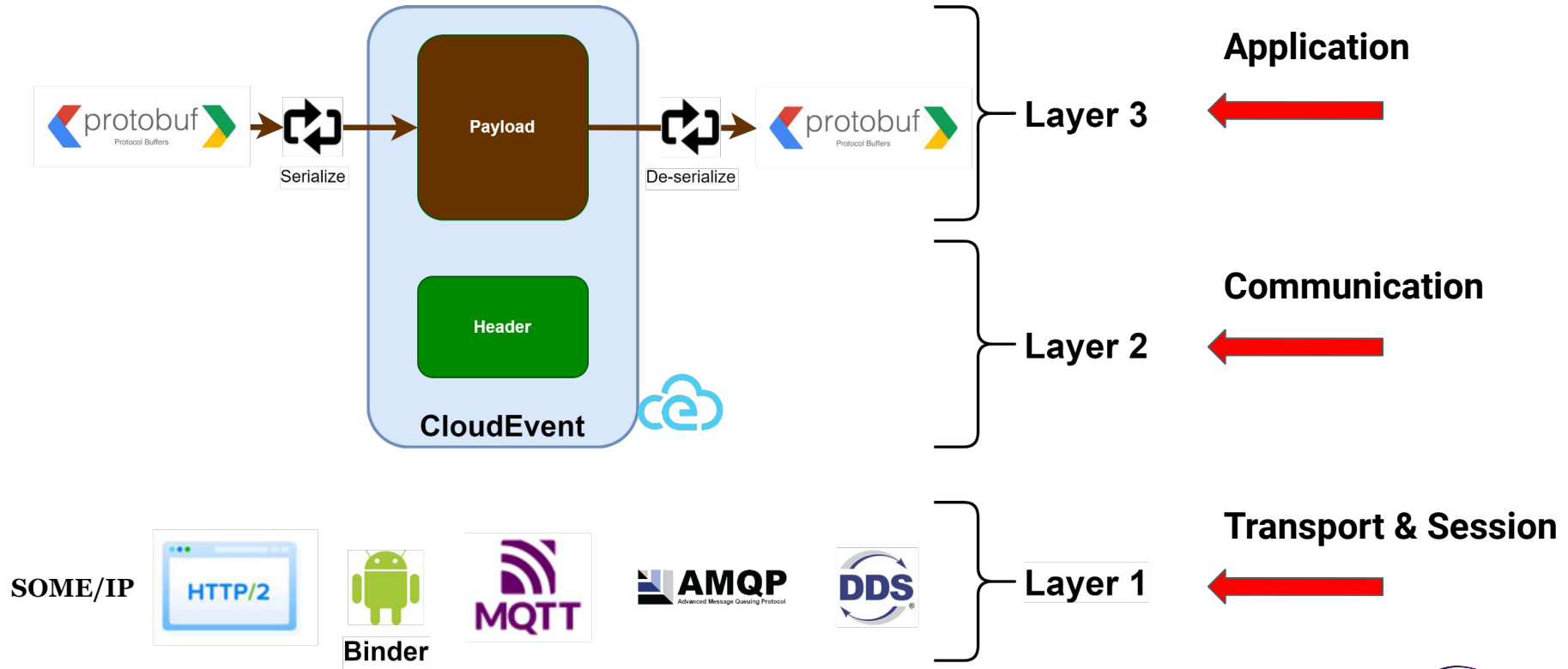
path

query

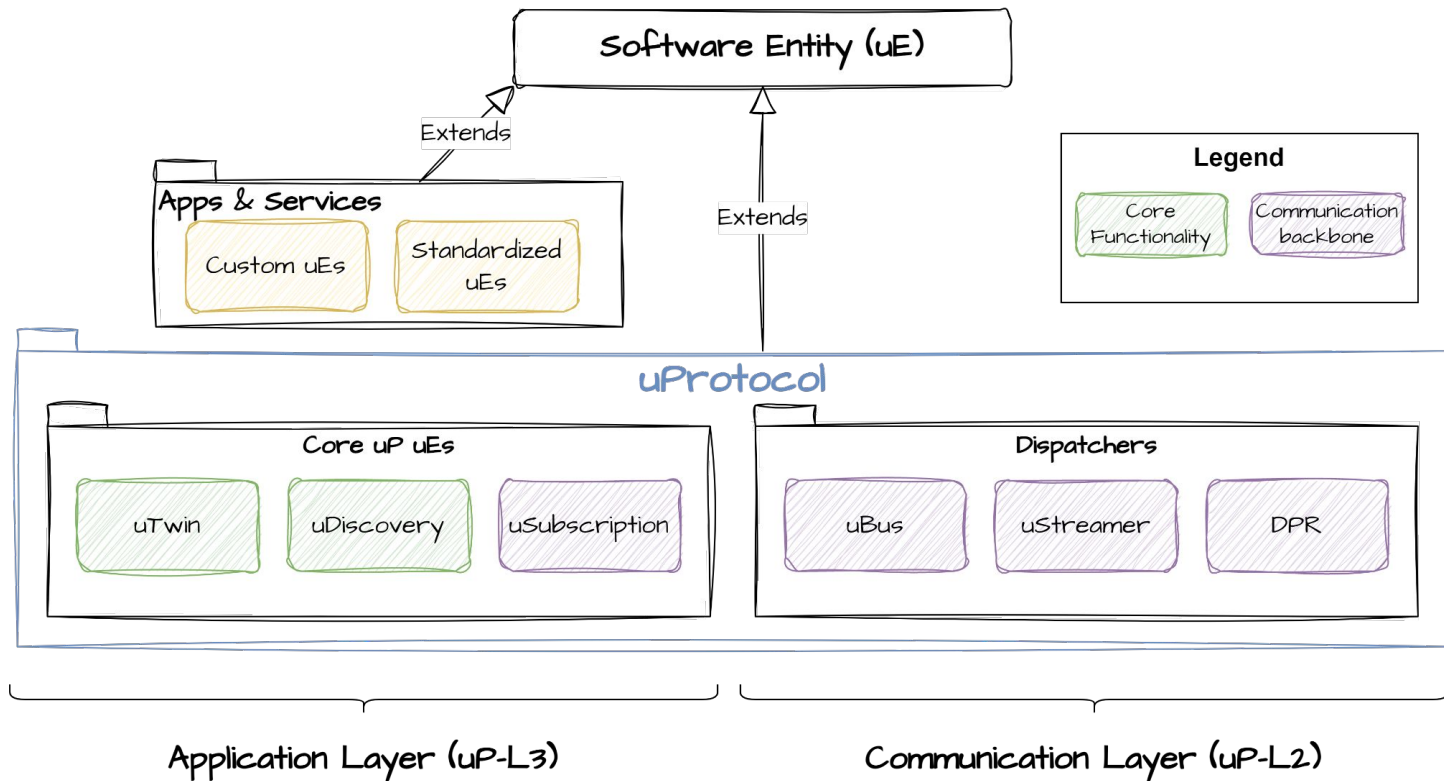
fragment

up : //<USERINFO@><UDEVICE>.<UDOMAIN><:PORT>/<UE>/<UE\_VERSION>/<RESOURCE | rpc.METHOD><?QUERY><#MESSAGE>

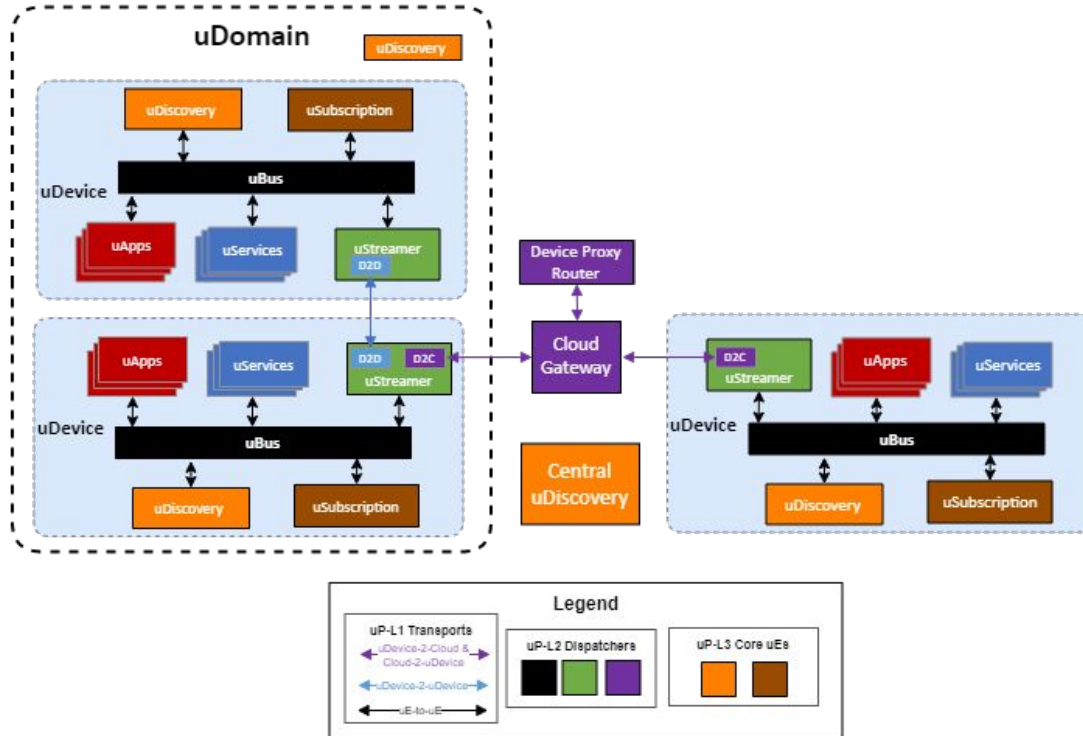
# Protocol Layers



# Software Entities



# Putting it all together





# Next Steps

- Lots more to share!



*To be continued...*



# THANK YOU!

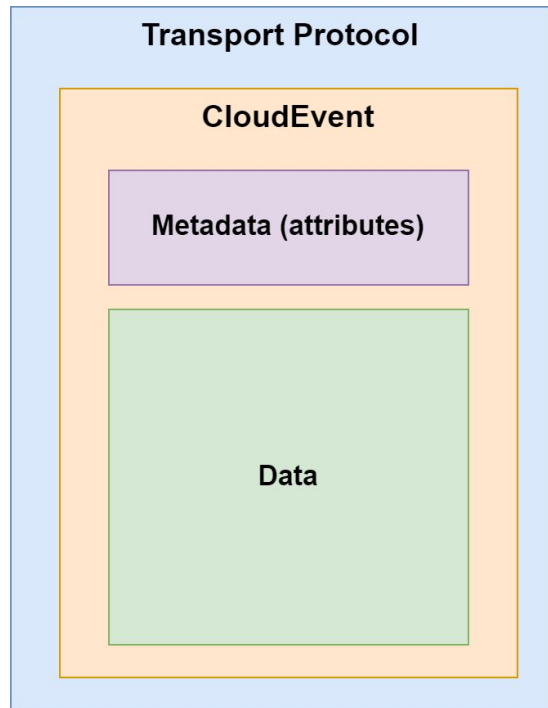
# Backup

# What is uProtocol?

Term	Definition
<b>uDevice</b>	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
<b>uDomain</b>	Collection (group) of uDevices using DNS nomenclature (ex. vehicle, Cloud, etc...)
<b>uE</b>	Software Entity that talks uProtocol
<b>uApp</b>	uE that performs the role of consumer
<b>uService</b>	uE that performs the role of producer
<b>Resources</b>	Something that can be manipulated/controlled/exposed by a service (ex. Door, window, camera, etc..)
<b>Topics</b>	Subject that a producer produces to and a subscriber subscribes to per the publisher-subscriber design pattern

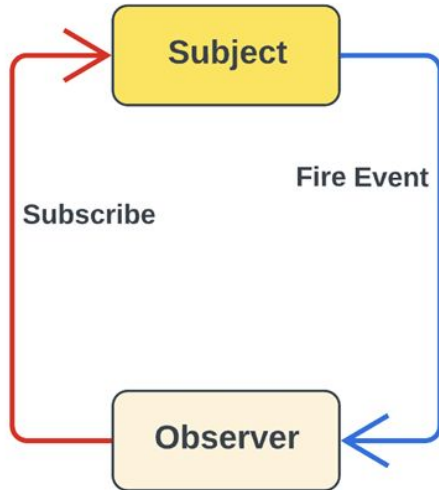


Purpose	CloudEvent Attributes
Where did it come from?	<b>source</b>
What is the destination?	<b>sink</b>
What type of event?	<b>type</b>
What is the unique identifier?	<b>id</b>
What format is the data?	<b>datacontenttype</b>
What schema is the data?	<b>dataschema</b>
What is the event data	<b>data</b>

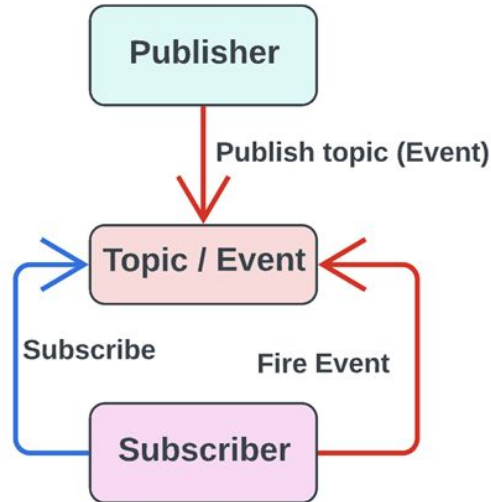


# Architecture Patterns

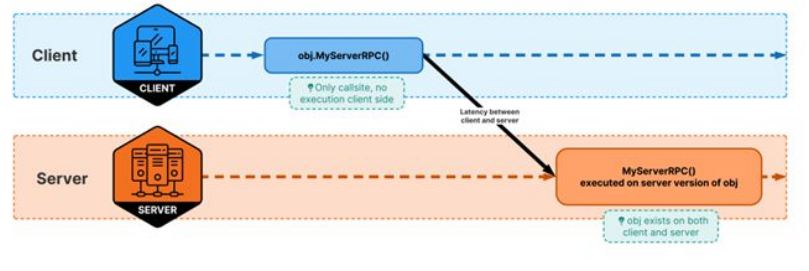
## Observer Pattern



## Publish-Subscribe Pattern



## Server RPCs



# Application Layer (uP-L3)

## Overview

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

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

- 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)

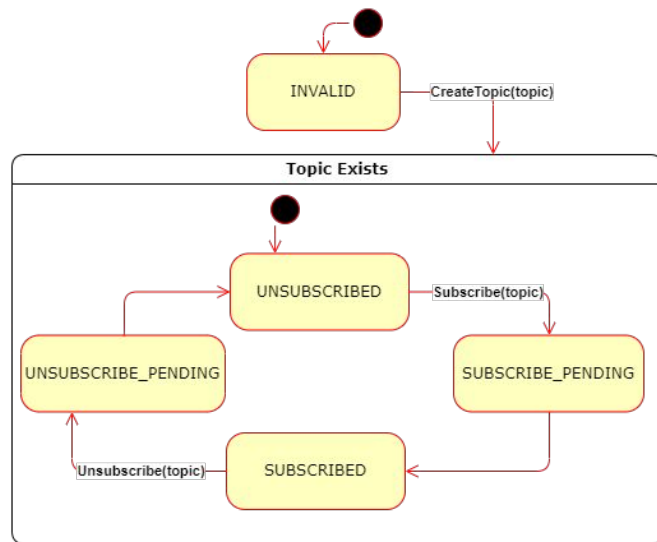
# uSubscription

- **Purpose**

- To allow local subscribers to subscribe to locally produced topics
  - Both the subscribers 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





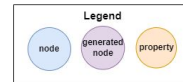
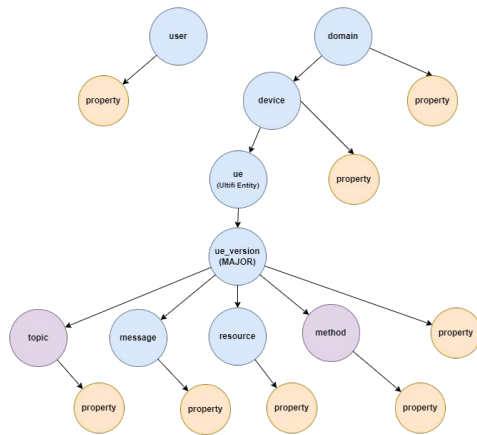
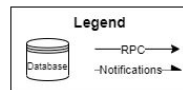
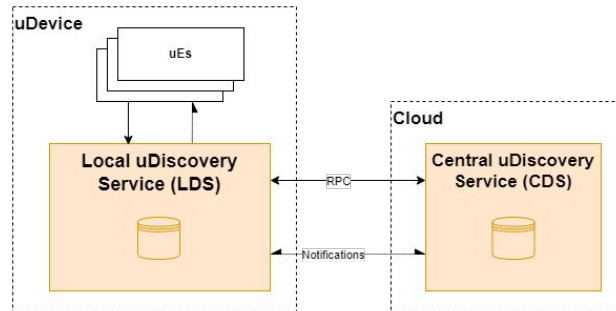
# uDiscovery

## ● 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)



# Communication Layer (uP-L2)



## 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)

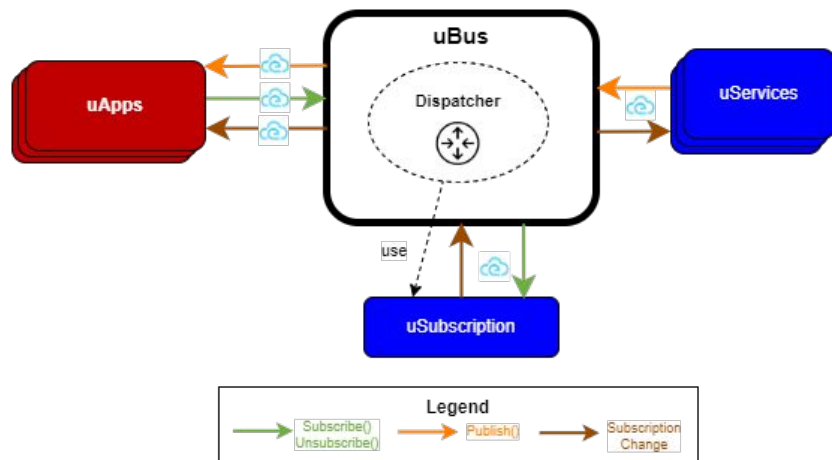
## uProtocol Event Types

Type	Use Case
<b>Publish</b>	Publish generic message
<b>File</b>	Publish file transfer event. Files are transferred automatically by publishing this message type
<b>Request</b>	RPC Request message
<b>Response</b>	RPC Response Message

# Communication Layer (uP-L2)

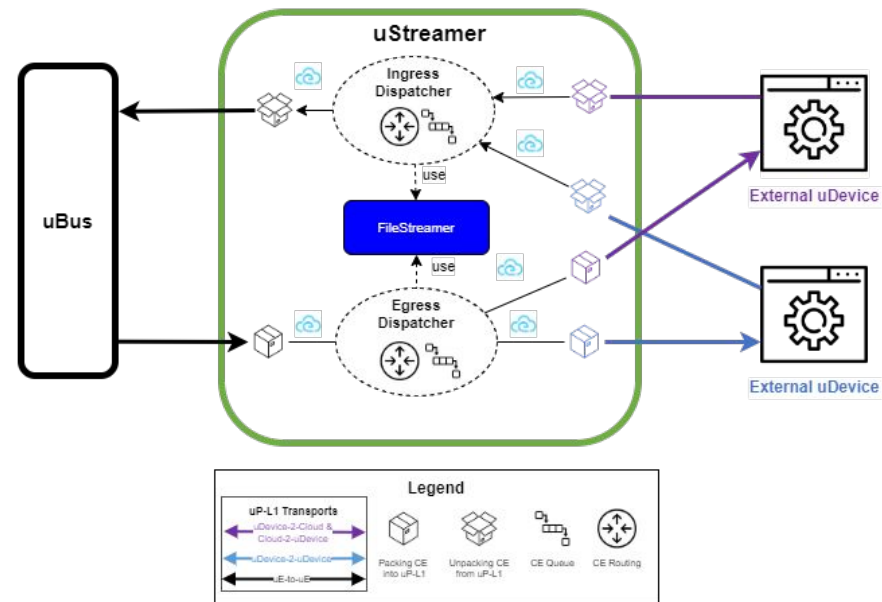
## uBus

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

Transports	Protocol	CE Format
<b>Android</b>	Binder	<a href="#">Protobuf</a>
<b>Vehicle</b> $\longleftrightarrow$ <b>Cloud</b>	MQTT	<a href="#">JSON</a>
<b>Within Cloud</b>	HTTP/2	
<b>Inter-Device (In-vehicle)</b>	SOME/IP	SOME/IP