from Service-*Mess* to Service-*Mesh*

with

uProtocol

CONNECTING AUTOMOTIVE APPS AND SERVICES, EVERYWHERE

Free stock photos - PxHere - 1055658

SDV

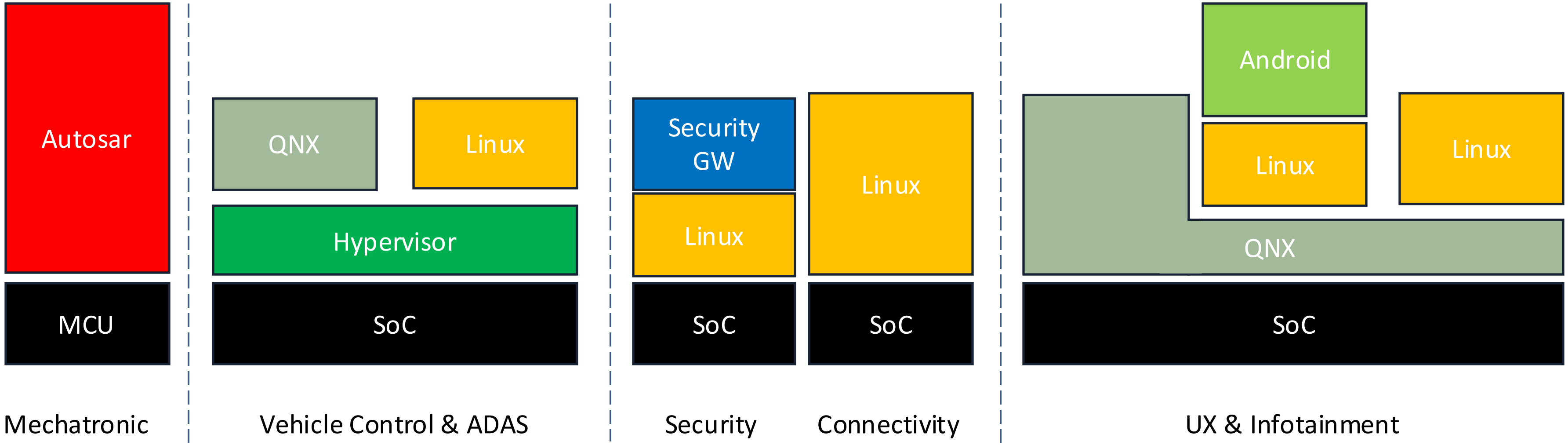
Eclipse Software Defined Vehicle

uProto-why?

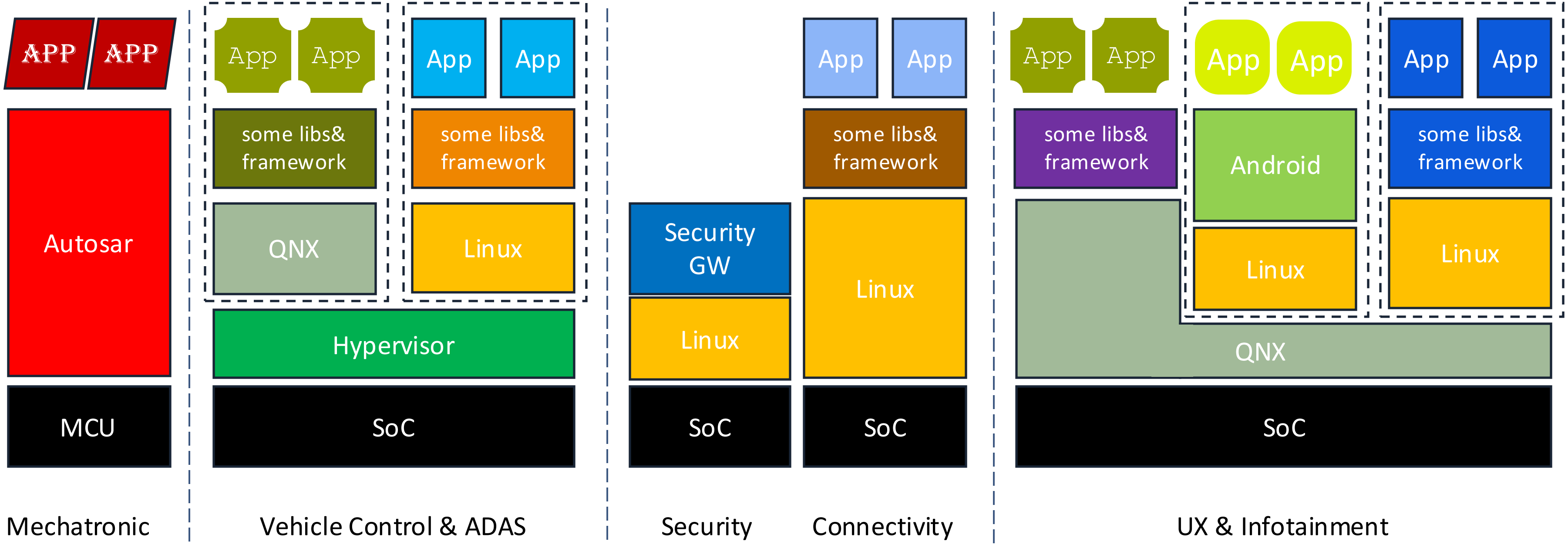
Free stock photos - PxHere - 1055658

Middleware, why do we care?

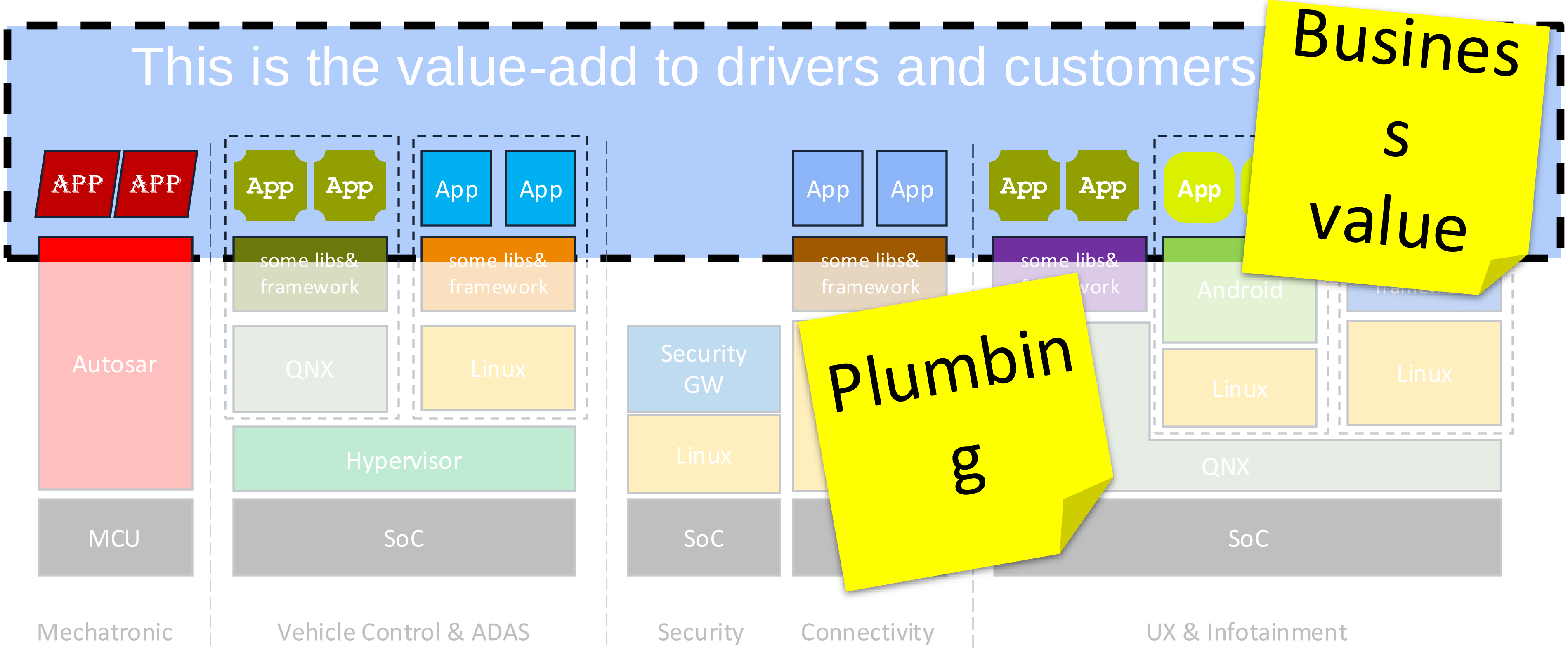
We have computers…



… now we want to run things!



Where is the customer value, what is just plumbing?

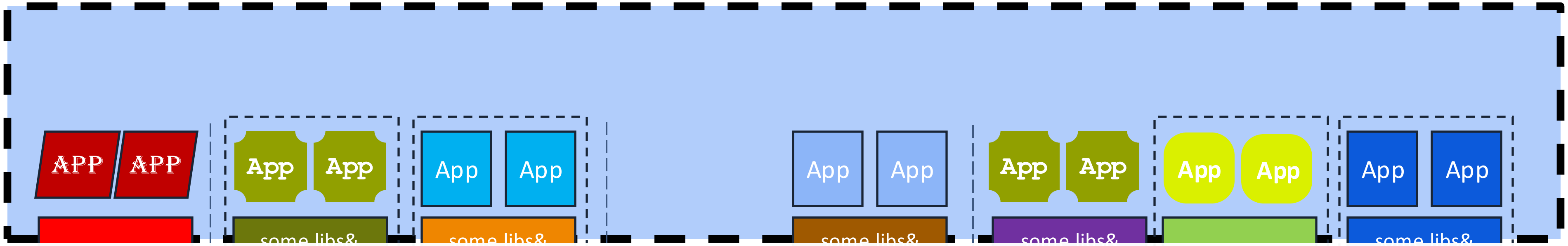


This is the value-add to drivers and customers

Busines
s
value

Plumbin
g

So this is all about Apps & Services!



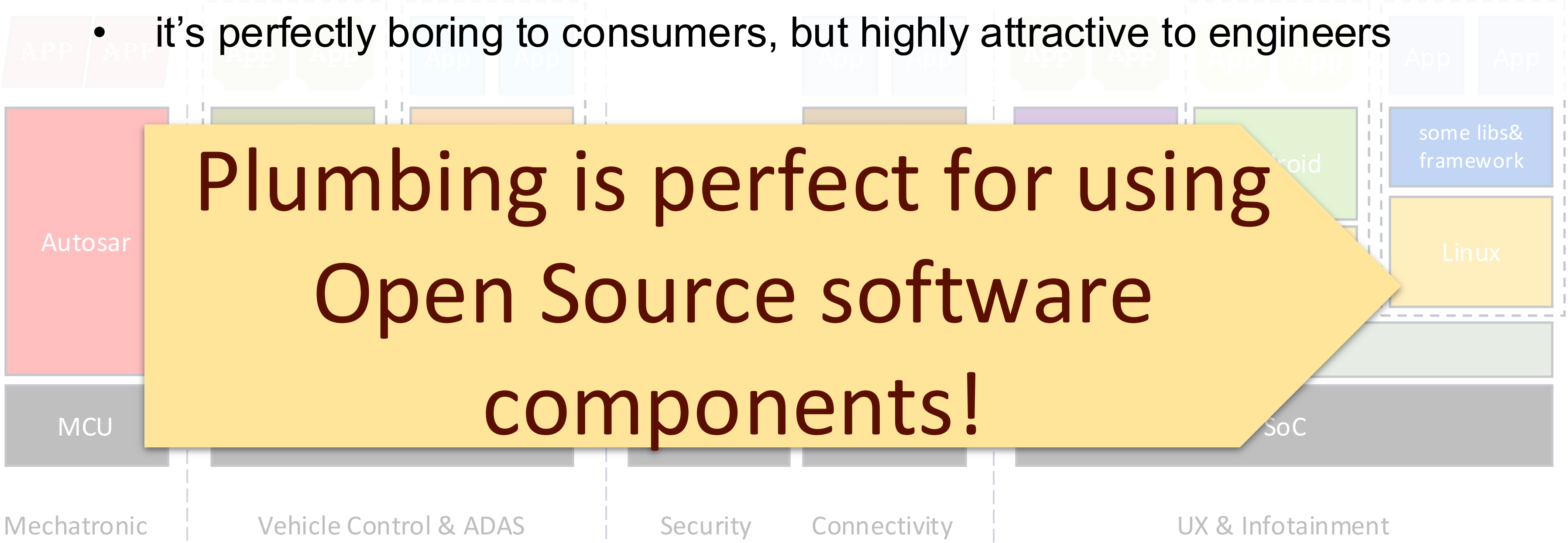
- Apps/Services represent the “vehicle surface” that users experience,
- they are built by very different kinds of organization (departments etc),
- who need to align around a common view of the vehicle,
- and provide long-term continuity of the business value they embody.

BUT: every tech stack brings its own plumbing

(programming model, vehicle view, communication paradigm, etc)

The deal with Plumbing (aka Software Infrastructure)

- Plumbing is mission critical, but not really differentiating
- it’s ubiquitous, but invisible
- it’s perfectly boring to consumers, but highly attractive to engineers



Plumbing is perfect for using Open Source software components!

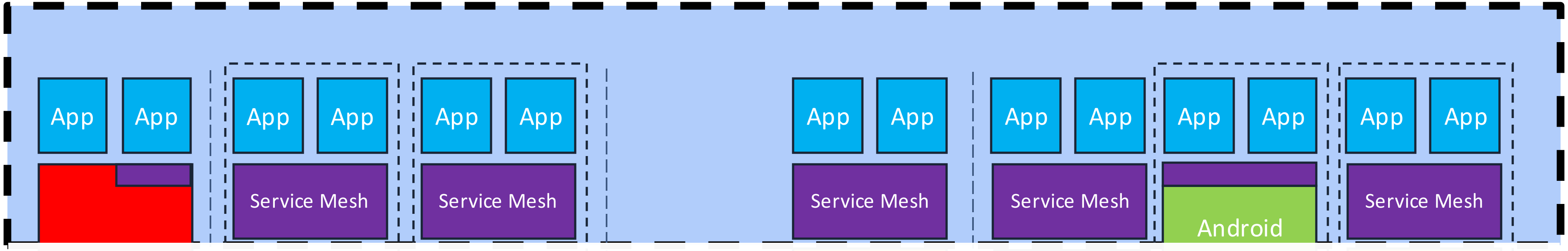
SDV™

Eclipse Software Defined Vehicle

uProto-what?

Free stock photos - PxHere - 1055658

How to do ”Apps and Services” successfully?



Plumbing needs to:

make communication simple and ubiquitous

→ Unified addressing, inter-domain pub/sub, transparent message routing

foster efficient long-term maintenance & evolution

→ Decouple programming model from proprietary, platform-specific libs & protocols

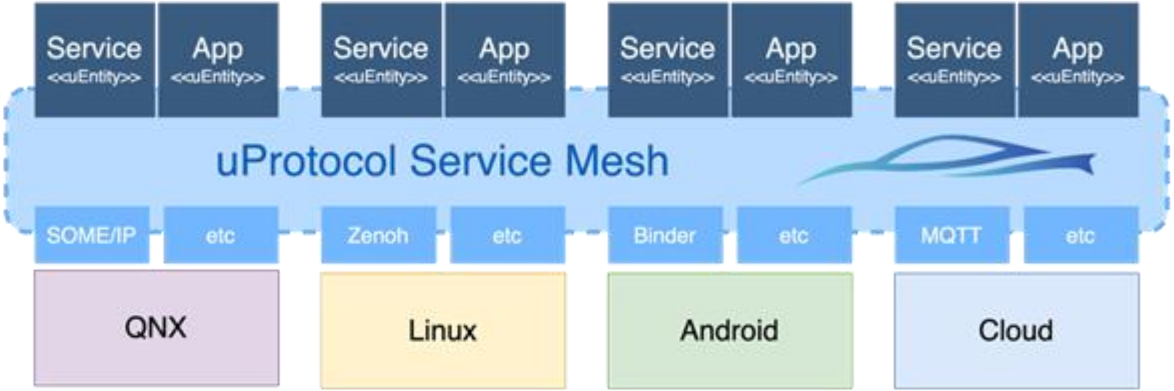
make App & Service development simple

→ Need ecosystem that caters to all kinds of programming languages, protocols, etc

There exist scores of OSes, communication stacks, middlewares, pubsub implementations, transport protocols, etc

We need something on top of all of that, providing us with:

- Unified Addressing
- Inter-domain message forwarding
- X-domain subscription tracking
- Cross-domain discovery
- Communication abstraction and programming model



→ We want an ***Open Source automotive Service Mesh***

SDV™

Eclipse Software Defined Vehicle

uProto-how?

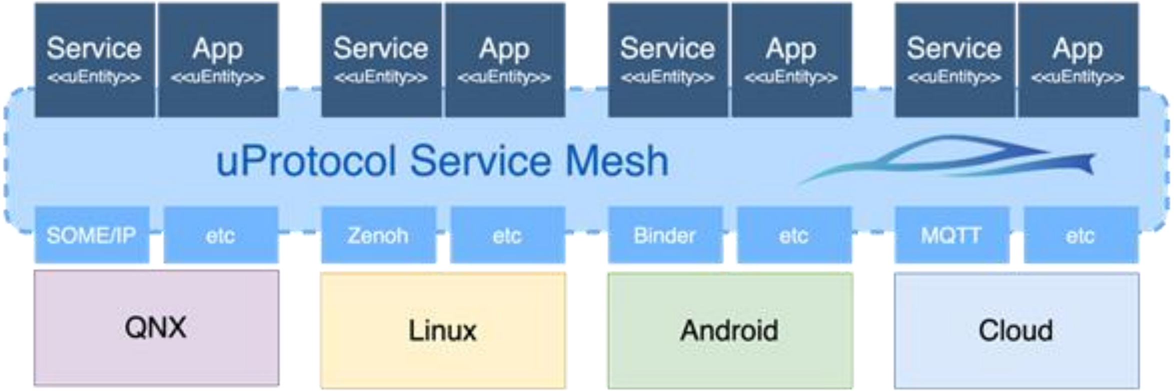
Free stock photos - PxHere - 1055658

uProtocol

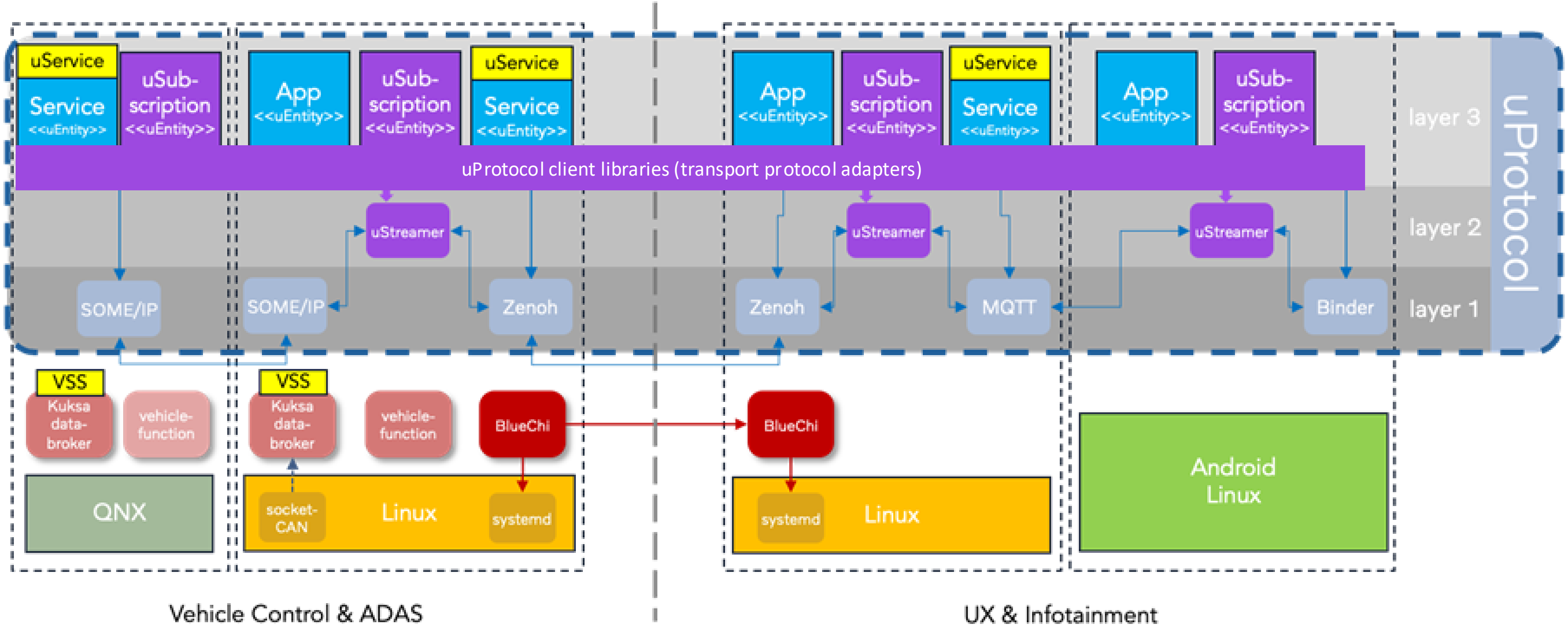
CONNECTING AUTOMOTIVE APPS AND SERVICES, EVERYWHERE

Feature spotlights 🔦

- Unified Addressing
- Inter-domain message forwarding
- X-domain subscription tracking
- Support for multiple programming languages and transport protocols
- Early adopter of Eclipse quality process initiative

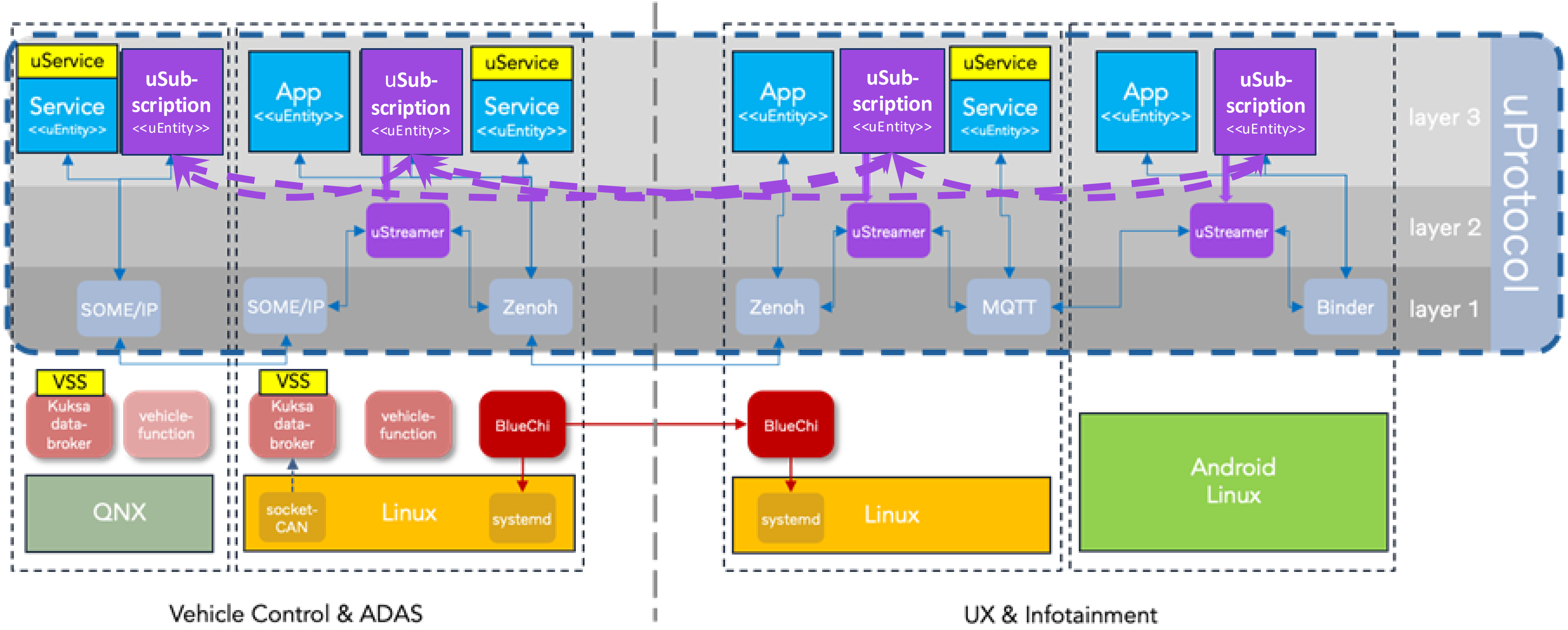


Unified addressing & Inter-domain message forwarding



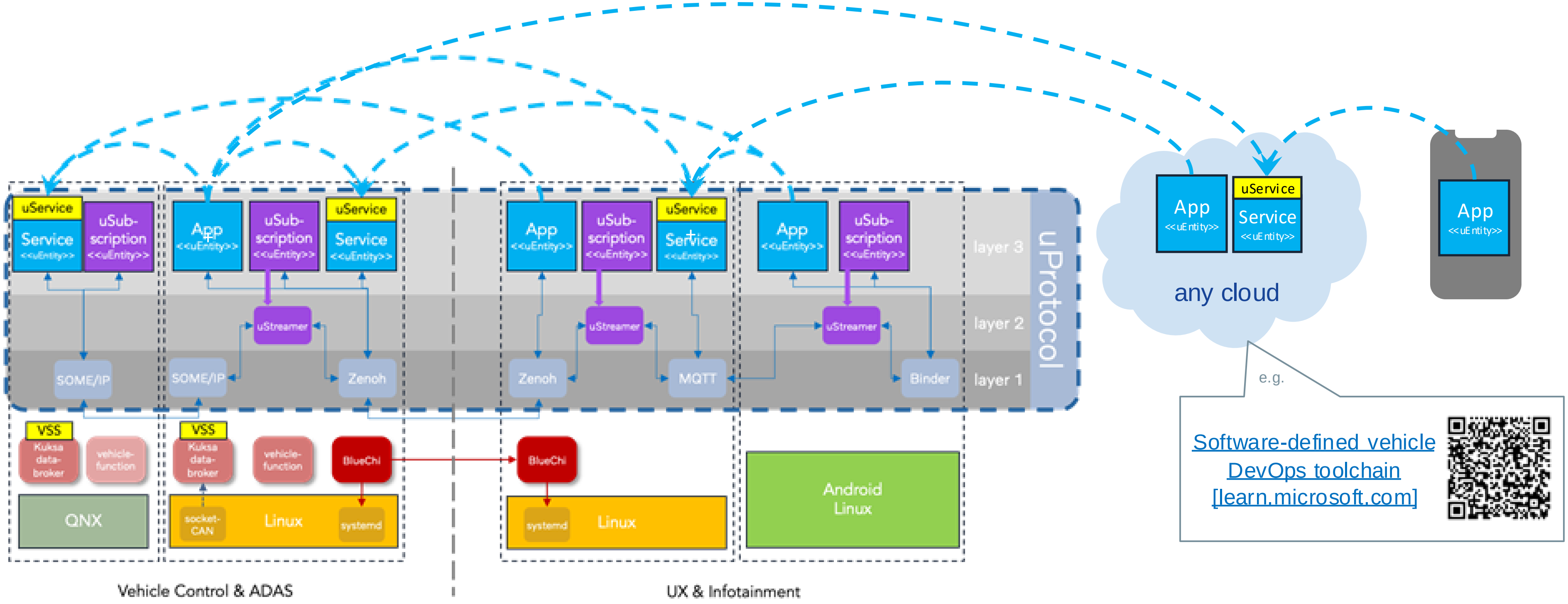
https://github.com/eclipse-uprotocol

X-domain subscription tracking



https://github.com/eclipse-uprotocol

Also: uProtocol extends to cloud/backend, mobile, etc



e.g.

[Software-defined vehicle DevOps toolchain [learn.microsoft.com]](https://learn.microsoft.com)

https://github.com/eclipse-uprotocol

SDV™

Eclipse Software Defined Vehicle

Thank You!

JOIN US ON GITHUB!

https://github.com/eclipse-uprotocol

https://github.com/eclipse-sdv-blueprints/service-to-signal