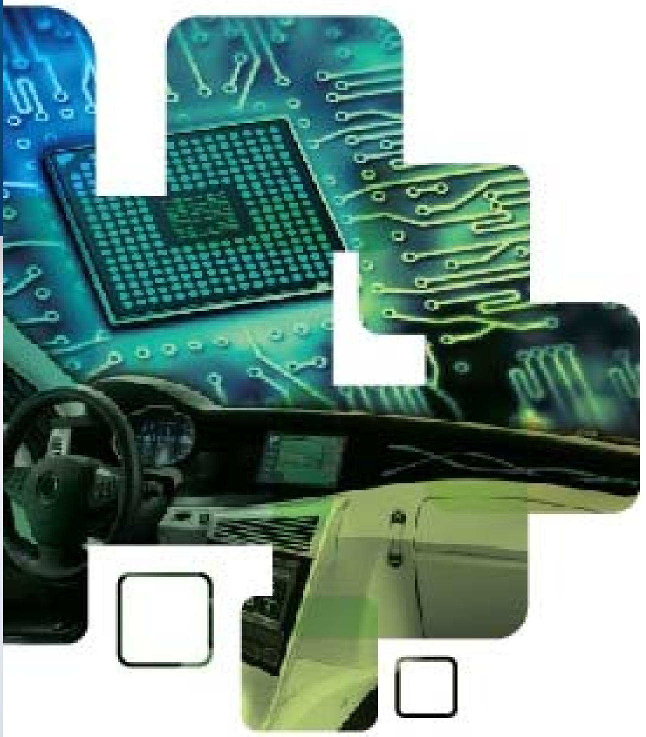


GENIVI®

AMM OPEN DAYS

RVI Expert Group Charter

2015-11-03

Magnus Feuer
Jaguar Land Rover

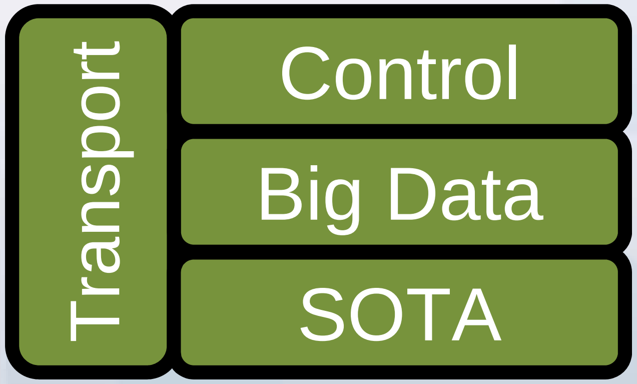
Proposed charter

Specify, Standardize, and Implement Core connectivity protocols and services between the IVI system and remote entities.

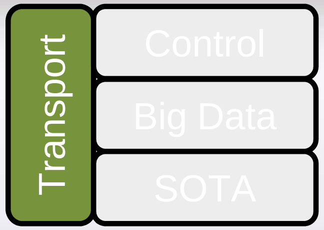
Use proven open source technologies to ensure ensure that all protocols and services can be implemented securely and robustly.

Collaborate with existing organizations (OMA, IEEE, etc) to ensure broad adoption and acceptance, and to avoid duplication and competition.

RVI Main Activities



Transport



Connectivity

- Utilize a wide array of data links to setup communication to and from vehicle, either P2P or via backend serve
- Provide encryption for secrecy, non repudiation, replay attack protection, etc
- Work with OMA, IEEE, and other organizations to standardize RVI and integrate existing communication standards

Authentication

- Prove the identity of communicating parties
- Use best-of-breed open source technologies to drive peer-reviewed security

Authorization

- Prove to remote parties the right to discover and invoke their services.

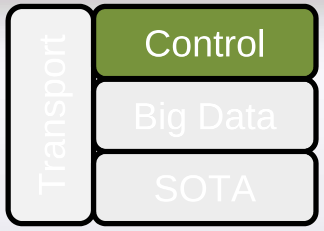
Service Discovery

- Announce services available to remote parties

Service Invocation

- Invoke services and report the result over unreliable data links that may change during execution
- Support retry and store & forward of service invocations to alleviate transient connectivity

RVI Control



Vehicle Integration

- Utilize Networking EG components to integrate with vehicle bus
- Use W3C-based signal standards to control vehicle

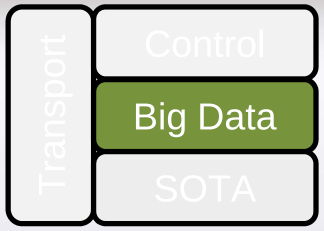
Service Protocol

- Define vehicle control protocols between vehicle and remote entities

Web Services

- Use W3C-based standards to define web services for remote vehicle control

RVI Big Data



Data collection

- Integrate with Genivi components to harvest data
- Use dynamically OTA-loadable code to securely collect and pre-process data

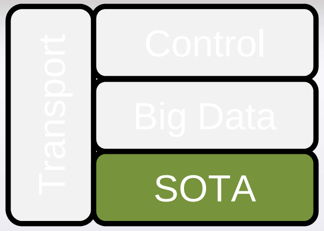
Reporting

- Specify and implement in-vehicle reporting services and their protocols

Data gathering

- Server-side data reception, storage, and web service access
- Work with big data industry actors to integrate analytics and data feeds into a larger ecosystem

RVI SOTA



Implement and standardize SOTA Client

- Integrate SOTA Client with System Infrastructure and its Software Management activities
- Implement existing transport protocol standards in addition to Transport

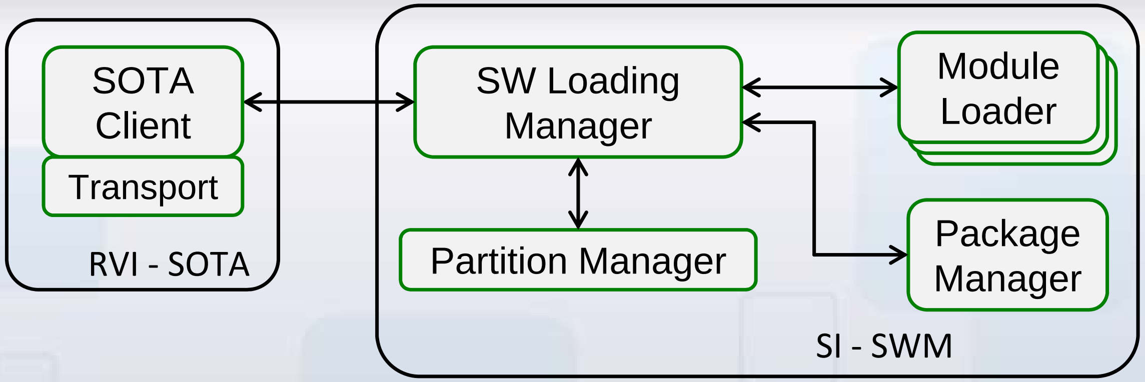
Implement and standardize SOTA Server

- Specify and implement reporting services and their protocols

Integrate with industry

- Work with vendors and community to drive RVI SOTA adoption in the automotive industry

RVI-SOTA vs. SI-SWM



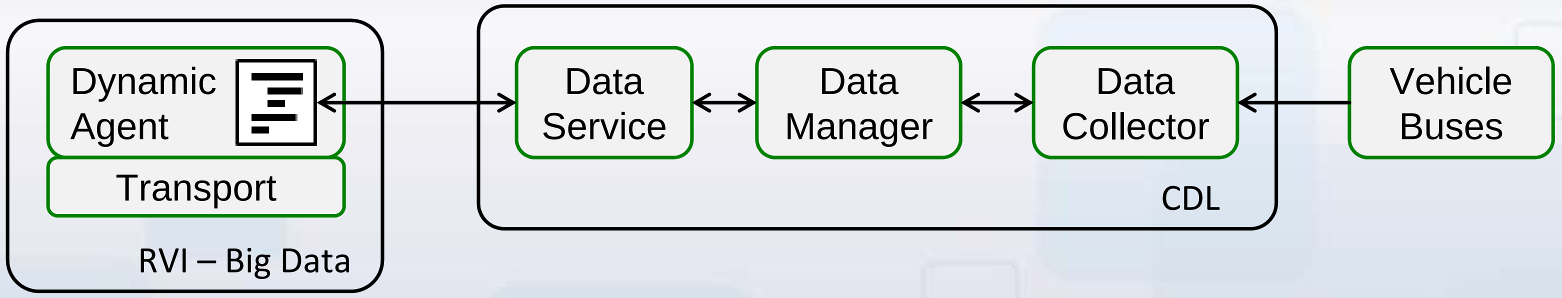
RVI - SOTA

- Contains SOTA Client, which uses Transport as a transport mechanism
- Interfaces SI Software Management through FrancaIDL

SI - SWM

- Software Loading Manager acts as a gateway interfacing SOTA Client while driving package management logic

RVI-Big Data vs. CDL



Car Data Logging (CDL)

- Data Collector harvests data from various vehicle buses and sensors
- Data is processed by Data Manager and is stored by Data Service

RVI Big Data

- Scripts executing inside the Dynamic Agent subscribe to CDL data points
- Scripts generate reports based on the received data and send those over RVI to backend server

Contributors

Role	Person	Company
Expert Group Lead	Magnus Feuer	Jaguar Land Rover
Expert Group Architect	TBD	Ericsson

Name	Company	Expertise	Participation
Naoki Ogishi	Harman	Data Analytics	TBD
Jasen Harada	Harman	Cloud / Server tecnologies	TBD
Vladimir Koshelev	Advanced Telematic Systems	SOTA	TBD
Other	TBD – Pending call for participation		

From requreiements to industry acceptance

Vehicle Requirement Phase

- Gather high-level purpose, objectives, and requirements from Genivi members

Discovery Phase

- Explore requirements and APIs through coding and experiments carried out in public

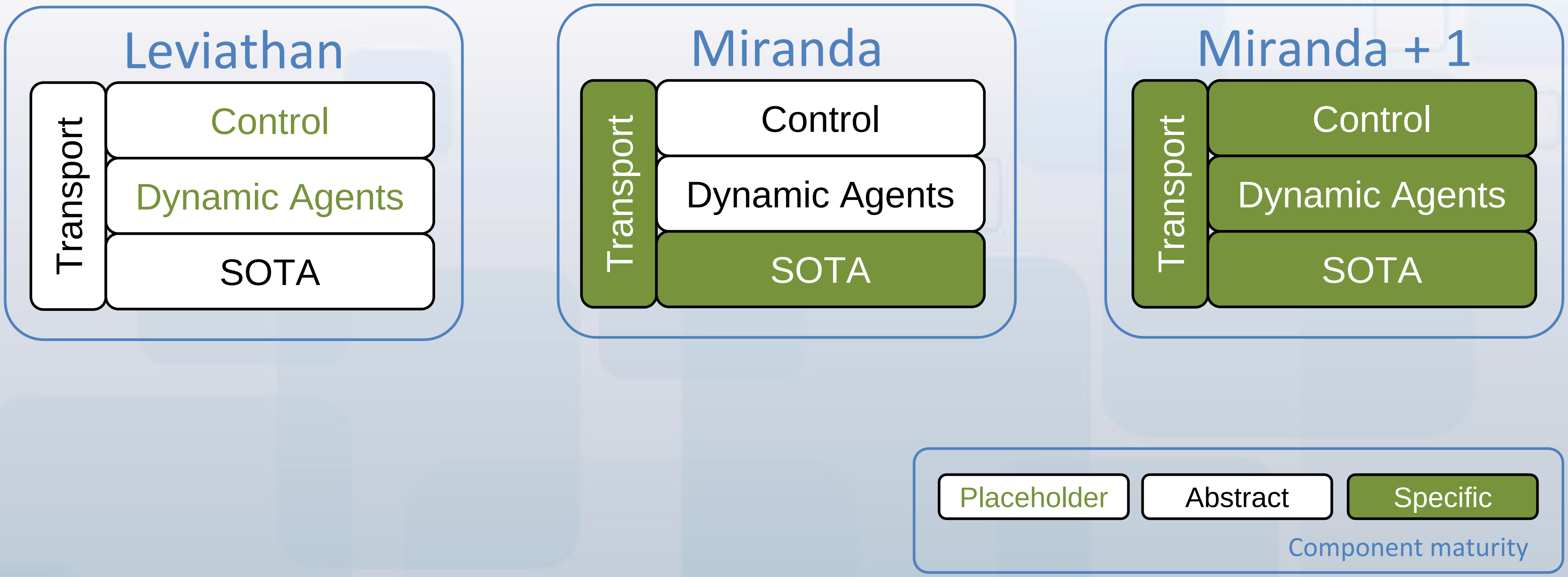
Specification Phase

- Use experience and consensus from discovery phase to create abstract and specific components

Acceptance Phase

- Use deliverables from specification phase to approach industry to gain wider acceptance

Proposed roadmap



GENIVI

Proposed scope

Transport protocol	• Encoding/Decoding. Data link management. Service Discovery. Service Invocation. Store and Forward. Retry.
Security	• Authentication, Authorization, Encryption, Integrity, Non-repudiation, Denial of service.
Provisioning	• Key, token, and certificate management, device provisioning and revocation. Backend provisioning API.
Sensor data reporting	• On-board processing of sensor data. Transmission. Backend data access API.
OTA updates	• Package management, queueing, transmission, validation, and reporting. Backend software management API.
Remote Control	• Remote interface to vehicle systems. Backend control API.

Thank You

Additional slides describing Transport, SOTA, Control, and Big Data

RVI - Transport

Open Source Automotive Connectivity

Magnus Feuer

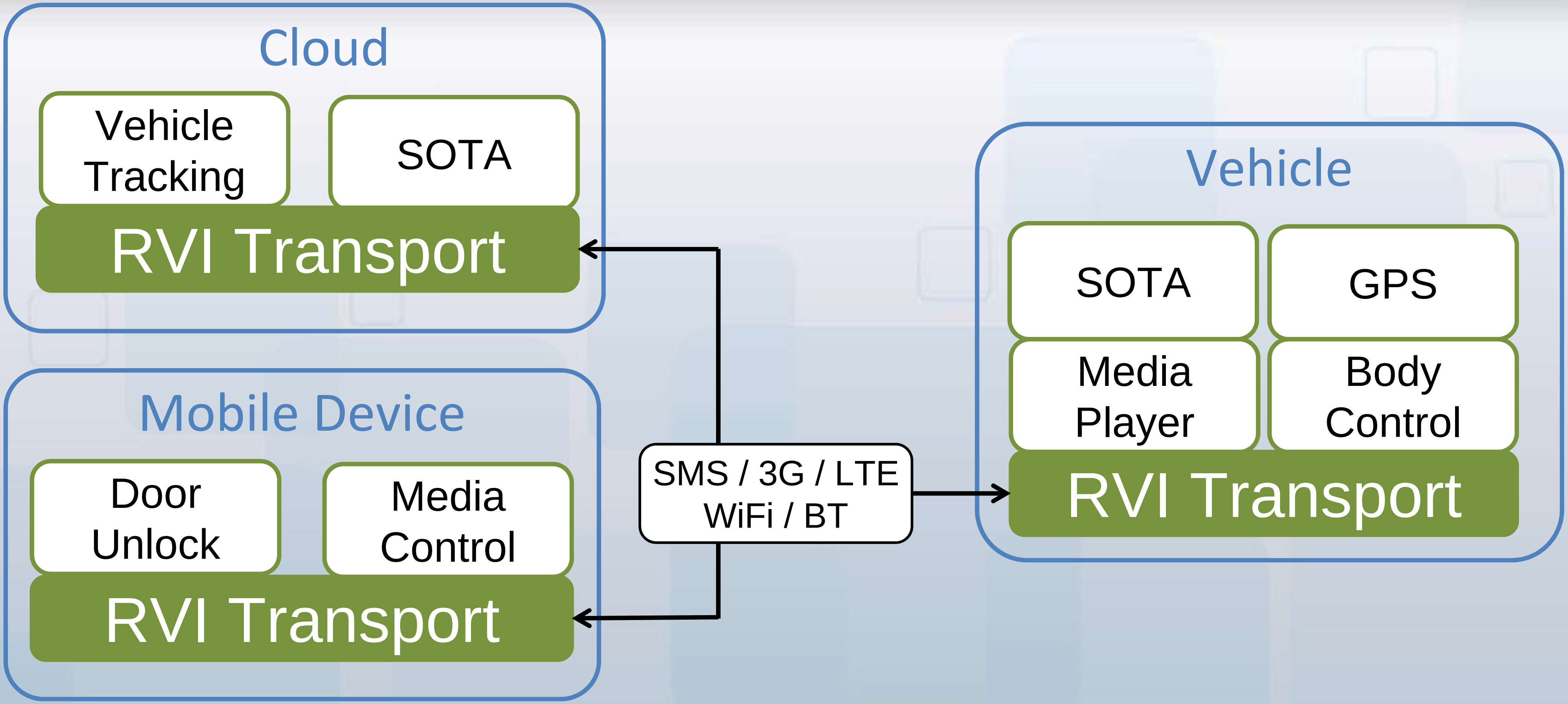
Head System Architect – Open Source Projects
Jaguar Land Rover Open Software Technology Center

Technical Scope

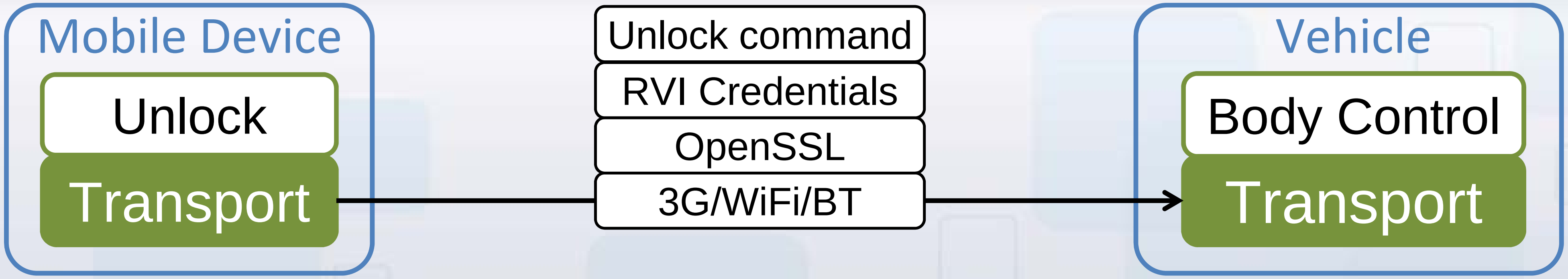
Provide P2P based provisioning, authentication, authorization, discovery and invocation between services running inside and outside a vehicle.

- **P2P**
  Internet connection not required for two peers to exchange services.
- **Provisioning**
  Add, delete, and modify services and network nodes.
- **Authentication and Authorization**
  Proves that a service is who it claims to be, and has the right to invoke another service.
- **Discovery and Invocation**
  Allow two peers to exchange services that the other party is authorized to use, and invoke those services over any data link.

Schematics

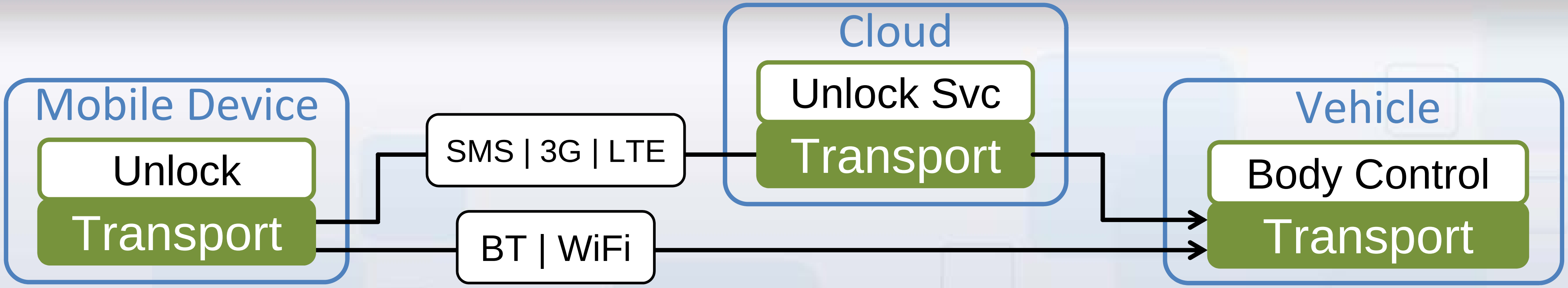

Cloud
Vehicle Tracking
SOTA
RVI Transport
Mobile Device
Door Unlock
Media Control
RVI Transport
SMS / 3G / LTE
WiFi / BT
Vehicle
SOTA
GPS
Media Player
Body Control
RVI Transport

Security



- **OpenSSL**
  TLS provides core eavesdropping and MITM attack protection
- **RVI Credentials**
  Signed by root server to prove device authenticity and its right to invoke unlock on the given vehicle
- **Unlock**
  Will only be accepted by vehicle if validated certificate specifies device's right to invoke unlock command

Robustness

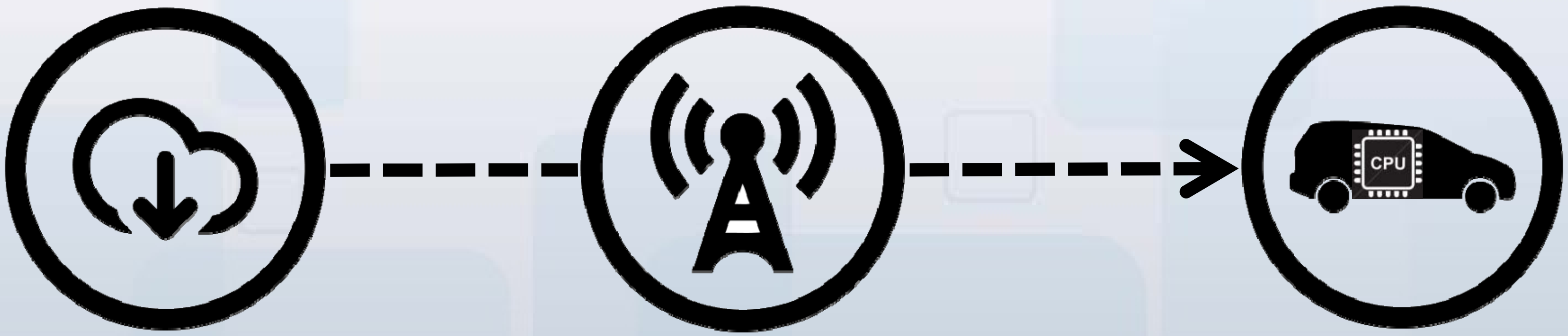


- **Multi-pathed commands**
  Commands can be tried over multiple data links, both as peer-to-peer and via backend server.

- **Traffic prioritization**
  Forces high-bandwidth services to use WiFi while allowing mission-critical services to use 3G and SMS.

- **Multi-protocol**
  A single service (such as unlock) can employ different protocols and data links depending on the targeted vehicle type.

SOTA Software Over The Air



Purpose

Use the cloud

to install software

on any vehicle

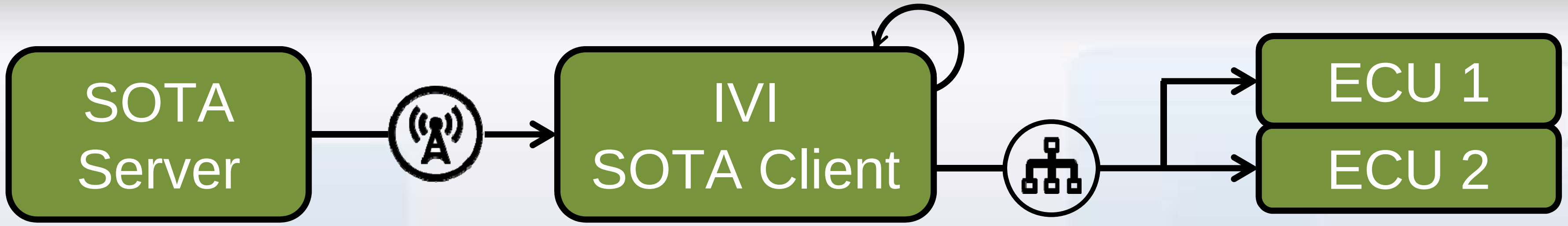
The user's story

When owner starts the car in the morning, an update icon is displayed on the head unit.

After learning that the update allows the car to analyse its daily commute to optimize drivetrain management, saving 6% fuel, the owner installs it.

When the owner drives home after work, the new app is available showing how the vehicle is adapting to the commute to minimize fuel consumption.

Overview

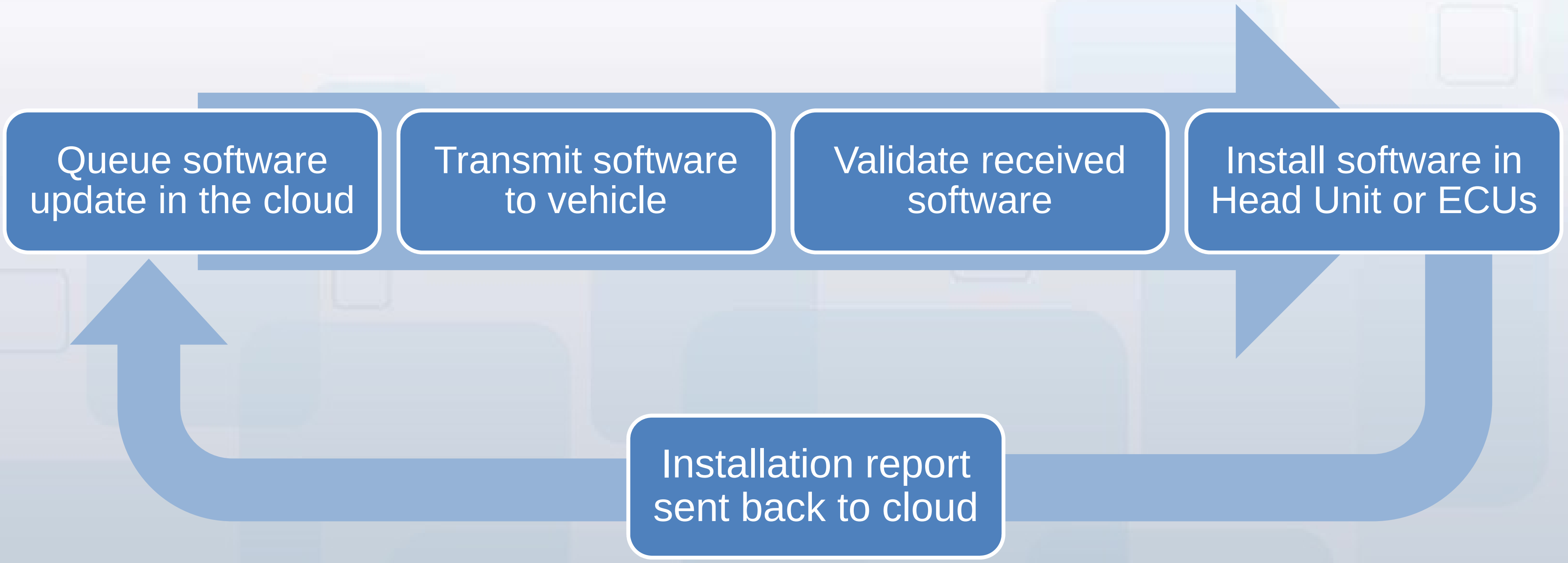


SOTA Server [Cloud]
Manages software images and vehicle database

SOTA Client [Head Unit]
Validates and installs software received from SOTA Server

Electronic Control Units
Installs firmware forwarded from SOTA Client

Use Case


Queue software update in the cloud
Transmit software to vehicle
Validate received software
Install software in Head Unit or ECUs
Installation report sent back to cloud

Feature - Benefit

Open Source	• Free to use. Free to modify. Free to integrate.
Reference end-to-end implementation	• Ready to deploy for evaluation and prototypes
Billing cycle aware	• Avoids data plan overrun costs
Mixed asset management	• One system to handle legacy and new fleets
Genivi reference system	• Vetted architecture with broad industry support
Logistic system separated from transport	• Allows for integration with existing production system
Vehicle configuration-based targeting	• Installs updates only on vehicles with given specification

Control Demo: Mobile Unlock

Magnus Feuer

Jaguar Land Rover
Open Software Technology Center

Purpose

Use any smartphone

to unlock

any vehicle

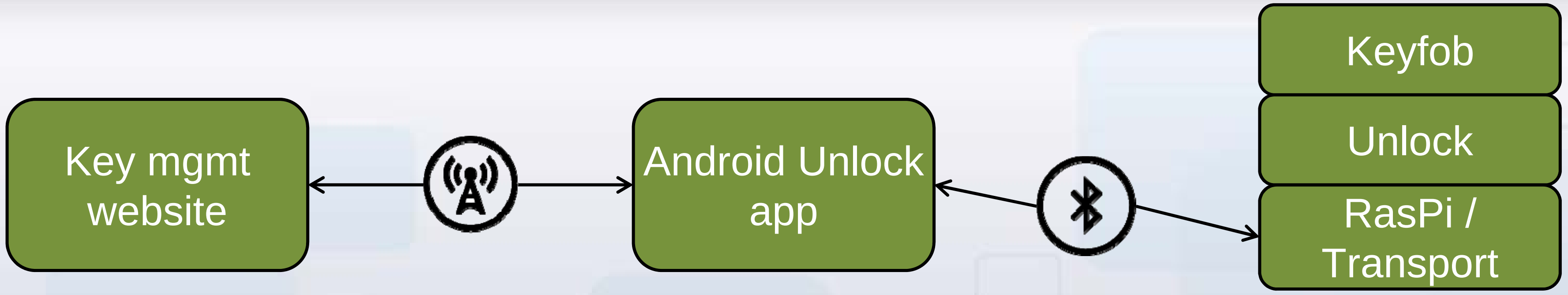
The user's story

An owner is travelling when a friend emails her to ask if he can borrow her car.

The owner brings up her guest driver app and sends a virtual car key, valid over the weekend, to her friend.

After the friend receives the key, he can walk up to the car to have his phone automatically unlock and start it.

Overview – Demo system

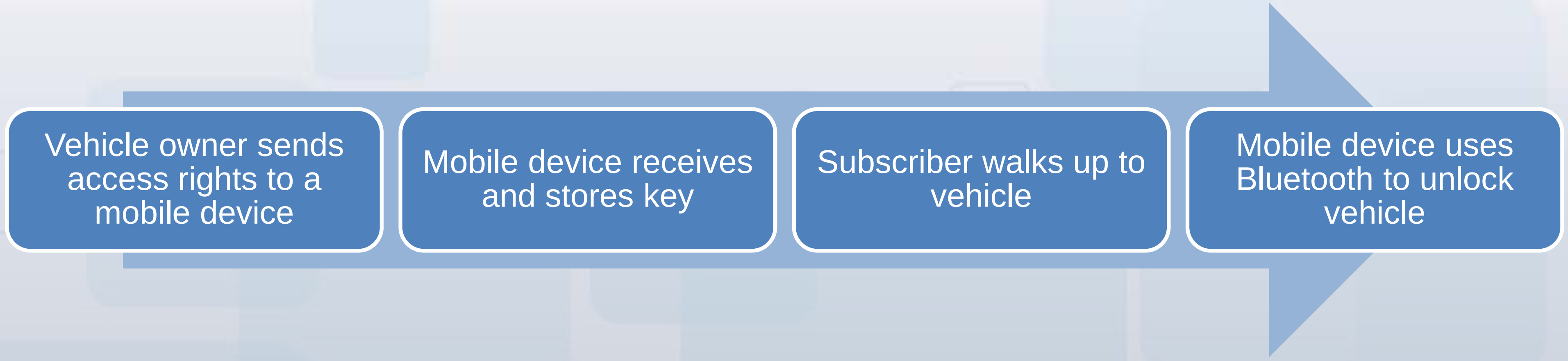


Key management website [Cloud]
Allow vehicle owners to provision and transmit keys to any android app

Android unlock app [Mobile Device]
Runs a background service using keys to lock/unlock select vehicles

Unlock / Keyfob [Raspberry Pi]
Receives, validates, and executes received lock/unlock command using a keyfob

Use Case


Vehicle owner sends access rights to a mobile device
Mobile device receives and stores key
Subscriber walks up to vehicle
Mobile device uses Bluetooth to unlock vehicle

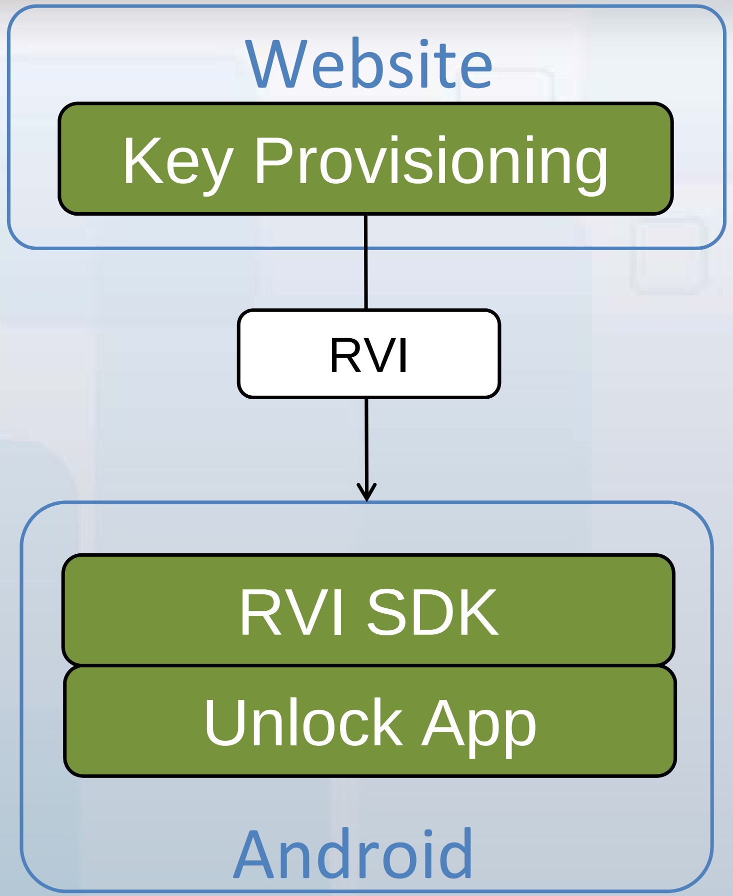
Key provisioning

A guest driver is provisioned with vehicle id and target Android device

An RVI credential containing target service (jlr.com/[VIN]/unlock) is created and signed by the root certificate

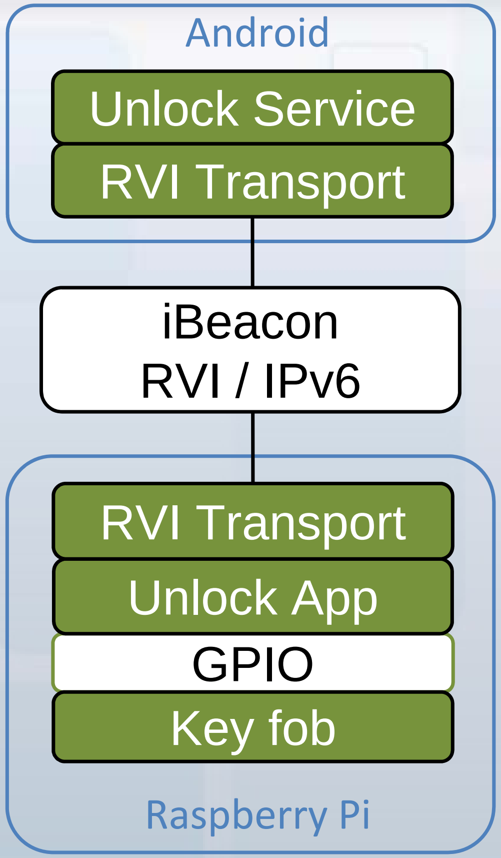
RVI credential is transmitted from website to target Android device using RVI

Unlock app on Android device receives credential and stores it as an RVI certificate

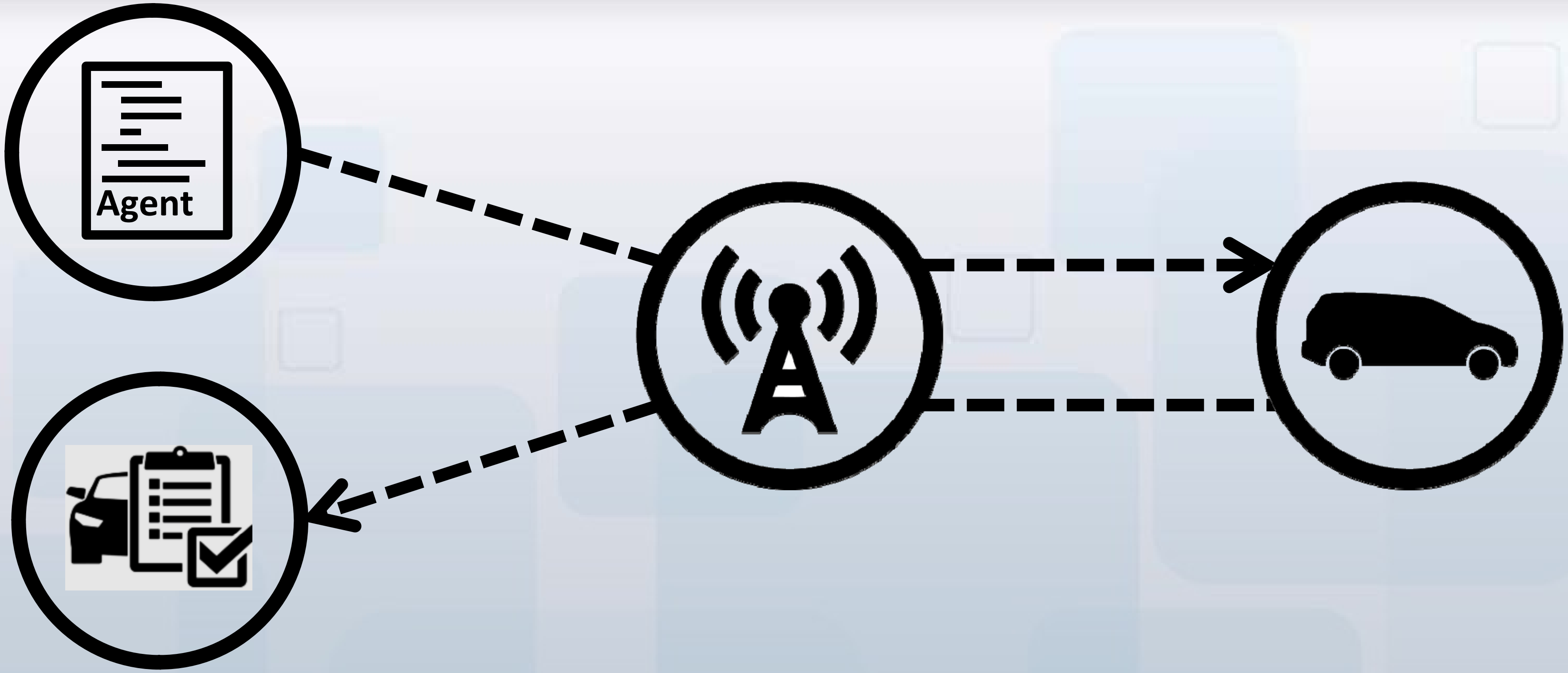


Vehicle

1. Unlock service detects vehicle presence
2. Unlock service sends unlock command to Android RVI
3. RVI forwards command, with provisioned credentials, to Raspberry RVI using Bluetooth
4. Raspberry RVI validates command and forwards it to Unlock app
5. Unlock app manipulates GPIO pins to press the unlock button on key fob



RVI Big Data – Dynamic Agents



Purpose

Deploy remote probes

to monitor vehicles

and report results

The user's story

A young entrepreneur comes up with an idea of letting people monitor their shared cars to make sure they are not abused.

By purchasing the V2XBoard and downloading the RVI – Dynamic Agent starter pack, she can develop her app using her own car as a test platform.

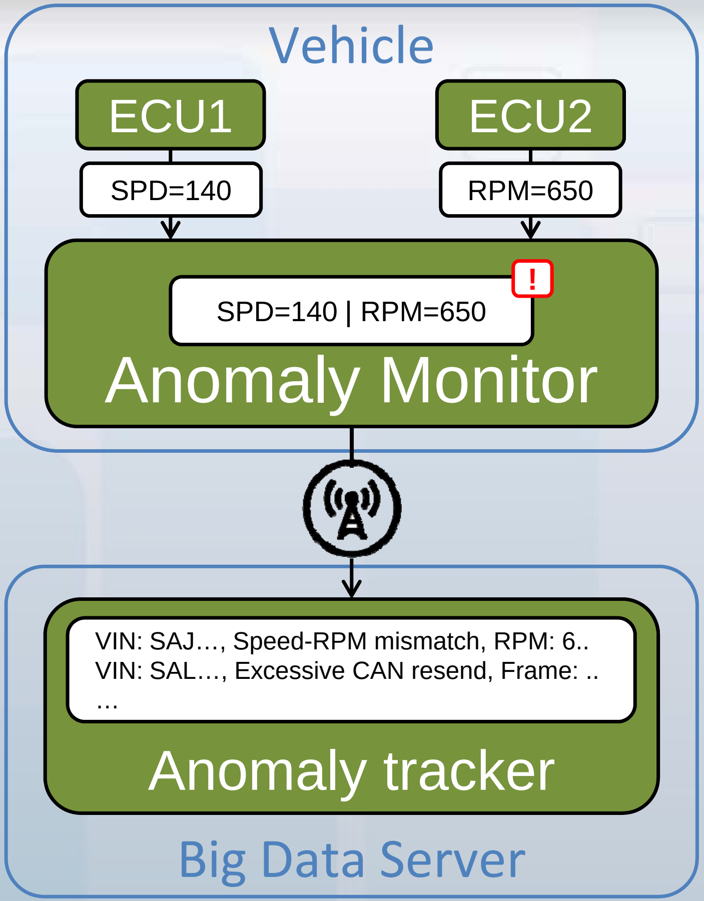
The entrepreneur demonstrates the finished application to an TaaS-oriented OEM who can bring it to production quickly since they both use the same connectivity stack.

Use case: Health Monitor

Anomaly monitor agent reads data from vehicle buses

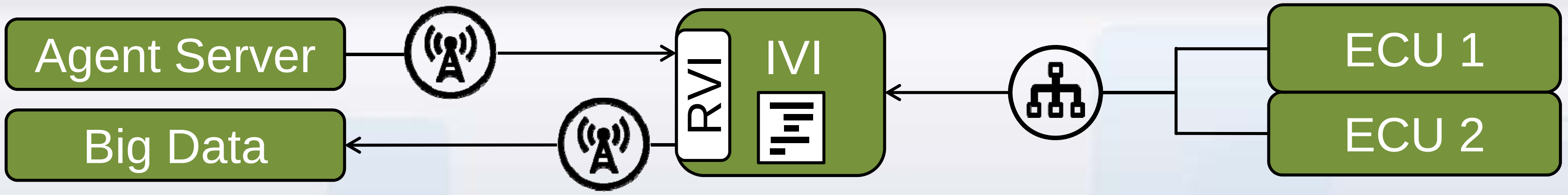
Any anomalous data detected by the agent is reported to the big data server

Big data server profiles all incoming data to generate failure trends and analysis



VIN: SAJ…, Speed-RPM mismatch, RPM: 6..
VIN: SAL…, Excessive CAN resend, Frame: ..
…

Overview – OEM solution

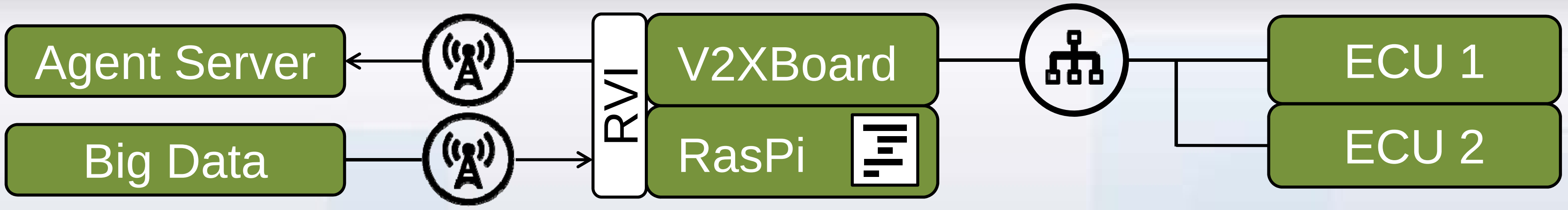


Agent Server [Cloud]
Distributes dynamic agents over the air, using RVI, to vehicle TCUs

IVI [Head Unit]
Executes dynamic agents in background to monitor ECU data streams

Big Data [Cloud]
Receives and analyses reports transmitted from dynamic agents through RVI

Overview – Entrepreneur solution



Agent Server [Cloud]
Distributes dynamic agents over the air, using RVI, to vehicle TCUs

Raspberry Pi / V2XBoard
Runs RVI and agent. V2XBoard for OBD2/CAN integration and M2M communication

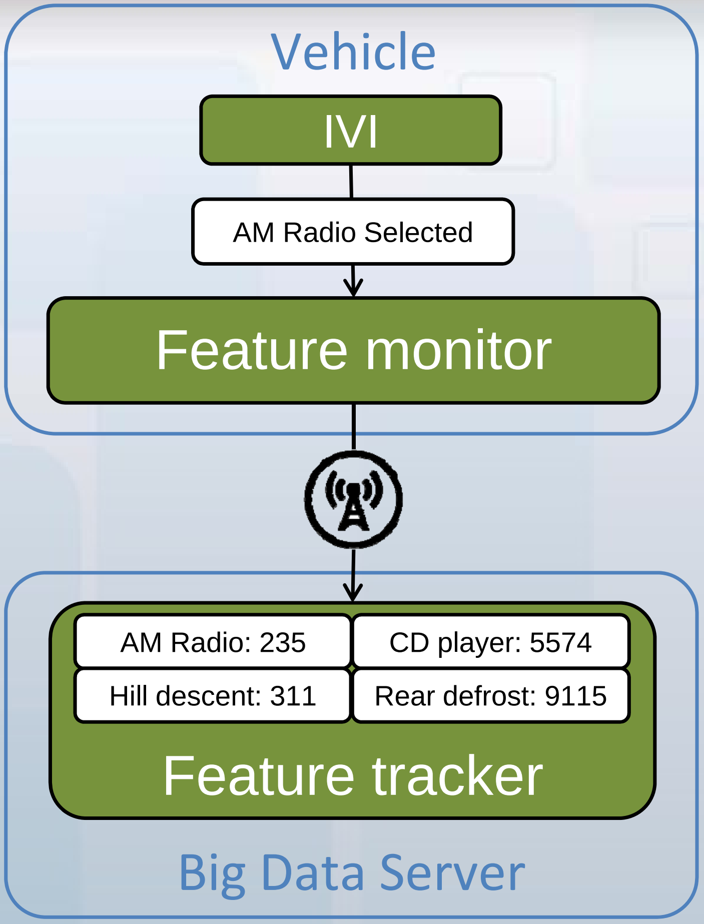
Big Data [Cloud]
Receives and analyses reports transmitted from dynamic agents through RVI

Use case: Feature usage

Feature monitor records all in-cabin feature usage

Periodic reports are sent by the feature monitor to the big data server

Feature tracker compiles report of fleet-wide feature usage

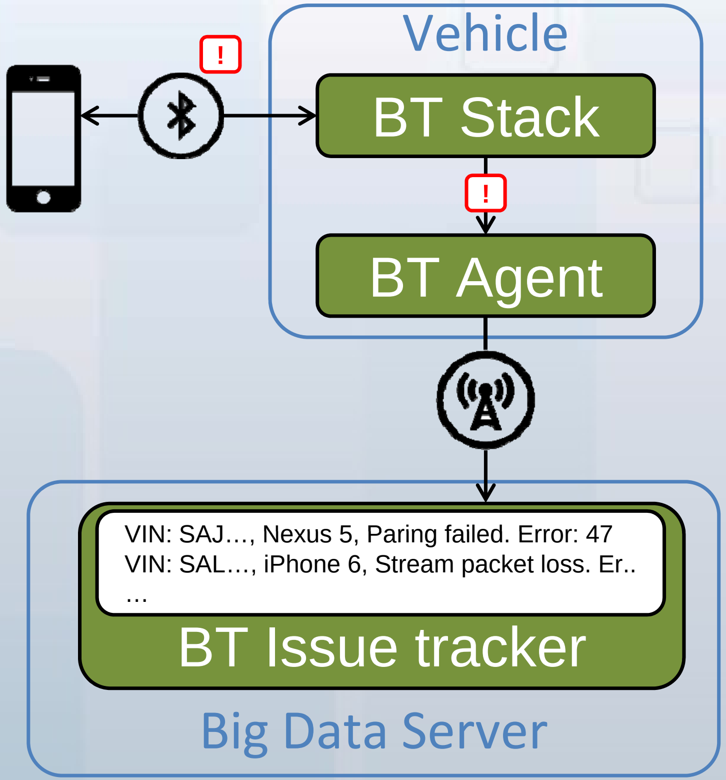


Use case: BT issue tracker

Bluetooth issue agent monitors pairings, streaming, and other possible failure points in the IVI BT stack

Any BT error is logged with circumstances and equipment ID

Error reports are transmitted to backend server for further analysis and bug fixes



VIN: SAJ…, Nexus 5, Paring failed. Error: 47
VIN: SAL…, iPhone 6, Stream packet loss. Er..
…