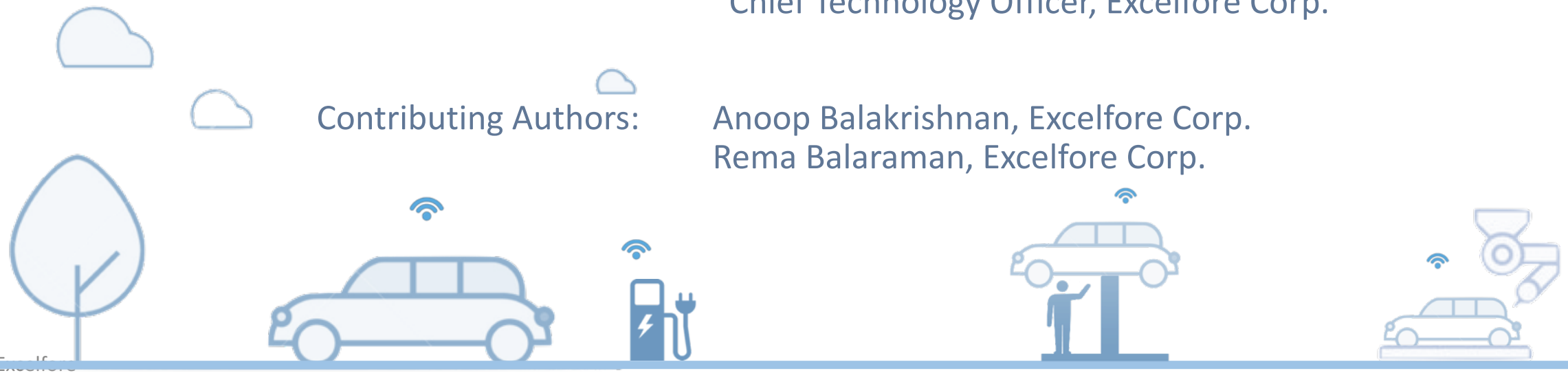
eSync Architecture and Programming Model for OTA and Diagnostics Reaching Non-Ethernet Devices Over an Ethernet Backbone

Presented by: Shrikant Acharya
Chief Technology Officer, Excelfore Corp.

Contributing Authors: Anoop Balakrishnan, Excelfore Corp.
Rema Balaraman, Excelfore Corp.



Agenda

1. Considerations – Objectives and Constraints
2. Architecture Review
3. Protocols, System Requirements, Security
4. Use Case Examples

eSync System Design Objectives

- Reach
  - From Cloud to End Device – Across Various Automotive Sub-Networks
- Bi-Directional
  - Pipeline for Data Push and Data Pull
  - Push Over-the-Air (OTA) Updates to the Vehicle
  - Pull Diagnostic and Telematics Data from the Vehicle
- Highly Secure
  - Vehicles can not be “Spoofed” or Compromised with Spurious Updates
  - Cloud Server can not be “Spoofed” with Spurious Vehicle Data
- Scalable
  - Scales to Many Devices in One Vehicle
  - Scales to Many Different Vehicle Configurations
  - Scales to Millions of Vehicles

Important Design Constraints

- Downtime
  - Full Vehicle Update Cycle Must Minimize Vehicle Downtime
- Resilience
  - Must be Resilient Against Errors / Interruptions in Over-the-Air Transmissions
- Efficient
  - Must Be Flexible for Different Processing and Memory Resources in Legacy ECUs
- Safe
  - Functional Safety Considerations, as Defined in ISO 26262 (ASIL levels)

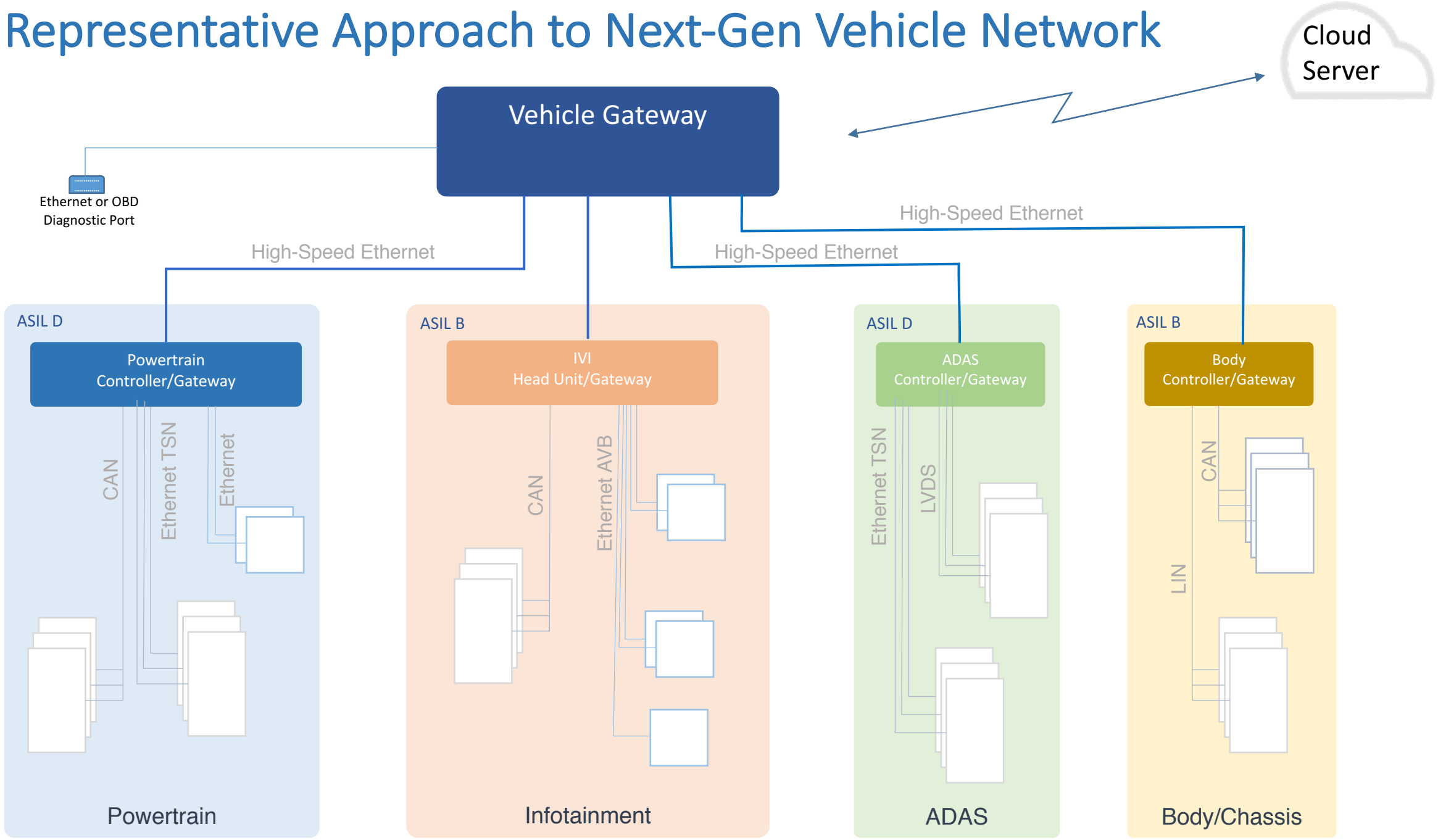
Important Considerations on Safety and Robustness

- ISO26262 Requirements:
  1. Non-Critical: The OTA Update System Does Not Reach Critical Elements at All
     - or -
  2. All Critical: The OTA Update System, and the Entire In-Vehicle Network, Operate Entirely as a Critical System
     - or -
  3. Isolate Critical: The In-Vehicle Network and the OTA Update System Isolate Critical and Non-Critical Elements of the Separate ASIL domains
     - Requires Parallel, Separate OTA Paths
- Robustness
  - Design for Modular Component Integration
  - Keep Up with Current Techniques by Using Latest Standards on Security and Network Protocols

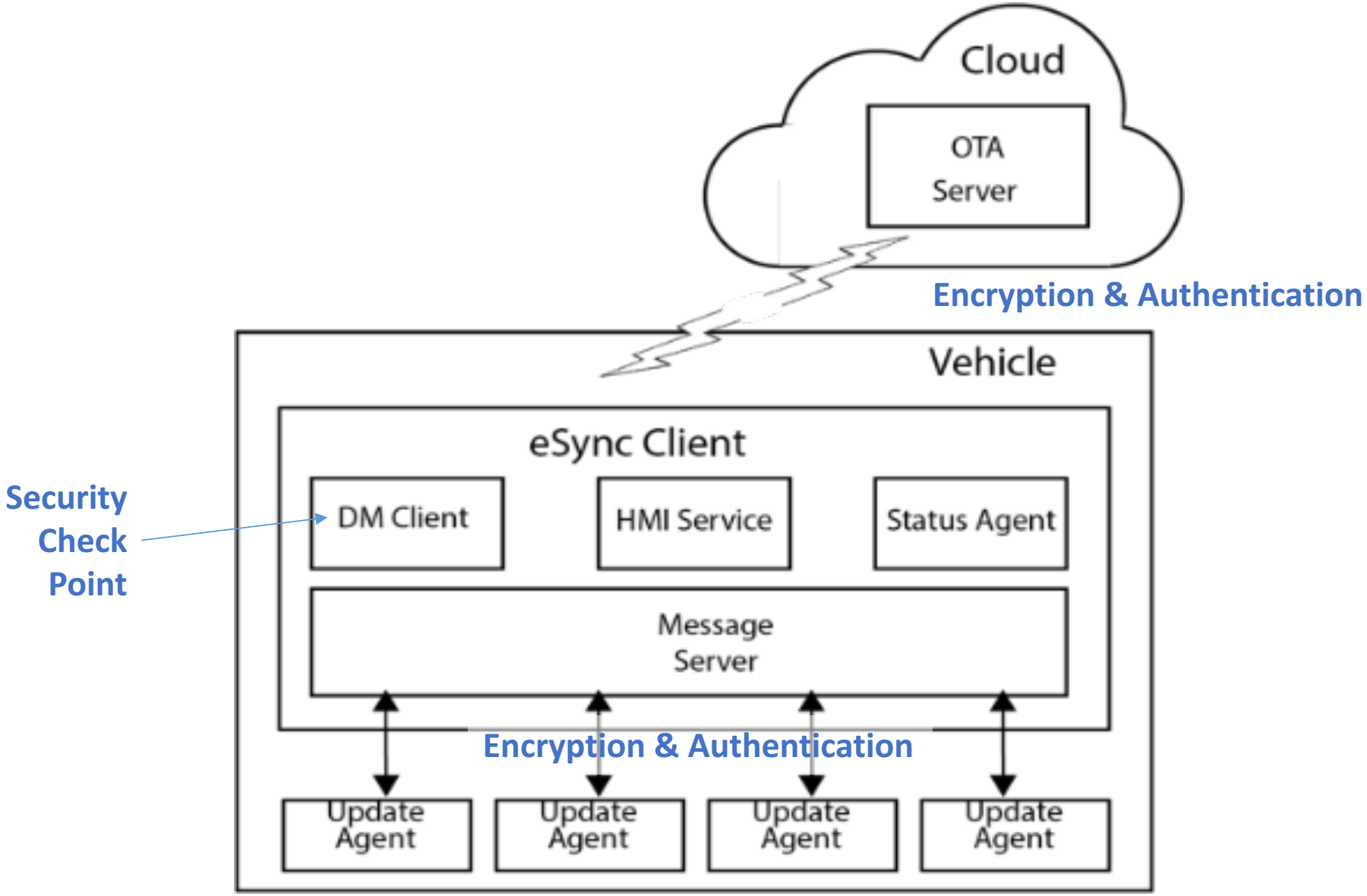
Agenda

1. Considerations – Objectives and Constraints
2. Architecture Review
3. Protocols, System Requirements, Security
4. Use Case Examples

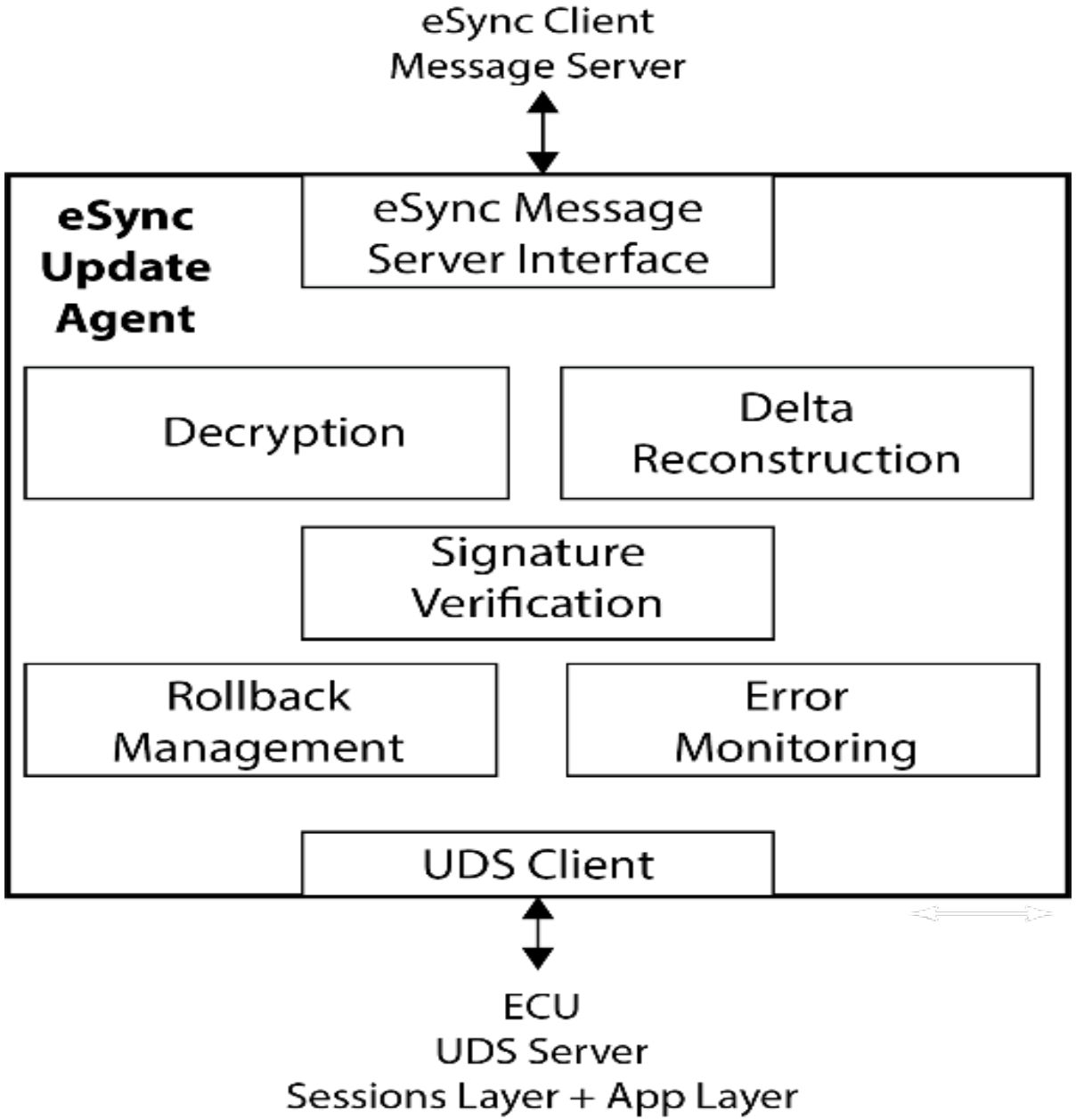
Representative Approach to Next-Gen Vehicle Network



The eSync System Architecture

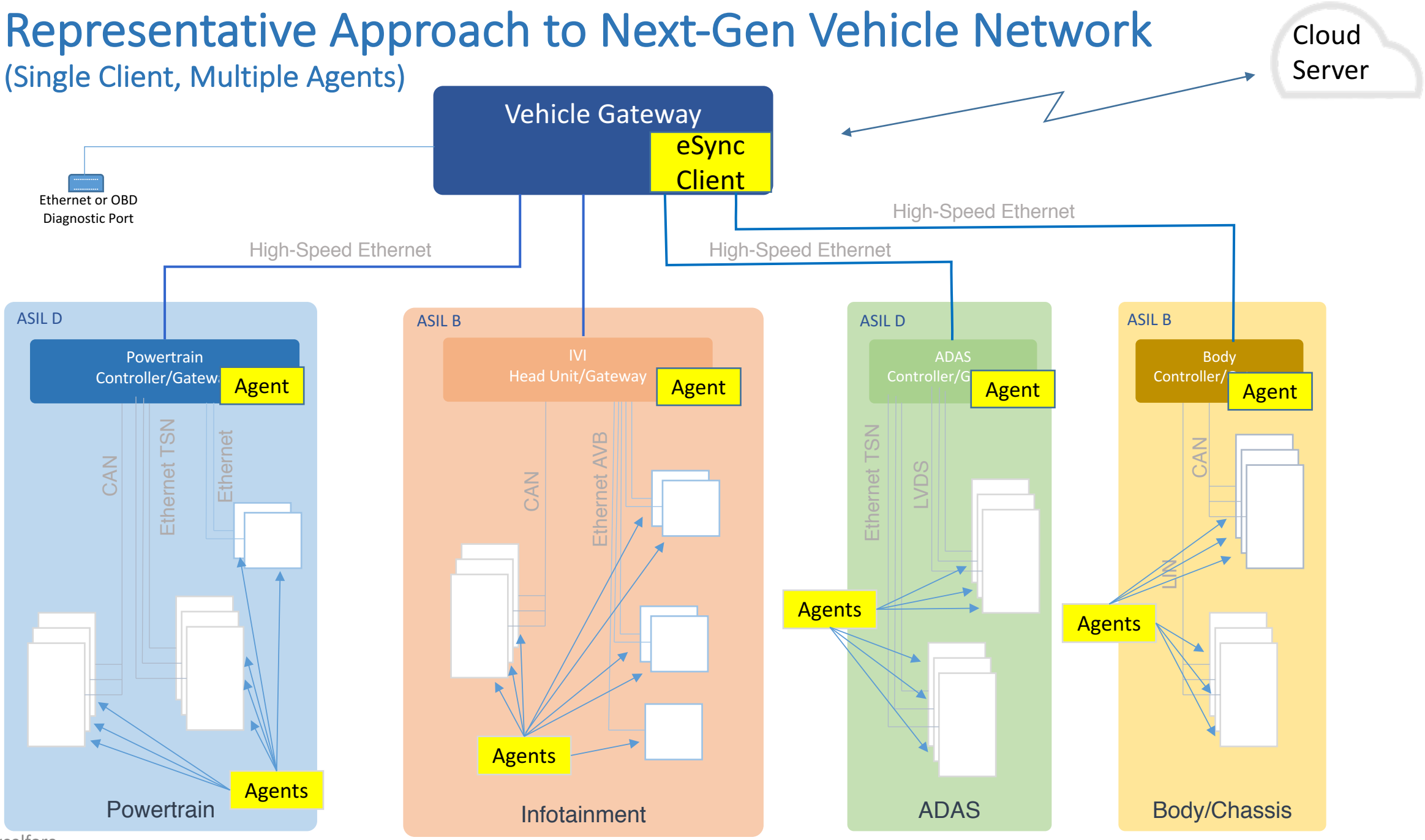


Update Agent

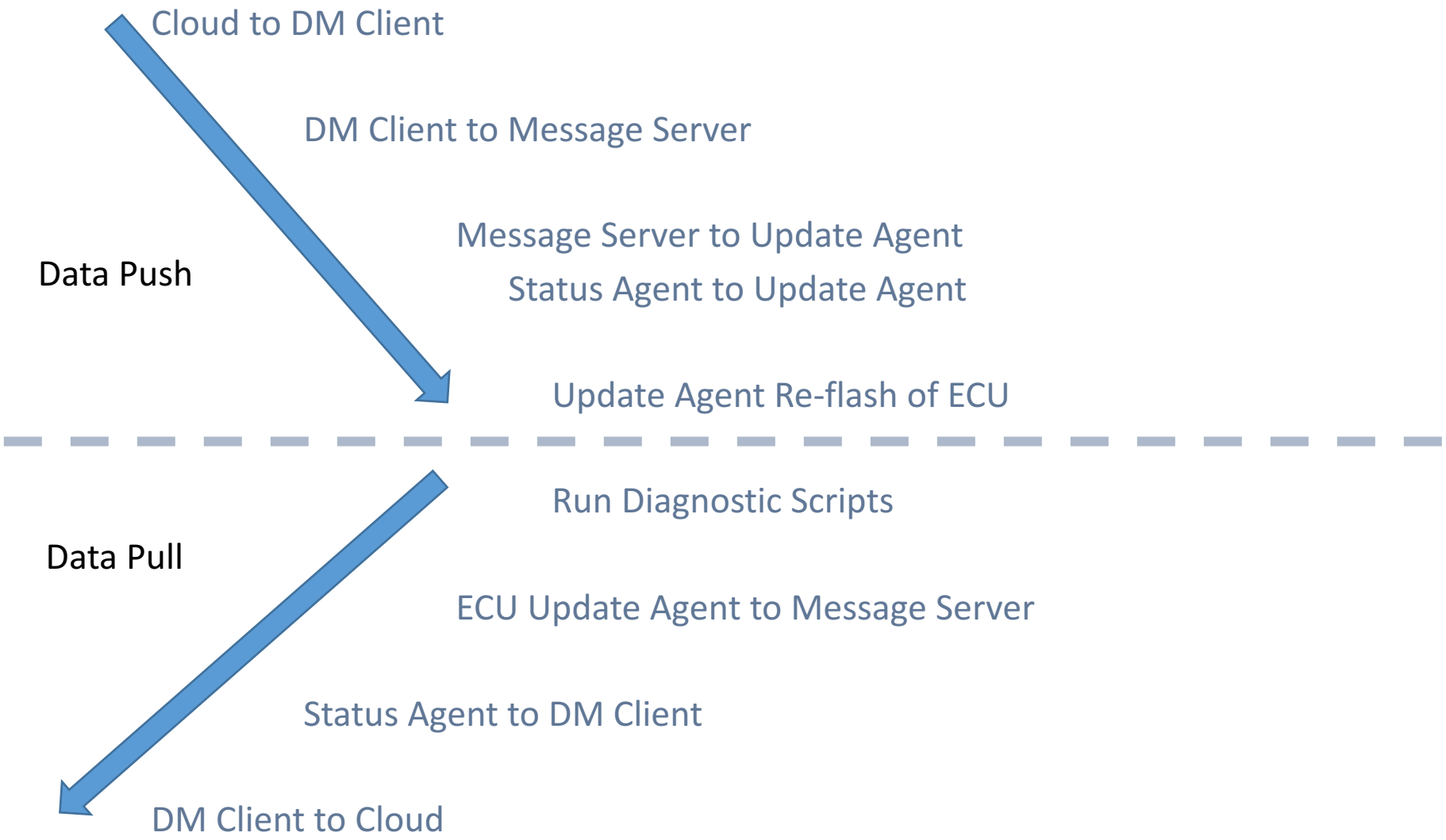


Representative Approach to Next-Gen Vehicle Network

(Single Client, Multiple Agents)



Operational Modes of OTA



Agenda

1. Considerations – Objectives and Constraints
2. Architecture Review
3. Protocols, System Requirements, Security
4. Use Case Examples

UDS Server Command Sequences

UDS Sessions Layer

1. Set the UDS server into program mode
2. Reset to new mode
3. Request Seed*
4. Send Key*
5. Transfer Data† (multiple data transfers)
6. Erase Memory†
7. Verify Memory
8. Set to Normal Mode
9. Reset to Normal Mode
10. End of Procedure

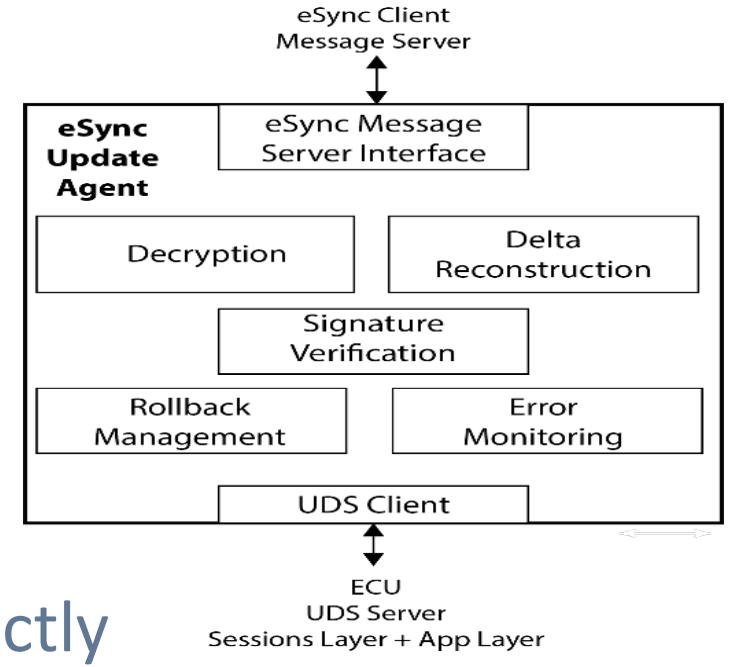
UDS Application Layer

1. Transfer Data
2. Read Data ID (even reading DTC codes)
3. Write Data ID
4. Upload Data
5. Erase
6. Verify

* May not be available on all ECUs

† Sequence may differ between UDS servers

Ethernet Based ECUs



- Newer ECUs May have Ethernet Interface
- Security Protocols can be Embedded into ECUs
- End to End Authentication can Go to the ECUs Directly
  - Payload can Remain Encrypted
- Simplifies the Security Architecture and Layout of Devices
  - Clear Segmentation of Functional Domains (using Ports and VLAN)
- No Change to UDS Client / UDS Server Handshake
  - Same as CAN-based ECU Transactions

Security Considerations

- DM Client Acts as Gate Keeper for Authentication
  - Preferred Location: In TCU
  - Can be in Gateway Switch – all External Connection are Authenticated
  - DM Client in a HeadUnit (Infotainment Gateway) Presents a Security Risk
- For ECUs located on FlexRay, CAN, LIN – Update Agents Can Reside in Gateways
  - Each ECU Authenticates with its Update Agent
- Newer ECUs on IP Networks can Host Update Agent within their Code Space
  - Isolate Legacy ECUs from Direct Connection to OBD Port
  - Use ECU Arbitration to Authenticate Legacy ECU Connections
- DM Client and each ECU have their own Unique Digital Certificates
  - Establish Bi-Directional Authentication
  - Difficult for Attackers to ‘Spoof' or Impersonate Any Element, Difficult to Gain Access to the System
    - Removes “man in the middle” Attacks
    - Impact on Cost and Performance

System Resource Requirements for eSync Client

- Operating System with Secure Non-Volatile File System
- Enough File System Memory for the Largest Expected Combination of Software Update Images, Plus Approximately 10%
- Enough Non-Volatile File System Memory to Buffer Diagnostic and Telematics Data
  - To Prevent Loss of Data when Connection is Interrupted
- Less than 500KB for eSync Client Code
- Typical: about 500KB for RAM
  - Additional RAM May be Needed for Many Update/Diagnostic Agents in the System

Agenda

1. Considerations – Objectives and Constraints
2. Architecture Review
3. Protocols, System Requirements, Security
4. Use Case Examples

Demonstrated Use Case Environments

eSync Client:

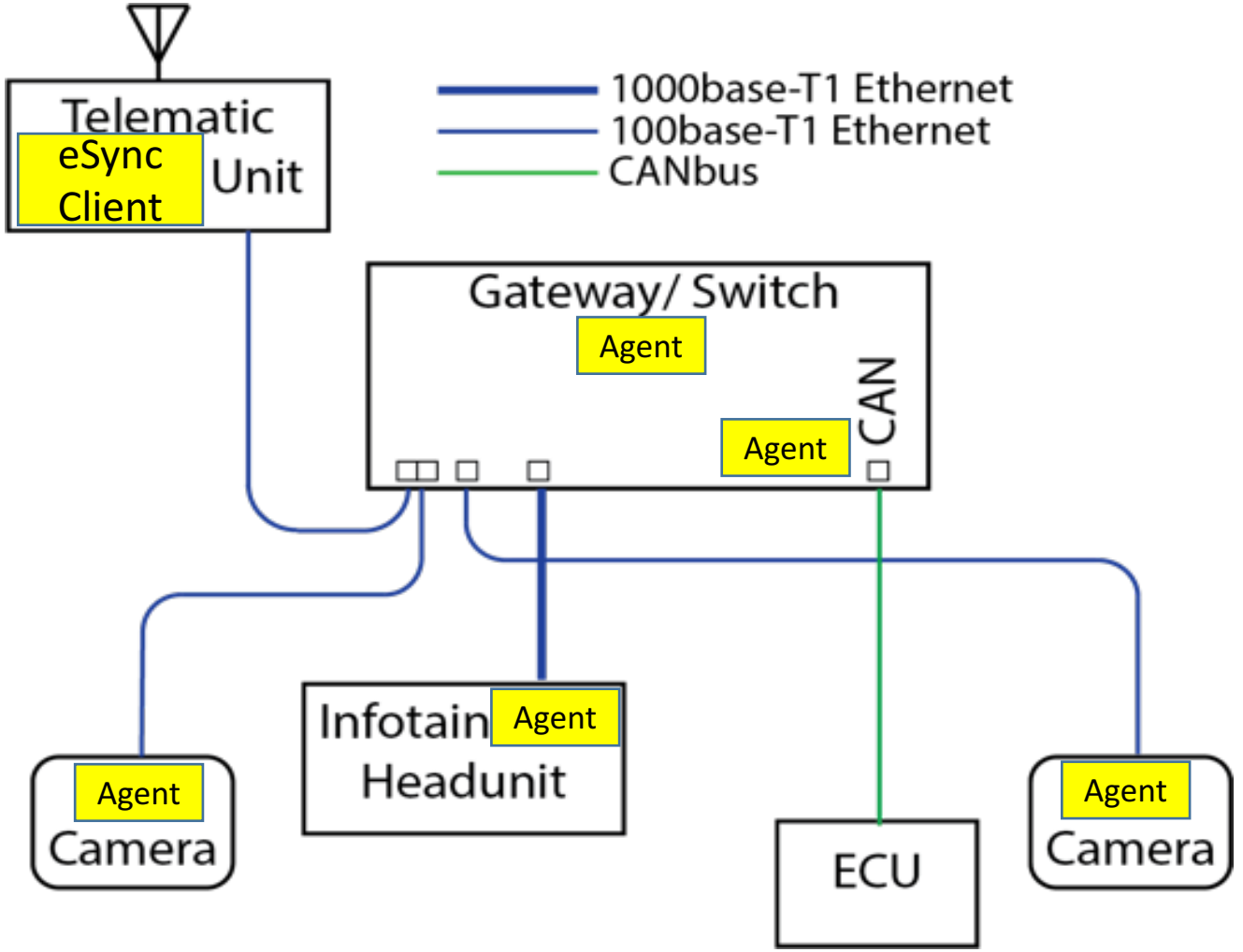
- Operating Systems: Linux, QNX, Integrity and Android
  - Other OS and File Systems are Possible
- Processors: Intel *Apollo Lake*; NXP *i.MX6*; Qualcomm *Snapdragon 820*; Renesas *R-Car3*

eSync Agent:

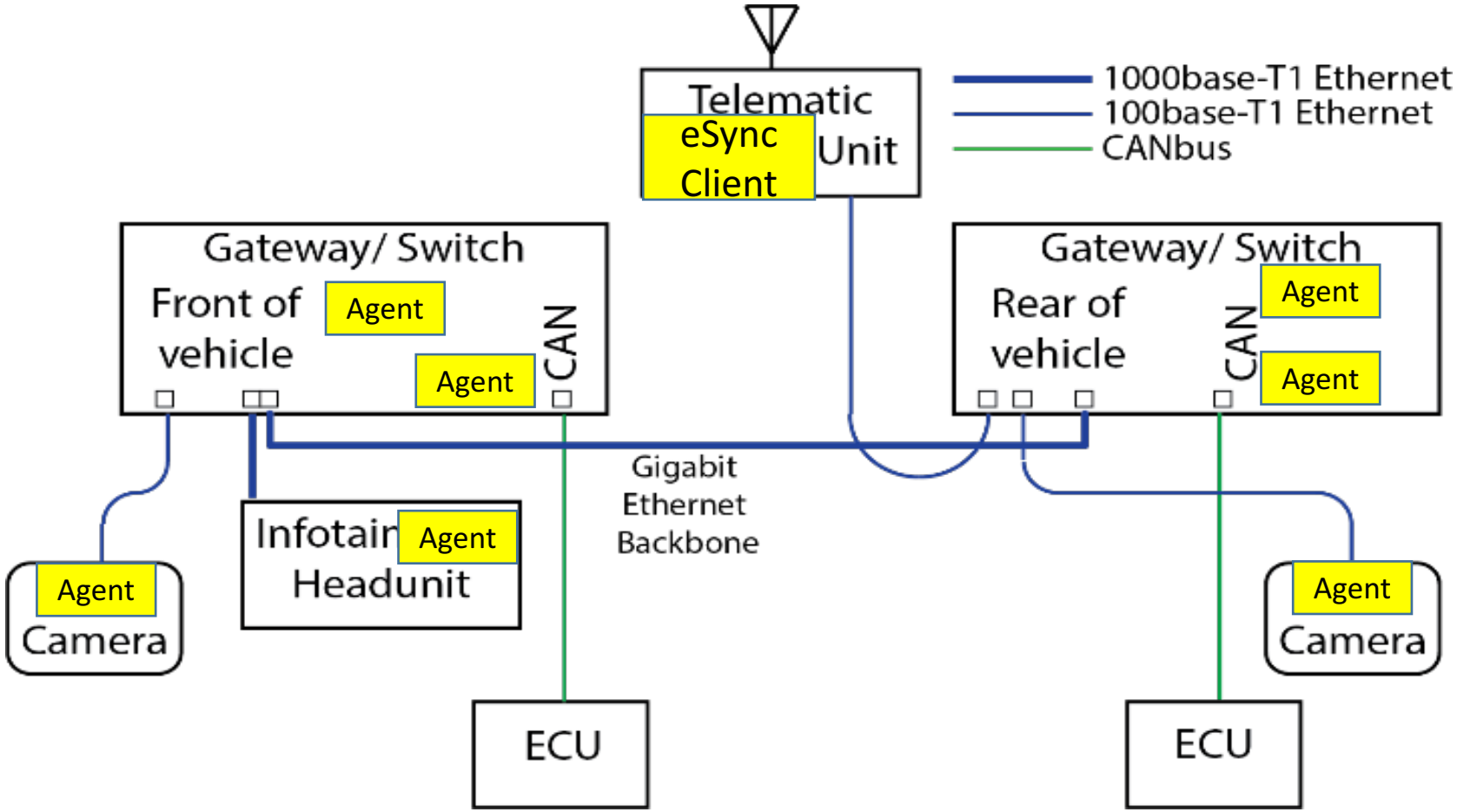
All OSs and Processors Used for the eSync Client, Plus:

- Operating Systems: AUTOSAR, Erika, FreeRTOS
- Processors and Controllers: NXP MPC5777 / 5648; Cortex R4 / Cortex M
- Bus / Networks: Ethernet (Broad-R Reach, AVB/TSN), CAN, LIN, FlexRay, USB

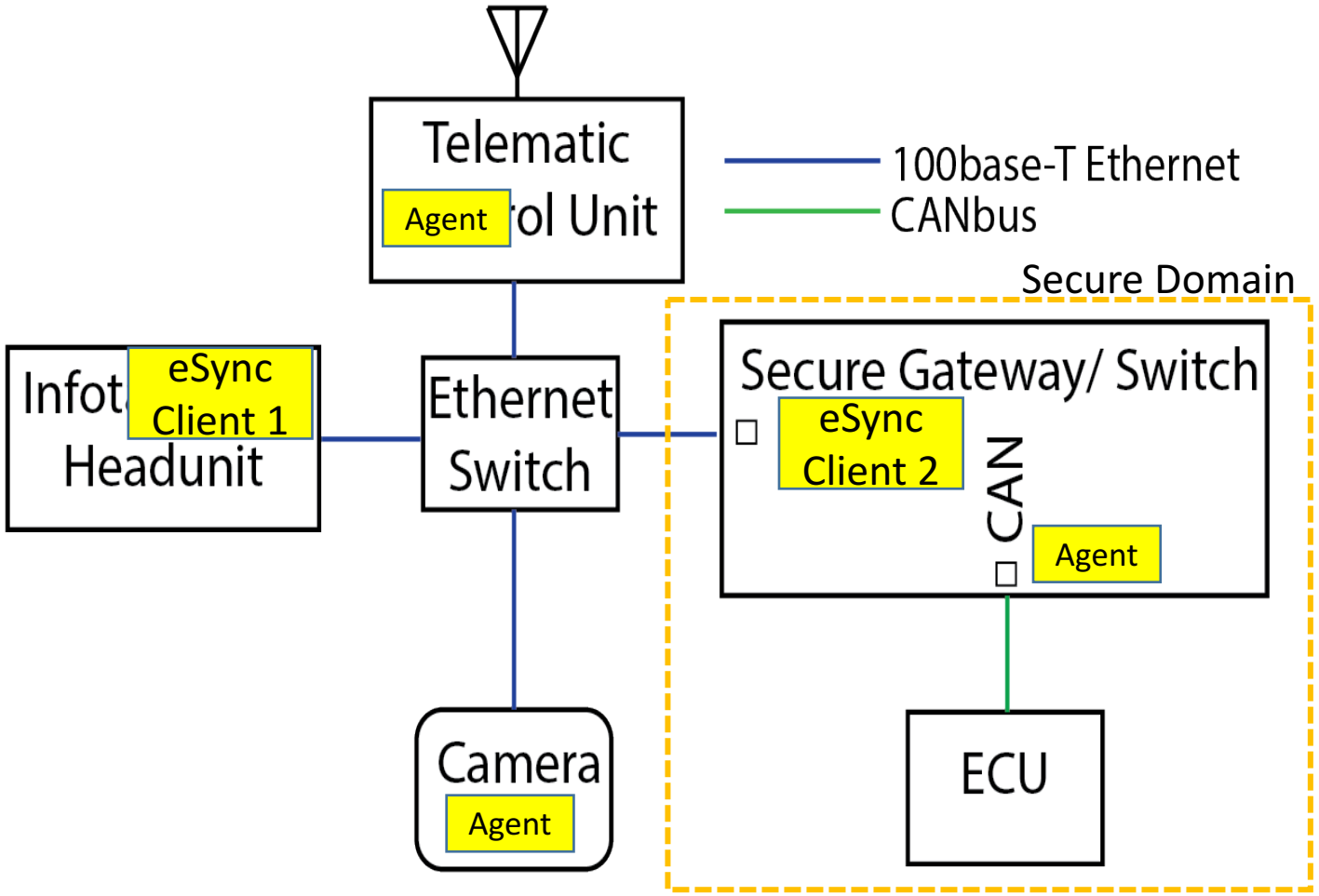
Use Case 1: Basic Vehicle System



Use Case 2: Vehicle Interconnect Using Ethernet for New Vehicle Platforms



Use Case 3: Multi-Domain eSync OTA System With Secure Gateway for Critical Domain



Summary of eSync System

- Bi-Directional and Transaction Based Information Transfer
- Modular Design with Update Agents for All Electronic Devices (ECUs, Sensors, etc.)
  - In the Device for IP Addressable Ethernet Devices
  - In the IP Addressable Port of the Gateway Switch for CAN, LIN Devices
  - Ensures System Reaches All Electronic Devices
- Layered Authentication and Encryption Between All Modules
  - Robust Security against Hackers
- Any Number of Update Agents, Update Any Number of ECUs in Parallel
  - Minimizes Vehicle Downtime during Updates
- Modular Design for Optimal Use of Limited CPU and Memory Resources