Why Traceability Matters in Automotive Embedded Systems

The automotive industry today faces a growing traceability gap. As vehicles have become rolling computers—with dozens of ECUs, increasingly complex software stacks, over-the-air updates, and globalized supply chains—many OEMs and suppliers struggle to maintain accurate, end-to-end visibility from requirements to code to hardware deployments in the field. This often leads to mismatches between intended and actual software configurations, slow root-cause analysis, costly recalls, and difficulty demonstrating compliance to standards like ISO 26262, ASPICE, and UNECE WP.29. In short, the industry lacks a reliable, automated way to know *what is running where*, at scale, across millions of vehicles and hardware variants.

Traceability is the ability to link each piece of software to its requirements, its tests, and—crucially—to the exact hardware it runs on. In safety-critical automotive systems (engine controllers, braking systems, ADAS, etc.), knowing which software version is on which hardware is essential for functional safety, regulatory compliance, debugging, and long-term maintenance.

To address this growing industry challenge, Datafrond's solution is an end-to-end, automated traceability platform designed specifically for complex automotive embedded systems. Our solution seamlessly connects requirements, software artifacts, test results, and deployed hardware configurations—creating a continuous, verifiable chain of custody from development through in-vehicle lifecycle. By providing real-time, hardware-specific software visibility at scale, it enables OEMs and suppliers to eliminate configuration uncertainty, accelerate diagnostics, simplify compliance, and ensure that every vehicle is running precisely what it should.

Key reasons why traceability is crucial:

- **Safety and Compliance:** Standards like **ISO 26262 (Functional Safety)** require configuration management and unique identification of all safety-related software and hardware items. Regulators (e.g. UNECE WP.29 **Regulation 156** on software updates) mandate that manufacturers maintain a *"unique identifier"* for every software installed on a vehicle type and record its associated hardware component. This ensures that any update or change can be audited and approved with full knowledge of the affected hardware. In fact, R156 requires automakers to record the *before-and-after configuration* of vehicle ECUs (hardware part number and software version) for any software update.
- **Quality and Recall Management:** If a defect is found in a particular software version, traceability allows pinpointing which vehicles (and ECUs) are affected by that software. Conversely, if a batch of hardware modules has an issue, traceability shows which software versions were running on them. This was highlighted by industry data showing that **about 50% of automotive warranty issues are due to electronic/embedded software problems**. Robust traceability helps limit the scope of recalls and ensures fixes address the right configuration.
- **Complex Integration:** A modern car can have **70–100+ ECUs networked together**, often provided by different suppliers. Each ECU's software must be compatible with the hardware's capabilities (CPU, memory, sensors/actuators) and with other ECUs. A traceable link between *software binaries* and *hardware part numbers* ensures that in manufacturing and service, the correct software is flashed onto the correct hardware module variant. *Automotive SPICE (ASPICE)*, a process quality framework, explicitly emphasizes bidirectional traceability from requirements through design, implementation, and testing – part of this best practice is being able to trace a delivered binary back to the hardware and requirements it fulfills.
- **Over-the-Air Updates and Cybersecurity:** With vehicles now supporting OTA updates, it's even more important to know exactly what software is on what hardware in each vehicle. **Software Bills of Materials (SBOMs)** and unique software IDs tied to hardware allow secure update management and vulnerability

tracking. For instance, **cryptographic signing** of software modules (as required by cybersecurity best practices) generates an identity for each binary; managing those identities involves linking them to the target hardware ECU to prevent mismatches.

Datafrond emphasizes **end-to-end traceability** between software artifacts and hardware components, ensuring compliance, safety, and efficient lifecycle management. This is achieved through **ALM–PLM integration**, configuration control, and adherence to industry standards.

Key Elements of the Solution

1. **Digital Thread via ALM–PLM Integration**
    - Connects **requirements → design → code → tests → build artifacts → hardware parts**.
    - Enables tracing from a regulation or requirement to the exact binary and associated hardware part number.
    - Tools like **PTC Codebeamer ALM** integrated with **Windchill PLM** provide a single source of truth for software–hardware linkage.
2. **Configuration Management**
    - Treats software as a **configuration item** with unique part numbers, managed alongside hardware in PLM systems.
    - Updates (e.g., OTA) require recording **before-and-after configurations**: hardware ID, old/new software version, and integrity hashes for compliance.
3. **Embedded Metadata & Manifests**
    - Software binaries include identifiers (hardware part number, software version) for in-field traceability.
    - Standards like **AUTOSAR** provide XML manifests mapping software components to ECUs.
4. **CI/CD Pipeline Integration**
    - Build pipelines embed traceability hooks, pulling metadata from PLM and storing binaries in artifact repositories with versioned metadata.

    - Automated updates link built binaries back to PLM records for instant traceability.

5. **Compliance with Standards**
    - **ISO 26262** and **ASPICE** mandate bidirectional traceability and configuration control.
    - **UNECE R156** requires unique software IDs and full configuration history for type approval.
    - AUTOSAR and cybersecurity frameworks support structured traceability.

Benefits

- **Safety & Regulatory Compliance:** Meets ISO 26262, ASPICE, and UNECE requirements.
- **Efficient Updates & Audits:** Maintains a clear history of software–hardware combinations.
- **Quality Management:** Enables precise recall and defect tracking.
- **Supports OTA & Cybersecurity:** Through SBOMs and cryptographic identifiers.