happiest minds
The Mindful IT Company
Born **Digital** . Born **Agile**

Quality Assurance
Vehicle to Cloud 5G Connected Car Enabled by **IoT**

In the era of fast evolving technology, the fusion of automotive excellence, cloud prowess & 5G connectivity has given rise to an excellent connected car solution. With the evolution in the automotive space, we need such an exclusive innovative solution that that gives the user an incredible and seamless experience. This whitepaper will explain about the key components of a connected car ecosystem as a solution and follow it up with a unique test validation approach which will reshape the future of transportation and driving experience through improved quality assurance.

1. Understanding the Connected Car Solution

Connected car solutions are built upon a sophisticated framework comprised of interconnected blocks that synergistically contribute to the seamless integration of vehicles with the digital world. These solution blocks form the foundation for creating an intelligent and interactive automotive ecosystem, enhancing safety, efficiency, and overall driving experience. Let's delve into a brief introduction to these key connected car solution blocks.

1.1. Telematics Control Unit (TCU)

Telematics Control Unit (TCU) is a fundamental component of modern connected vehicles, serving as a central communication hub that enables the integration of telecommunications and informatics. Essentially,the TCU functions as the brain of the connected car, responsible for collecting, processing, and transmitting data.
Equipped with an array of sensors such as GPS, accelerometers, odometers, environmental sensors, and cameras, connected cars gather real-time data about their surroundings, driving conditions, and internal components.

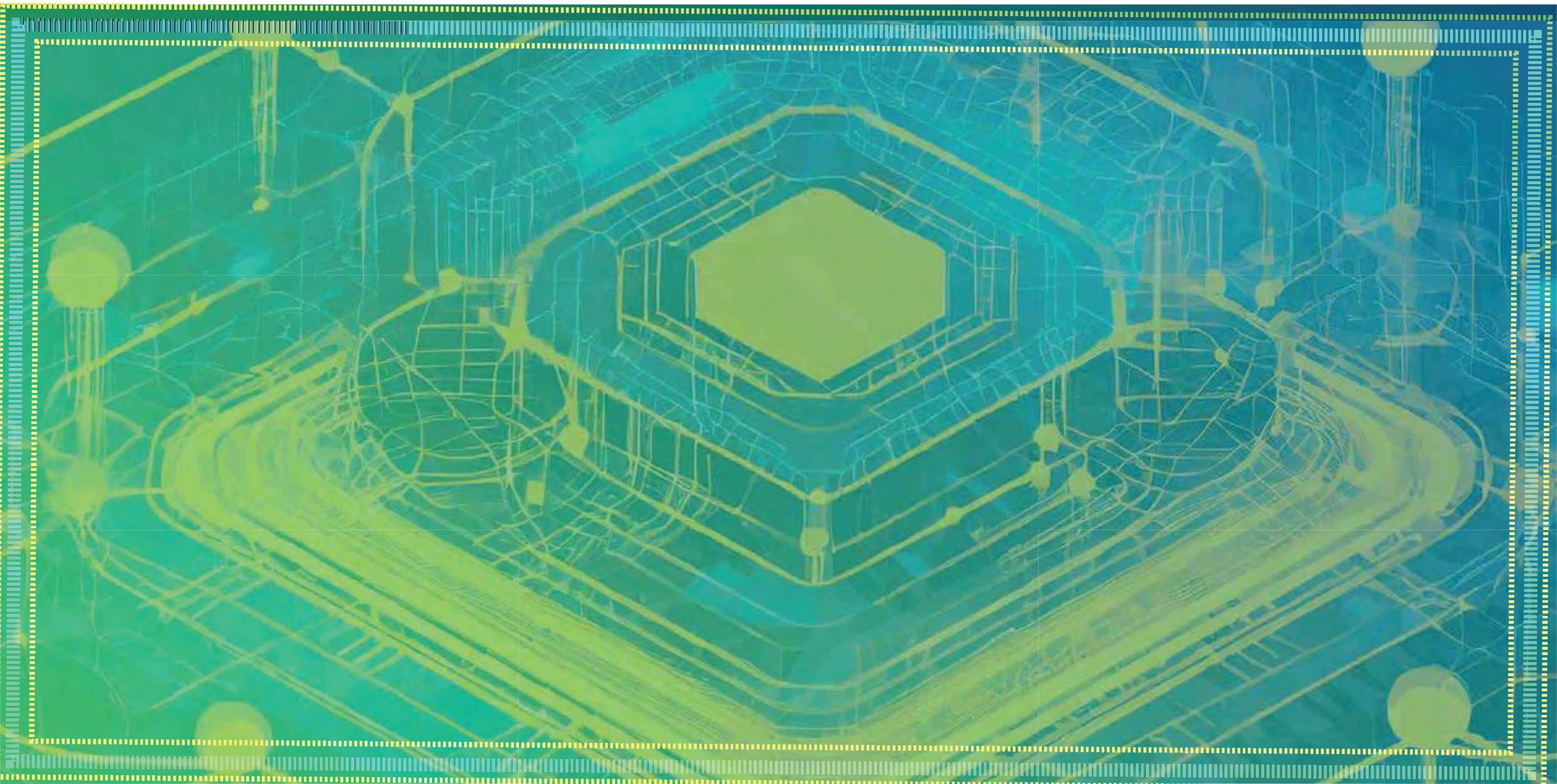
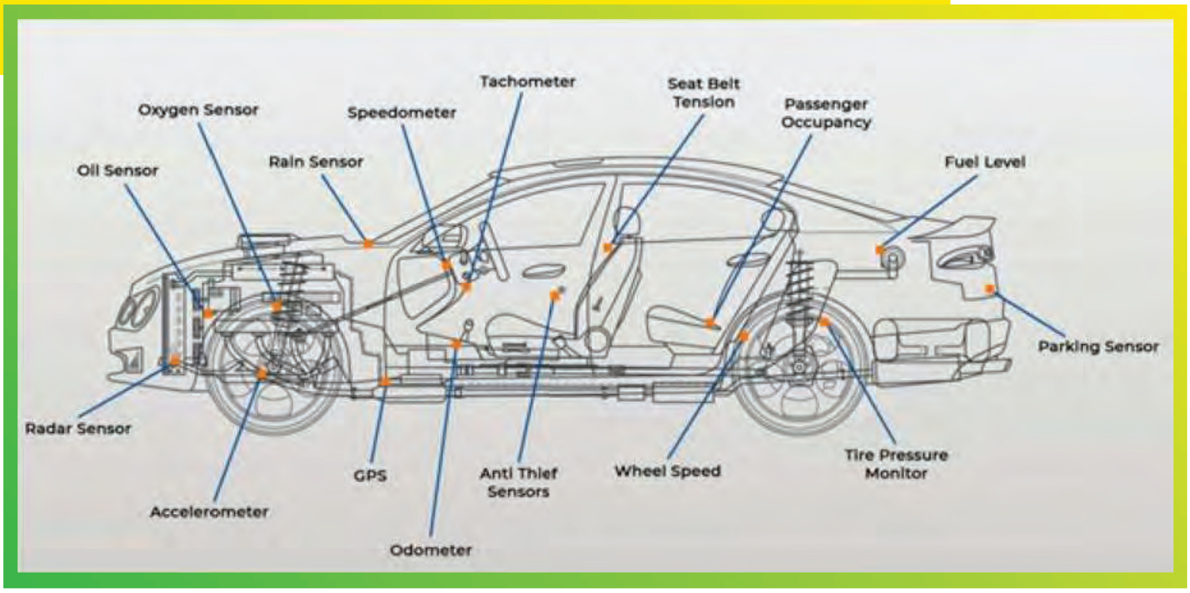


Figure 1 : Sensors that are available within any passenger vehicle



It manages the vehicle's connectivity to external networks, such as 5G cellular and satellite systems, facilitating communication between the vehicle and various entities, including other vehicles, infrastructure, and cloud-based servers.

1.2. IoT Connectivity - TCU to Cloud Integration

IoT connectivity establishes the vital link between the vehicle and the broader digital landscape. It enables real-time data exchange, infrastructure, and cloud-based platforms. This connectivity is the backbone of the entire connected car ecosystem. Integration of TCU to the cloud is the first and most important step to enable the TCU to perform data transfer, maintain bidirectional communication, and synchronize the car state to the application.

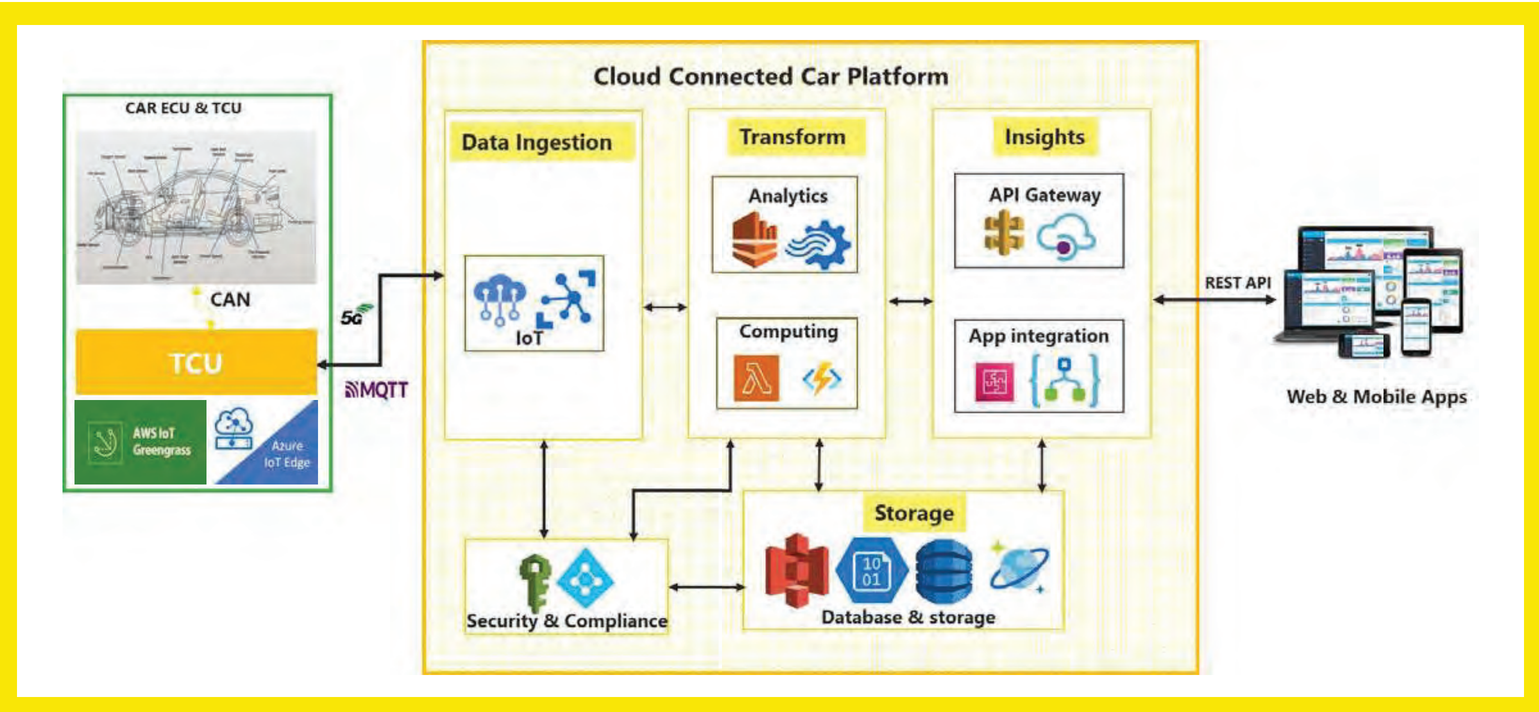
Each TCU needs to be registered as an IoT device in the AWS IoT Core for creating X.509 device certificates which can be configured in the TCU to initiate the successful connection with the AWS IoT Hub. TCU, equipped with AWS IoT Core SDKs and device certificates supports MQTT (Message Queuing Telemetry Transport) and enables secure communication between the car and the cloud. Once a secure connection is initiated, the TCU will publish data to the predefined topics to the AWS IoT Hub and subscribe to required topics for receiving commands or updates.

1.3. Cloud Platform Services

The cloud platform for connected cars plays a pivotal role in shaping the intelligence, scalability, and functionality of the entire connected car ecosystem. This cloud-based infrastructure serves as a centralized hub for managing and processing the vast volumes of data generated by connected vehicles, enabling various services, applications, and insights. It empowers features like data analytics, remote monitoring, and the delivery of over-the-air software updates, fostering a scalable and intelligent infrastructure.

Connected Car Service	Description	Reference Azure / AWS service
Device Management	Streamlines the management of connected devices throughout their lifecycle.	AWS IoT Device Management Azure IoT Device Provisioning Service
IoT Hub	IoT Hub is a scalable and fully managed IoT service that facilitates communication between IoT applications and the devices it manages.	AWS IoT Hub Azure IoT Hub
Serverless Computing for Ingested Data	Serverless computing service for running code without managing servers, suitable for processing ingested data from connected cars.Ingested data triggers functions, allowing developers to implement business logic, perform real-time analytics, or interact with cloud other platform services.	AWS Lambda Azure Functions
Data Management and Storage	Storing and managing large volumes of data generated by connected cars, including telemetry, logs, and multimedia content.	Amazon S3 (Simple Storage Service) Azure Blob Storage
Database	Database for storing and retrieving structured data with low-latency access for data related to connected car configurations, user profiles,and real-time telemetry with high performance and scalability.	Azure Cosmos DB Amazon DynamoDB
Real-Time Analytics	Enables the processing and analysis of streaming data in real-time. Analyzing live telemetry data from connected cars to detect patterns, anomalies, and trends for immediate insights.	Amazon Kinesis Azure Stream Analytics
API Management	Facilitates the creation, deployment, and management of APIs. Exposing APIs for accessing connected car services, managing access, and tracking API usage.	Amazon API Gateway Azure API Management
Over-the-Air (OTA)	Manages and delivers software/firmware updates to connected devices and updating software in connected cars for bug fixes, feature enhancements, and security patches.	AWS IoT Device Management Azure IoT Hub
Security and Compliance:	Securing access to connected car data, applications, and services to prevent unauthorized usage using the identity management, Role Based Access Control (RBAC) etc.	AWS Identity and Access Management (IAM) Azure Active Directory (AD)
Integration with External Services:	Facilitates the integration of connected car services with external applications and services. Integrating weather data, mapping services, or external APIs into the connected car solution.	AWS App Integration Azure Logic Apps

Figure 2. Sample reference implementation of a basic 5G connected car leveraging AWS



1.4. Edge Computing and OTA updates

Edge computing in connected car Telematics Control Units (TCUs) significantly reduces the amount of data transmitted to the cloud compared to the pace at which data is collected from the car. By processing and analyzing data locally within the TCU, only relevant and essential information is sent to the cloud. This approach optimizes bandwidth usage and minimizes the need for extensive data transfer, ensuring the network is not overwhelmed with unnecessary information. As a result, edge computing enhances efficiency by prioritizing crucial data, reducing latency, and improving the overall responsiveness of the connected car platform.

Services like AWS Greengrass and Azure IoT Edge provide TCUs with these crucial edge computing capabilities, culminating in an efficient, prioritized data flow that enhances overall responsiveness within the connected car platform. Additionally, these platforms facilitate Over-The-Air (OTA) updates seamlessly. This ensures that the TCUs can receive software updates and improvements remotely, further enhancing their functionality and adaptability without requiring physical intervention.

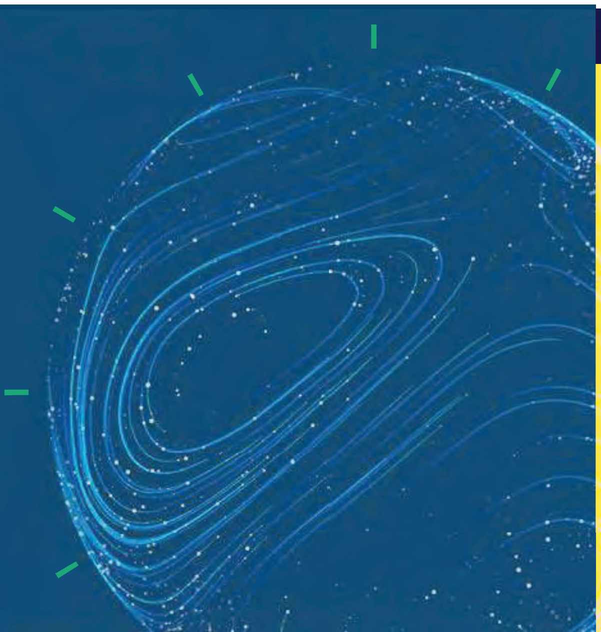
2. Testing Approach

The complexity of interconnected components, ranging from TCUs and IoT Hubs to cloud platforms and serverless computing, necessitates a robust testing strategy to guarantee optimal performance, reliability, and safety. Comprehensive testing becomes imperative to address the unique challenges posed by the convergence of automotive engineering, digital technologies, and cloud-based solutions. Creating a comprehensive test strategy for a connected car platform involves defining the overall approach and methodologies to ensure the quality of the system.

Connected car platforms demand various testing levels: Unit Testing for individual components, Integration Testing for system interactions, System Testing for overall functionality, End-to-End Testing for real-world scenarios, and Performance Testing for optimal responsiveness. These levels collectively ensure data accuracy, system reliability, and security, contributing to a seamless and trustworthy connected driving experience.

2.1. Module Level Testing

During module level testing for a connected car platform, critical modules like the Telematics Control Unit (TCU) and the cloud platform must be thoroughly evaluated.



2.1.1. Telematics Control Unit (TCU)

Module-level testing of the Vehicle Telematics Control Unit (TCU) is a crucial step in designing effective test cases to ensure that the TCU, which acts as the central intelligence of a connected vehicle, functions as intended to perform the below actions.

- TCU configuration tests
- Communication with various in-vehicle sensors.
- TCU to ECU communication
- Cellular (3G/4G/5G) connectivity checks
- Data encryption validation
- Secure boot
- Firmware update
- Data storage and propagation

TCU testing will require automotive expertise and can employ tools such as Vector CANoe and CAPL to simulate the vehicle sensor data and validate the TCU data collection.

2.1.2. Cloud Platform

Connected car cloud platform module level testing is paramount to validate the robustness of cloud services integral to seamless vehicle connectivity. This process ensures the reliability of services such as IoT Hubs, functions, and databases, guaranteeing functionality in the connected car ecosystem.

- Device provisioning and onboarding
- Telemetry data acceptance by IoT Hubs
- Data processing by the cloud functions as intended
- Data storage in the database

Cloud platform testing should employ the usage of TCU simulator to simulate the vehicle telemetry data. These simulators are generally designed using the cloud platform SDKs provided by the cloud services providers. Simulators will ensure the creation of the telemetry data and send the data over to the IoT Hubs in a defined format as per the TCU specification. These Telematics Control Unit (TCU) simulators are instrumental in testing the connected car cloud platform for several crucial reasons listed below.

- Isolation of TCU functionality
- Ingestion of customized data into the cloud
- Scenario reproduction
- Scalability testing
- Faster test cycles
- Enhanced test coverage

In addition to the TCU simulators, cloud platform tools, such as the AWS IoT console and Azure IoT explorer, shall be used to monitor and visualize IoT data. Correctness of the cloud functions and the writing of the telemetry data to the databases shall employ the usage of tools such as the AWS DynamoDB Console/Azure Cosmos DB Explorer. In addition to the above, cloud platform providers SDKs can be used to extract the data programmatically for validation.

2.1.3. Cloud API Testing

API testing of the cloud platform shall consider the validation of the API gateways and the exposed APIs to the API marketplace. Testing of the APIs in the cloud platform AWS and Azure involves ensuring that the gateways effectively manage and secure the APIs, handle traffic efficiently, and adhere to the specified configurations.

- Endpoint validation
- API response code validation
- Security and compliance
- CORS (Cross-Origin Resource Sharing) testing
- API performance testing

2.2. Integration Testing

Connected Car Platform has mainly two critical integration points that play a pivotal role in ensuring the seamless flow of data and functionalities within the connected car ecosystem. These integration points are the Telematics Control Unit (TCU) integration with the Cloud Platform's IoT Hubs and the Mobile application integration with the Cloud Platform through exposed APIs. Each integration serves a distinct purpose, enhancing the overall connected car experience. The focus of TCU integration is as follows.

- IoT Hub Connectivity via 5G
- Data transmission and reliability validation via defined protocol HTTP/MQTT/WebSocket
- Security and authentication
- Event reporting
- Firmware OTA updates

Mobile app integration testing with the connected car cloud platform will focus on the below:

- API interaction and functionality
- Authentication & Authorization
- Data synchronization checks
- Real-time notification

2.3. System Testing

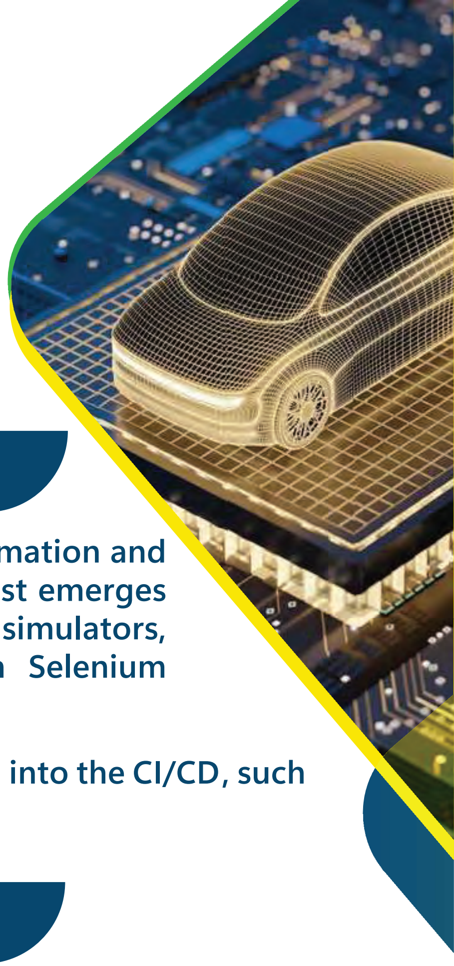
Connected car system testing involves a comprehensive evaluation of the entire ecosystem, including integrating hardware, software, communication protocols, and external interfaces. The goal is to ensure the connected car system's functionality, reliability, and security. This testing will be carried out with the use of a web/mobile application with a focus on the application use cases.

2.4. Test Automation

Test automation plays a crucial role and is paramount in efficiently executing repetitive and complex test scenarios, ensuring the reliability and stability of the connected car platform. As discussed in the above sections, TCU simulators will play a very vital role in the test automation of the platform.

2.4.1. TCU Simulators in Test Automation

TCU simulators, acting as MQTT clients, take center stage in replicating data in the requisite format for ingestion by cloud platform IoT Hubs. This simulation facilitates controlled testing scenarios, ensuring that the platform responds as expected under various conditions.



2.4.2. Choosing the Right Automation Framework

The next crucial step involves strategically deciding the areas for automation and selecting an adept framework. A customized version of Robot or Pytest emerges as an optimal choice, providing essential hooks to control TCU simulators, supporting API test automation, and seamlessly integrating with Selenium WebDriver for application test automation.

Most importantly, the test suites and the framework shall be integrated into the CI/CD, such as the Jenkins/Azure DevOps, as needed.

2.4.2. Choosing the Right Automation Framework

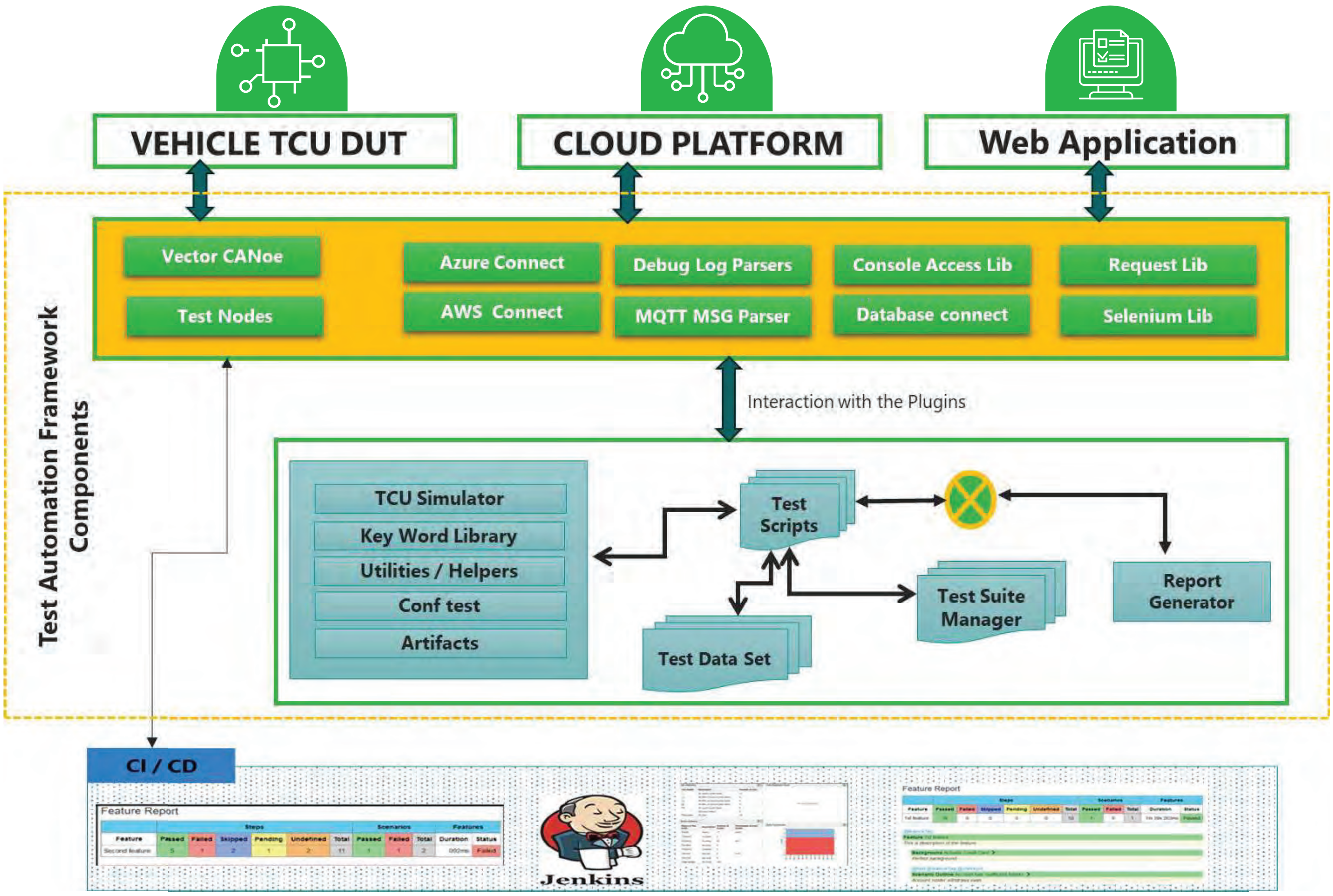


Figure 3: Reference test framework for 5G connected car platform

TCU Simulator Control Hooks

- The framework incorporates specialized hooks to control TCU simulators effectively, allowing for precise and controlled testing scenarios.
- This shall be achieved by the design of the Python-based TCU simulator using the Paho MQTT library.

API Test Automation

- Ensuring comprehensive coverage, the framework shall support API test automation, verifying the seamless communication between various components of the connected car platform.
- Python request library or inbuilt keywords of the robot framework can be used for this purpose.

Application Test Automation

- Harnessing the power of Selenium WebDriver, the framework extends automation to the application layer, validating the user interface and overall user experience.

Test Data Management

- Comprehensive management of test data is essential for executing varied test scenarios. The framework shall include robust mechanisms for generating, storing, and manipulating test data, ensuring diverse and realistic testing conditions.
- In the robot framework, the framework leverages the built-in resource files and external data sources for efficient test data management. In PyTest, custom data generation scripts can be seamlessly integrated for dynamic and diverse test data.

Test Script Repository

- A centralized repository for storing and versioning test scripts. This ensures easy access, collaboration, and version control, promoting a systematic approach to script management.

Test Suites

- Organized collections of test scripts designed to validate specific functionalities or scenarios. Test suites enable efficient test execution and provide a structured approach to testing different aspects of the connected car platform.
- In the robot framework, test suites can be created using the test suite and test case structures, providing a hierarchical organization.PyTest uses Python classes and functions to structure test suites and also supports tags offering flexibility in organizing and executing tests.

Test Reports and Logging

- Comprehensive test reporting mechanisms offer detailed insights into test results, aiding in quickly identifying and resolving issues. Logging mechanisms capture detailed information during test execution, facilitating effective debugging and analysis.
- In the robot framework, built-in reporting tools like a robot and log files provide detailed reports. In PyTest, plugins such as pytest-html or pytest-json provide customizable reporting options.

Integration with CI/CD

- To streamline the testing process and align with modern development practices, the test suites and the automation framework shall seamlessly integrate with Continuous Integration/Continuous Deployment (CI/CD) pipelines. Whether utilizing Jenkins or Azure DevOps, this integration ensures that automated tests are integral to the development lifecycle.

2.4.4. Benefits of Test Automation

- Increased test coverage
- Reduction in time and effort
- Quick debugging and issue resolution
- Enhanced release confidence
- Efficient test data management
- Reusable test scripts
- Enhanced test maintenance
- Faster feedback with efficient integration with CI/CD

Conclusion:

At the forefront of ensuring the reliability and stability of connected car platforms is a comprehensive validation approach, covering functional, API testing, and integration with API marketplaces. Integrated seamlessly with TCU simulators, a robust automation framework becomes the linchpin for executing efficiently thorough tests, including test data management and integration with CI/CD pipelines. Essential components not only accelerate the development cycle but also foster agility in this dynamic landscape. The synergy between comprehensive validation and test automation, encompassing functional and API testing, not only meets but surpasses user expectations, shaping the driving experience of the future. This holistic approach ensures that connected car platforms not only meet but exceed the highest performance and user satisfaction standards, setting the stage for the next frontier in smart mobility.

ABOUT THE AUTHORS

SNEHA BATI

Senior Test Lead, PDES

Sneha Bati has 16+ years of experience in validation of embedded & networking products, cloud, web & mobile-based apps. The senior test lead, designed and implemented multiple test automation frameworks using Keyword-Driven Development (KDD). Her expertise is in delivering quality tested products and solutions to clients across varied domains such as networking, telecom, retail and industrial IoT. She holds a Bachelor's degree in Computer Science & Engineering from VTU, Belgaum.

RAVINAGAKUMAR K

Principal Test Architect, PDES

Ravinagakumar K, Principal Test Architect of Product & Digital Engineering Services (PDES) at Happiest Minds Technologies, brings 18+ years of experience spearheading innovation in automotive technology. He brings an unique expertise that spans Embedded Systems, Industrial Automotive, IoT, IIoT, Azure, and AWS while working with several global companies in the Industrial and Manufacturing space. He is proficient in Test Automation, Simulator Designing, Test Strategy, Test Planning, Test Automation Framework Design, Cloud Platform Automation, and diverse IoT protocols such as MQTT and CoAP. With a deep understanding of automotive protocols like CAN and LIN, along with proficiency in CANoe Network Simulation, CAPL, he navigates the intricacies of vehicular communication, contributing significantly to advancing testing methodologies. Ravi's rich experience positions him in driving innovation and setting industry benchmarks in Test Automation practices.

JASON CHANDRALAL

General Manager, PDES

Jason Chandralal is the General Manager of Product & Digital Engineering Services (PDES) at Happiest Minds Technologies. He is responsible for Embedded and Quality Engineering solutions focused on IoT, Industrial Automation & Manufacturing, Hi-Tech and SaaS, Medical Embedded Systems. He has over 30 years of experience in Telecom, Networking, Medical and Industrial product design; development and validation working across OEM, Telecom Service Providers & Hi-Tech and Medical customers. He is currently engaged in designing AI & ML methodologies for embedded system validation and is regular contributor to various international forums.