

FAHRZEUG DIAGNOSTIK

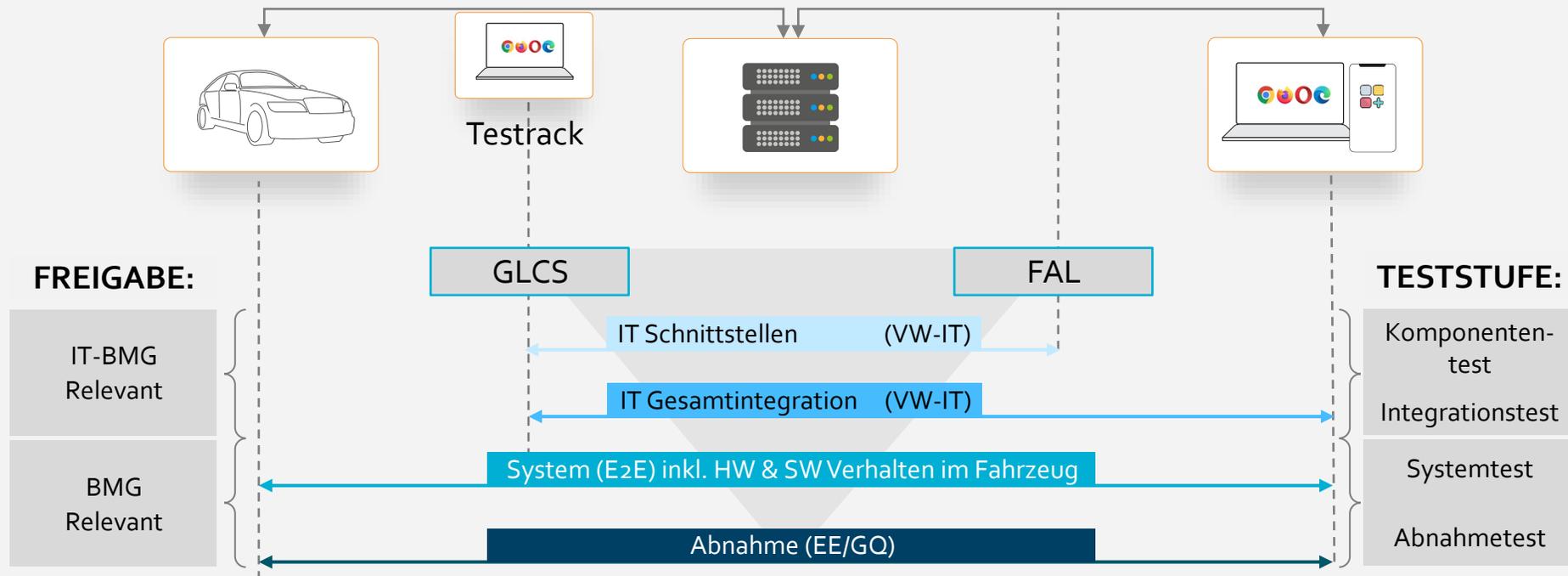
Testing – MOS
1TNC GmbH



Testansatz und -stufen – MOS

Integrativer Ansatz

1. Applikation wird in das bestehende System integriert.
 - Fokus dabei zunächst auf IT und dessen Schnittstellen
2. Nach Abnahme von Schritt 1, wird Applikation auf Gesamtsystem ausgerollt
 - Fokus wird erweitert im Rahmen des Testings

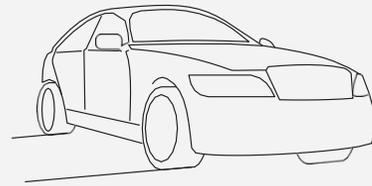


Teststrategie in der Praxis

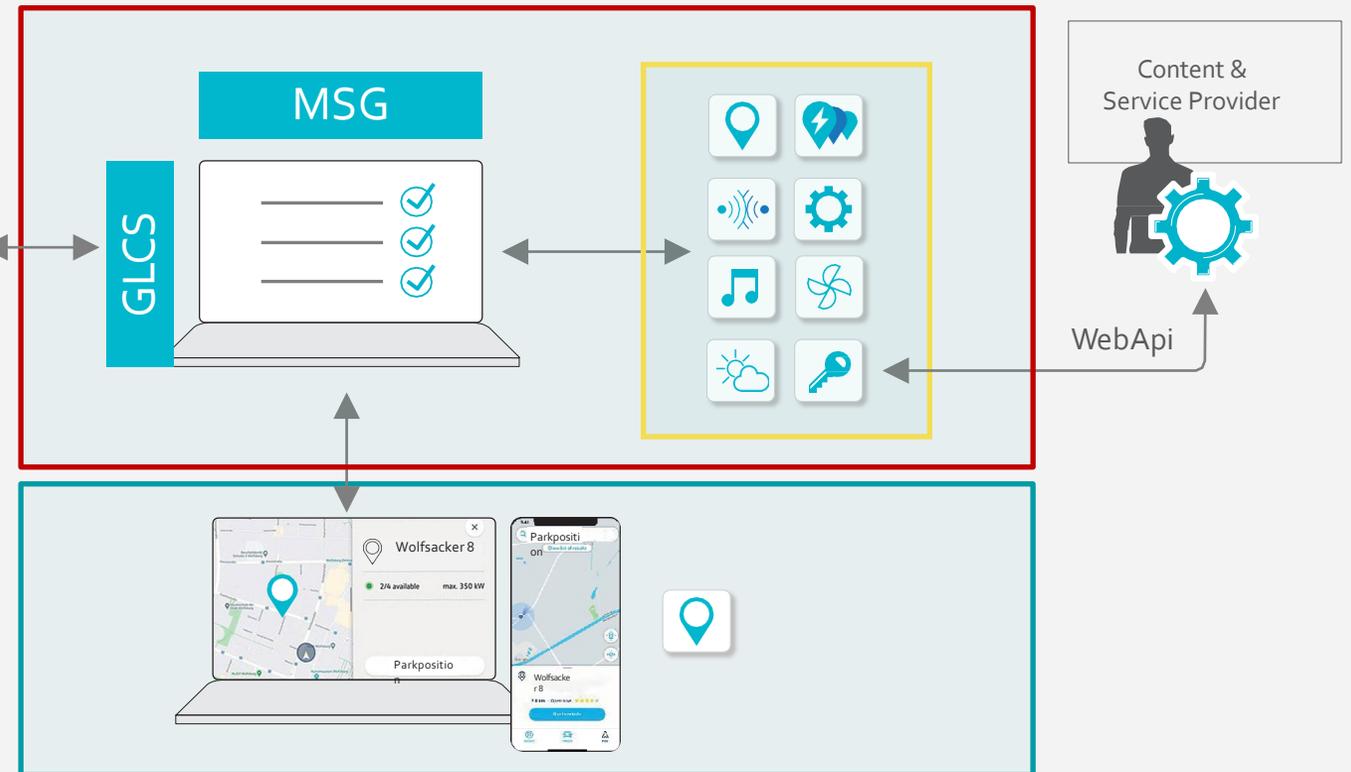
Unterscheidung zwischen INT und GINT

Scope ①

- Integrationstests
- Schnittstellen-Tests mit Scope auf IT-Backend
- IT-BMG-relevant



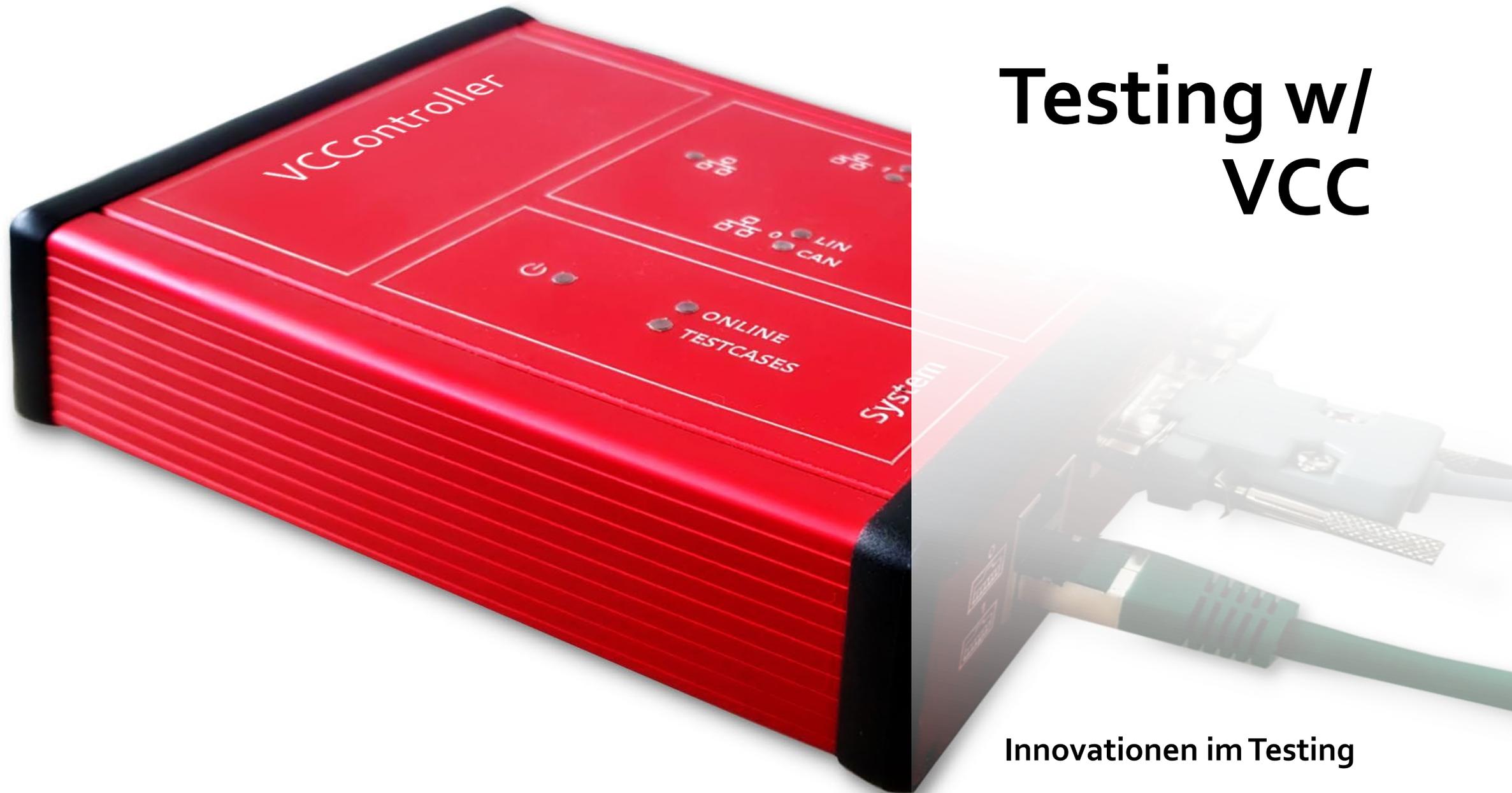
$$\textcircled{1} + \textcircled{2} + \textcircled{3} = \text{IT Gesamtintegrationstests (GINT)}$$



Scope ① + ② + ③

- Gesamtintegrationstests
- Schnittstellentests unter Zuhilfenahme der Hardware
- BMG-relevant

Testing w/ VCC



Innovationen im Testing

Softwarequalität ↑ Kosten ↓

Kontinuierliche Qualitätskontrolle während des gesamten Entwicklungsprozesses automatisierbar und reproduzierbar auf allen Ebenen.

Von TUI über Approval zu Live (reduzierter Umfang)

Kosteneinsparung durch frühzeitige Fehlererkennung und -beseitigung.

Vermeidung von Fehlern und Ausfällen vor Kunden

Bis zu ca. 480 Tests pro VCController und Tag machen ca. 140.000 Tests pro VCController pro Jahr!

Automatisierbare Log und Fehlerauswertung im Fehlerfall möglich.

Jeder durchgeführte Test ist ein Schritt hin zur fehlerfreien Software!

(Der Berechnung liegt eine Ausführungsdauer pro Testfall von 3 Minuten bei ca. 80% Verfügbarkeit zu Grunde)

SiL – HiL Ansatz

Unser SiL – HiL Ansatz sieht vor:

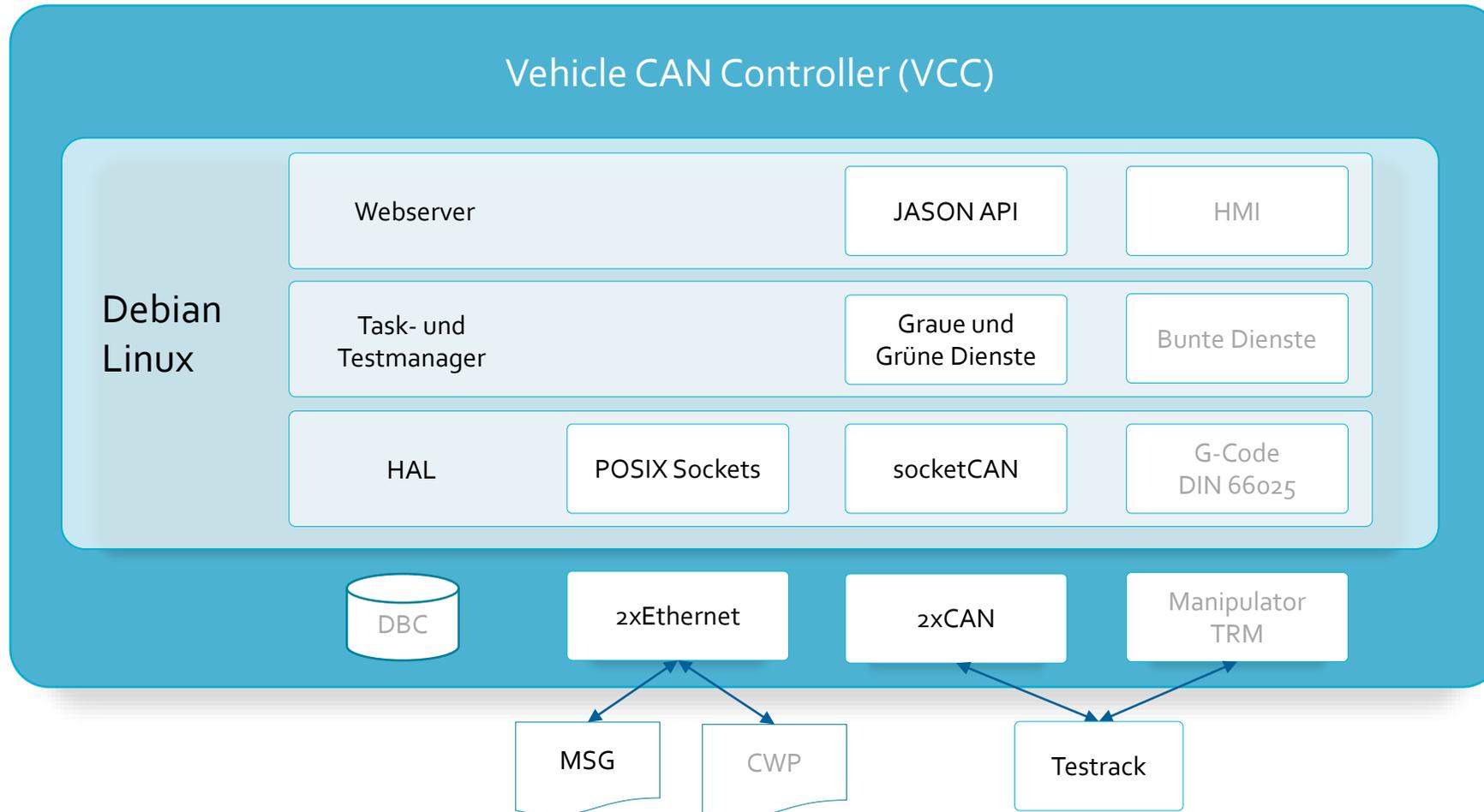
- Die Lösung erlaubt eine Absicherung vom Frontend bis hin zum CAN-Strang im Fahrzeug sowohl manuell, halbautomatisiert sowie vollautomatisiert. Die Kommunikation findet direkt mit der VW-Infrastruktur mit der Channel App bis hin zum CAN-Strang (nicht lediglich Hardware-Tests) statt.
- Der Service benötigt keine weitere Hardware wie Framegrabber, externe Eingabehardware oder weitere Eingriffe in die abzusichernde Versuchsträger. Die Nutzung findet über vorhandene System-Schnittstellen statt.
- Kompatibel mit allen Versuchsträgern – keine direkte Abhängigkeit zwischen Testobjekt und genutzter Testhardware.
- Architektur erlaubt Implementierung und Modellierung softwareseitiger Funktionen und standardisierter HW-Schnittstellen

VCC = Vehicle CAN Controller



Eine cloudbasierte Hardware- und Softwarelösung zur Erstellung skalierbarer MOD-Testfälle für eine umfassende Testabdeckung

Struktur Overview



Testtiefe und Testbreite unter Zuhilfenahme des VCC's

API-Testing (SiL): Direkte Absicherung von Schnittstellenfunktionen über den Channel APP über die MAL-/MSG-Schnittstelle auf QS als auch auf Live sowie Mocking der GLCS-Schnittstelle und weiterer Umsysteme (bspw. MQTT-Broker).

Steuergeräte Tests (HiL): Absicherung von Hardwarefunktionen im Bereich der online Dienste. Testfallevaluierung bis hin auf CAN-Ebene.

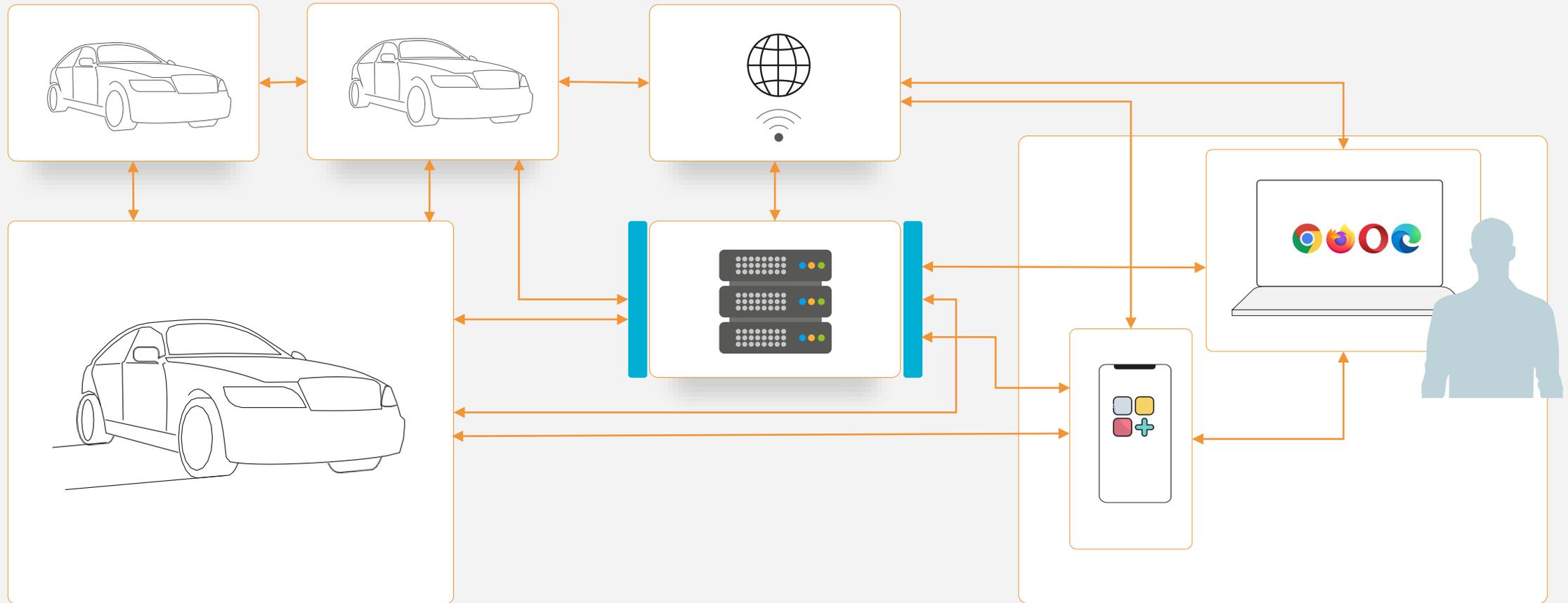
Use-Case-Tests: Absicherung von direkten Kunden-Use-Cases (sequenzielle Absicherung der Kundenprozesse) basierend auf API-Schnittstellen in sehr kurzer Zeit durch vorab parametrisierte Testfälle

Cron-Job-Tests: Automatische Durchführung ausgewählter Testsets zu vorab eingestellten Parametern (bspw.: Täglich um 06:00 Uhr Testset X ausführen)

Remote-Testing: Testen unabhängig von Standort des abzusichernden Versuchsträgers

Dauer-Tests: Durchführung von wiederholten und dauerhaften Tests und Möglichkeiten des Monitorings

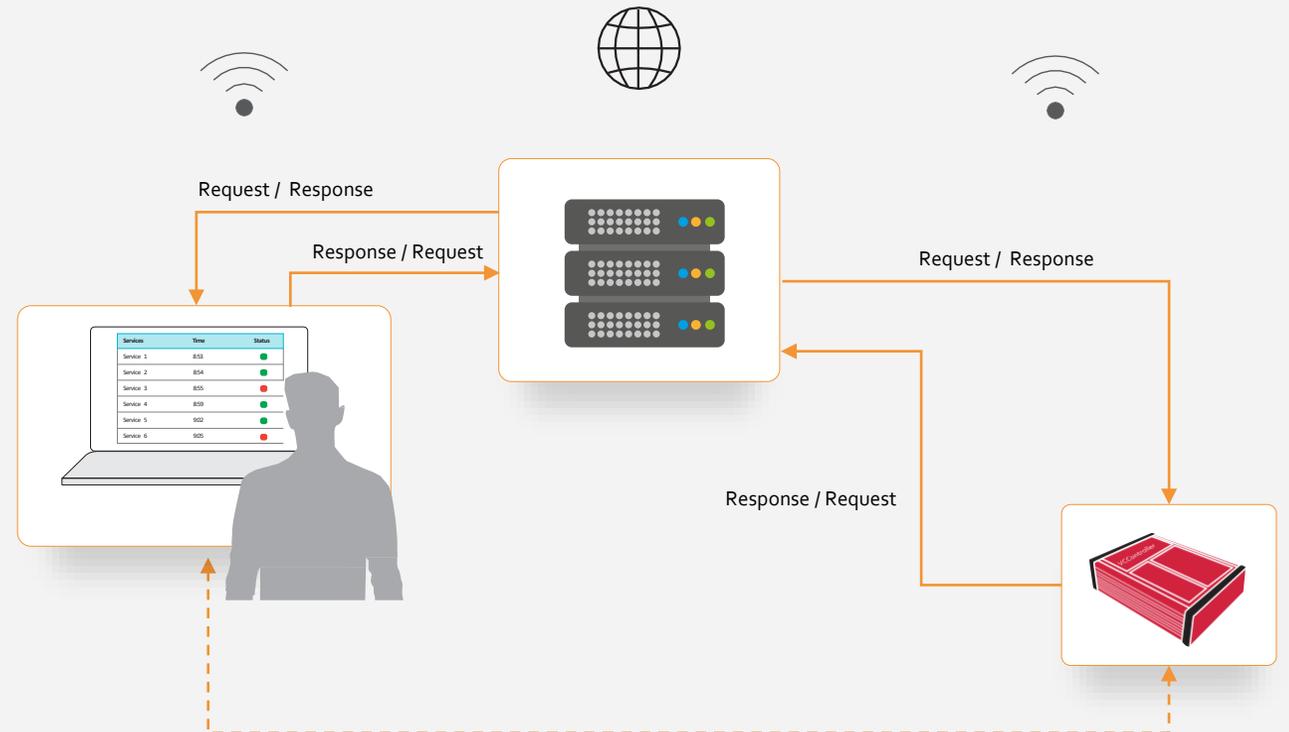
Aktuelle Umgebungen



SiL – HiL Ansatz

Testdurchführung Schnittstelle – vollautomatisiert (Fokus: SW)

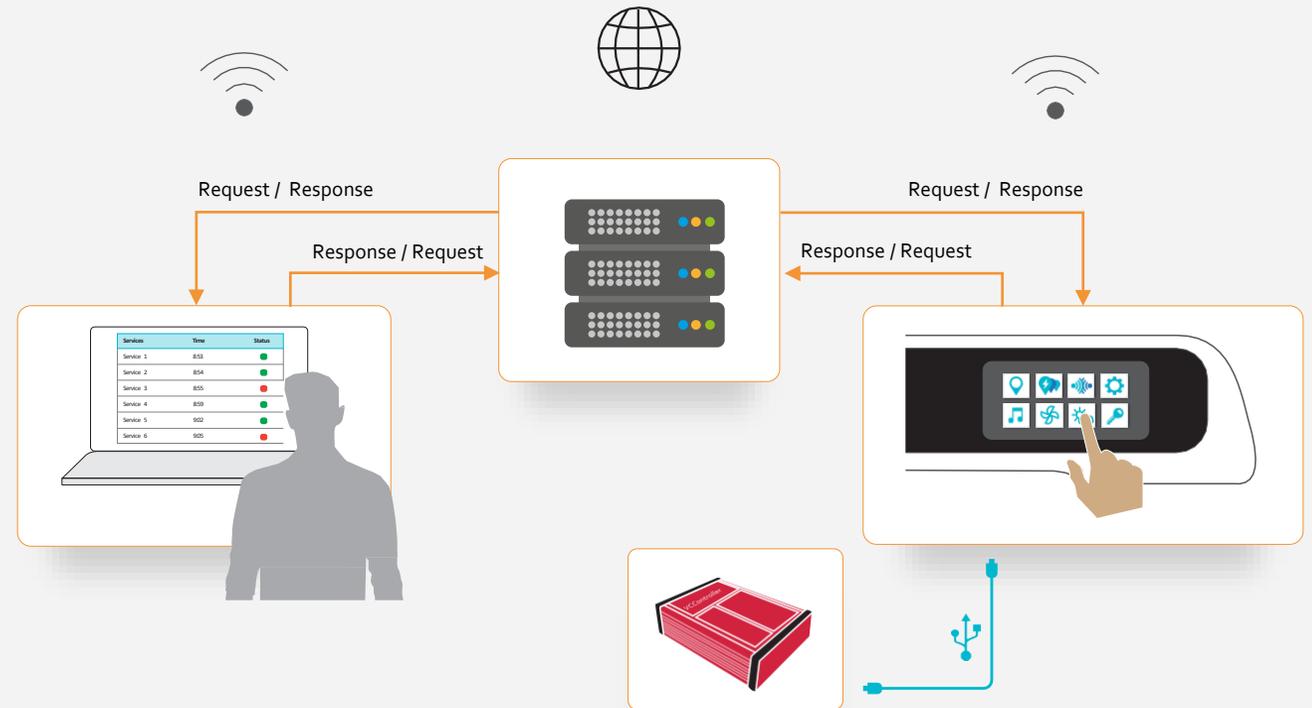
- Über das VCC-Frontend wird der Testfall am VCC angestoßen. Als Medium wird die Channel ‚App‘ auf User-Ebene verwendet.
- Der Testfall stellt dabei sicher, dass alle notwendigen abgestimmten Vorbedingungen vorhanden sind. Bei erfolgreicher Prüfung wird der Testfall angestoßen. Falls nicht, werden bei Möglichkeit die Parameter gesetzt.
- Der VCC setzt den Request (abhängig von Test / Use-Case) an der dafür vorgesehenen Schnittstelle ab (GLCS / MAL) und quittiert die Anfrage gemäß Use-Case an der ebenfalls dafür vorgesehenen Schnittstelle gemäß Spezifikation der Komponente Fahrzeug / SOP.
- Fokus liegt dabei wesentlich auf dem IT-Backend und somit die Integration der Komponente im Backend. Das Verhalten des Backends auf einkommende Anfragen sowie dessen Ausgabe liegt im Fokus der API-Tests.



SiL – HiL Ansatz

Testdurchführung Versuchsträger – vollautomatisiert (Fokus: SW + HW = GINT)

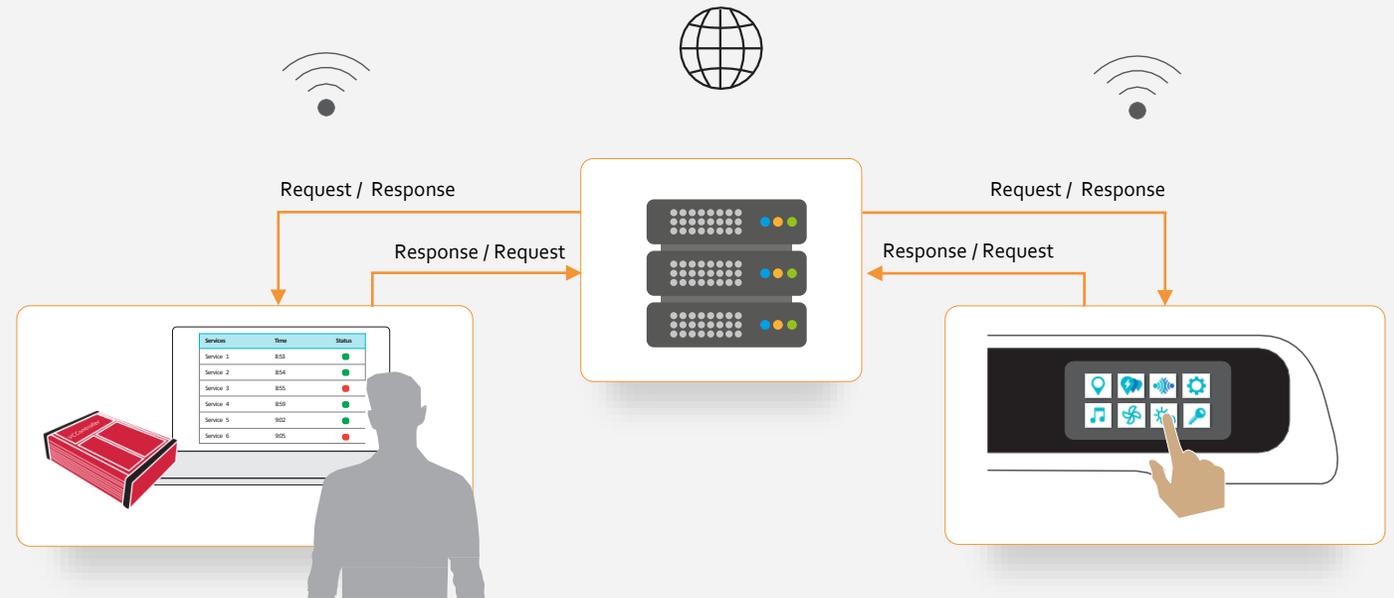
- Über das VCC-Frontend wird der Testfall angestoßen. Als Medium wird die Channel ‚App‘ auf User-Ebene verwendet.
- Der Testfall stellt dabei sicher, dass alle notwendigen abgestimmten Vorbedingungen vorhanden sind. Bei erfolgreicher Prüfung wird der Testfall angestoßen. Falls nicht, werden bei Möglichkeit die Parameter gesetzt.
- Über die OCU nimmt der Versuchsträger nach wie vor den Request entgegen. Der am Versuchsträger angeschlossene VCC überprüft dabei auf CAN-Ebene, ob die Botschaft zum Beispiel zum Hupen und Blinken auf dem CAN abgesetzt wird und gibt dies bereits als ein Erfolgspunkt zurück.
- Ferner wird auf Backendebene über entsprechende API-Schnittstellenfunktionen überprüft, ob der Response der OCU am Backend angekommen ist und an die App quittiert wird.



SiL – HiL Ansatz

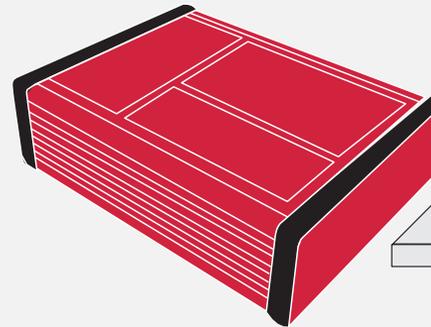
Testdurchführung Versuchsträger – halbautomatisiert (Fokus: SW + HW = GINT)

- Über das VCC-Frontend wird der Testfall angestoßen. Als Medium wird die Channel ‚App‘ auf User-Ebene verwendet.
- Der Testfall stellt dabei sicher, dass alle notwendigen abgestimmten Vorbedingungen vorhanden sind. Bei erfolgreicher Prüfung wird der Testfall angestoßen. Falls nicht, werden bei Möglichkeit die Parameter gesetzt.
- Über die OCU nimmt der Versuchsträger nach wie vor den Request entgegen. Der VCC prüft dabei anhand des im Backend abgelegten Response des Fahrzeugs die erfolgreiche Testdurchführung.



Testdurchführung Versuchsträger – Monitoring

- Über das VCC-Frontend wird der Testfall angestoßen. Als Medium wird die Channel ‚App‘ auf User-Ebene verwendet.
- Der Testfall stellt dabei sicher, dass alle notwendigen abgestimmten Vorbedingungen vorhanden sind. Bei erfolgreicher Prüfung wird der Testfall angestoßen. Falls nicht, werden bei Möglichkeit die Parameter gesetzt.
- Der VCC kann je nach Auslegung des Monitorings-Setups die Informationen auswerten und ein Ausfall schnell identifizieren – entweder durch den Anschluss an ein Rack, die direkte Fahrzeug und Backend-Kommunikation oder lediglich Backend-Kommunikation.

A stylized illustration of a laptop computer. The screen displays a table with three columns: Services, Time, and Status. The table contains six rows of data, with the status of each service indicated by a colored dot (green for success, red for failure).

| Services | Time | Status |
|-----------|------|--------|
| Service 1 | 8:53 | ● |
| Service 2 | 8:54 | ● |
| Service 3 | 8:55 | ● |
| Service 4 | 8:59 | ● |
| Service 5 | 9:02 | ● |
| Service 6 | 9:05 | ● |

Vorteile der Automation

Die Automatisierung der Prozesse ermöglicht es Testern, sich auf wichtigere Qualitätssicherungsaufgaben zu konzentrieren.

Mit kontinuierlichen Tests können Ressourcen effizienter und flexibler eingesetzt werden (Offshore oder Onshore).

Die Anzahl an Tests ist problemlos skalierbar.

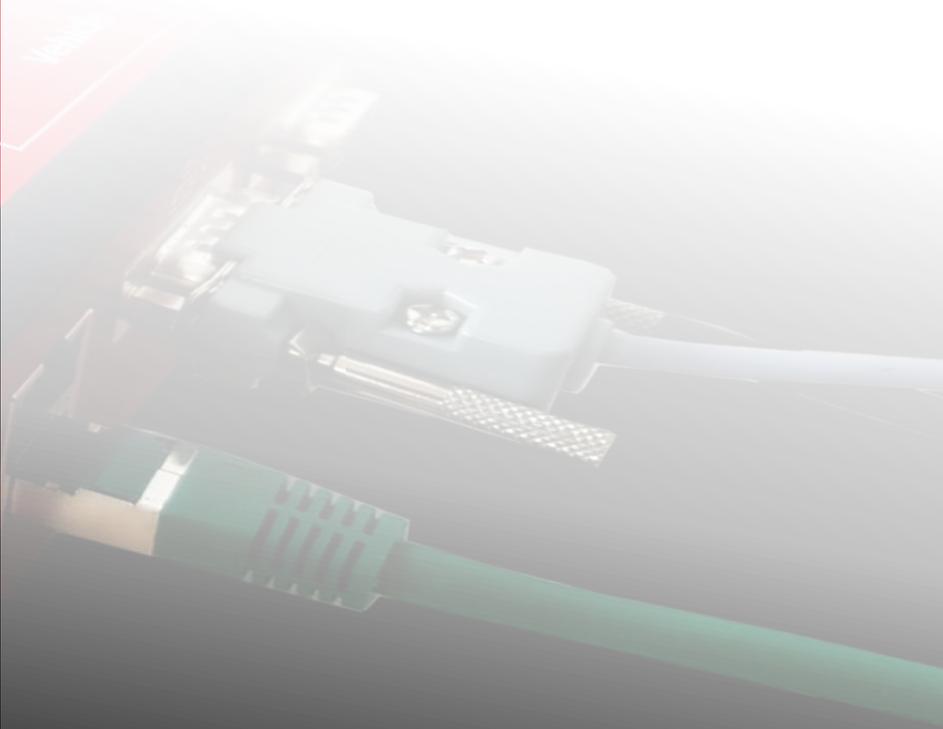
Das Monitoring hilft bei der Identifizierung von Störfällen.

Erhöhung der Testtiefe (mehr Testfälle werden in derselben Zeit durchgeführt) und der Testbreite (Erhöhung der Anzahl an Versuchsträgern und Testobjekten).

Das Defectmanagement ermöglicht eine strukturierte Behebung der Fehler, die im Rahmen des Testens gefunden werden.



FAZIT



Lösungen erlauben den Herausforderungen und Trends des Marktes entgegenzuwirken

Möglichkeit der Remote-Diagnose im Bereich Car + Car-IT.

Durch Testautomatisierung können Entwicklungsstände regressiv abgesichert bis zum Monitoring von Bestandsfunktionen durch Cron-Jobs.

Frühe Absicherung möglich im Entwicklungsprozess.

Unmittelbare Behebung von Fehler durch Testautomatisierung (Verlagerung des Aufwandes (€) ans Fehlermanagement) – Steigerung Qualität