

From Heiligs Blechle to the container

How Virtual Testing brings Vehicle and Cloud together

07.10.2026 | Bertrandt Group | Nic Eckstein

AI makes writing code faster. Who makes testing faster?

WRITING CODE

67%

report a significant increase in code volume

TESTING CHALLENGE

70%

consider maintaining their test suite to be a major challenge

More Code



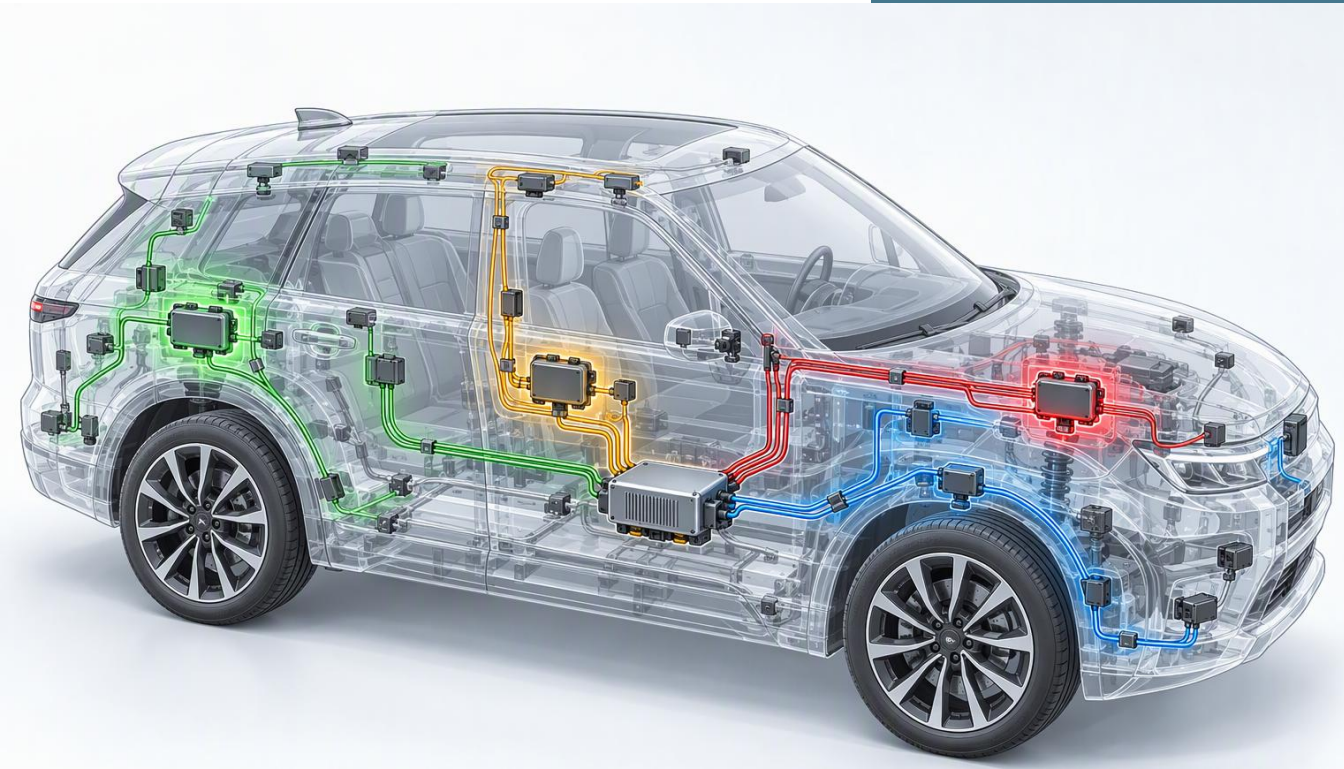
Greater Testing Needs

The bottleneck is shifting from writing to review, testing, and approval

01

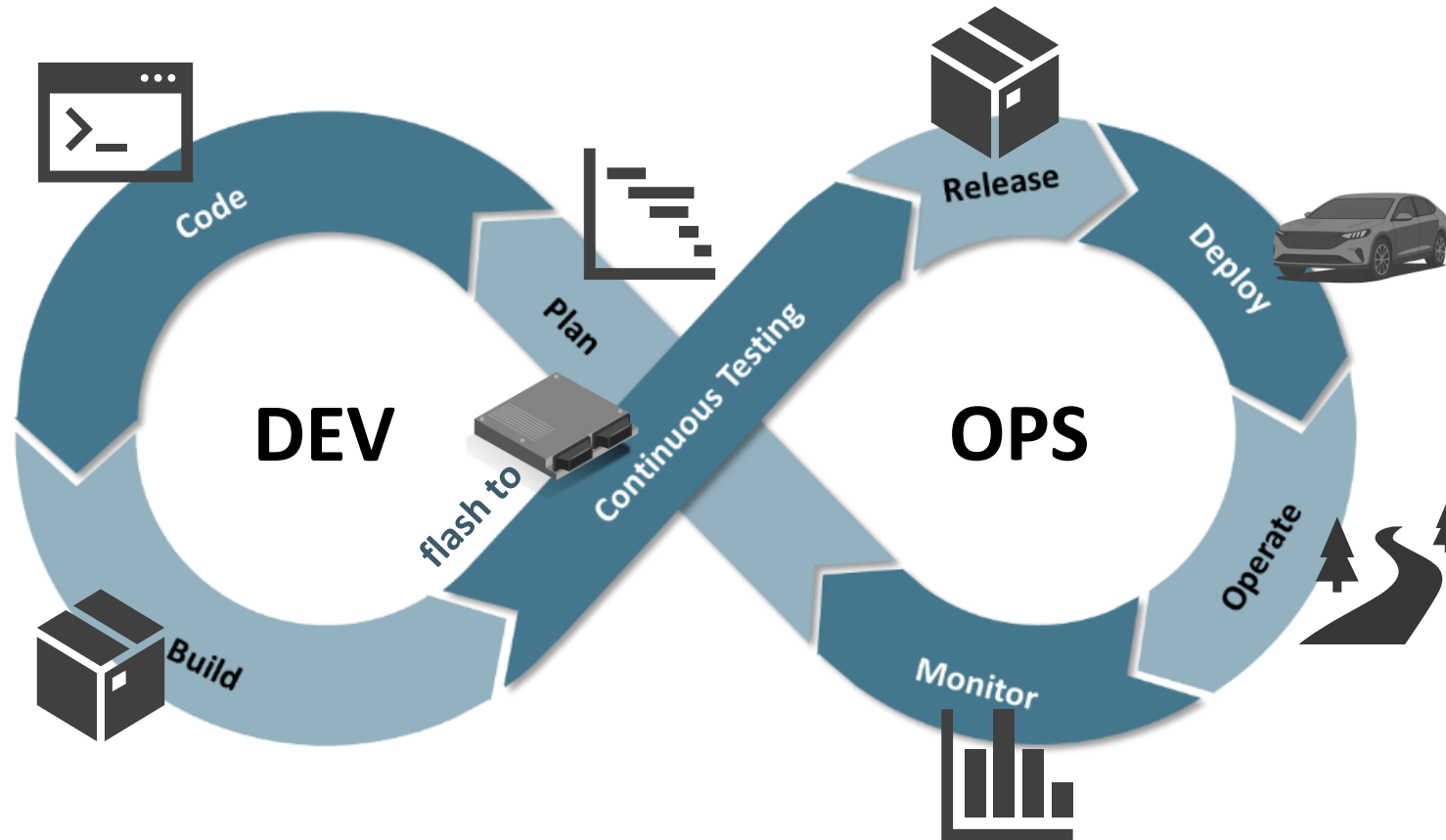
In the SDV age

Where do we really stand?



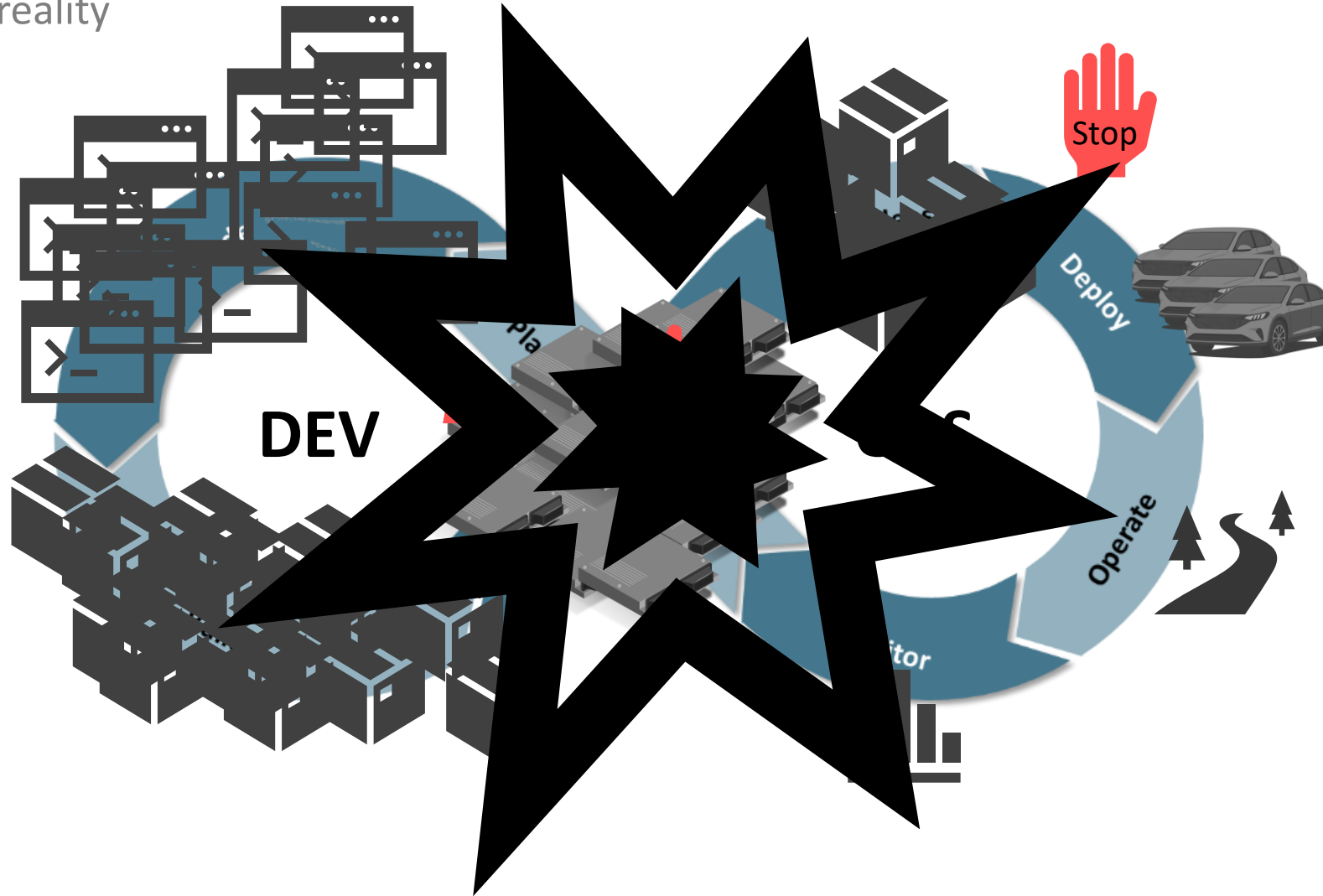
DevOps: Short, controlled learning loops

The Perfect Theory



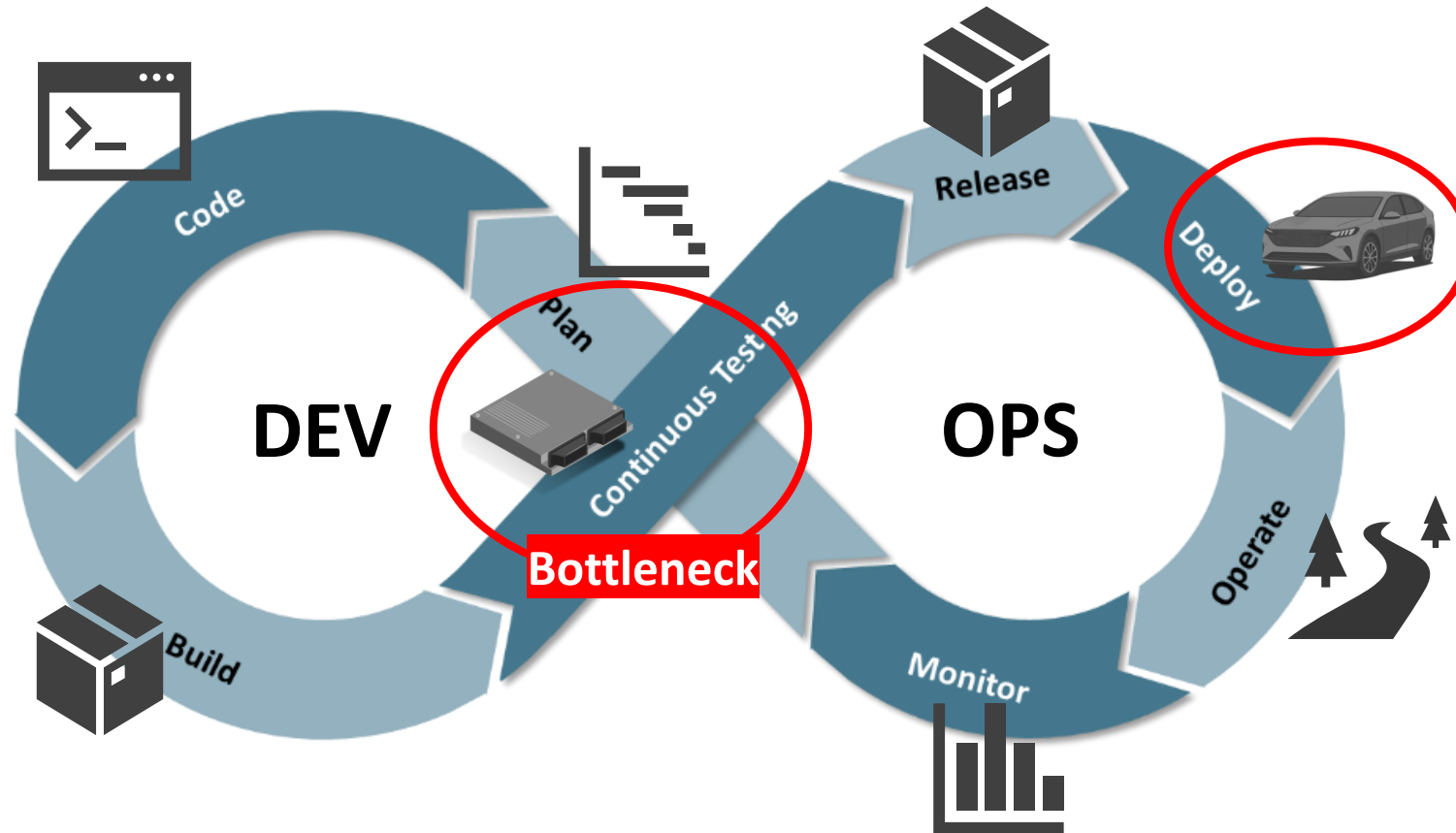
DevOps

The Automotive reality



DevOps

What can we improve?



Not every change goes immediately into hardware or into the vehicle.
But every change can be checked early, automatically and comprehensibly.



02

How can a container help us now?

The story from the harbour

Why does everything have to go into a "container" these days?

A brief history of the container

1956 - In the Harbour

Before: bags, barrels, boxes. Loaded differently for each ship.

With containers: a **standard size**. No matter what's in it — everything fits on every ship, every truck, every crane.

2013 – in IT

Before: Every installation a little different.

With Docker Container: Code + libraries + configuration in a box.

Works the same on any device:
data center or laptop of the colleague or in the vehicle.



Application

Libraries

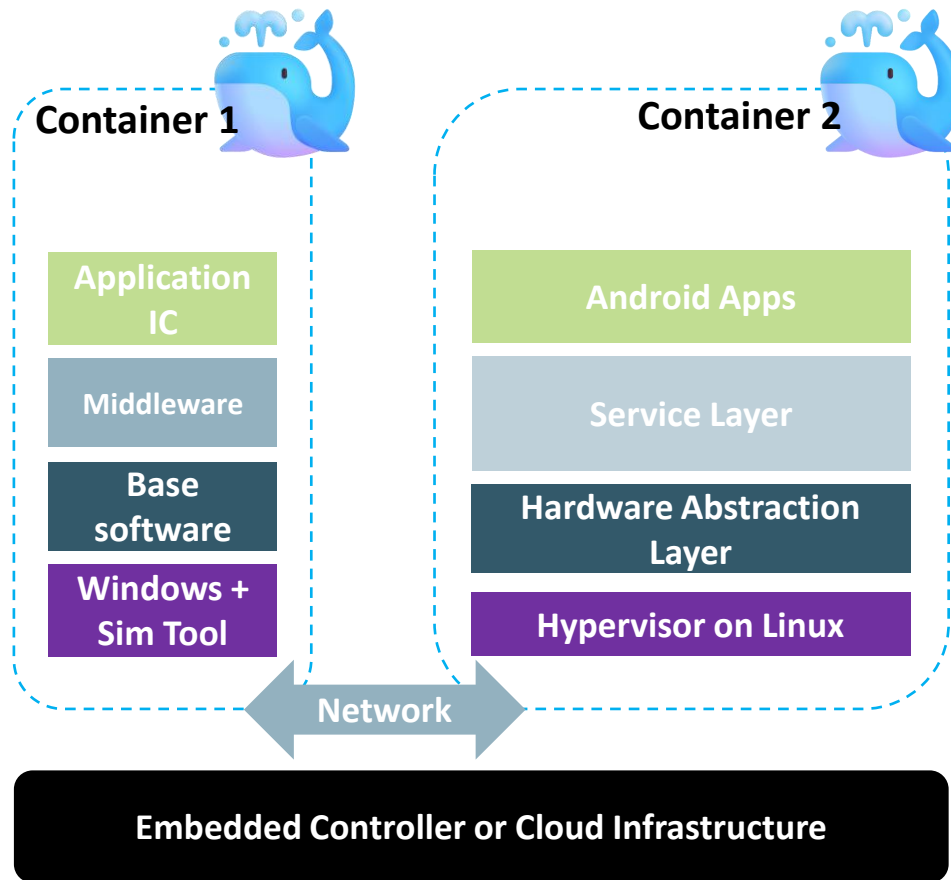
Configurations

Operating system
Linux, Windows & Co.

Whether it's a
local computer or a cloud

Carryover to Automotive

Using the Example of Instrument Cluster + Android Automotive OS (Infotainment)



A container does not automatically make software good. It makes delivery + execution more reproducible.



03

Scaling to the Moon

How modern cloud technologies help

Two options to scale



- **Principle:**
Upgrade existing hardware
(more CPU, RAM, memory)
- **Advantages:**
Easy to implement, often cost-effective for smaller performance increases
- **Challenges:**
Limited hardware cap, risk of failure

VS



- **Principle:**
Add more machines instead of improving existing ones
- **Advantages:**
Almost unlimited scaling, high reliability, good load distribution
- **Challenges:**
Increased complexity in software architecture (load sharing)

Scaling in practice

Using the example of...

A new software version for a brake controller.
10,000 traffic scenarios are to be tested overnight.

Classic

1 HiL test bench - sequential - ~30 s per scenario

Duration: 83 hours plus commissioning/set-up time

In the cloud

1.000 containers - parallel - ~30 s per scenario, same code

Duration: ~5 minutes plus commissioning

... and after the test?

- ✓ Results saved
- ✓ Container deleted
- ✓ No additional costs

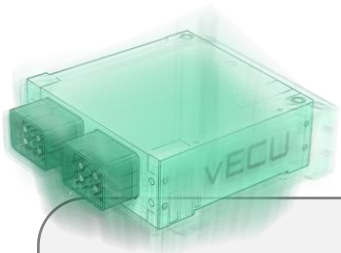


Physical hardware or virtual testing?

What now?

Hardware testing remains essential

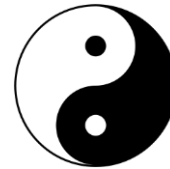
Virtual takes care of the volume, not the physics



Virtual Testing

vECU, SiL

Function variants, regression, daily builds



Hardware-Defined Testing

HiL, Testbench, Car

Timing, EMC, safety-critical functions,
final acceptance

No “either-or”. Virtual testing **does not replace physics**.

It reduces the load on hardware test instances so they can focus on what matters most.



04

**So how do I bring this
into my everyday life?**

Smart platforms help

Let's imagine



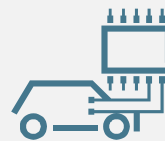
Each employee has a
HiL test bench available **at any
time**



Any test implementation can be
validated **immediately**



All variants are tested



The hardware test bench is free
for major releases and necessary
hardwaretesting

violis Platform

vECU
Testing

Data

Pipeline &
Workflows

Supported



SYNOPSYS®

MATLAB®
& SIMULINK®

Tools



python™



ecu.test™



PROVEtech:TA

Hosting



Azure

STACKIT





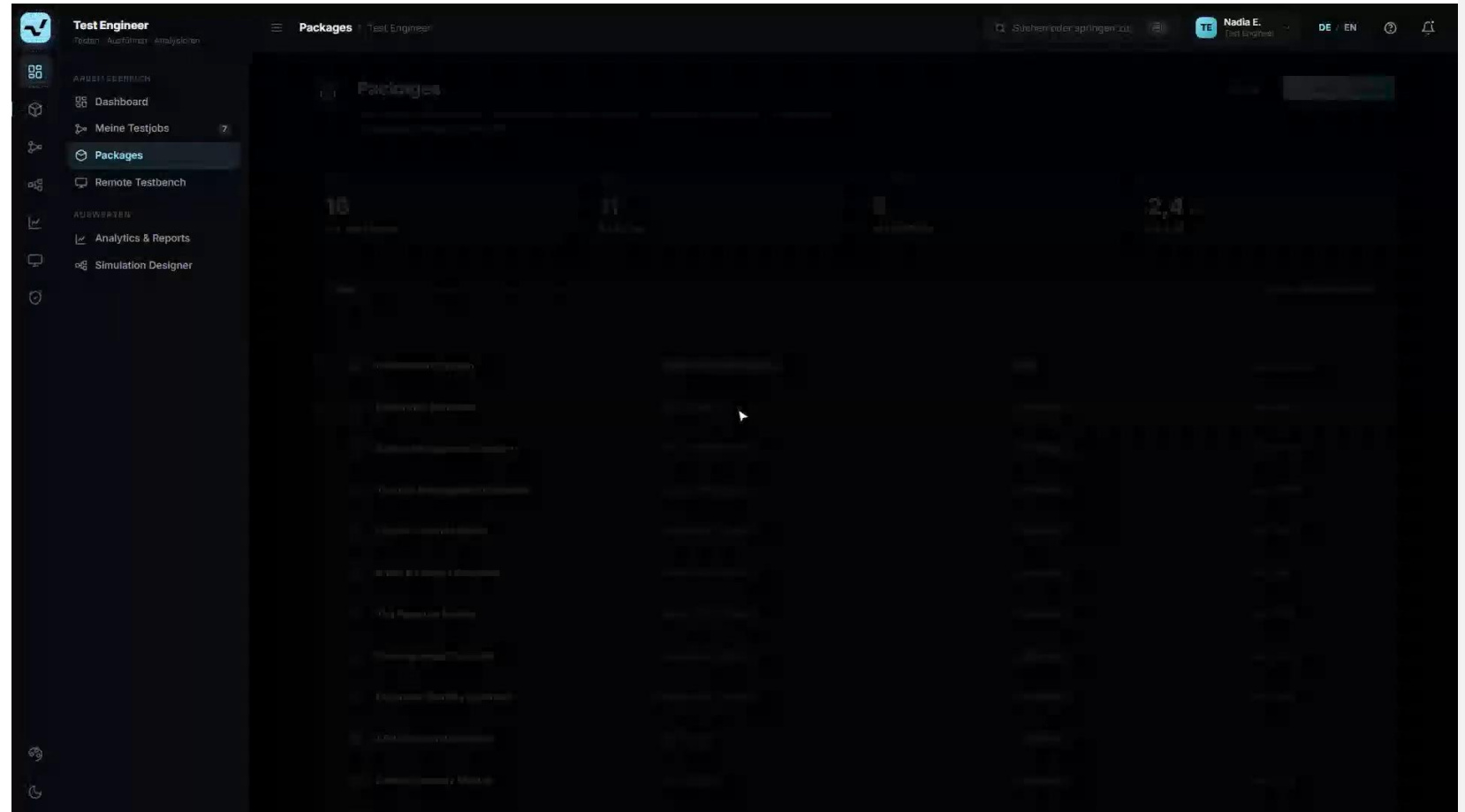
How does violis.io work?

Step 1: Delivery of your virtual device



A **package** is defined once.
Packages can be connected to systems.

Then you upload any new version.

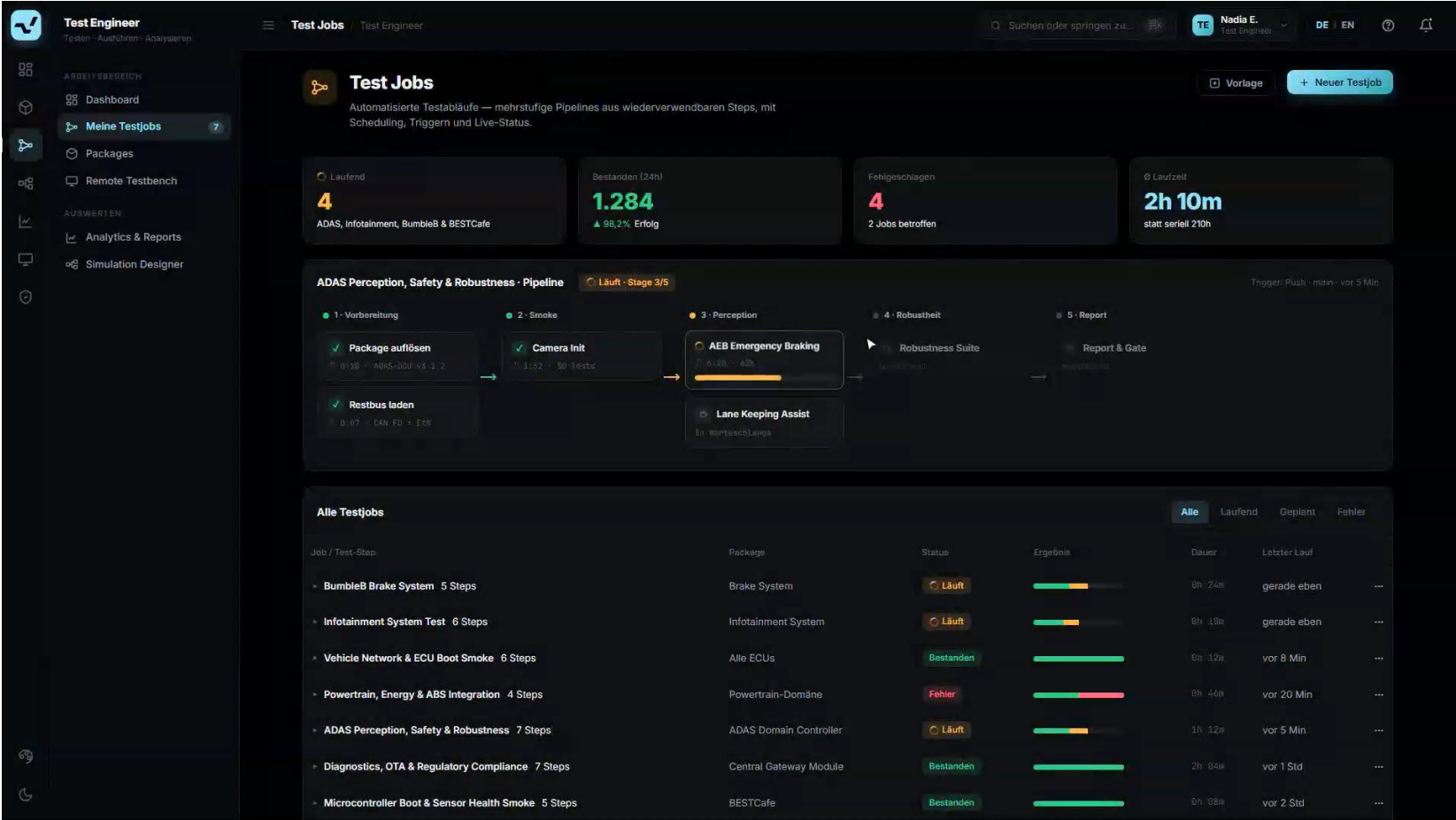


Step 2: Configure the Test Job



Configure your **Test Job** with your existing and linked tests.

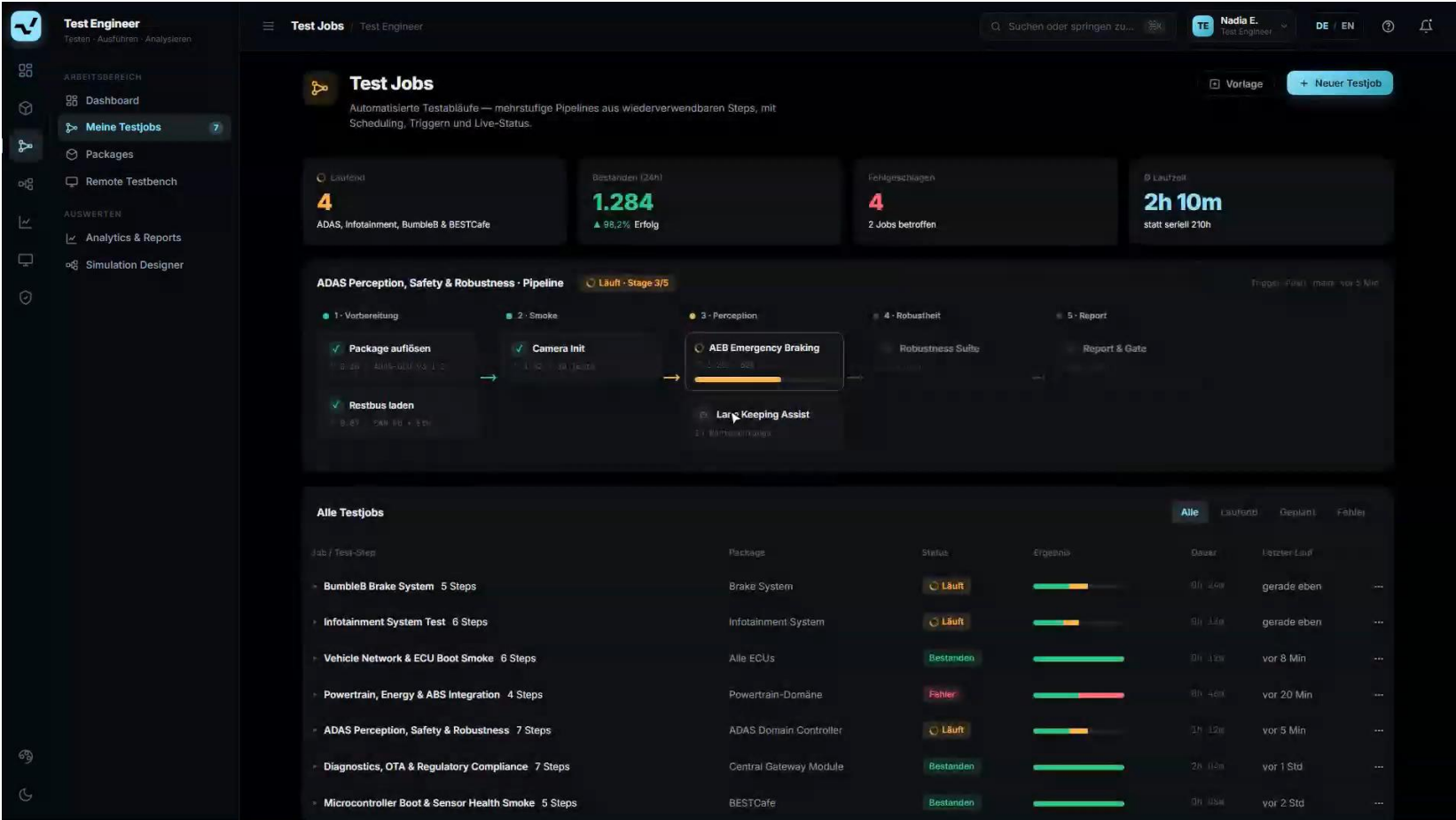
The test job starts automatically with each delivery.



Step 3: Build and scale



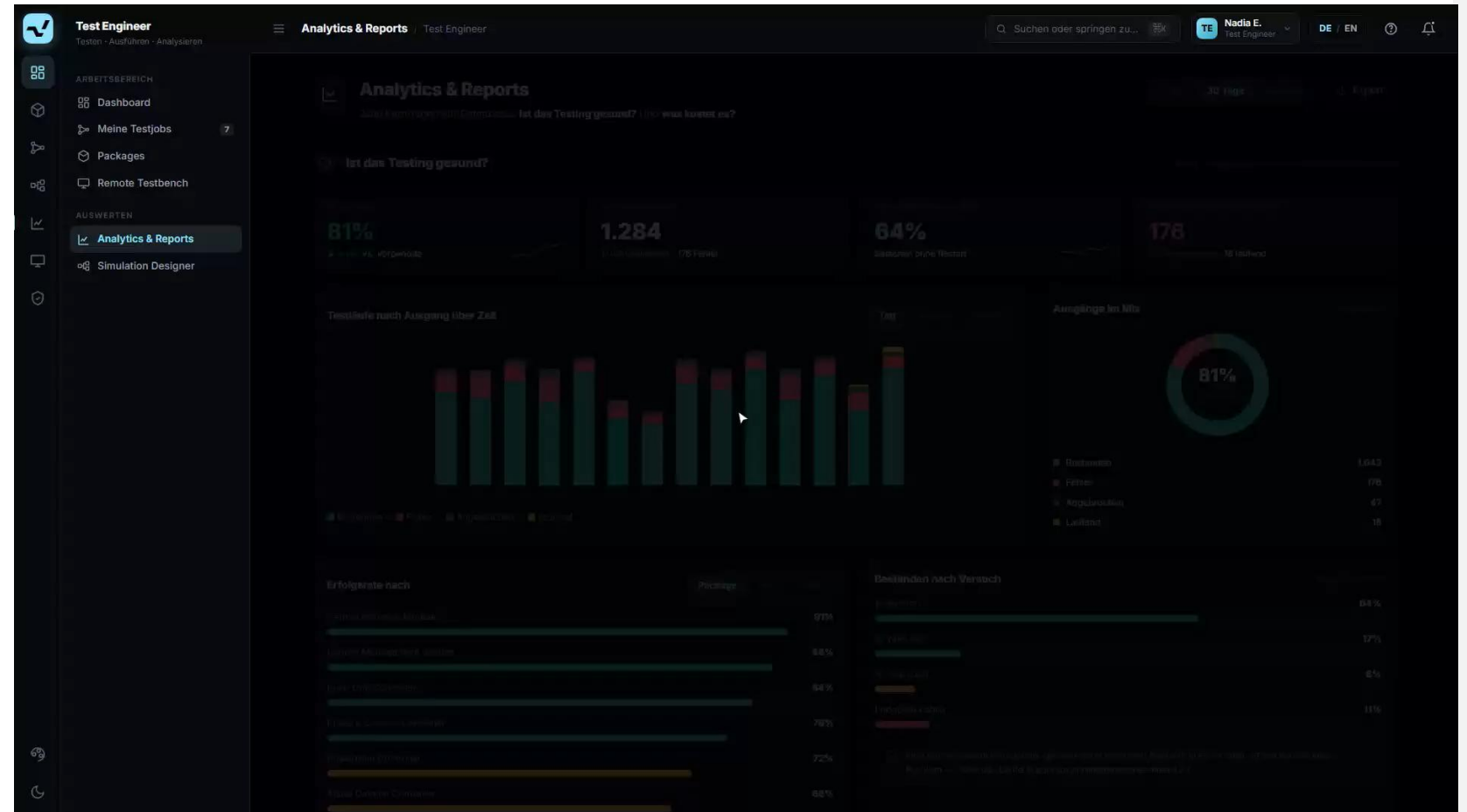
violis.io automatically creates the virtual test environment and **starts dynamically Containers** in the cloud.



Step 4: Analyze & debug



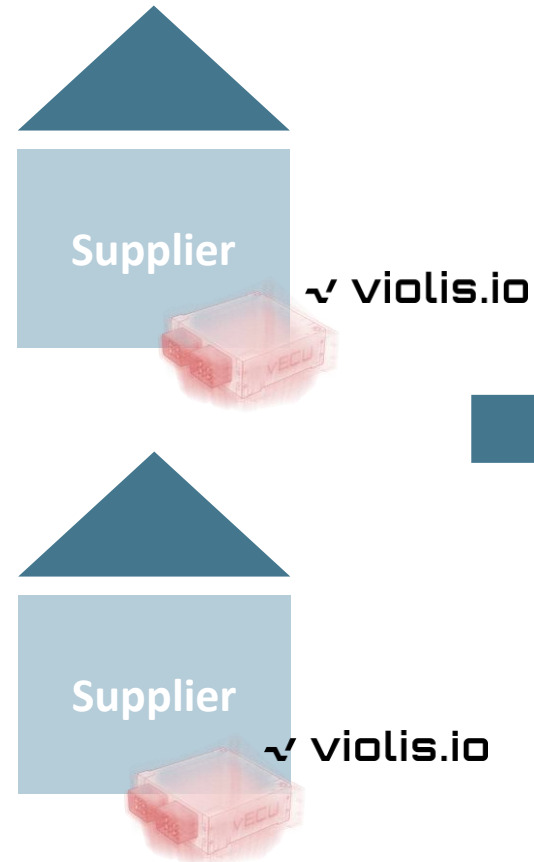
Smart analysis of the results and debugging with the tools you're already using.



End-to-end testing

from supplier to integrated system

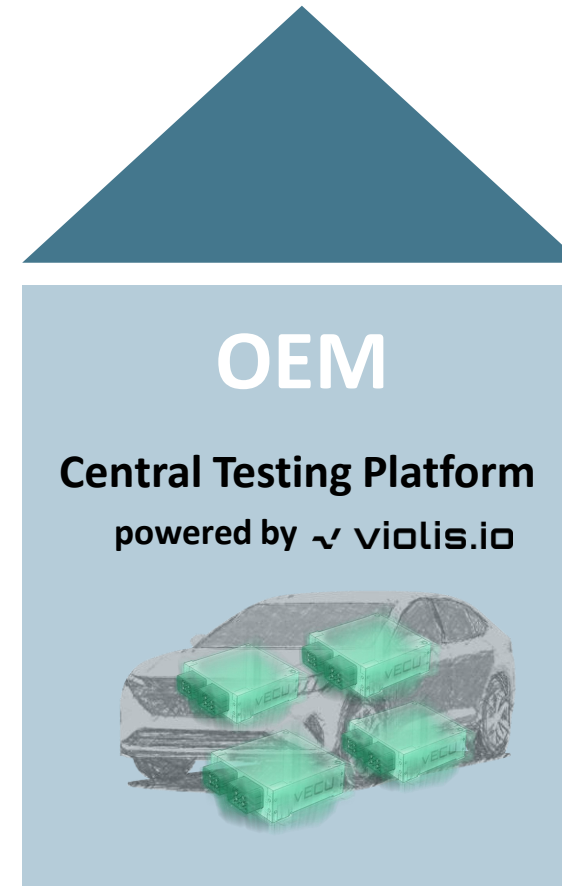
vECU creation and
preliminary testing with violis



Ready for
integration

Q-Gate

Virtual testing
in the integrated system



Ready for hardware
testing

Q-Gate

One shared platform, secure workspaces, clear handovers. Identify and fix defects early.

Concrete results from a customer project

Reference values of a highly complex zone control unit

5x
faster

than the execution on a hardware testbench by parallel testing of all buses

Daily
tests

compared to weekly test runs on the hardware testbenches

>40%
Cost
savings

compared without the use of virtual test solutions

Take-aways

for next Monday



1. Give it a try

You don't need everything. Start with a SW-component, one sub-system...

Scaling comes naturally.



2. Smart platforms help

violis.io & Co. take the IT complexity off your hands.

As an engineer, you continue to work in familiar tools.

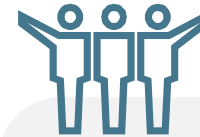


3. Both and

Hardware remains the final anchor.

Cloud is the scaling layer before and after.

No either-or.



4. Accept both languages

The bridge between engineering and IT is built by people, not tools.

Change is organization.



Nic Eckstein

Team Manager
Electronics & VirtualTesting Solutions



+49 175 6006910



nic.eckstein@bertrandt.com

