© Softing Automotive

Biggest Possible Development Efficiency with Remote Engineering

Development projects are increasingly being handled by teams whose members are spread all over the world. Remote Engineering, meaning remote access to test objects, allows efficient networking of global work capacities. However, there are a number of aspects that need to be considered during implementation to ensure a smooth process, as Softing Automotive explains.

AUTHOR

Markus Steffelbauer
is Head of Product Management and Marketing at Softing Automotive Electronics GmbH in Haar near Munich (Germany).

GLOBAL ENGINEERING LANDSCAPE

Throughout all industries modern vehicles have one thing in common: increasing complexity. This can be put down on the one hand to an ever increasing number of convenience/comfort functions which make driving more pleasant. But, above all, safety functions are playing an ever greater role: Lane keeping assist and blind spot warnings have gained a foothold in PCs, the emergency brake assistant in trucks has become an important feature, and construction and agricultural machines are being offered with numerous new functions. The growing trend of autonomous driving is ultimately causing this phenomenon to explode because at autonomy level 3 at the latest, the driver is remote from the driving situation which means the vehicle has to be able to guarantee safety itself.

But increasing complexity has not seen a lengthening of the engineering cycles. Attempts to master this situation see Electronic Control Units (ECUs) and the overall vehicle system now being engineered across several sites all over the world. Basically, this means that engineering can be carried out 24 hours a day: A colleague in Central Europe takes over a task from the Asian engineer and then hands his work on to America. Competence centers are also often set up to take on specific functions. For example, control functions are engineered in the Far East, testing takes place in India, prototyping in Germany.

This situation also highlights a bottleneck. Hardware is scarce particularly in the early phases of engineering; ECUs, test benches and vehicles always have to be shared. Deliveries can be organized, which takes a lot of time, or employees sent elsewhere by plane which is both inefficient and expensive.

Electric vehicles bring with them an additional challenge: The high-voltage technology means that in many cases only specially trained experts should access the vehicle.

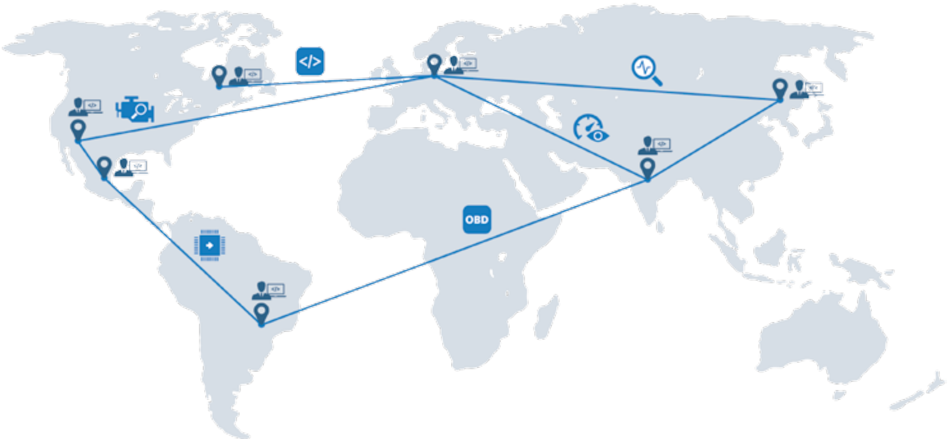
The name of the solution is Remote Engineering: There is a Device Under Test (DUT), which can be accessed by lots of people with a different focus from all parts of the world. Remote diagnostic access is an excellent example of this, **FIGURE 1**.

REMOTE DIAGNOSTICS IN ENGINEERING

As soon as an ECU is no longer developed independently of its environment, several people tend to be involved. Validation takes place at test benches which are as varied as the underlying vehicle functions. For example, Hardware-in-the-Loop (HiL) testers with a simulated ECU environment, component/function testers and endurance test benches are used. To guarantee communication between the ECUs, an integration test is run on test benches and what are referred to as Functional Mockup Units (FMUs), and ultimately the vehicle is validated at several stages using prototypes.

Diagnostics certainly always plays a role: It is first developed and released, but is also used to check functions. In addition, it plays a further decisive role: It is the way the ECU software is updated. This usually takes place with the diagnostic communication protocols. The same is true of variant coding in which uniform ECU software can be specialized, for example to satisfy regional characteristics (steering wheel on the left or right, daytime running lamp on or off) or marketing specifications (same engine with different performance variants). Moreover, all other diagnostic functions are used in the test environment: addressing actuators, reading out measurement values, setting parameterization variables and error memory operations. All these tasks can be run by the engineer directly on the DUT, the ECU with the environment required in each test case. However, with distributed methodologies, in which a number of engineers process different subfunctions worldwide, this is no longer possible in this way. A large number of DUT would be required which would have to be in the same state at all times: in terms of their environment, their software status and their parameterization. An obviously better way would be the provision of a test bench which, if required, could be accessed remotely via the network from a whole range of engineering sites. Incidentally, with a smart test procedure, the time at the DUT can be optimized: First of all the DUT is set up (for example by updating the ECU software) and then the actual test is run. While this test is being evaluated and documented offline, another colleague can start testing.

FIGURE 1 Global engineering collaboration (© Softing Automotive)



NUMEROUS CHALLENGES

In practice, it quickly becomes apparent that remote access is not always immediately possible. A range of factors – also in their various permutations – play a major role:

- infrastructure
- vehicle condition
- tools
- Information Technology (IT) security.

In terms of the infrastructure, bandwidth and latency have to be taken into consideration depending on the application case. The bandwidth can become a limiting factor particularly when it comes to flash programming the ECUs. This will not usually be the case at test benches as work mostly takes place within an internal network. In the case of test vehicles, particularly in summer and winter testing, it is important to consider the times at which large amounts of data are exchanged. Latency particularly limits real-time behavior when reading variables, although the requirements of latency have to be defined according to the distance in any case. Last but not least the availability of the network has to be taken into account. Particularly when tests are run on moving vehicles, the test sequence has to be separate from data transmission as interruptions/failure can otherwise often lead to unusable results.

In support scenarios, the vehicle condition or state is also key as ECUs are not stateless with respect to diagnostics. A number of functions are protected using security mechanisms and can only be run in special diagnostic sessions. If a specialist wants to connect up a system to help a colleague on site, the exact situation must be clear in advance – not an easy task particularly with automatic test runs. Addressing actuators, such as engines, in remote cases is also an aspect that needs to be considered in detail. It must be made certain that there can be no injury or any impairment to people.

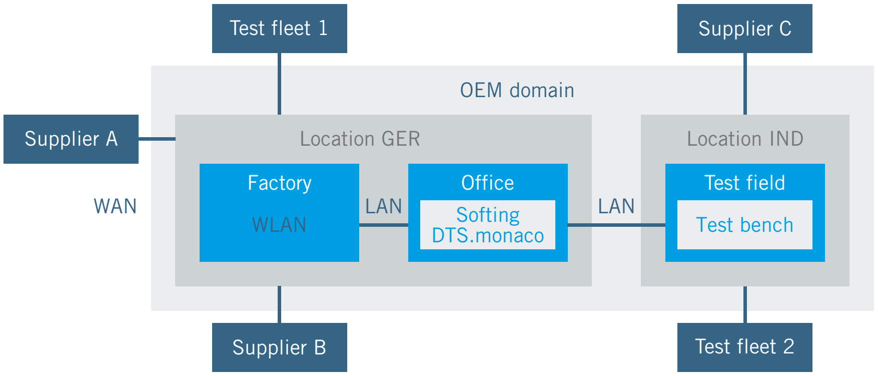


FIGURE 2 Location-independent working (© Softing Automotive)

When it comes to the tools used, at least the compatibility in terms of data and configurations must be ensured depending on the architecture. If, as explained later, remote access to the installation is direct, licensing and versioning are important. The best case would see configuration management ensuring that the same version and license are used everywhere. Nowadays this is easy to implement with corresponding update strategies.

Another obstacle is now developing: the necessary protection of the vehicle. This involves the actual access to the vehicle being protected with server authentication if this is not taken care of via the standardized On-board Diagnostics (OBD) jack. But authorization is also required for access to the relevant ECUs. The bus communication itself is also increasingly being encrypted to enable better protection against undesired access in the vehicle functions. All mechanisms are relevant to diagnostics, in remote scenarios all participants must therefore have the necessary authorizations.

But there is one problem the best strategies cannot solve: If hardware has to be replaced – ECUs, sensors, actuators – this can only be carried out on site by someone responsible for the DUT.

MANIFOLD APPROACHES

There are various methods used to carry out the remote access described, **FIGURE 2**. Some of the technology necessary is already available or easy to extend, but for some application cases it still has to be created. In terms of the network, a distinction has to be made between some of the basic technologies:

- Wireless Local Area Network (WLAN, so-called WiFi) within a particular area: A direct WiFi connection can be established within a repair shop or production hall.
- A Local Area Network (LAN) within a site enables a connection between two network-compatible devices, even if wireless adapters are used. These usually have to be registered in the network, in other words are known to the relevant IT.
- Cross-site connections provide an extended LAN area. Under the control of the particular IT – and thus without the challenges of a real Wide Area Network (WAN) – information can also be transmitted between sites that are at a great distance from one another.
- In a WAN, information can also be exchanged outside a company domain, for example with a supplier or – perhaps via 4G – with a test vehicle in winter/summer testing.

Remote Desktop Protocol Application (RDPA) is already being used today as a basis for applications, **FIGURE 3**. It is used, for example, in the Windows Remote Desktop, but also in add-ons such as Skype and TeamViewer. The advantage is that it is quickly available as a pure software solution and exactly mirrors the image of the connected system. In a support scenario, the expert thus sees exactly the environment his colleague has. Today, data can usually be exchanged directly. However, data often flows over systems

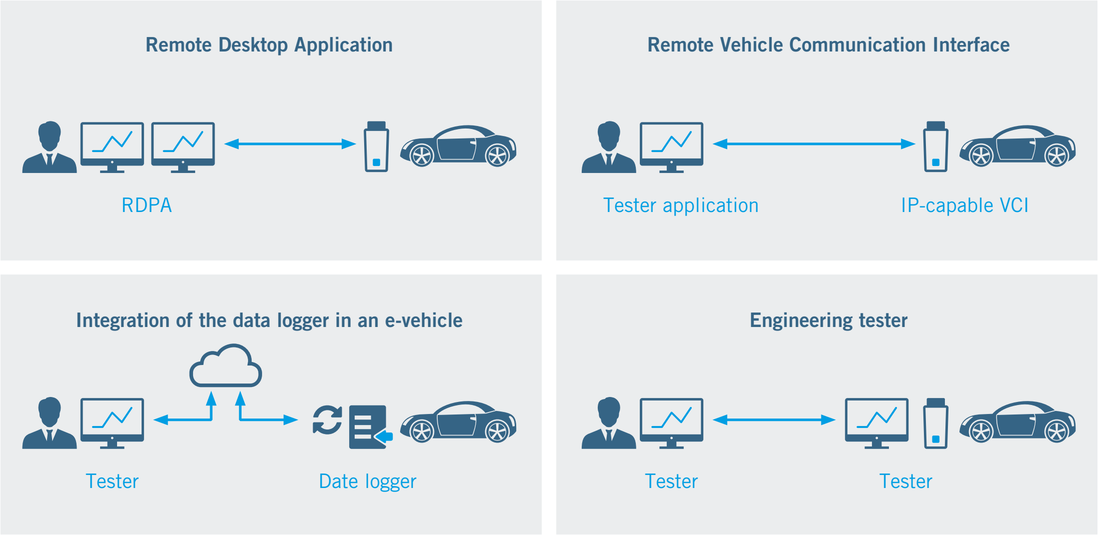

Remote Desktop Application
RDPA
Remote Vehicle Communication Interface
Tester application
IP-capable VCI
Integration of the data logger in an e-vehicle
Tester
Date logger
Engineering tester
Tester
Tester

FIGURE 3 Different Remote Engineering solution approaches (© Softing Automotive)

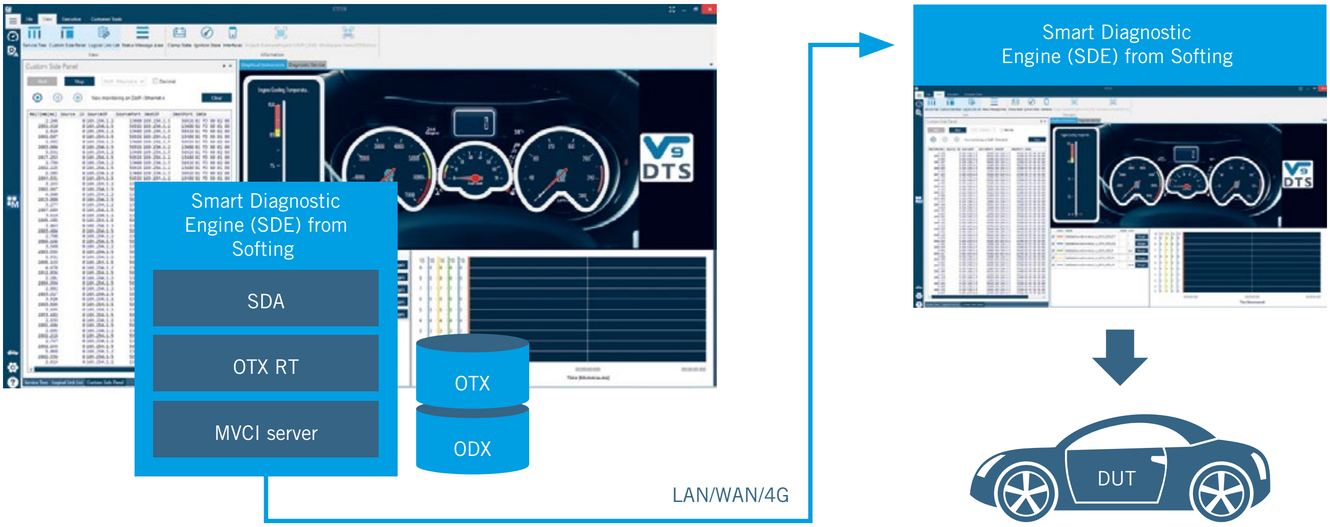

Smart Diagnostic Engine (SDE) from Softing
SDA
OTX RT
MVCI server
OTX
ODX
LAN/WAN/4G
Smart Diagnostic Engine (SDE) from Softing
DTS
DUT

FIGURE 4 Softing DTS.monaco: remote use (© Softing Automotive)

of external companies, something prohibited by many information security regulations.

In many cases, it is easy to use a modern WiFi-capable Vehicle Communication Interface (VCI). Basically, as soon as it is registered in the company network, it can be addressed locally. The real usability is highly dependent on implementation because a stable connection is required in modern protocols such as Diagnostics over Internet Protocol (DoIP); but the protocols of the Controller Area Network (CAN), such as Unified Diagnostic Services (UDS), used in most applications today, require precise time responses. This can only be enabled with suitable VCI implementation.

The extension of existing data logger solutions with diagnostic issues basically functions in a similar way. Today they are often installed in test benches and nearly always in test vehicles and, by definition, permit the exchange of data. This usually takes place asynchronously as the connection quality can often vary greatly particularly on test drives. A diagnostic solution must thus be adapted to this behavior and more or less be able to run self-sufficiently on the logger.

Furthermore, a modern engineering tester can open up communication Application Programming interfaces (APIs) for remote access. Then, the right tool can be used in engineering, so that an expert can access the existing environment in a support scenario as described above in the Remote Desktop procedure. Unlike with Remote Desktop, however, engineers can work in their own Graphical User Interface (GUI) environment, permitting advanced tests.

UNIVERSAL ENGINEERING TESTER

An example of this kind of engineering tester is Softing DTS.monaco. The tool can be used universally for all diagnostic and communication application cases:
- identification
- error memory operations
- flash programming
- measurement technology
- variant coding
- actuator test.

These functions are implemented internally using a functional API which extends the runtime environment for the standards OTX and ODX. This interface, called Smart Diagnostic API (SDA), can also be reached remotely which makes it possible for a second entity of the tester to connect up from a distance, **FIGURE 4**.

This facilitates a whole range of new collaboration models: The engineer develops new software and makes this available centrally. A tester implements the software at an appropriate time and starts testing. If problems occur, the engineer can connect up remotely, carry out his/her own tests and correct any problems directly (remote software update) so that the actual tests in hand can be continued.

REMOTE ENGINEERING IS ALREADY POSSIBLE

Today's engineering landscapes require new global collaboration models in which engineers can work together on common results in teams all over the world. Engineering testers such as Softing DTS.monaco make diagnostics possible both at a local level as well as in remote scenarios and thus support these models perfectly with very simple implementation on the IT side.