



## Software in the loop simulation at Daimler

### Application area

- Testing and deployment of functional code
- Version update safeguarding of functional code
- „Desktop“-application / -calibration
- Fault simulation
- Virtual endurance testing
  - ➔ safeguarding of drivetrain components
  - ➔ calculation of load collectives for gearbox and drivetrain

### Requirements

- Powerful, stable and fast simulation environment
- Easy to use by any engineer

### Tool chain

#### SIL-tool

- Backbone (in house development)
- Silver (QTronic GmbH)

#### Plant model

- MSL 2.2 in Dymola 6.2 (Dassault Systèmes)
- In the future MSL > 3.1 with Dymola from v. 7.4 or SimulationX from v. 3.4 (ITI GmbH)

#### Test generator

- TestWeaver (QTronic GmbH)

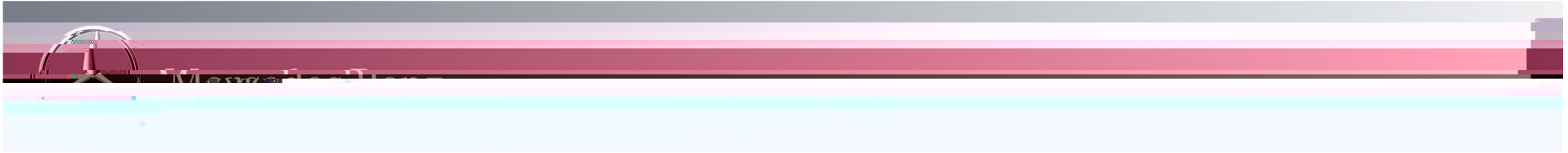
#### Software integration platform

- Microsoft Visual Studio 2005 or 2008



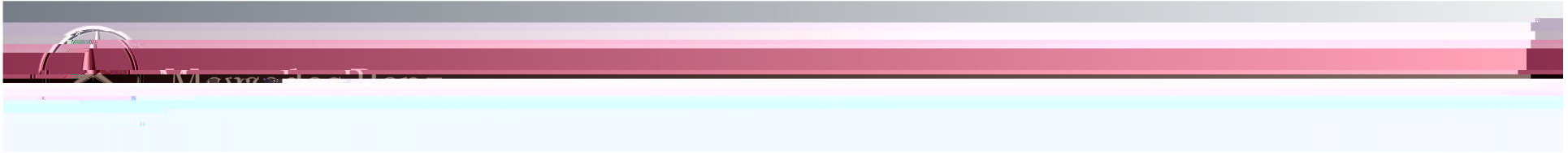
## SIL-environment / functionality

- The simulation is controlled by a special program (e.g. Silver) which guards the single modules
- Every module (called „Client“) sends its Outputs to Backbone und reads its Inputs from him, i.e. no direct communication between the modules occurs (except for the CANape-coupling with the control software )
- The communication step time is fixed and represents the lowest task time step of the functional code (5, 10 or 20 ms)
- The plant model is wrapped with a numerical solver which calculates with smaller time steps
- Backbone waits after each communication step until all clients are finished so that the next step



# Plant model

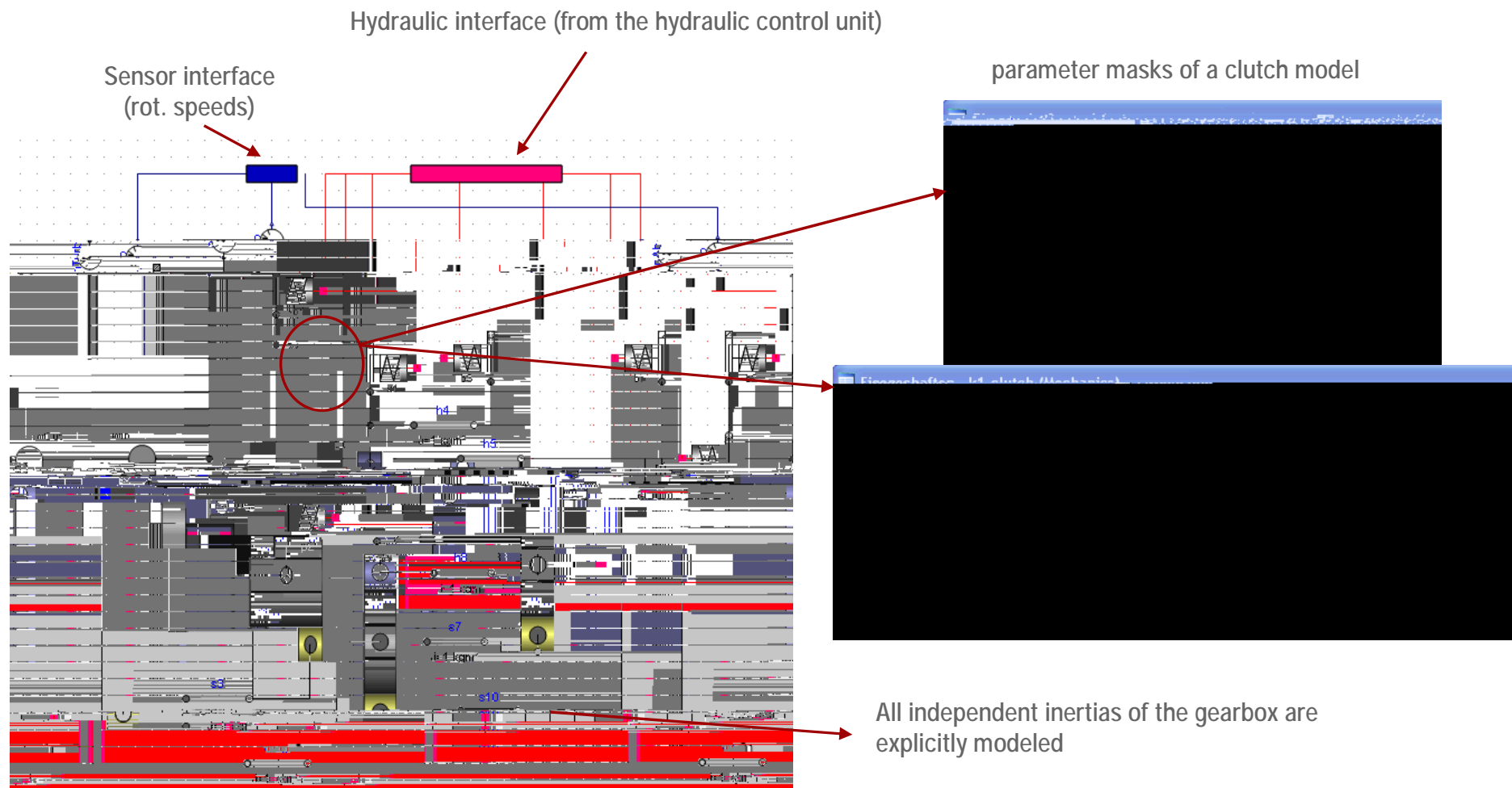
Description



## Plant model: Modelica Libraries

For the creation of the plant model, own devised

# Plant model: mechanics



## Plant model: hydraulics (control unit)

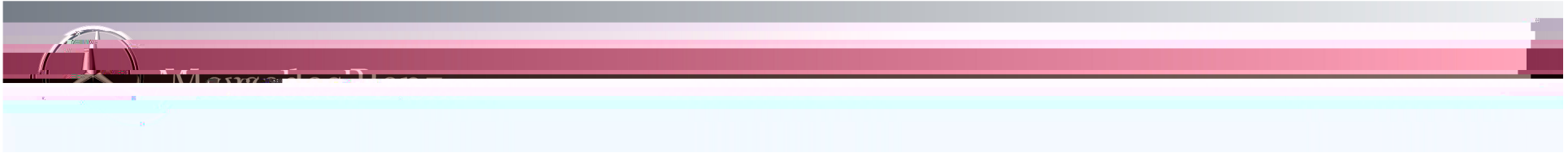


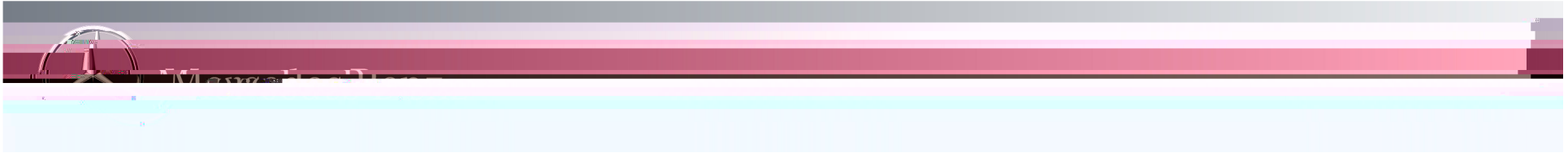
The electro-hydraulic control model has been modeled phenomenologically (control logic, no dynamics) for the sake of simulation performance. However, many components such as orifices, shifting valves, fluid volumes etc. have a physical model description in order to accurately describe important effects in the simulation (filling, draining, pressure switch, sticking valves etc.)

Parameter mask of a control valve









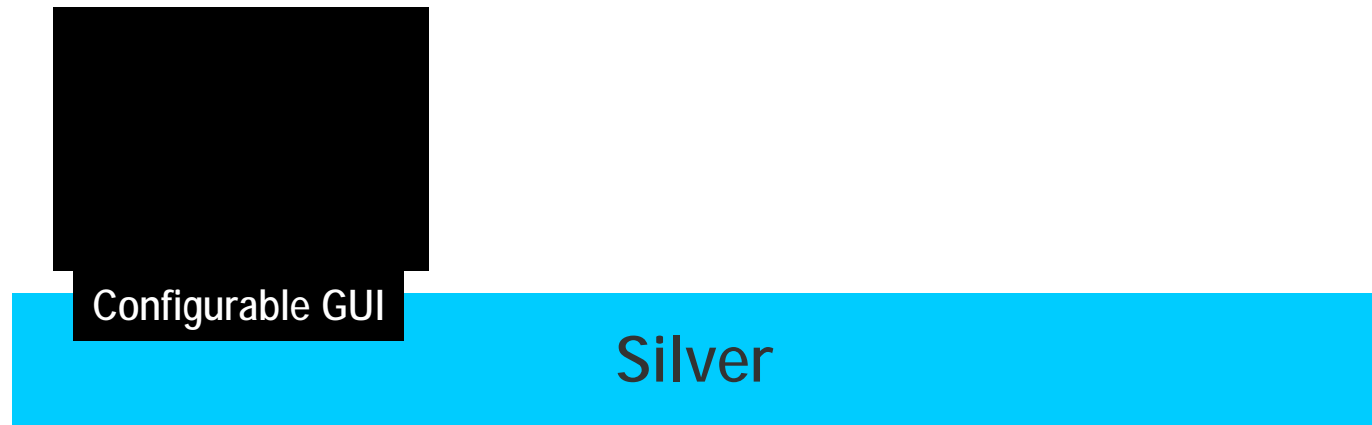
# Integration of a SiL-Project

## SiL environment

- Simulation: Silver (QTronic)
- Measurement: CANape (Vector)
- Debugging: Visual Studio (Microsoft)
- Automated Test: TestWeaver (QTronic)
- Code Coverage: CTC++ (Verifysoft)

## graphical user-interface:

- interaction of driver/user with simulated car
- accel pedal, steering, etc. can be controlled
- plotter, breakpoints, scripting, file in/out, ...



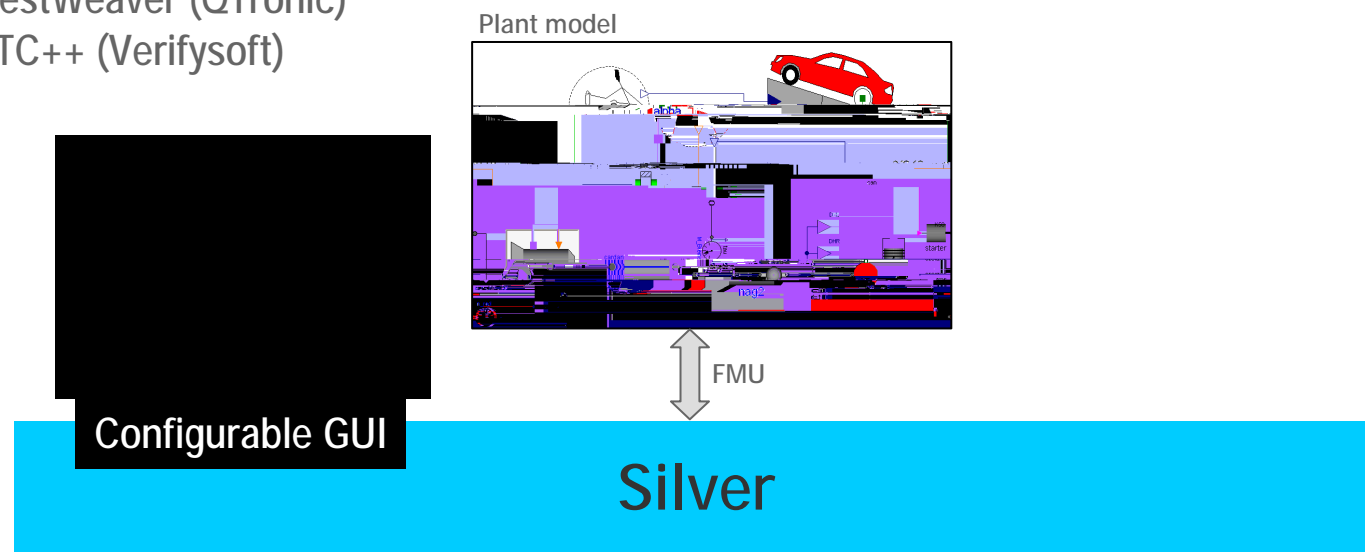
# Integration of a SIL-Project

## SiL environment

- Simulation: Silver (QTronic)
- Measurement: CANape (Vector)
- Debugging: Visual Studio (Microsoft)
- Automated Test: TestWeaver (QTronic)
- Code Coverage: CTC++ (Verifysoft)

## hardware DLL:

- simulated vehicle, engine and transmission
- Dymola/SimulationX



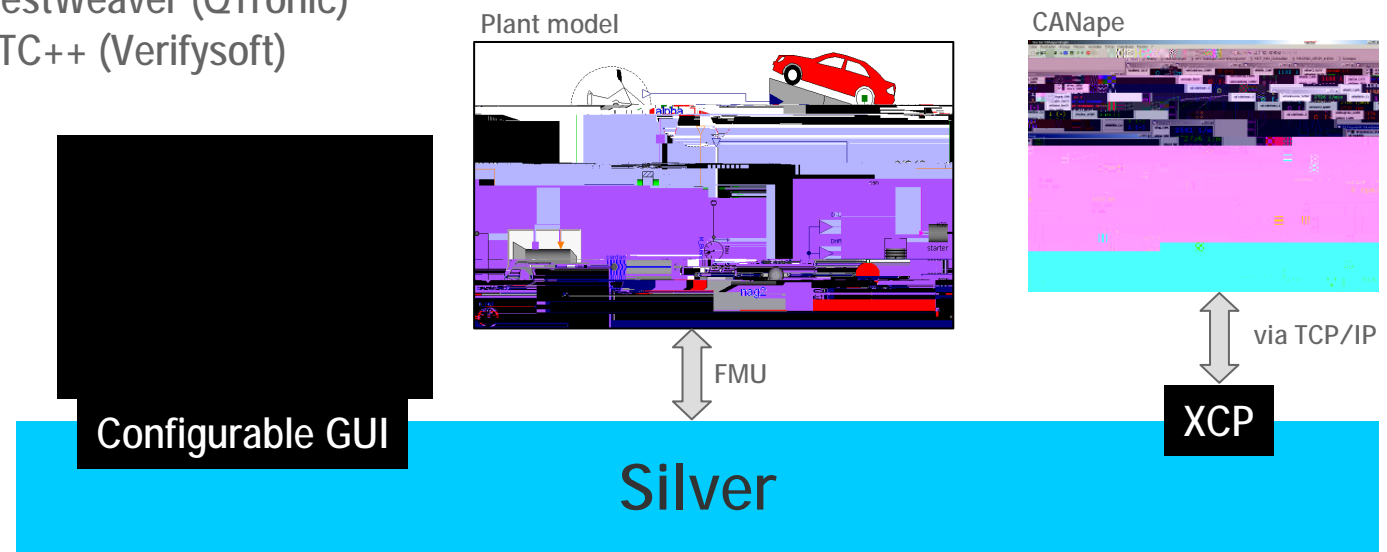
# Integration of a SIL-Project

## SiL environment

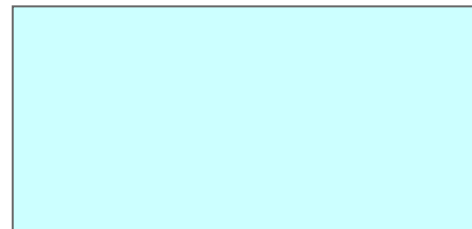
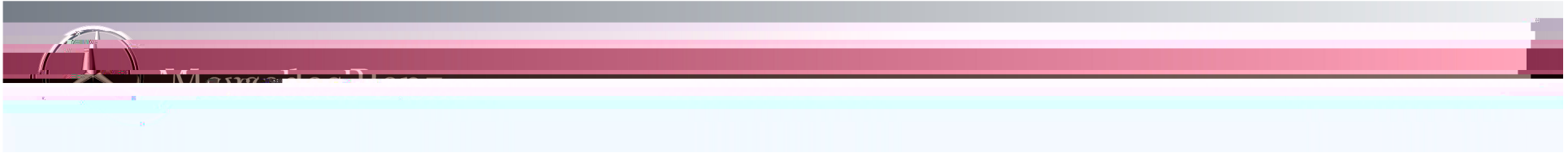
- Simulation: Silver (QTronic)
- Measurement: CANape (Vector)
- Debugging: Visual Studio (Microsoft)
- Automated Test: TestWeaver (QTronic)
- Code Coverage: CTC++ (Verifysoft)

## XCP with Canape/INCA:

- XCP measurements via TCP/IP
- no limitation of bandwidth as with CAN
- online calibration of parameters



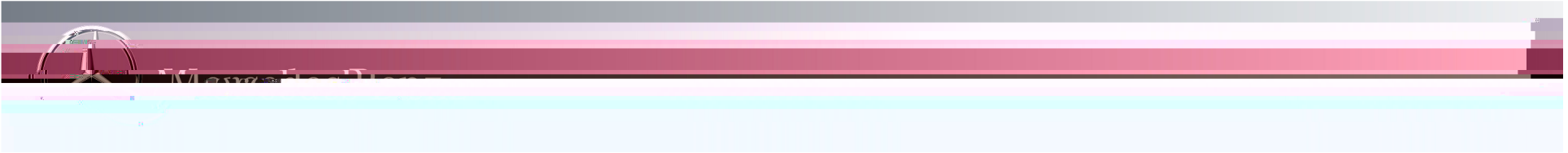












## Summary

- SIL is an essential tool in the gearbox development at Daimler
- For the creation of the SIL plant model, Dymola (MSL 2.x) has been used
- Upgrade of the model to MSL 3.1 enables full compatibility to SimulationX v. 3.4
- For the plant model export to SIL the new Modelisar-FMI can be applied
- SIL integration of the functional code (TCU) is done by wrapping the original code with the Silver-API and emulating the frame software
- Silver offers the possibility to measure and calibrate TCU-internal signals either directly in the Silver GUI