About the speakers

Ricardo Gonzalez Ramos

- Born 1973
- Joined ZF Summer 2003 (Master Thesis Mechanical Engineering)
- From 2004 onwards Embedded SW Developer Passenger Car Transmission systems
- Senior Manager Infrastructure and Tooling

Eric Bieber

- Born 1968
- 1990 Studies Electrical Systems (Telecommunications) HTWG Konstanz
- 1995 ZF Friedrichshafen AG Embedded Software developer
- 2001 ZF Friedrichshafen AG Infrastructure and tooling
AGENDA SLIDE

- What is OpenXilEnv and where it comes from
- Usecases for OpenXilEnv
- What will be part of open source
- Main Working principle of OpenXilEnv
- Demonstration based on Electric car
What is OpenXilEnv

Eclipse OpenXilEnv provides an environment for creating Software In the Loop (SIL) systems for the Software Defined Vehicle ecosystem. OpenXilEnv is primarily developed for setup a SIL (digital twin), through its versatile nature it is also possible to use it in a HIL environment.

OpenXilEnv provides also an interface to Matlab/Simulink where it can be used for co-simulation between Code and functional models in a Model in the Loop environment (MIL).

OpenXilEnv also provides a lightweight Hardware in the Loop (HIL) system option for MiniHils (CAN,CN-FD)

Where comes the the Name from?

- Open Source - > Open
- Environment -> Env
- Capabilities - > HIL,SIL,MIL
- Heritage Name - > Softcar

Summ of MIL and SIL and HIL is Xil
Softcar -> OpenXilEnv

First Softcar for DOS/Windows 3.11

1995

Memory protection
Separate executables

2001

09. Jan. 2007 first iPhone

2001

New user front end
Network layer
RPC

2016

64 bit
Multicore
Calibration XCP

2020

Rename to OpenXilEnv
Open source

2023

COPYRIGHT (C) 2023, ECLIPSE FOUNDATION. | THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)
Use cases and features of OpenXilEnv

- **SIL Software In the Loop**
  - No Target hardware, compiler, … needed

- All parts separated in own executables
  - Communication over a network layer

- Distributed digital twin, the control units/ models must not run on the same location

- Residual bus simulation for emulating ECUs not present
  - Fault injection

- FMI interface for FMUs

- XilEnv (without GUI)
  - No installation is needed.
  - Docker container, or/and in the cloud.
  - Automated Simulations incl. result evaluation

- XilEnvGui (with a Qt GUI)
  - Expansive configurable sheets
Use cases and features of OpenXilEnv

- Parallel execution schedulers/barriers (configuration must done by the user)
- Cosimulation interface to Matlab/Simulink
- Recording
- Stimulation through measurement data or script
- Pre calibration with XCP over ethernet to interact with an external calibration system
  - Or a small internal Calibration system
- RPC (Remote Procedure Call Interface) for automation

Target platform is Windows or Linux.
Mixed 32/64bit Windows/Linux executable.
What will be part of open source

What is our goal

To promote open source digital twin environment and simplify interaction between digital twin participants

What will be part of OpenXilEnv

The sources of OpenXilEnv to build:
- XilEnv
- XilEnvGui
- XilEnvRpc
- XilEnvExtProc
And some small example
You have to build executables your own.

What you can

Use it as it is, change it if you need, give back changes if you want. Contribution will be welcome

Eclipse OpenXilEnv
Working principles

- **Engine Model**
- **FMU**
- **Transmission Model**
- **Vehicle Model**
- **Battery Model**
- **Battery ECU**
- **Engine ECU**
- **Transmission ECU**
- **Carla**
- **XilEnv (GUI)**
- **CAN bus**
- **RPC**

**Input signals**

- **Engine IDE**
- **Debug**
```c
Internal
```
THANK YOU!