

Lessons Learned in Building Large, Distributed HIL Systems

Hans Nyström

Manager, Test System Design group

Prevas

hans.nystrom@prevas.se

Prevas provides innovative solutions helping our customers to create growth.

Prevas have supported our customers with solutions for over 15 years.



Reasons for Simulated Test



Testing abnormal behavior

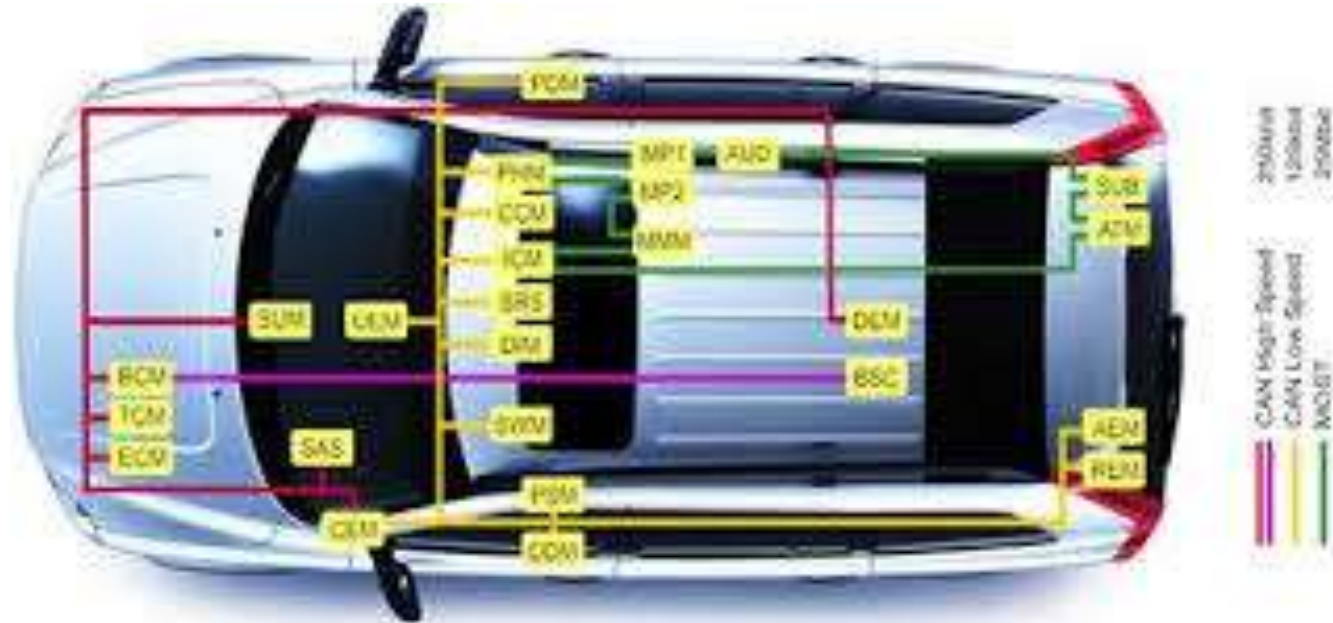


Ordinary testing is dangerous



Costly to run the system
(big machines)

Typical Network of ECU

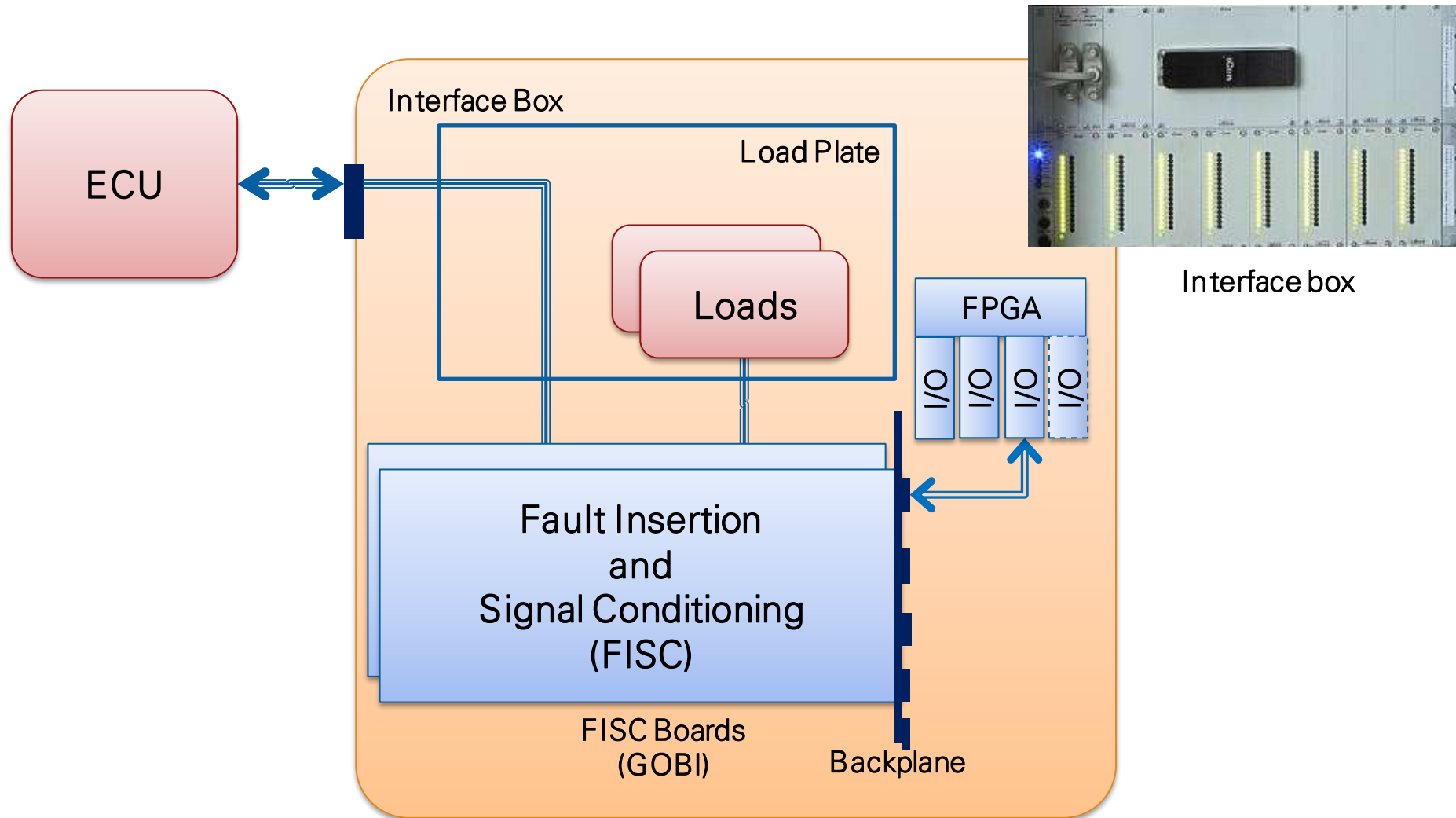


Challenge #1, Hardware

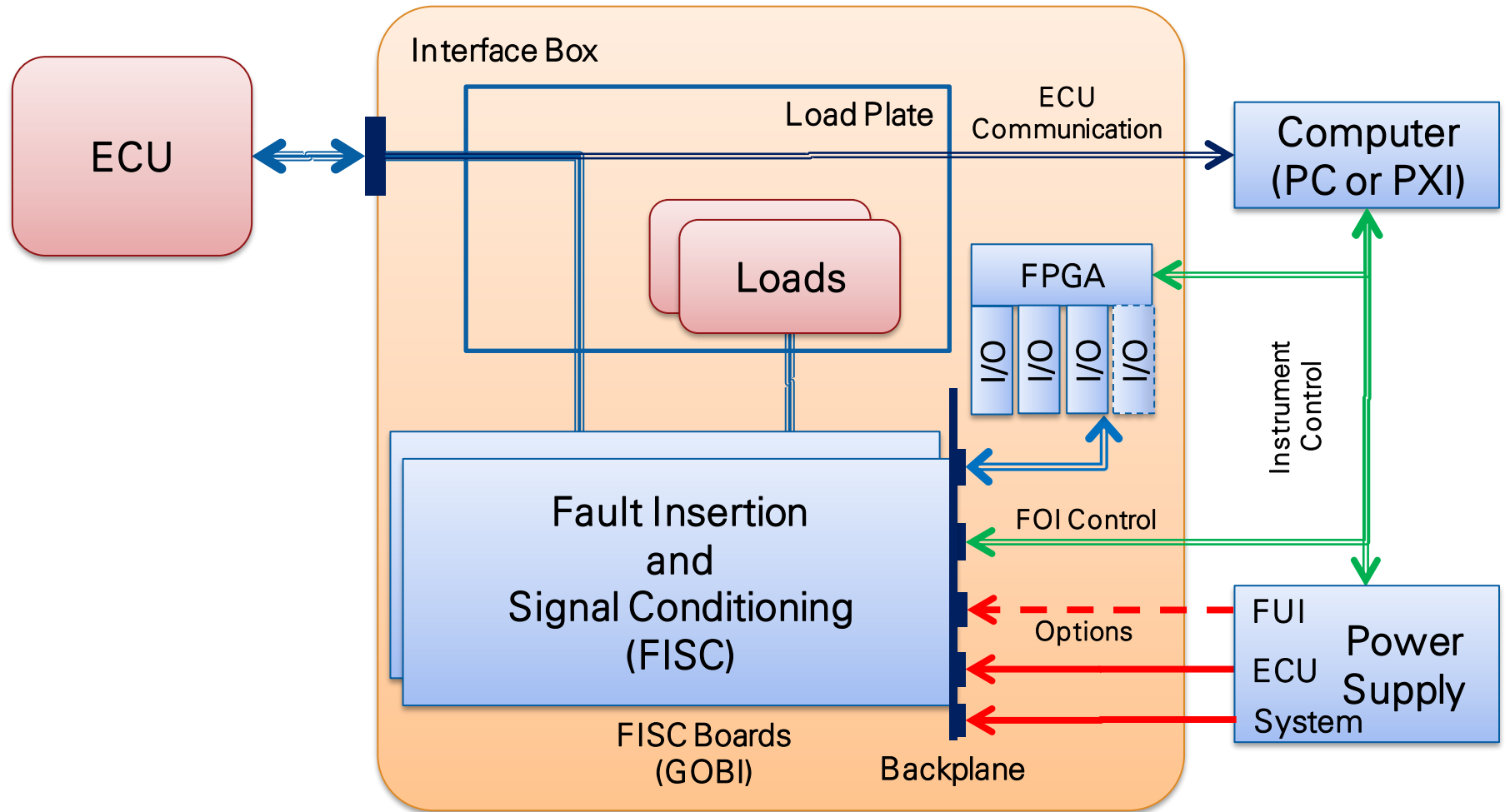
User requirements (identified and addressed):

- Flexibility
- Large amount of I/O
- Each signal needs to be equal to real signal
- Capable of testing robustness.

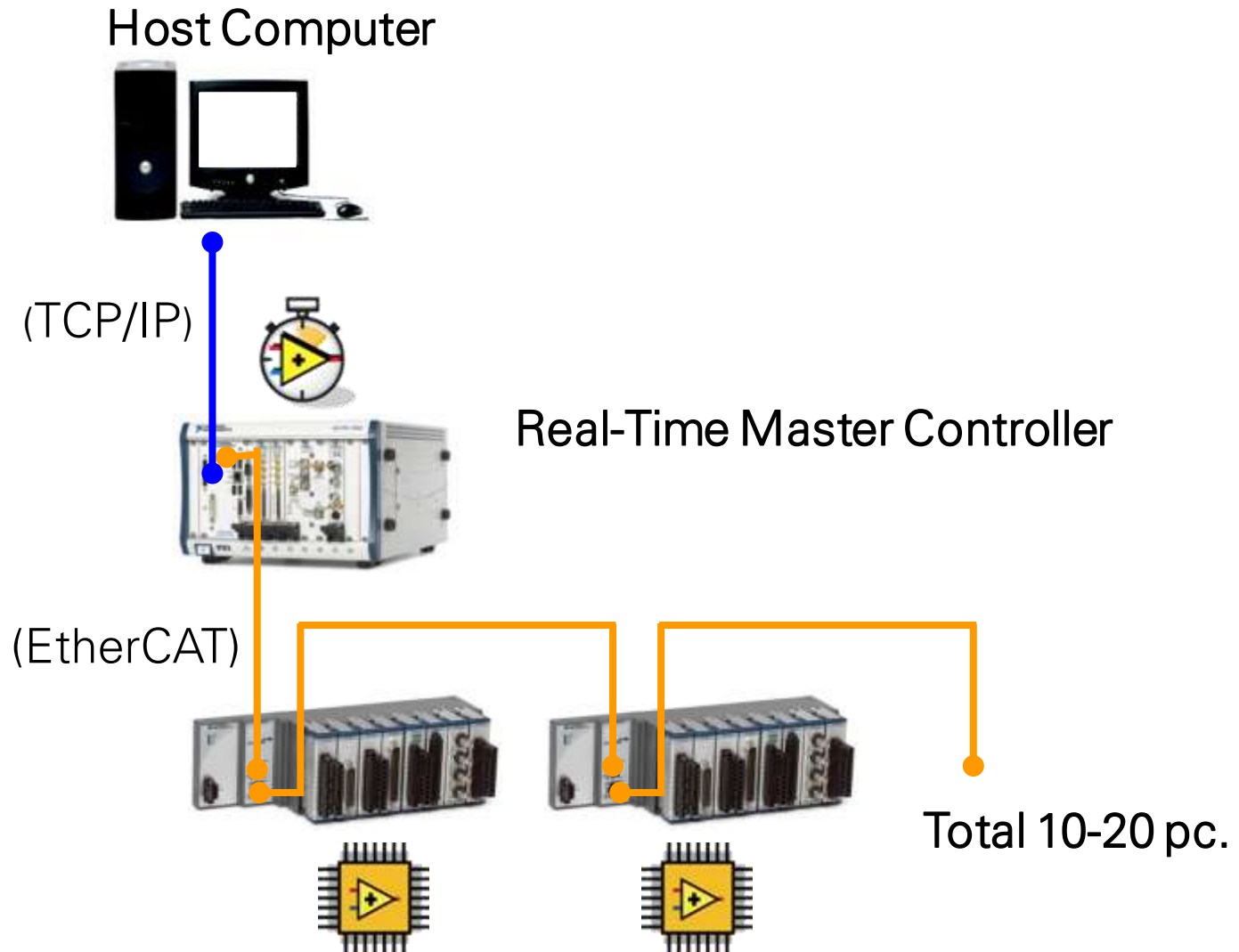
xMove System Architecture (Hardware)



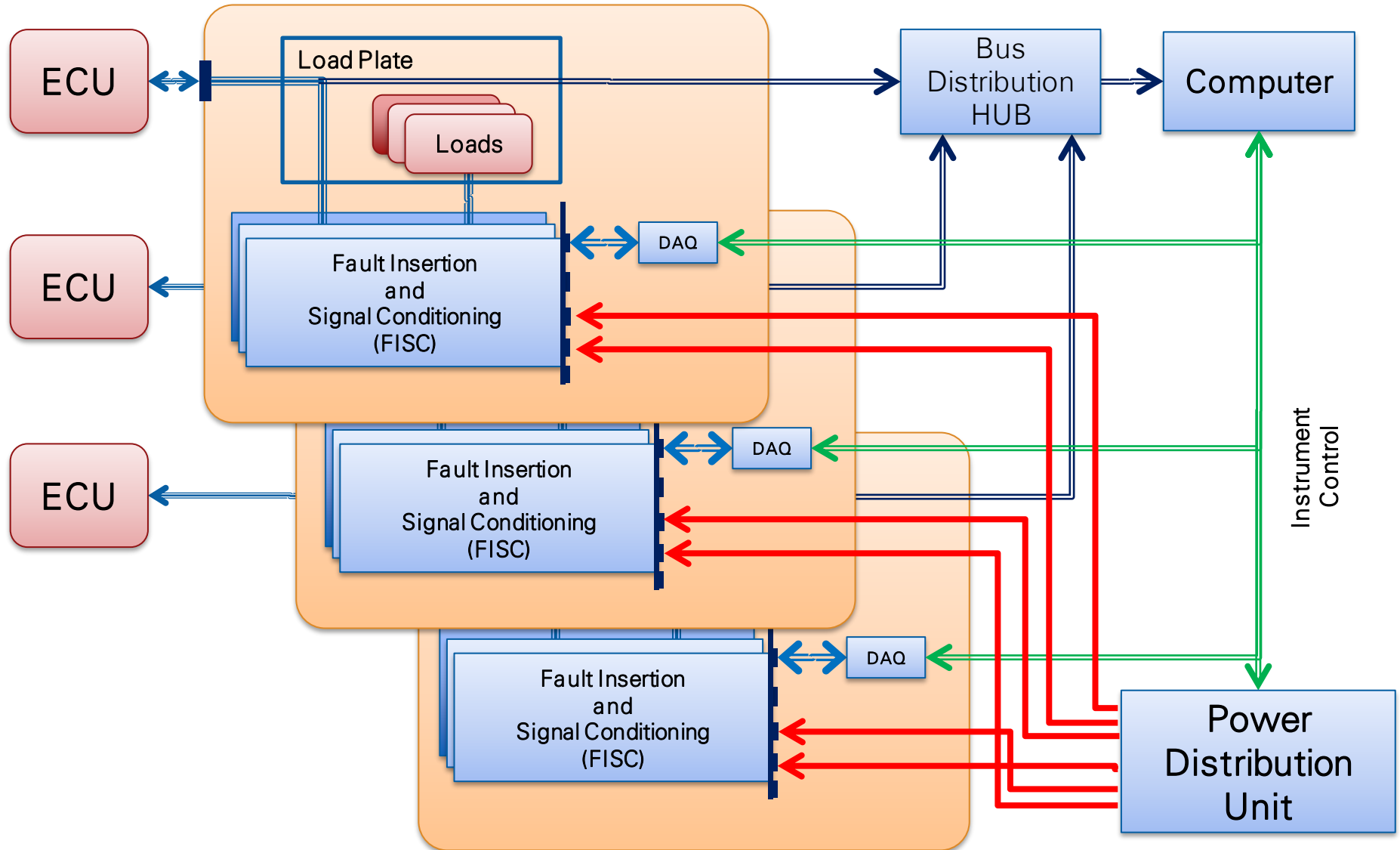
xMove System Architecture (Hardware)



Distributed I/O Using EtherCAT and LabVIEW



xMove System Architecture (Hardware)



Challenge #2

Mechanics and Wiring

User requirements (identified and addressed):

- Flexibility
- Capable of handling high currents
- Good work environment

Prevas large rack details



Silent, high capacity fan



Interface boxes mounted on sliding shelves for easy access



Power distribution



CAN and 1587 distribution units

Prevas Large-Rack Solution



Rear View



Challenge #3, Software

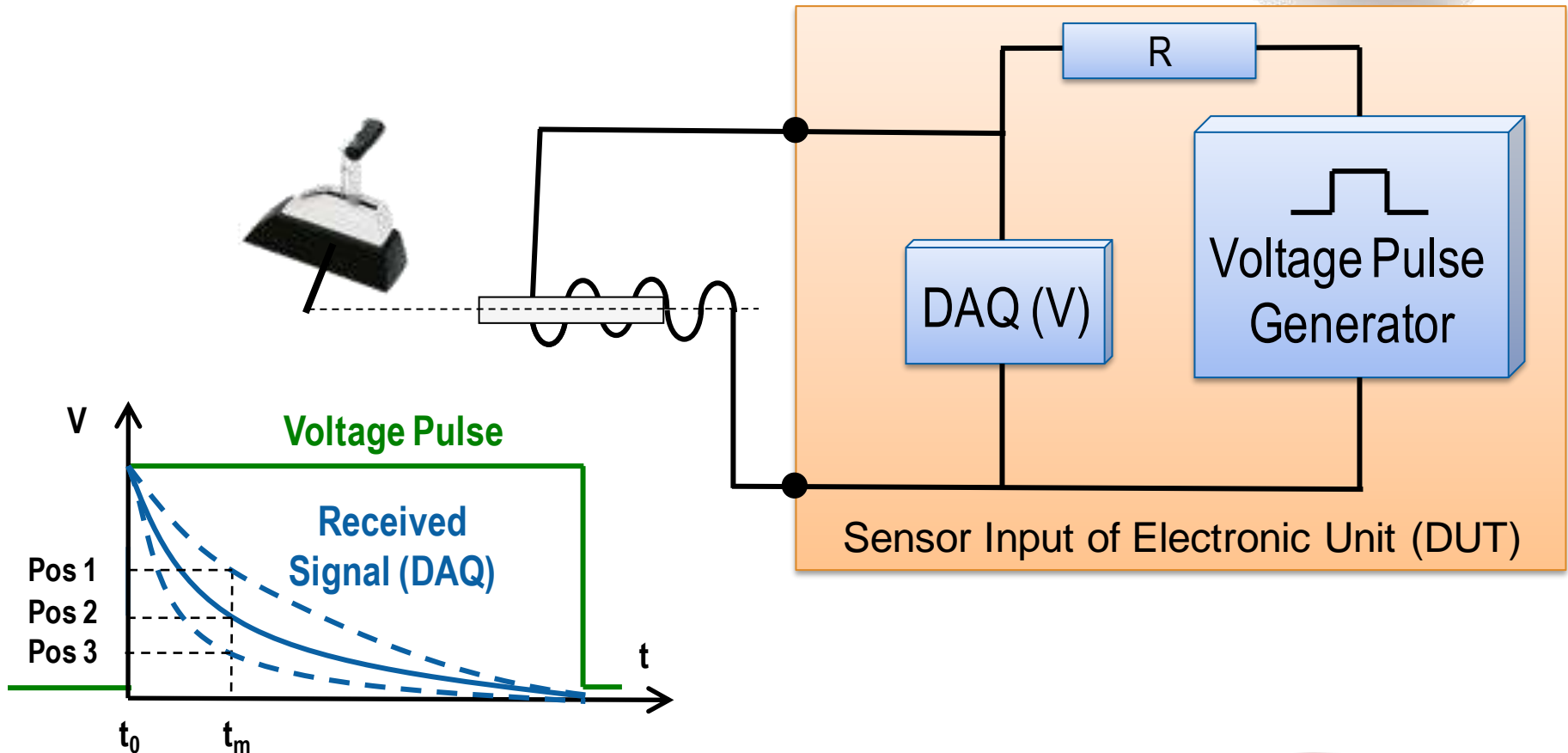
User requirements:

- Conduct both manual and automatic testing
- Flexibility
- Fast to get up and running
=> Addressed by VeriStand
- Performance
=> Secured by NI architecture and preprocessing using FPGA

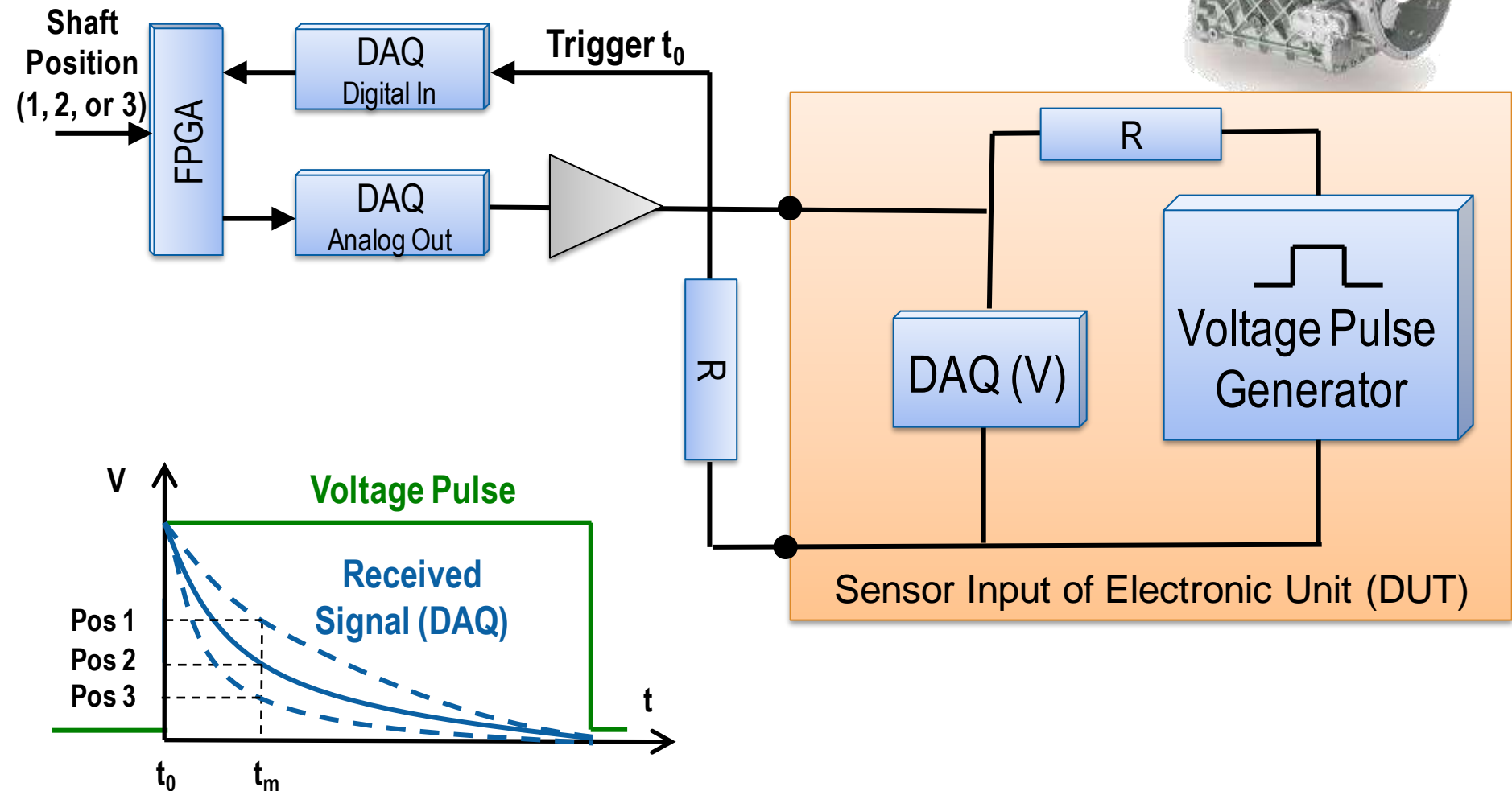
Tips:

- Build the models less accurate as functional testing do not require high resolution models
- Exclude sensors models from the core model and if possible use the FPGA.

Sensor example, Position Sensor



Position Sensor Emulator



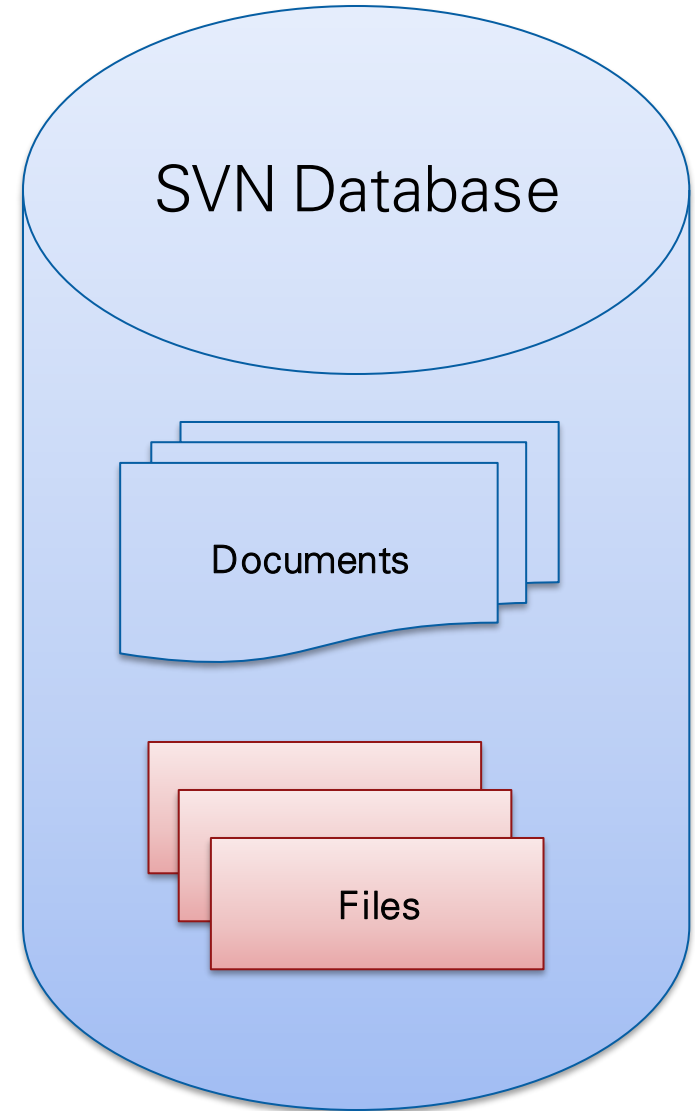
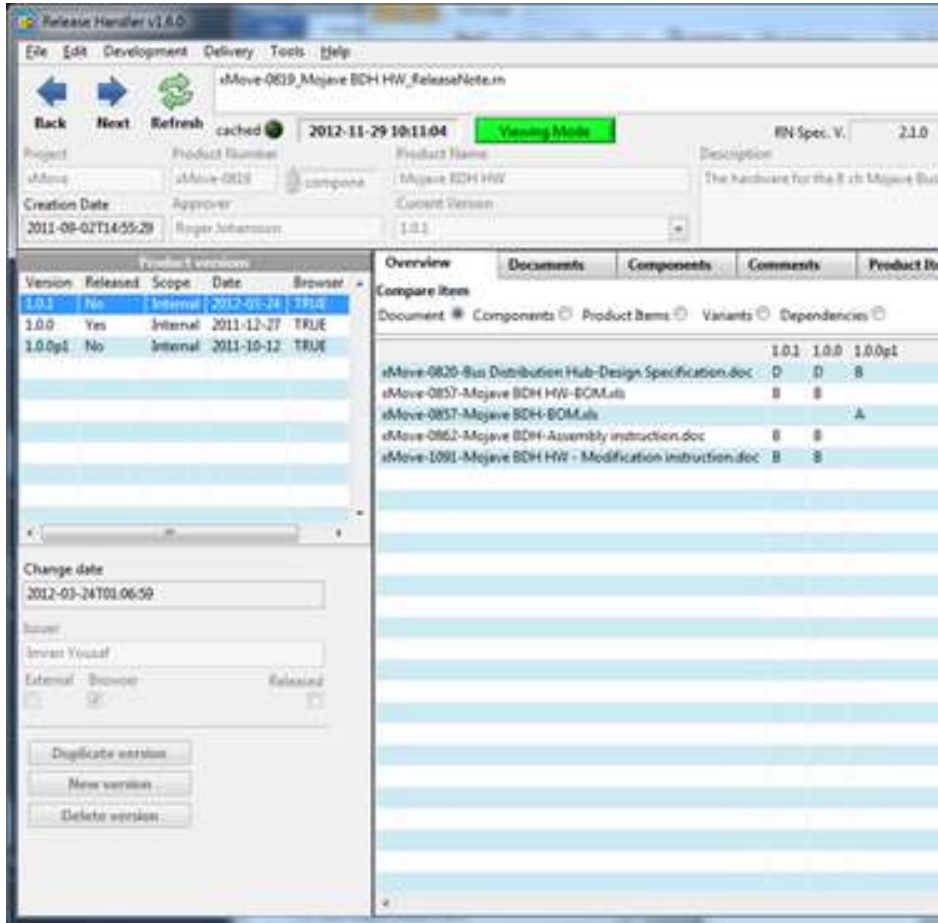
Challenge #4

Configuration management

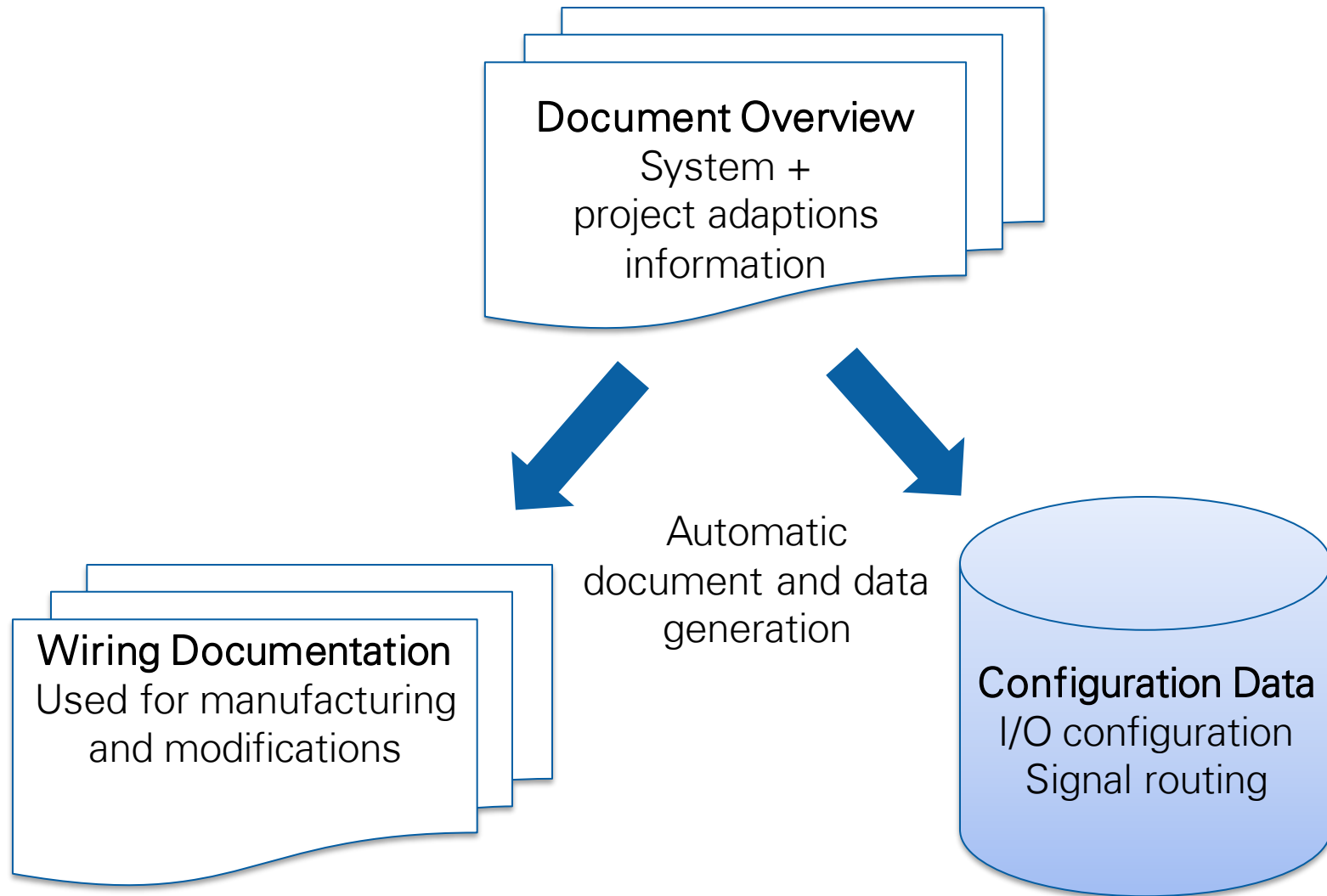
User requirements (identified and addressed):

- Full traceability during design (or purchase), operation and maintenance
- Flexibility

Configuration Management



Automatic Documentation and Configuration



Summary; lesson learned

- Build modular
- Distribute the I/O
- Use the NI/FPGA technology
 - Minimizes need of custom hardware
 - Provide flexibility
 - Provide pre-processing, (offloads the RT CPU)
- Create good documentation structure
 - Automatic configuration generation if possible !
- Use Configuration Management with Release Handling of all part of the system

• Questions ?

• Hans Nystrom

• hans.nystrom@prevas.se