Exploring New Synergies in Simulation and EGSE

SESP, 25-27 September 2012

Michiel Haye, Dutch Space B.V.
Chris Plummer, OHB System AG

What are we going to do and why

INTRODUCTION

Introduction #1

- Collaboration Dutch Space and OHB
- Aim
  - Explore solutions for simulation in AIV
  - Increase awareness of simulation and EGSE concepts
- Route
  - Develop a demonstrator Simulator-EGSE system
  - Use this system to explore further concepts
    - Model Integration Infrastructure.
    - Precision Time Protocol for EGSE time synchronisation.

Introduction #2

Contents

- The Demonstrator Simulator-EGSE system
- Model Integration Infrastructure
- Precision Time Protocol
- Conclusion

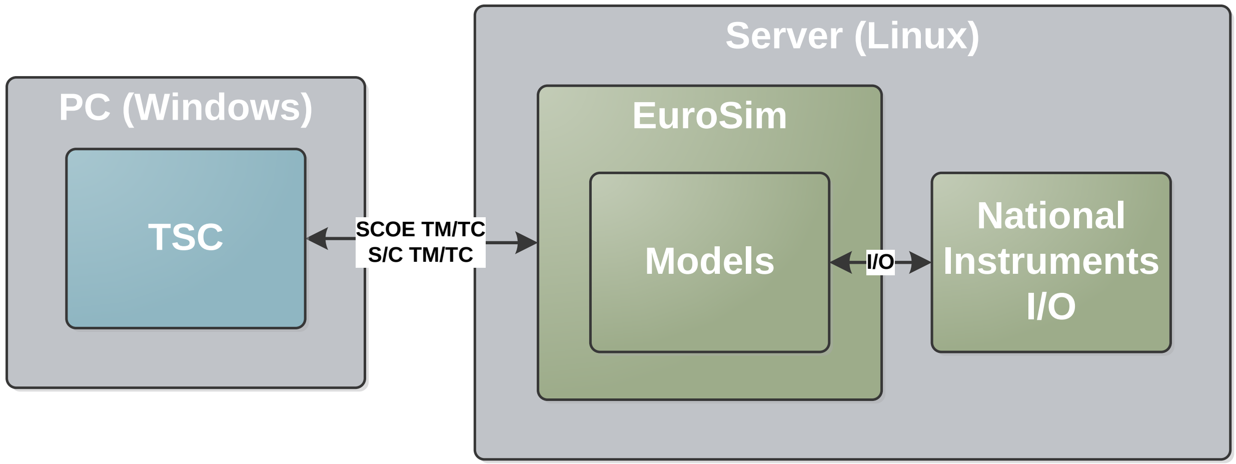
Requirements, architecture, implementation

DEMONSTRATOR

Requirements

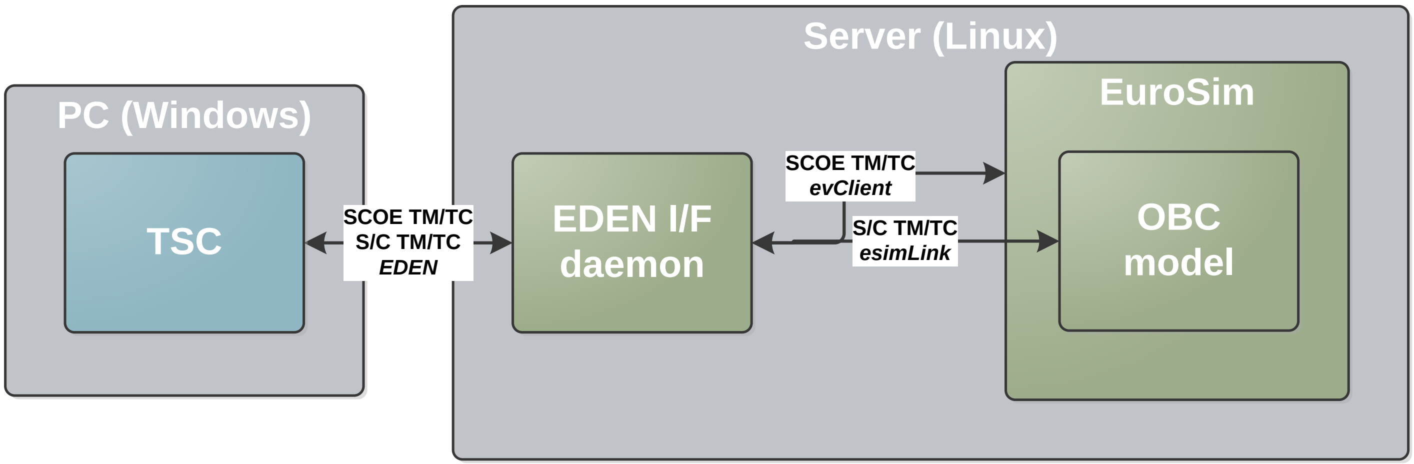
- General
  - Representative of real AIV usage
  - Avoid over-complexity
- Features
  - EDEN interface
    - To integrate simulator with test environment in use at OHB (TSC)
  - Simple OBC model implementing PUS services
    - To use real spacecraft TM/TC database.
  - Simple sensor model
    - To show model integration
  - Something with hardware in the loop
    - To show specific issues related to “hard real-time”

Architecture



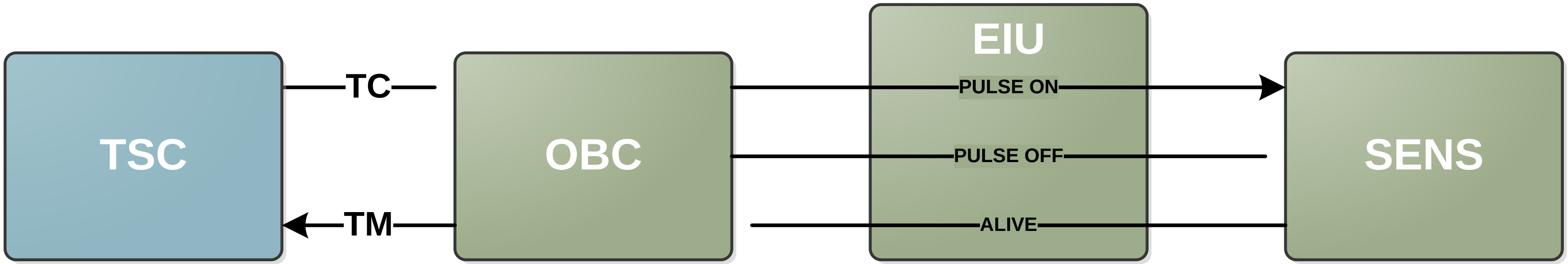
- Terma TSC as CCS
- EuroSim as Simulation environment
- National Instruments Digital I/O card

CCS Interface



- EDEN protocol
  - Single link for SCOE and S/C TM/TC
- Interface application EDEN-EuroSim

Models



OBC model

- Send on/off pulses (service 8, service 1)
- Periodic HK(service 3)
- Event generation (service 5)

EIU model

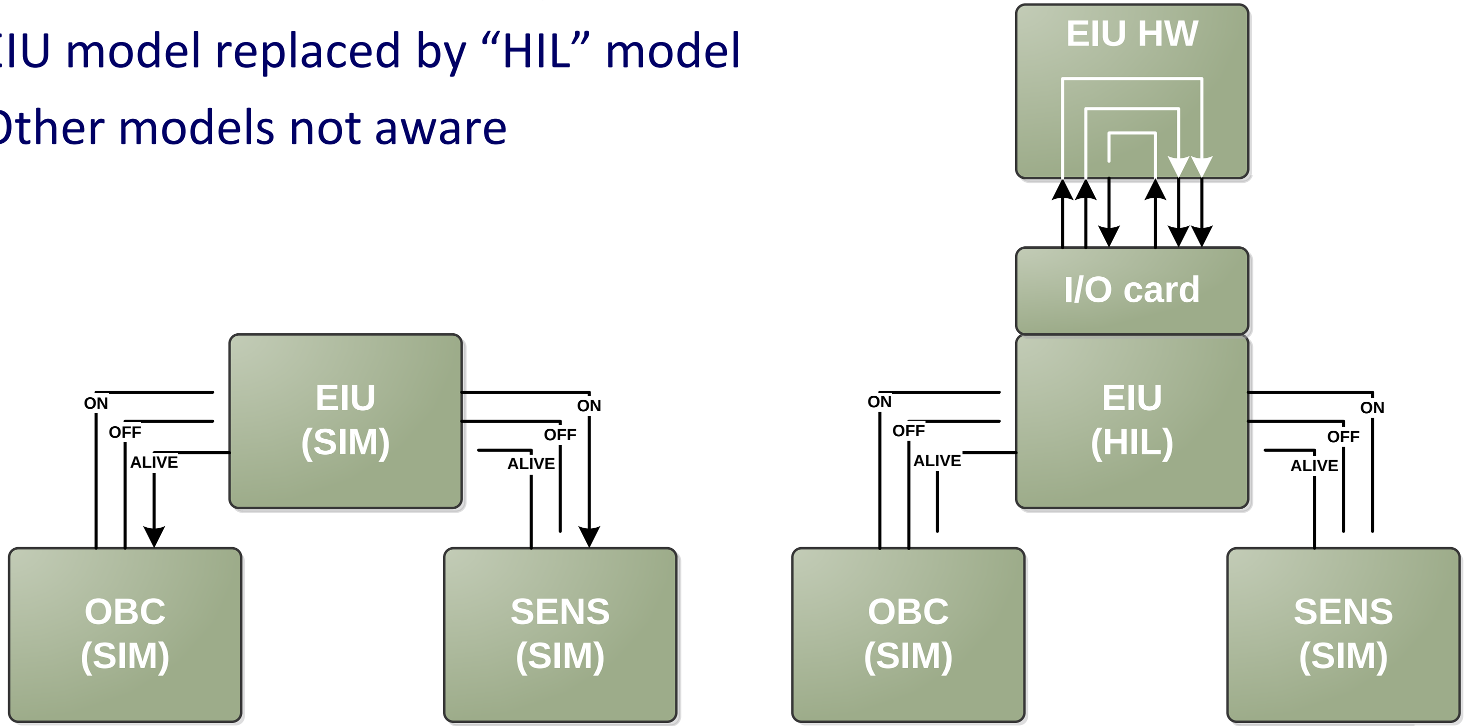
- Wire (for HIL demonstration)

Sensor model

- Generate alive based on on/off status

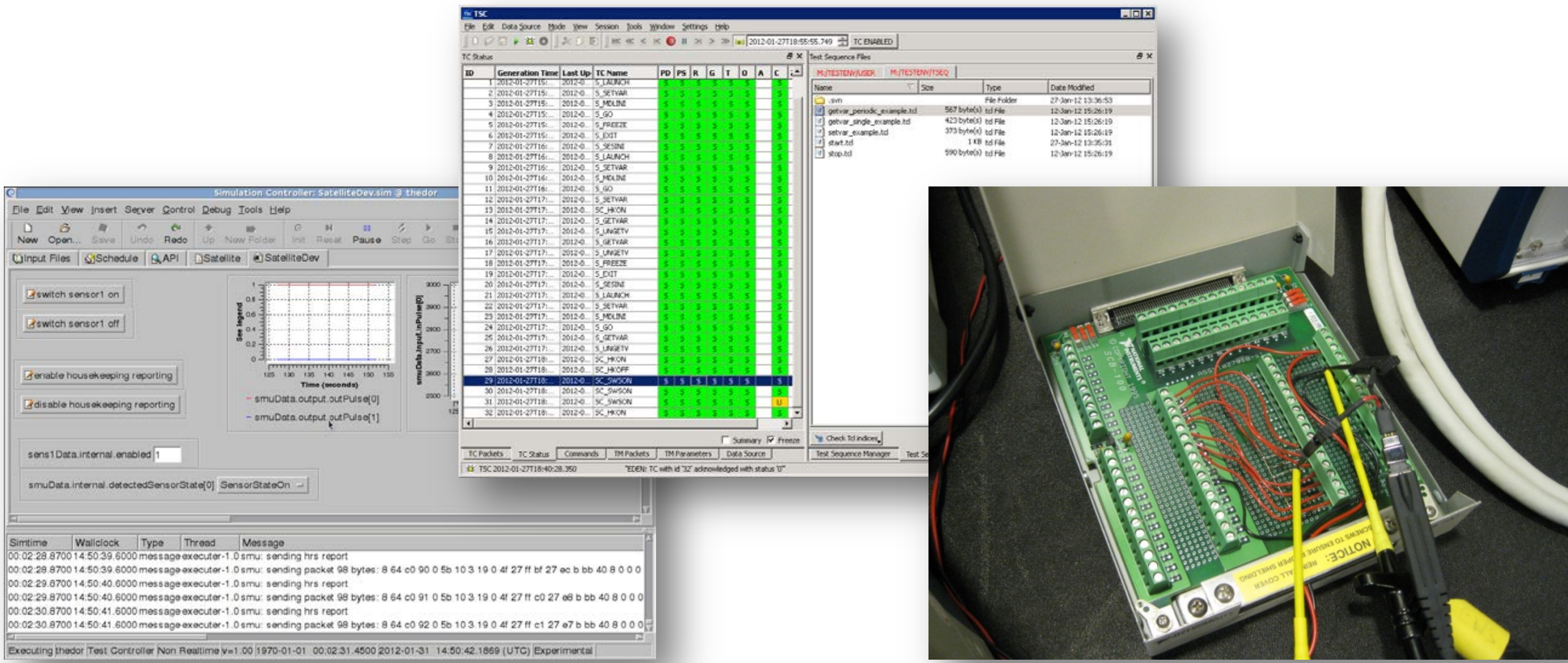
Hardware in the loop

- Real EIU (wires) in the loop
- EIU model replaced by “HIL” model
- Other models not aware



Demonstrator

- See EuroSim exhibition boot



New infra for going from model to simulator

MODEL INTEGRATION

Requirements

"Legacy features" to be supported

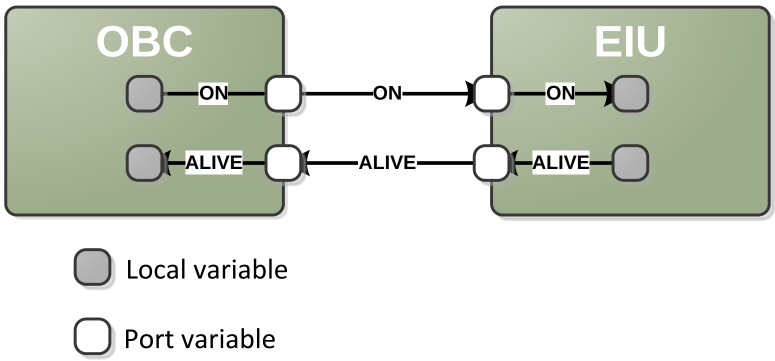
- Data flow oriented model integration with error injection
- Dynamic configuration of models and data flows
- Support for scheduling synchronized to multiple timelines

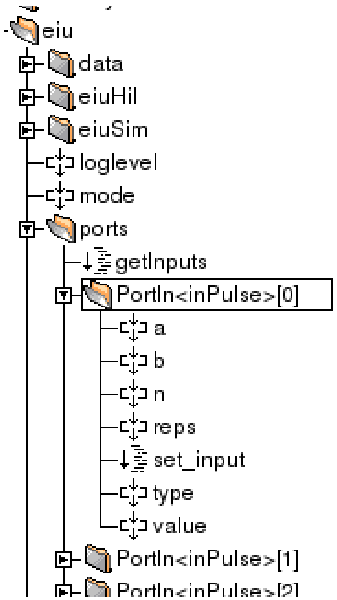
New

- Object oriented approach for above features
- Support libraries to ease integration effort

Data flow oriented model integration

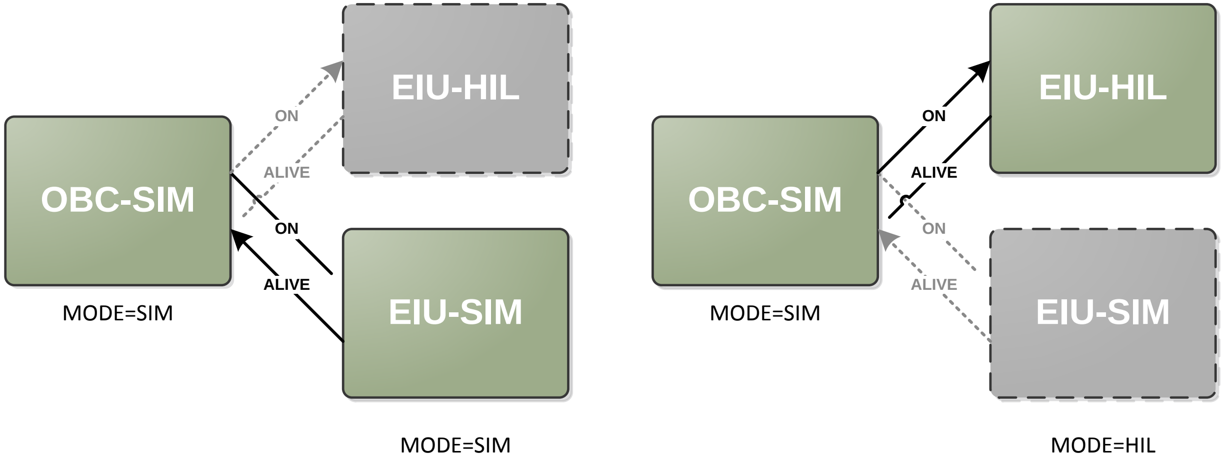
- Models publish their external interfaces
  - Results in dedicated interface variables: ports
  - Generic error injection from local to port and vice versa.
  - Alternative for EuroSim datapool and model description files
- Simulator publishes data exchanges between models
  - Data exchanges between ports
  - Alternative for EuroSim parameter exchange file





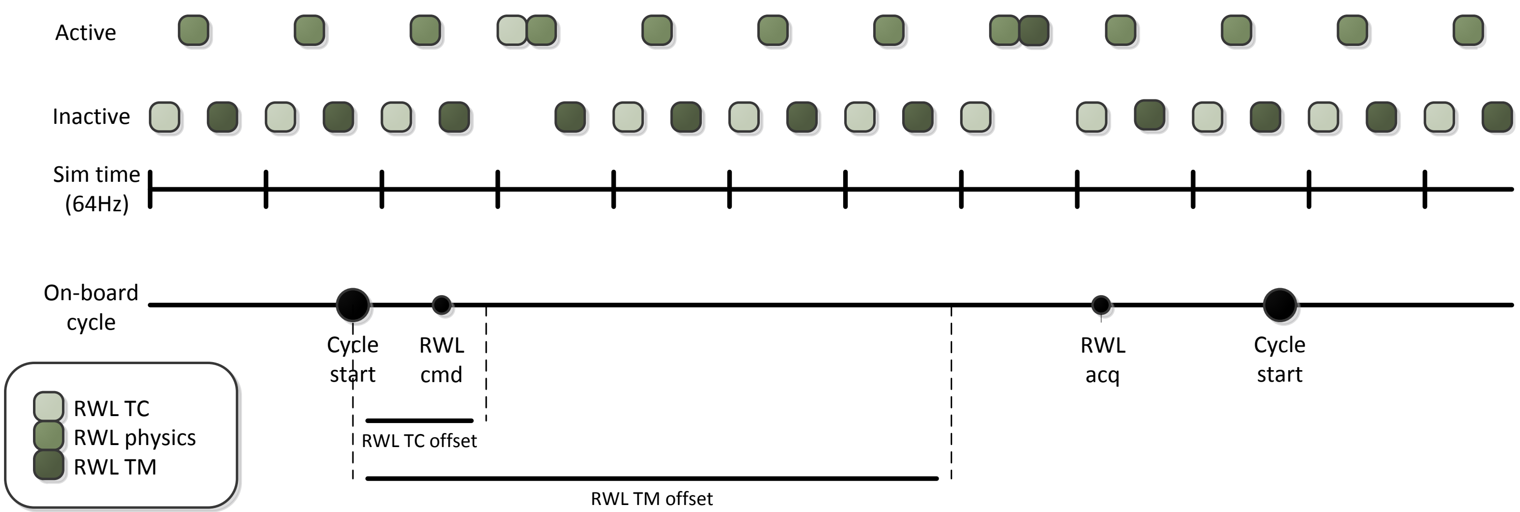
Dynamic configuration of models and data flows

- Models register entrypoints to “model mode” value
  - Model mode = user configurable variable
  - Entrypoints active/inactive based on mode setting
- Simulator registers data exchanges to pair of model mode values
  - Exchange active/inactive based on mode sender AND receiver



Support for scheduling synchronized to multiple timelines

- Multiple timeline concept
  - All entrypoints “statically” scheduled at highest simulator frequency
  - All entrypoints “dynamically” scheduled at frequency/offset of a timeline
  - At highest simulator frequency enabling/disabling depending on timeline propagation
- Advantages
  - Deterministic execution order possible
  - Avoid need for data locks
  - Can also be applied for “software only” SVF → reuse of schedule!



Summary of the integration approach

- Models publish data and entrypoints
- Models publish data flow interfaces
- Models register entrypoint with model modes
- Simulator instantiates models
- Simulator publishes data exchanges
- Simulator registers data exchanges with model modes
- Simulator defines timelines
- Simulator schedules models and exchanges at timelines

A word on SMP2

- SMP2 not fully suitable for all outlined integration options
- But … presented OO approach maps well on SMP2 approach
- Thus …
  - Integration framework needs to be expanded to support easy integration of SMP2 compliant models
  - Use native tooling for model integration

Can we keep the EGSE time synchronised?

PRECISION TIME PROTOCOL

Approaches

Conventional

- Single master time server
- Software time synchronisation via NTP
- Hardware time synchronisation via IRIG-B

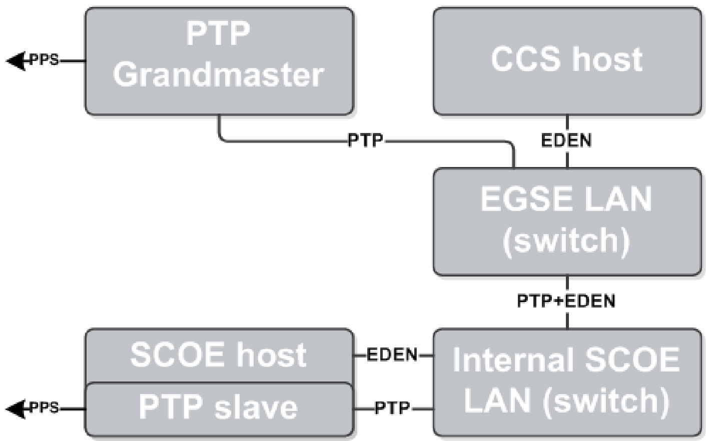
Alternative

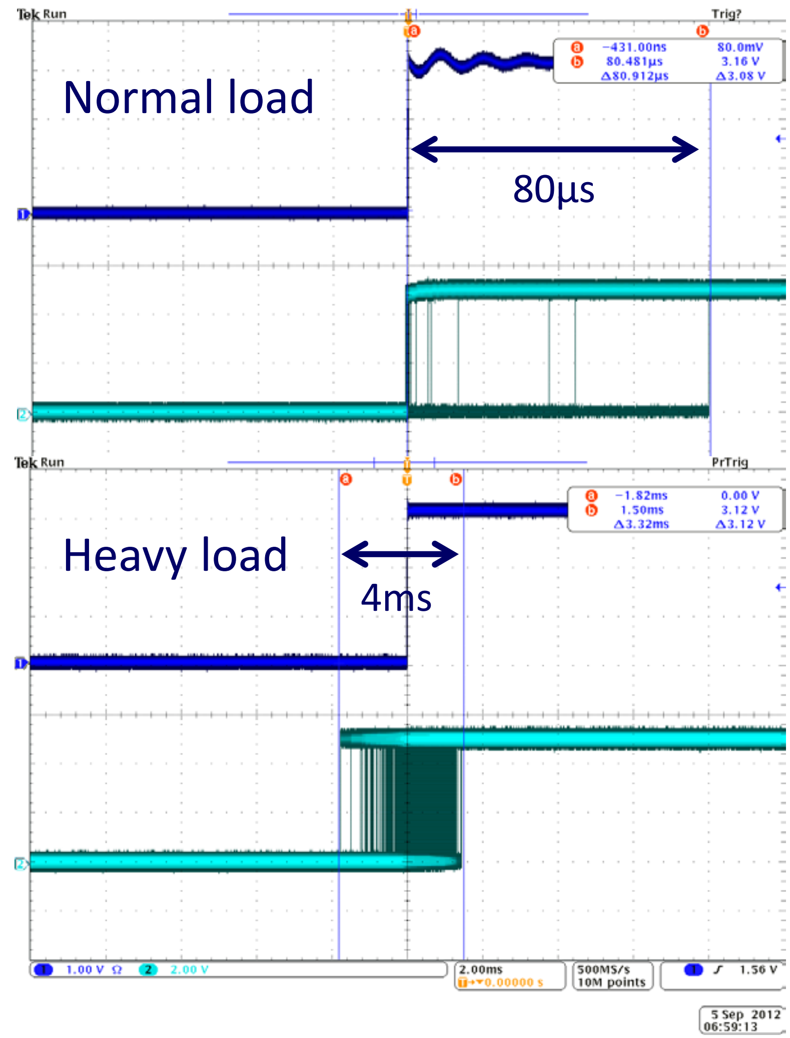
- PTP grandmaster time server
- Hardware and software time synchronisation via network

Advantages (?)

- Simplified (cheaper ?) infrastructure
- Better accuracy and quicker convergence for software time synchronisation
- The future (?)

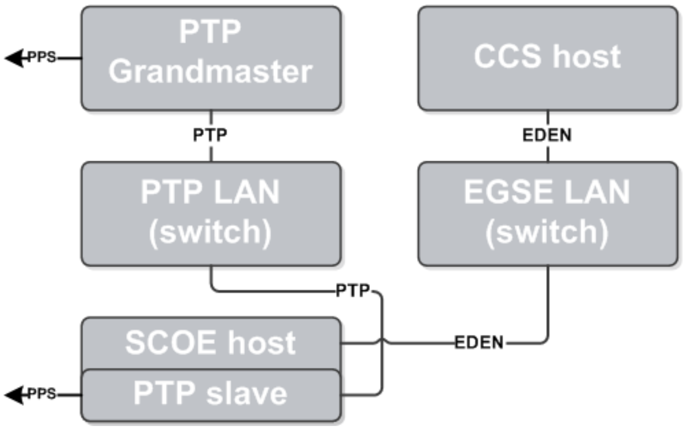
Measurements single LAN

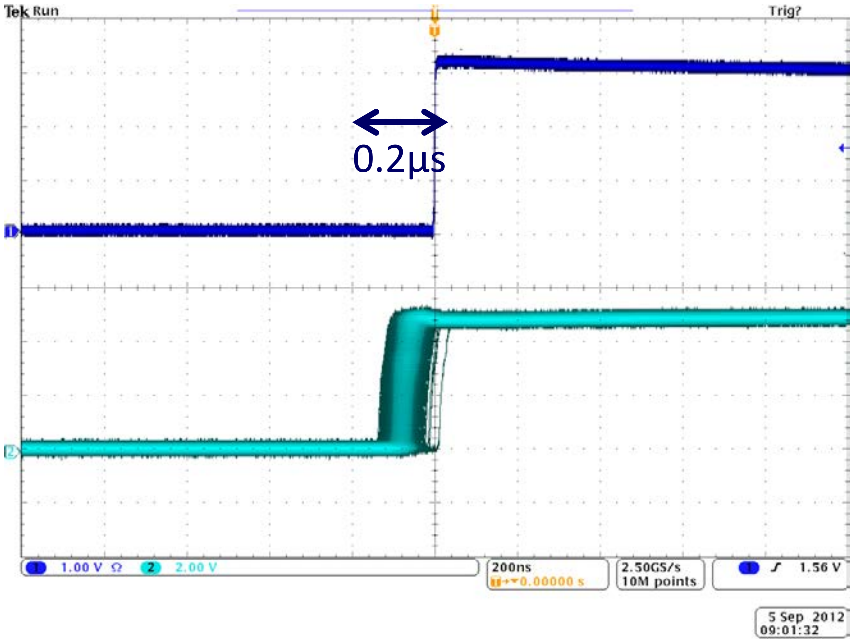




- Results depending on network load
- Typical µs accuracy
- Peaks up to ±2ms (fully saturated network)
- (note one 100Mb switch)

Measurements separate PTP LAN





- Results independent EGSE LAN load
- Sustained sub microsecond accuracy
- Much more to explore:
  - HW vs SW timestamping
  - Standard OS support

What does this lead to and who did the real work

CONCLUSIONS AND THANKS

Conclusions

- Demonstrator successful
  - Creates awareness
  - Good test bed for new simulation concepts in AIV
- New model integration infrastructure for AIV simulators simplifies simulator development.
- PTP seems good candidate for EGSE synchronisation (preliminary)

Big thanks to

- Matthijs van der Kooij, André Glas, Leon Bremer, Piet Vriend