OBUs' Development and Maintenance of a Train Control System for Low Density Traffic Lines

Gerhard Hanis, Burkhard Stadlmann

University of Applied Sciences Upper Austria, School of Engineering and Environmental Sciences

Ada-Europe 2014, Paris

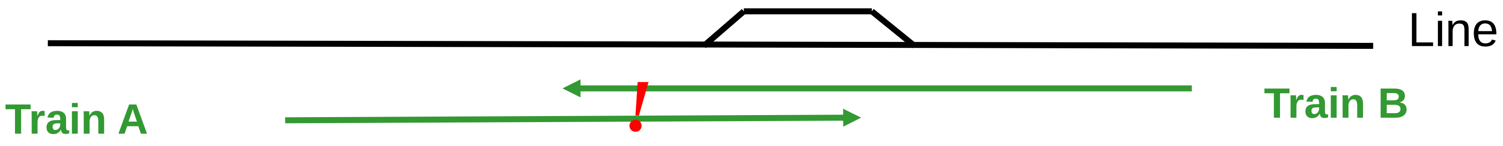
Train Control System for Low Density Traffic Lines

Why?

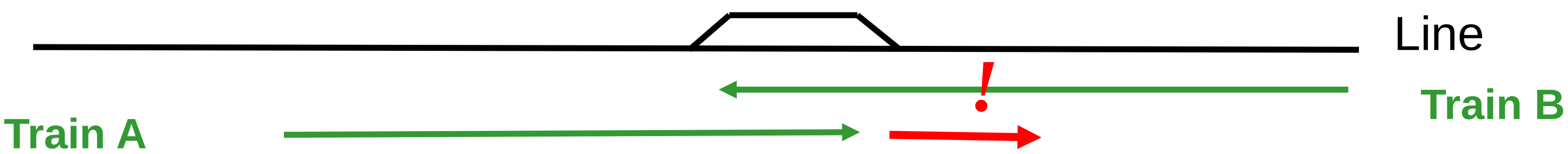
- Voice communication and predefined operational procedures
- Often no hardware interlocking
- Failure of one single person can cause an accident

Examples of accidents in Austria:

Failure of the traffic controller



Driver did not wait for the crossing train



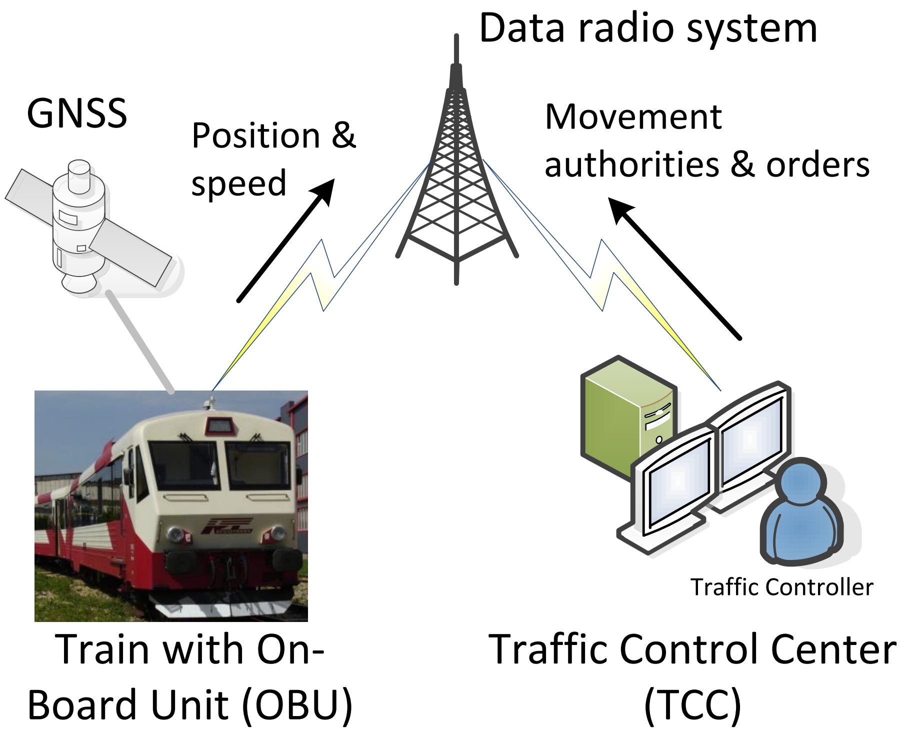
Train Control System for Low Density Traffic Lines

Basic ideas of the TCS

- Computerized version of radio-based operational train control
- Distributed real-time system
- Digital data radio communication
- No need for cost-intensive track-side installations

History

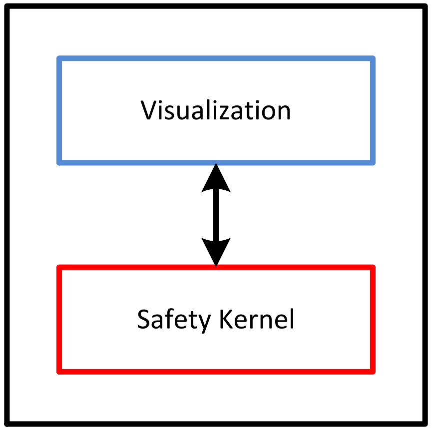
- Development started in 1999
- Ada95 due to recommendations in EN 50128
- First deployment in 2005
- SIL0 approval in 2006
- In operation on 4 lines in Austria



System architecture

Traffic Control Center (TCC)

- Safety Kernel
  - Language: Ada95
  - Management of trains and Communication with the trains (including authentication)
  - Management and issuing of movement authorities and shunting authorities
  - Sends real-time view of the line to the visualization



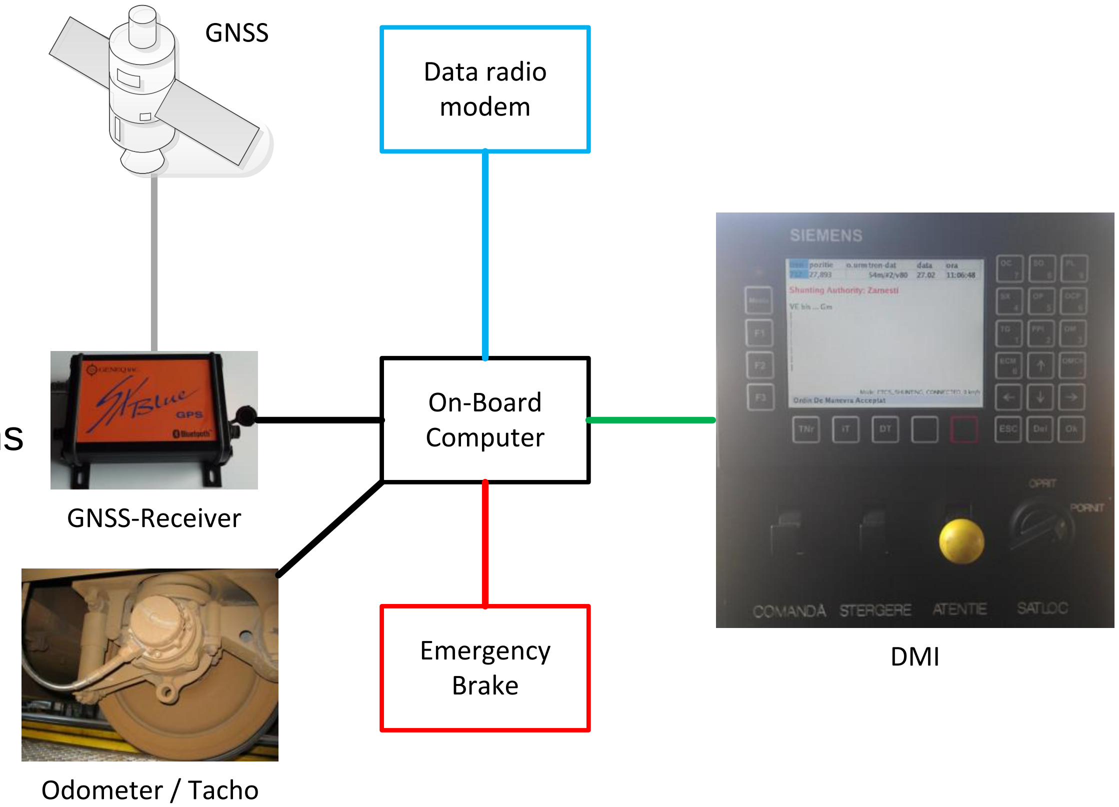
Traffic Control Center

- Visualization
  - Language: Java
  - Display data received from Safety Kernel
  - Provide “dumb” HMI for the dispatcher

System architecture

On-board Unit

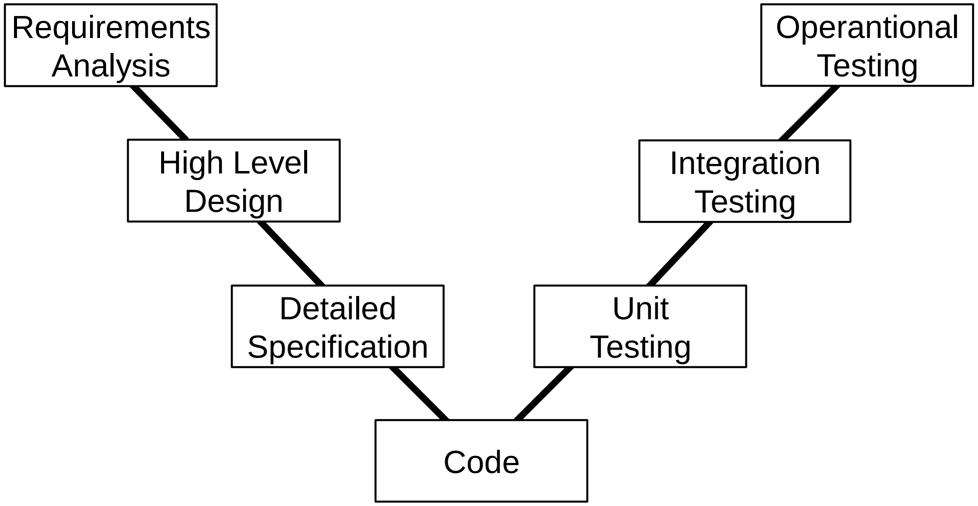
- On-Board Computer (OBC)
  - Language: Ada95
  - Location determination
  - Communication with TCC
  - Supervision of Movement authority
  - Application of emergency brake in dangerous situations
- Driver Machine Interface (DMI)
  - Interaction with driver
- COTS hardware as far as possible



Software design

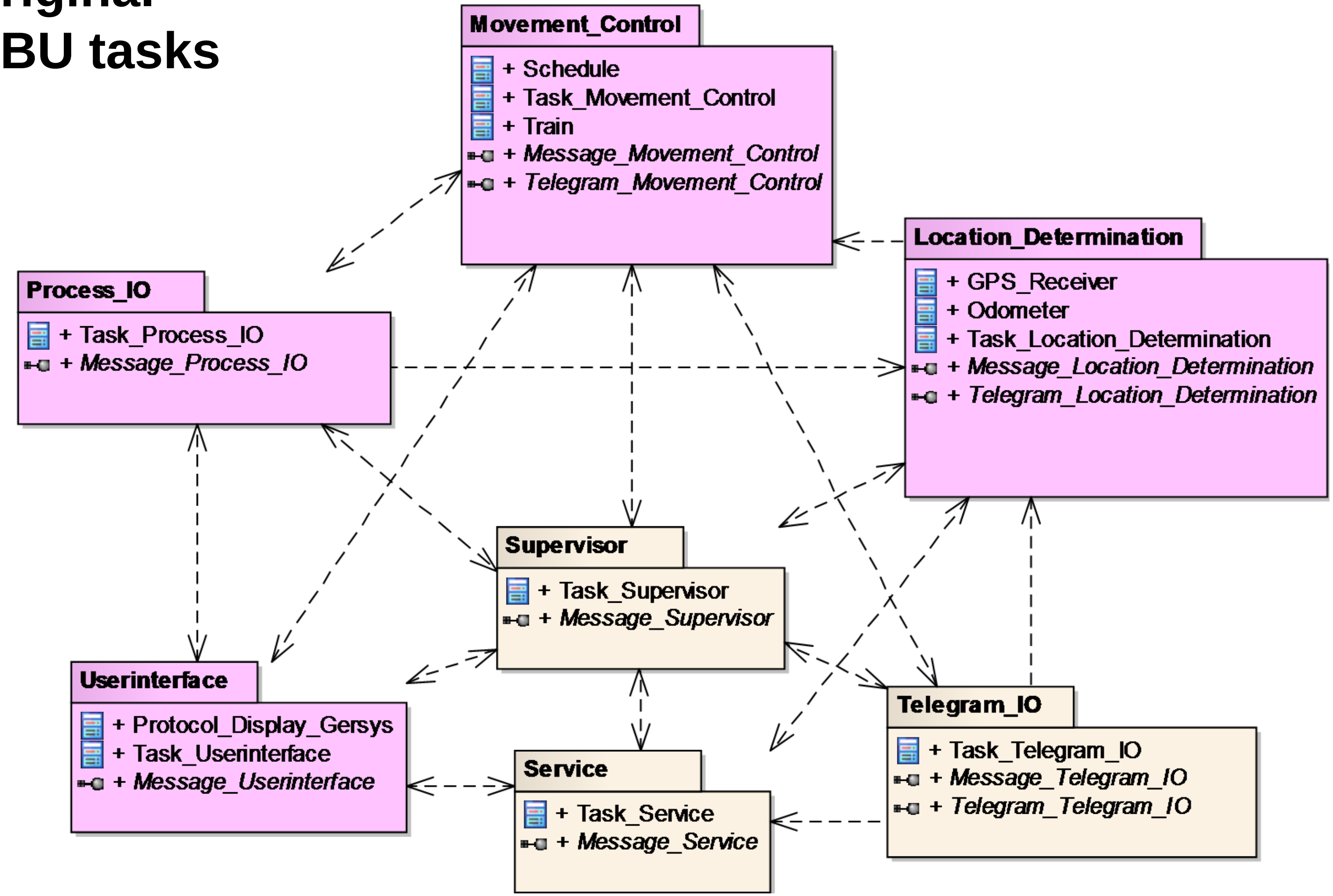
Software development process

- According to a simplified version of railway relevant parts of CENELEC but without formal approval for a certain SIL-Level
- Detailed system specification
- Use Case driven software design
- UML design using
  - Sequence diagrams
  - State and activity diagrams
  - Class diagrams



System architecture

Original OBU tasks



Software architecture

Principles of Task structure

- Task_Supervisor
  - Load system configuration
  - Start (productive) tasks in defined order
  - Monitoring activity of other tasks via heartbeats
  - Triggering of the hardware watchdog
- Tasks offer rendezvous method “Start”
- Communication between tasks via asynchronous messages
- Pre-defined static task priorities and cycle times

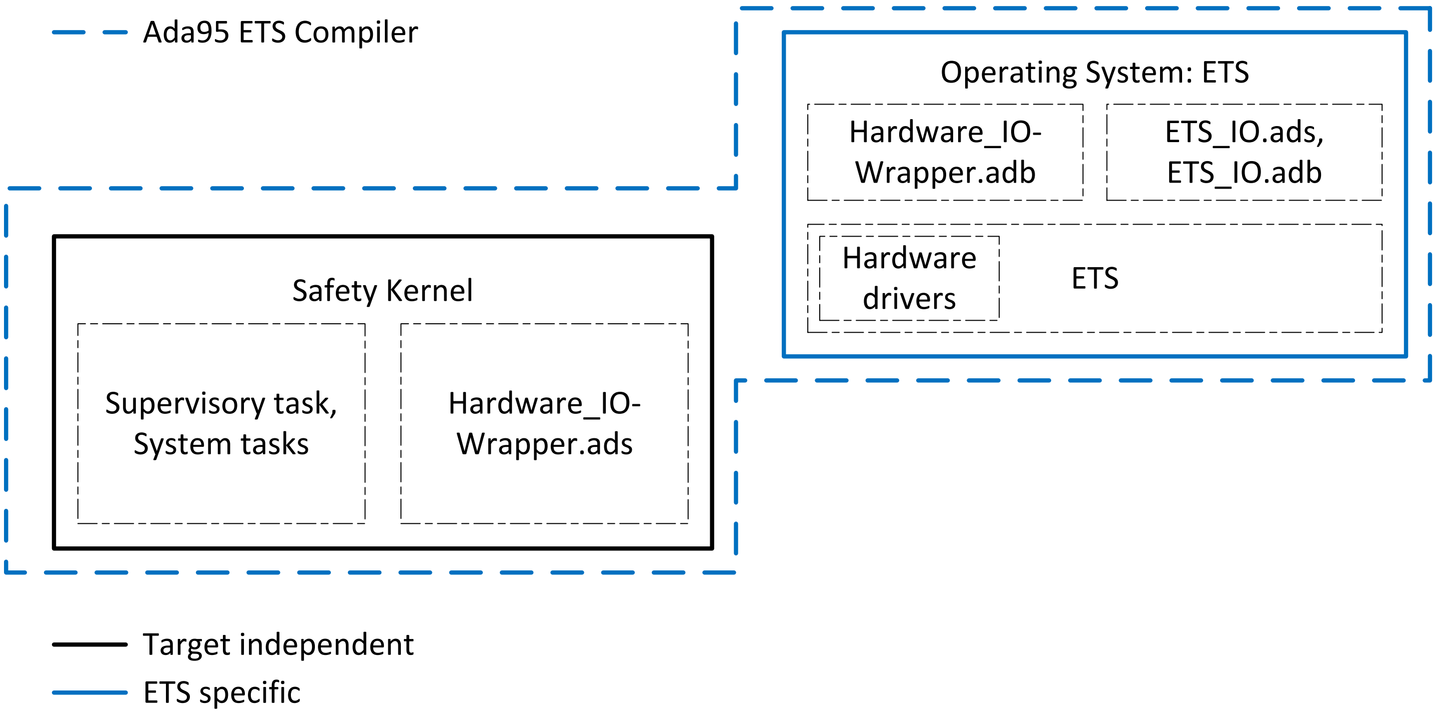
Software architecture

Communication

- Generic protected buffers
- Message consist of
  - ID as enumeration type
  - Data as Unbounded_String
    (EN 50128 recommends avoiding dynamic structures for SIL3/4)
- One message buffer per task
  - Only owner is allowed to perform read
  - Owner has to query buffer every cycle
    and process received messages
  - Case structure where “others” is not allowed

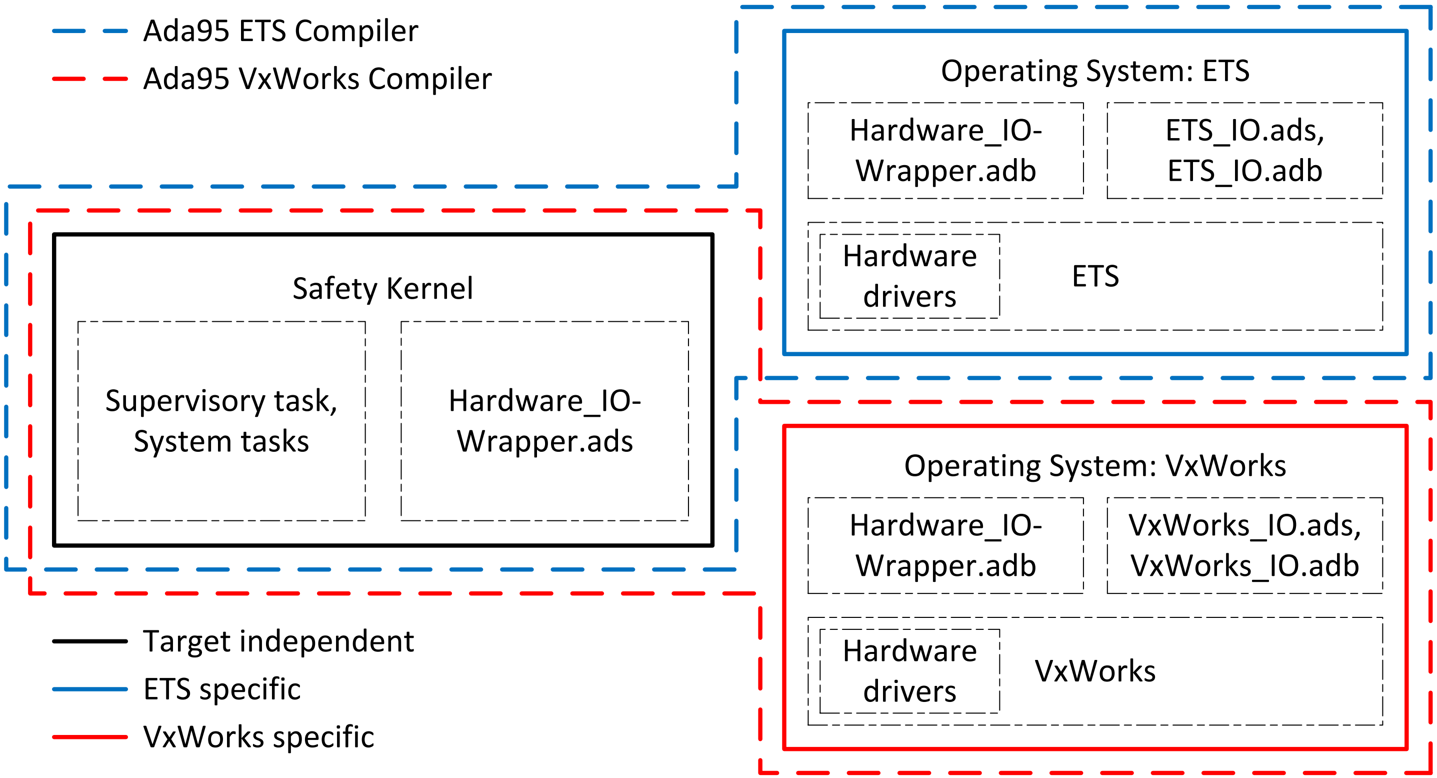
Software architecture

IO-Encapsulation



System evolution

Change of operating system



System evolution – PZB

Project Pinzgaubahn

- Project started in 2010
  - Cooperation with Siemens
- Customer requested SIL2
- Additional features
  - Track selectivity -> integrate balises (RFID reader and tags)
  - Track selectivity for all operational procedures and GUI
  - Implementation of Euroradio CBC-MAC
  - Integration of multipurpose station controllers in selected stations on the line
  - Enhanced functionality for special operational sequences
- Development process according to CENELEC with formal SIL2 approval

System evolution – PZB

Pinzgaubahn – Balises

- Integrate balises into digital line atlas
- Implement Task_Balise_Reader
  - Handles hardware interface and communication protocol
  - “Sub task” of Task_Location_Determination
  - Report detected balises via balise-buffer
- Integrate balises into
  - Location Determination algorithm
  - Communication with TCC

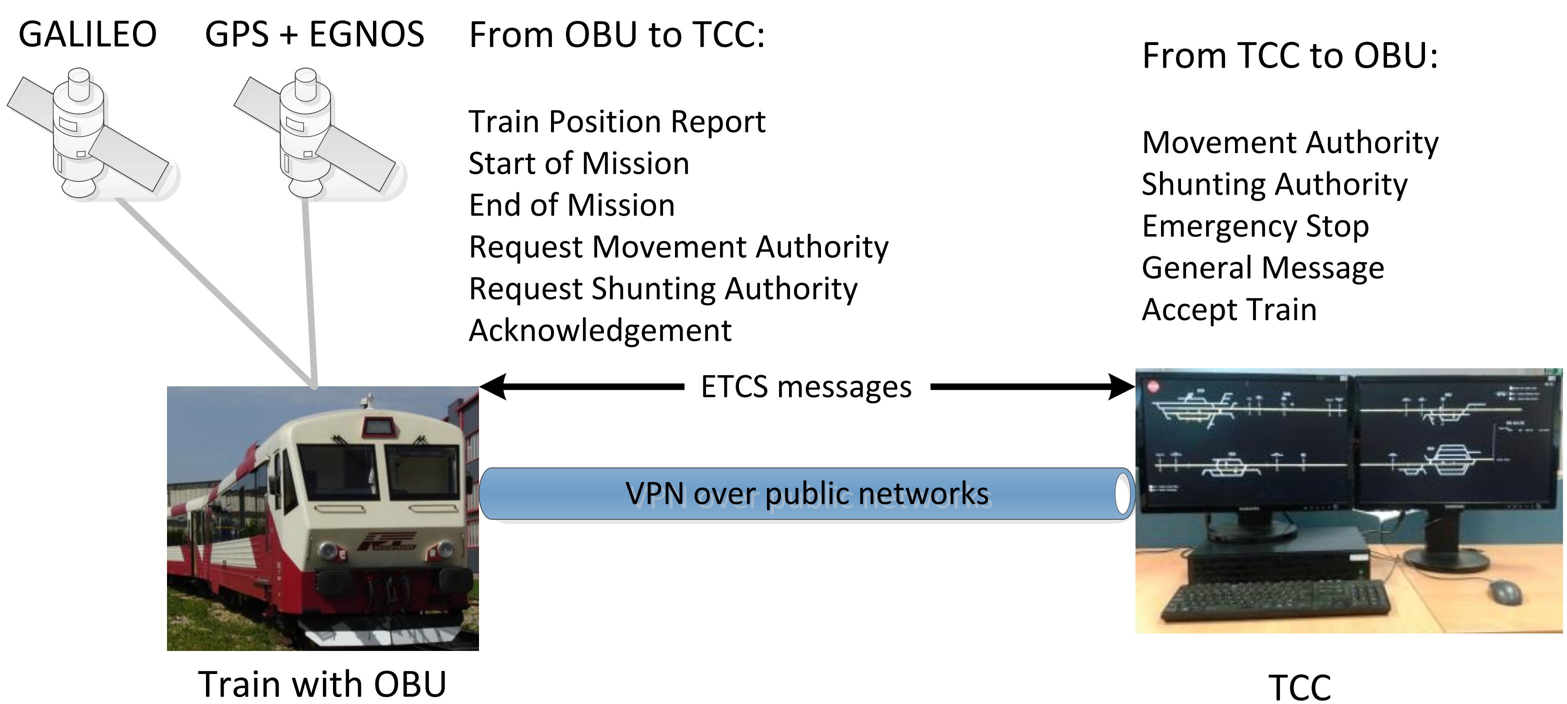
System evolution – Satloc

SATLOC

- EU FP7 funded research project
  - Start 1/2012
  - End 4/2014
  - Test site in Romania
- Promote usage of GNSS in the railway domain
  - 40% of Europe's railway network are low density traffic lines
- Implement a TCS demonstrator at a real line
  - Mainly use GPS + EGNOS and in future Galileo
  - Low investment and operational costs
  - Communication via public networks
  - European Train Control System (ETCS) compatibility considered

System evolution – Satloc

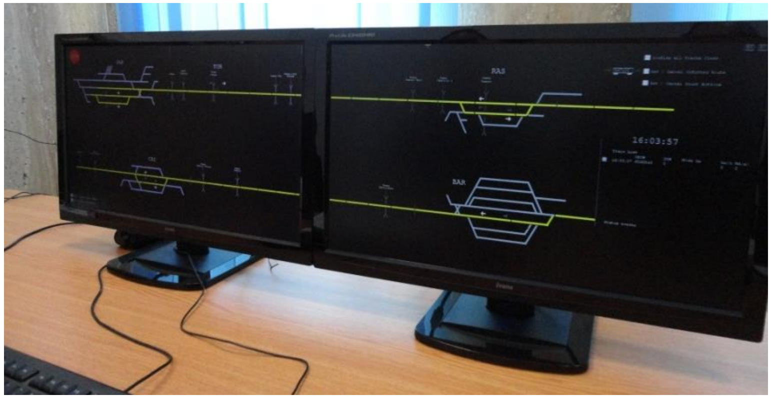
SATLOC – System design



System evolution – Satloc

SATLOC – TCC

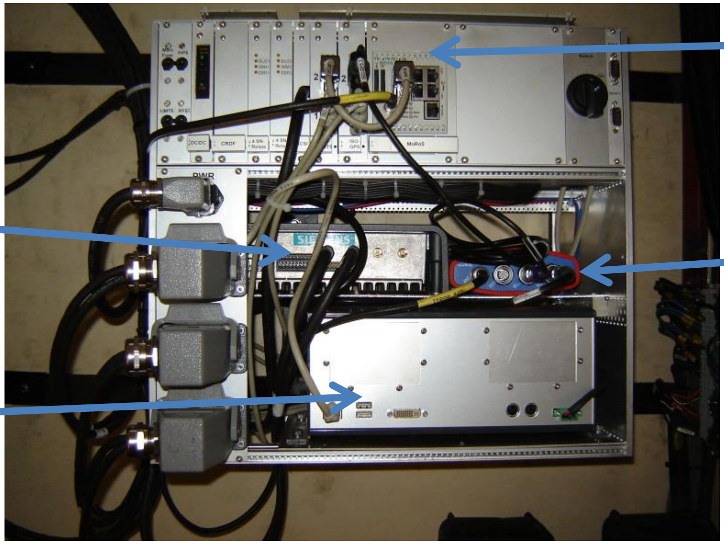
- Usage of available ETCS Radio Block Centre (RBC)
- Siemens UK (formerly Invensys Rail)
- Parts of RBC developed in Ada
- Communication adopted to specific needs of SATLOC



System evolution – Satloc

SATLOC – OBU

- Use existing OBU of project Pinzgaubahn
- Integrate new hardware
  - Balise reader
  - GNSS receiver
  - Data Radio Modem

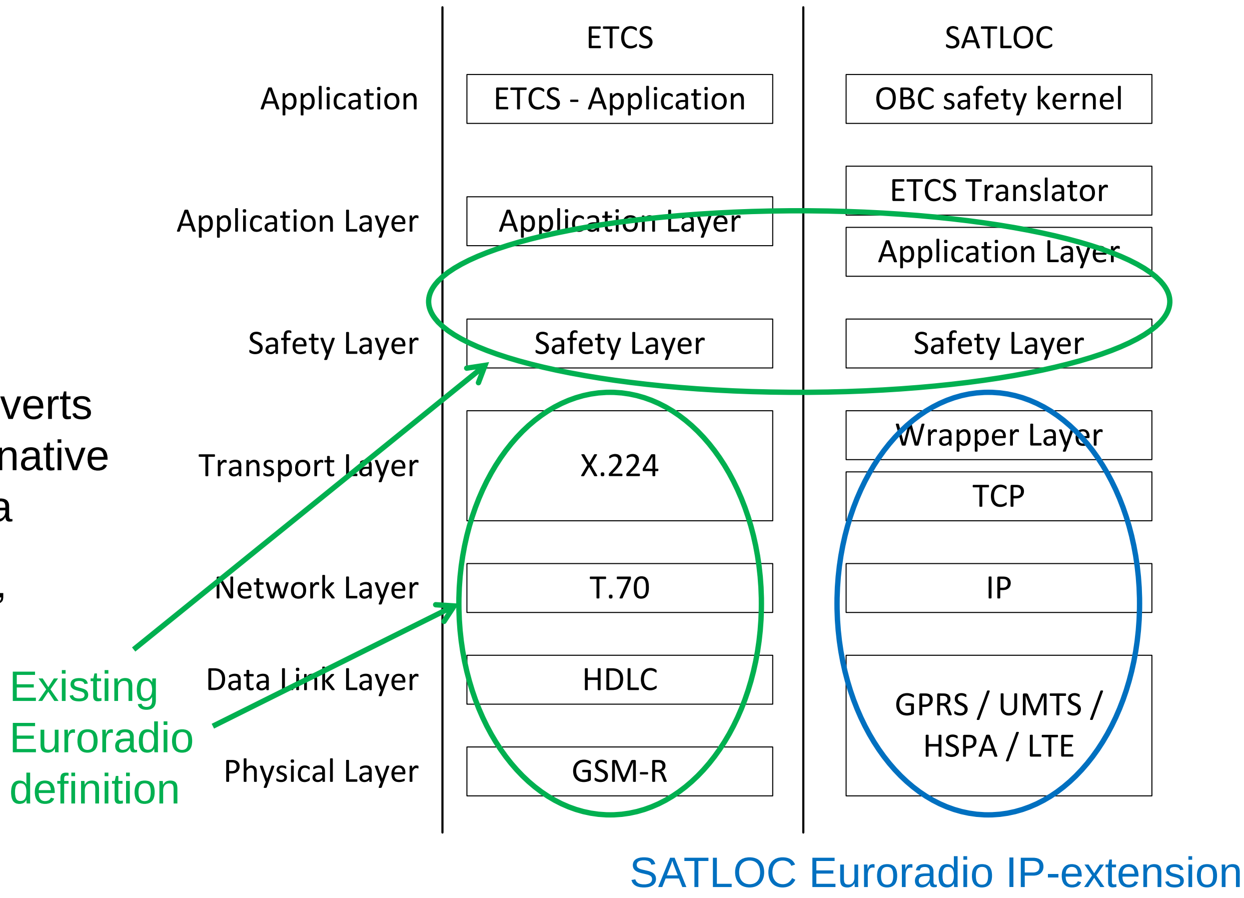


OBU rack mounted in train cabinet

System evolution – Satloc

SATLOC – OBU

- Added 3 Tasks
  - Application Layer
  - Safety Layer
  - Transport Layer
- ETCS Translator converts messages into OBC native format and vice versa
- Implemented a “light” version of the stack



System evolution – Satloc

SATLOC – OBU

changeable

English only

Additional data from the TCC

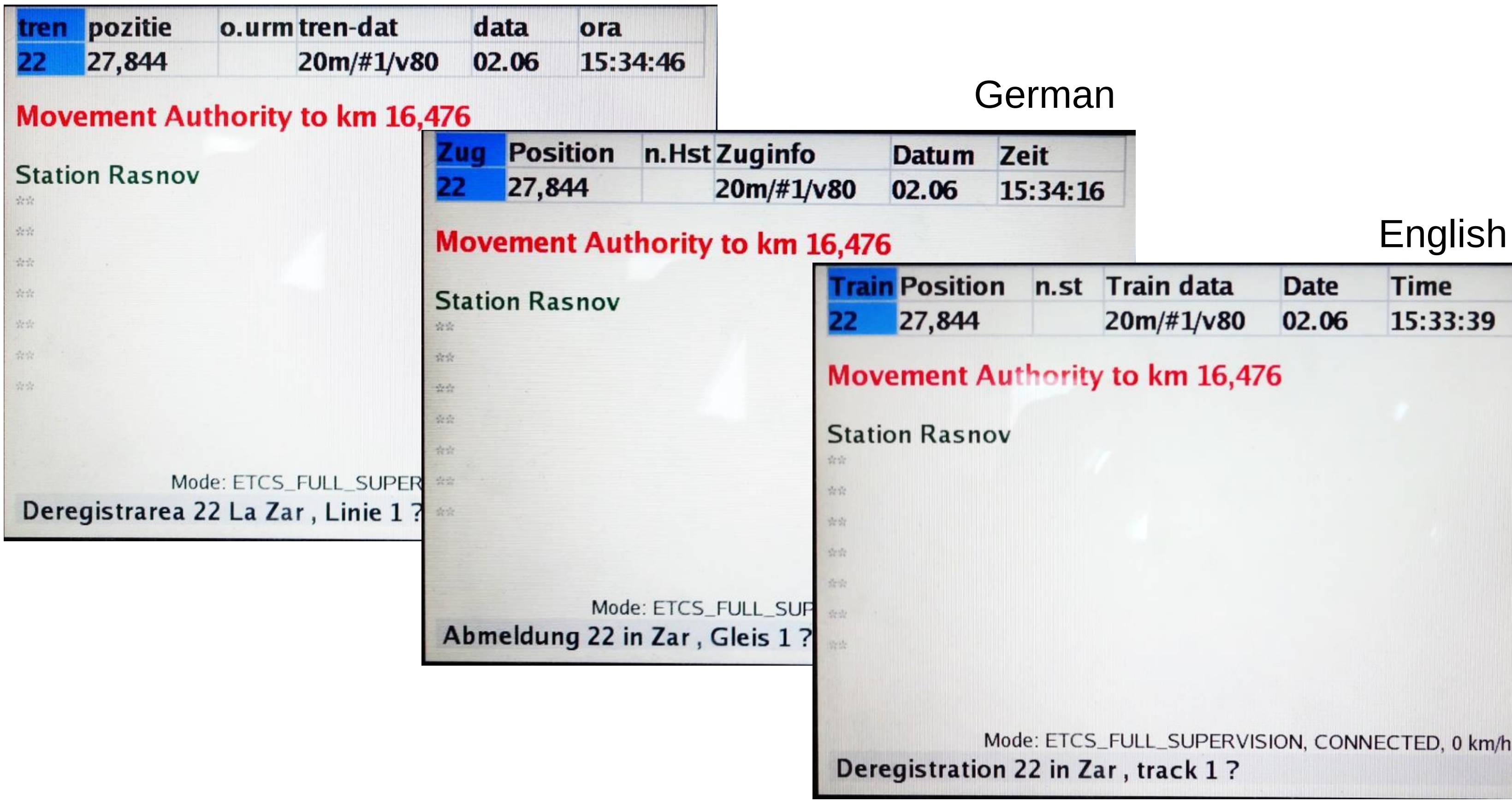
English and changeable

changeable

Romanian

German

English



System evolution – Satloc

SATLOC – Field tests

- Implemented a “light” version of ETCS over TCP/IP
- Mobile carriers provided a fabulous coverage along the test line
- Hardware exchange was of limited work
- Possibility to test off track at local offices
  - VPN over mobile carriers
  - Lab tests between Austria and United Kingdom

- On-line change of displayed language proved to be of high value
  - Field tests took place in Romania
  - Test crew multi-national

Experiences & Conclusion

- System development started in 1999
- SIL0 approval in 2006
- SIL2 approval in 2012
- Low cost Train Control System is in operation

- Base software architecture remained untouched
- IO-encapsulation proved to be of high value
- Changes in: Operating system, hardware and compiler

- New features are constantly added
  - Track selectivity, further movement supervision
  - “light” version of ETCS communication
  - Multiple languages on the DMI

Thank you for your Attention!