



LiCAS meeting

software considerations

Andreas Herty
Deutsches Elektronen-Synchrotron DESY, ZMEA2



LiCAS meeting – software considerations

Software engineering

- A concept for the hardware and software for a train has been made by WERUM and DESY. It is the base for the design of the stretched-wire train
- For the LiCAS train many aspects, e.g. network, hardware, sensor controlling, can be used as well.

→ **It is the aim to create two identical systems as far as possible**

- a) for some sensors, motion stages and drives**
- b) for geodetic calculations**
- c) for user interface (display / input)**

Operating System

- Operating system is **Linux**.
- Database system will be freeware, e.g. MySQL or Postgres



LiCAS meeting – software considerations

Network

- For communication, data transfer e.g. to central database a wireless LAN (ethernet communication) will be used
- For communication between the master and the measuring cars an ethernet (wire-connection) will be used
- For communication between the measuring cars a CAN bus will be used

programming language

- data aquisition, calculations, control of train in **ANSI C**
- control panel and data displaying in **HTML** and **JAVA applets** (WERUM)



LiCAS meeting – software considerations

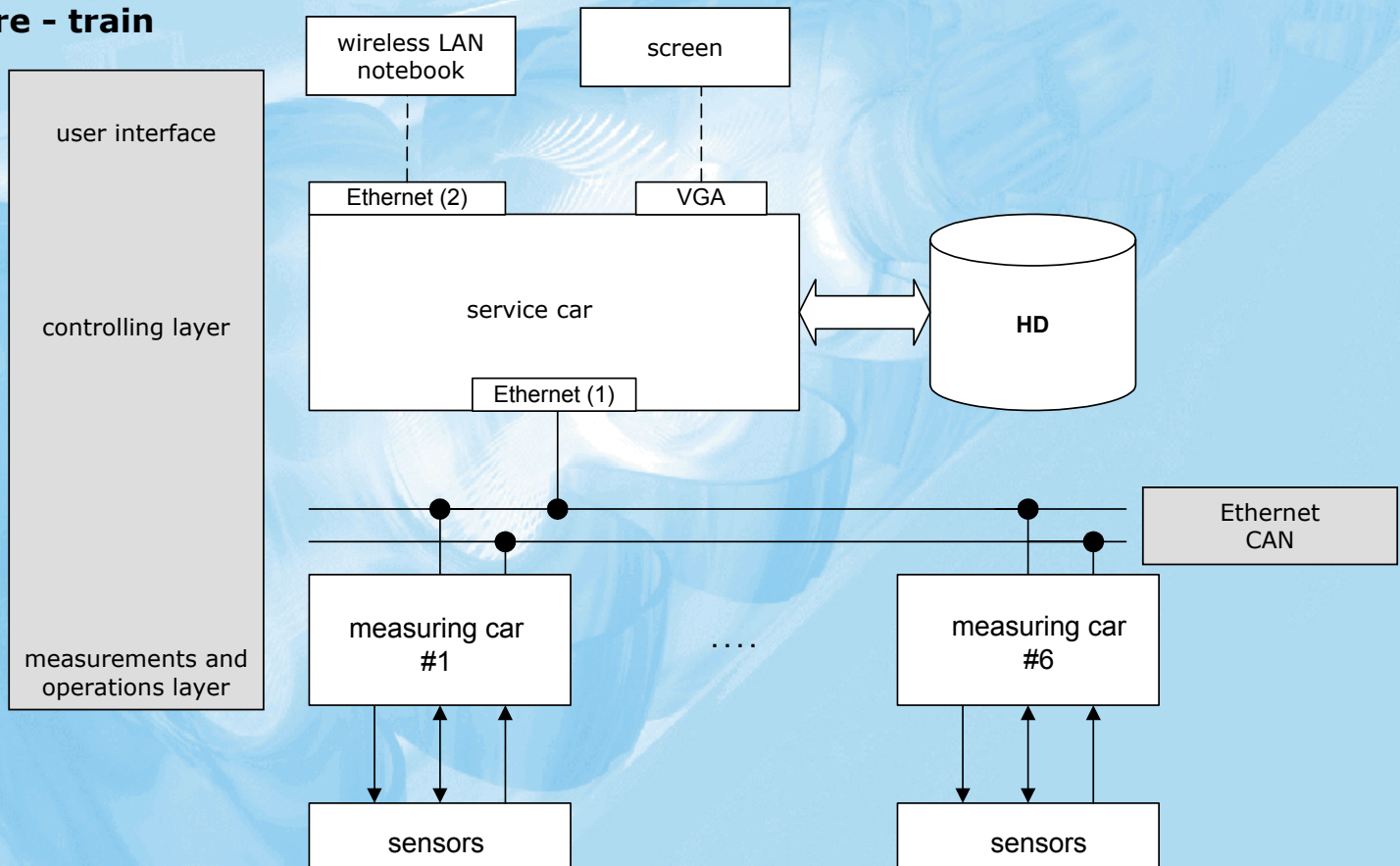
hardware

- normal, rack mounted 19" PC with power supply 8.5VDC to 40VDC
- save data storage with RAID Level 1 HD system
- HD capacities depending on the amount of data
- PCI interfaces
- Power interfaces with 15 VDC, 24VDC and 230VAC
- Power supply with 230VAC from power socket (prototype)
- Components will be linked by a bus system
- 100Mbit/s ethernet for connection the cars with each other
- Every car gets an own computer for controlling
- Data storage and central commands will be given by one separate computer



LiCAS meeting – software considerations

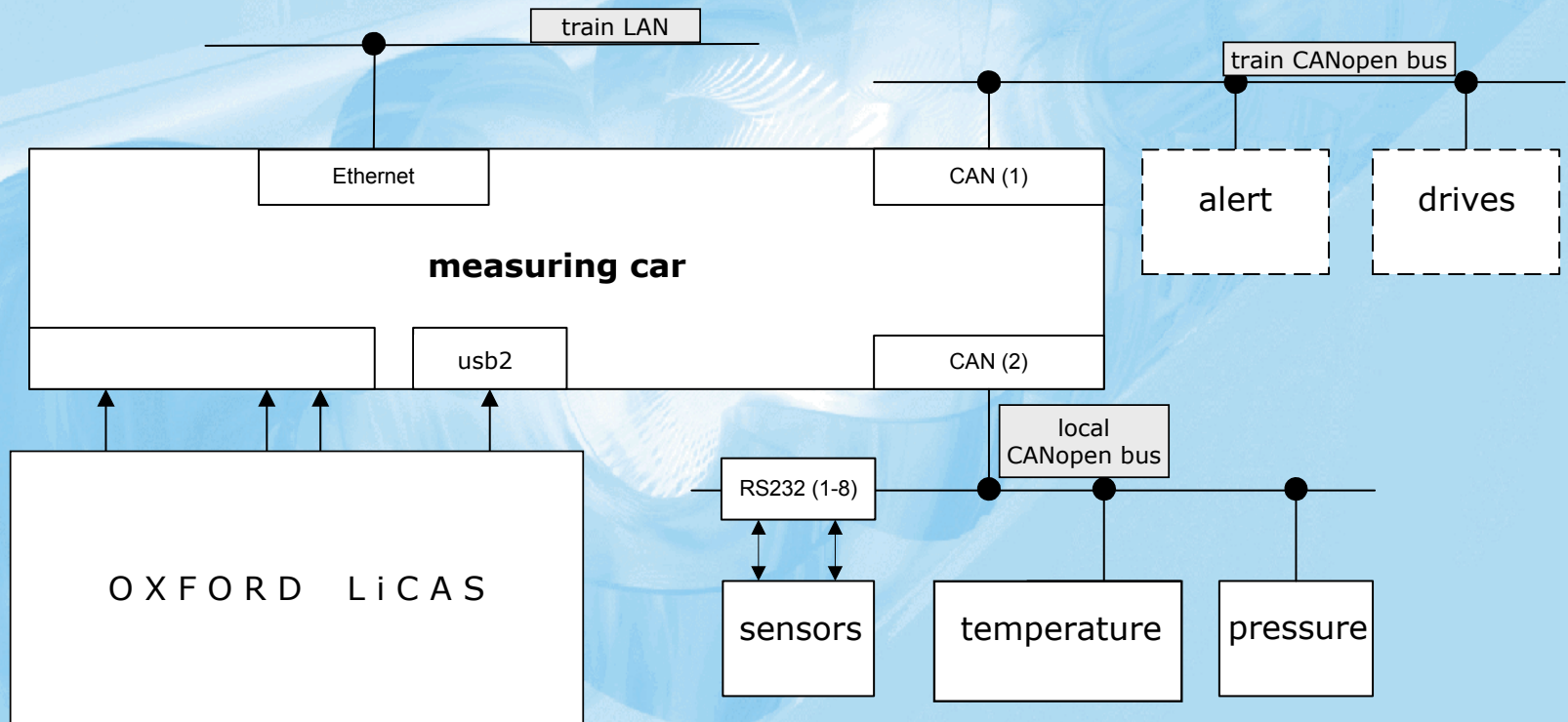
Hardware - train





LiCAS meeting – software considerations

Hardware - car





LiCAS meeting – software considerations

coding

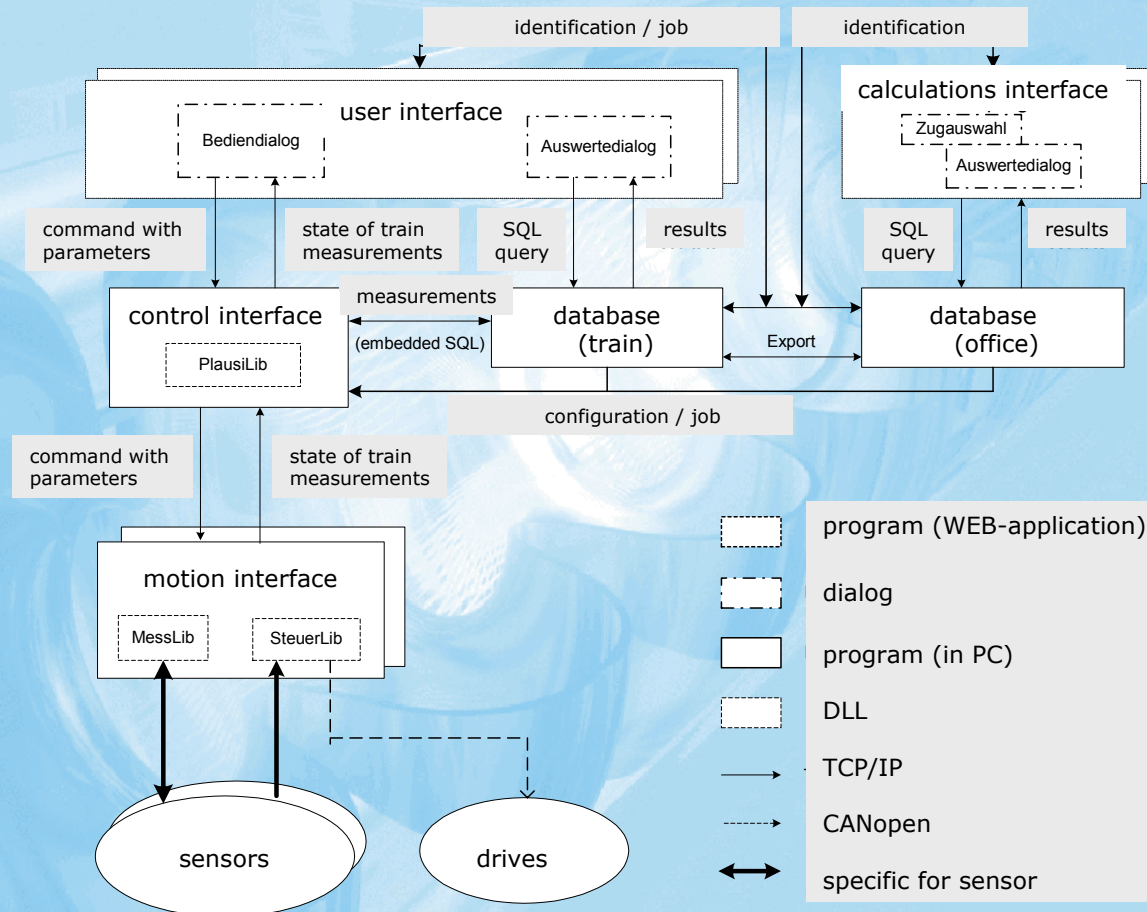
- global positioning coding will be
 - a) starting at the one side with a continuous increment (collider)
 - b) starting at the injector with a continuous increment (XFEL)

point classes

- definition of several types of targets
 - a) **stop** – point is a point where the train has to stop
 - b) **boundary** – point is a point where the train reaches the end
of a section
 - c) **reference** – point is every point on the tunnel wall
- probably needed for LiCAS
 - d) **marked** – point is an additional LiCAS prism



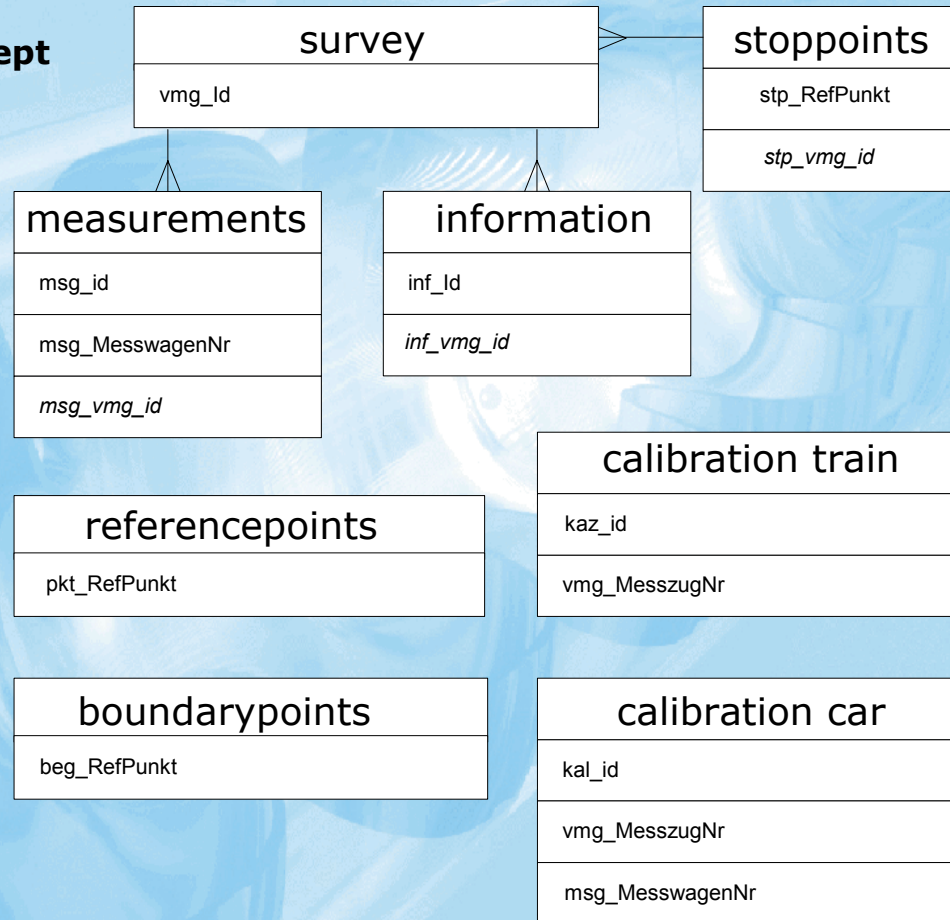
LiCAS meeting – software considerations





LiCAS meeting – software considerations

Database concept





LiCAS meeting – software considerations

user interface

TESLA: Bedienprogramm - Messwerte

Vermessungsinfo

Kennung

Anwender

	Beginn	Ende	Aktuell	Stopp
Punktnummer	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Datum	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Uhrzeit	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Rechnerstatus

Kontrollrechner

Messwagen 1

Messwagen 2

Messwagen 3

Messwagen 4

Messwagen 5

Messwagen 6

Fehlermeldungen

1 2 3 4 5 6

Punktnummer	<input type="text"/>
Linearführungen	Z <input type="text"/> y <input type="text"/>
Neigungssensor	Z <input type="text"/> X <input type="text"/>
Feinmesskamera Zielzeichen	linkes Teilbild: x <input type="text"/> y <input type="text"/> rechtes Teilbild: x <input type="text"/> y <input type="text"/>
Feinmesskamera Draht	oberes Teilbild: x <input type="text"/> y <input type="text"/> unteres Teilbild: x <input type="text"/> y <input type="text"/>
Inkrementaltaster	oberer Taster: x <input type="text"/> unterer Taster: x <input type="text"/>
Ultraschall	OF <input type="text"/> RF2 <input type="text"/> RF1 <input type="text"/>
Grobmesskamera Draht	Draht <input type="text"/> A <input type="text"/> B <input type="text"/>