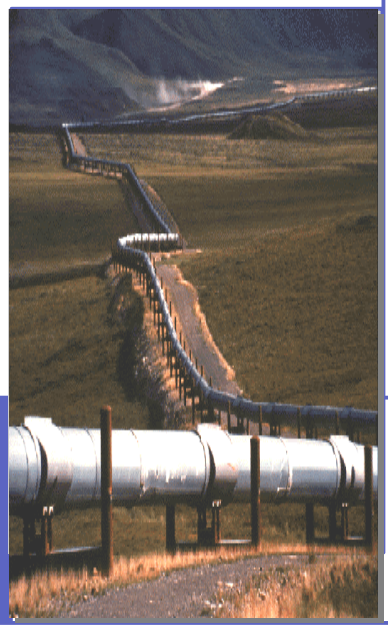
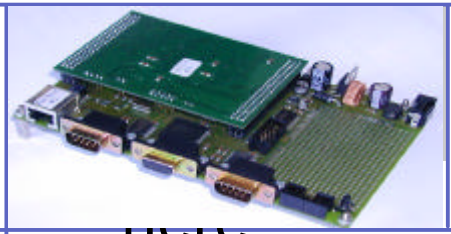
Web based services for embedded devices

Asset management on the level of field devices





here

here

Dr.U.Topp,
P.Müller

ABB Corporate Research

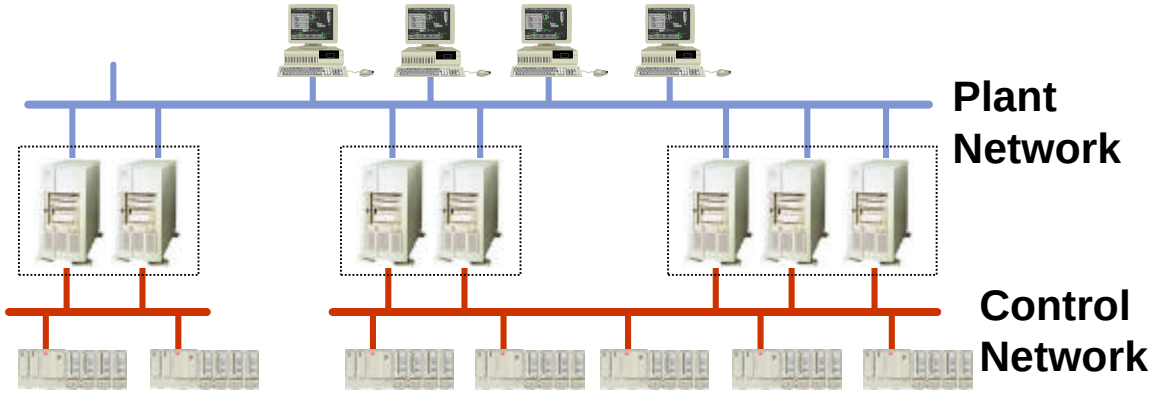
J.Konnertz,
A.Pick

IAS Uni Stuttgart

WHY ?

Current status and issue (I)

- Field devices widely used in industrial applications



- Many vendors
- Many protocols at all levels of ISO/OSI model

⇨ Many, many engineering tools

- design time of a systems
- during life time for service purposes

⇨ **Unacceptable costs since alternatives are arising!**

Curre

- Fie
- Ma
- Ma

⇨ Ur

ant
etwork

Control
Network

ng! d

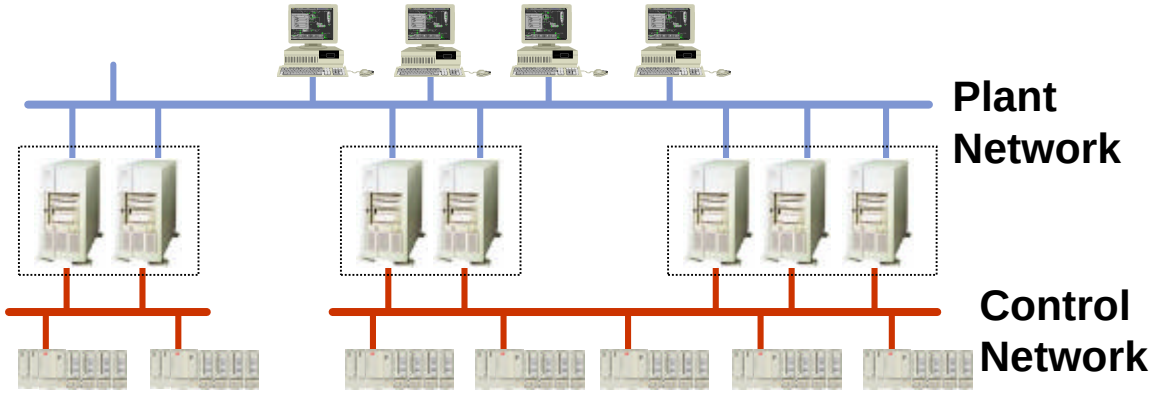
Field devices are Embedded Devices

Small electronic devices with a processor and environment dedicated to a special class of problems

- Control applications: derive direct actions from some sensor values (e.g. Opening/closing a valve)
- Analog and digital IO directly on board or via fieldbus connected
- CPU: MC683xx, ColdFire, StrongARM, ...
- Memory:
  - RAM: few hundred kilobytes to some Megabytes
  - ROM: Flash memory 1-16MB
- Application must run under „hard realtime“ restrictions
- other restricions like
  - areas with explosive gases
  - Often power consumption restrictions
  - Rough environment

Current status and issue (I)

- Field devices widely used in industrial applications



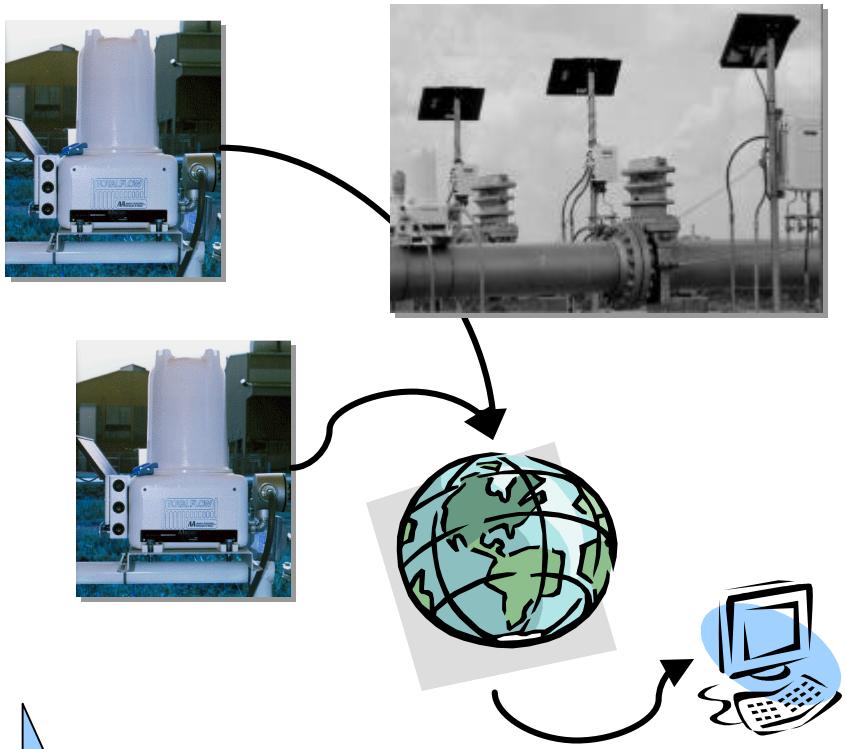
- Many vendors
- Many protocols at all levels of ISO/OSI model

⇨ Many, many engineering tools

- design time of a systems
- during life time for service purposes

⇨ **Unacceptable costs since alternatives are arising!**

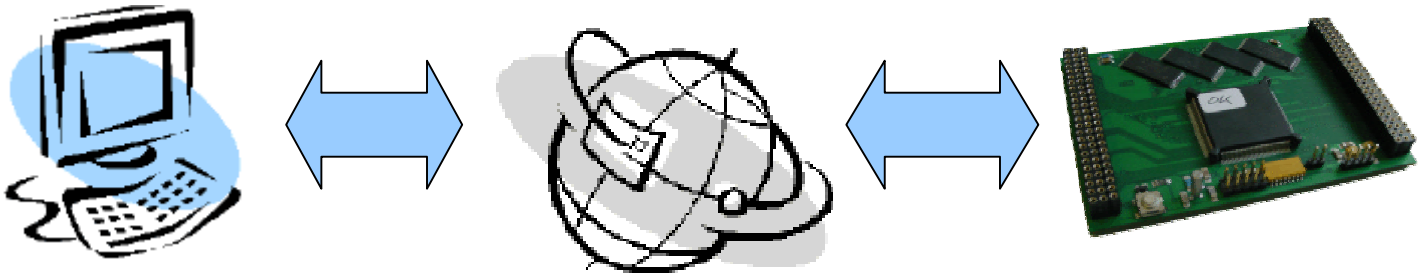
Current status and issue (II)



- Locally wide spread bunches of devices
  (not neccessarily the same manufacturer)
- Operational status and measurement values are to be monitored remotely
- Detect malfunction in advance, avoid unneccessary maintenance visits

Remote access crucial

Current situation and issues (III)



What is provided by todays solutions?

- Static information enriched with process data
- Focussed on presentation

Remote access of process values

What are the main requierements on future solutions?

Services:

- New services
- Highly flexible
- Interoperable

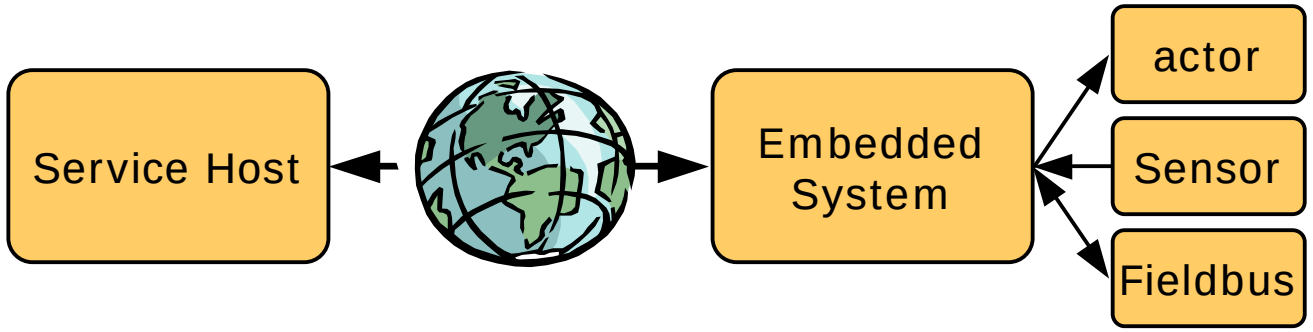
Technologies:

- Vendor independent
- Use existing standards
- Generic data description

Integrated remote services

WHAT ?

Remote Service Scenarios

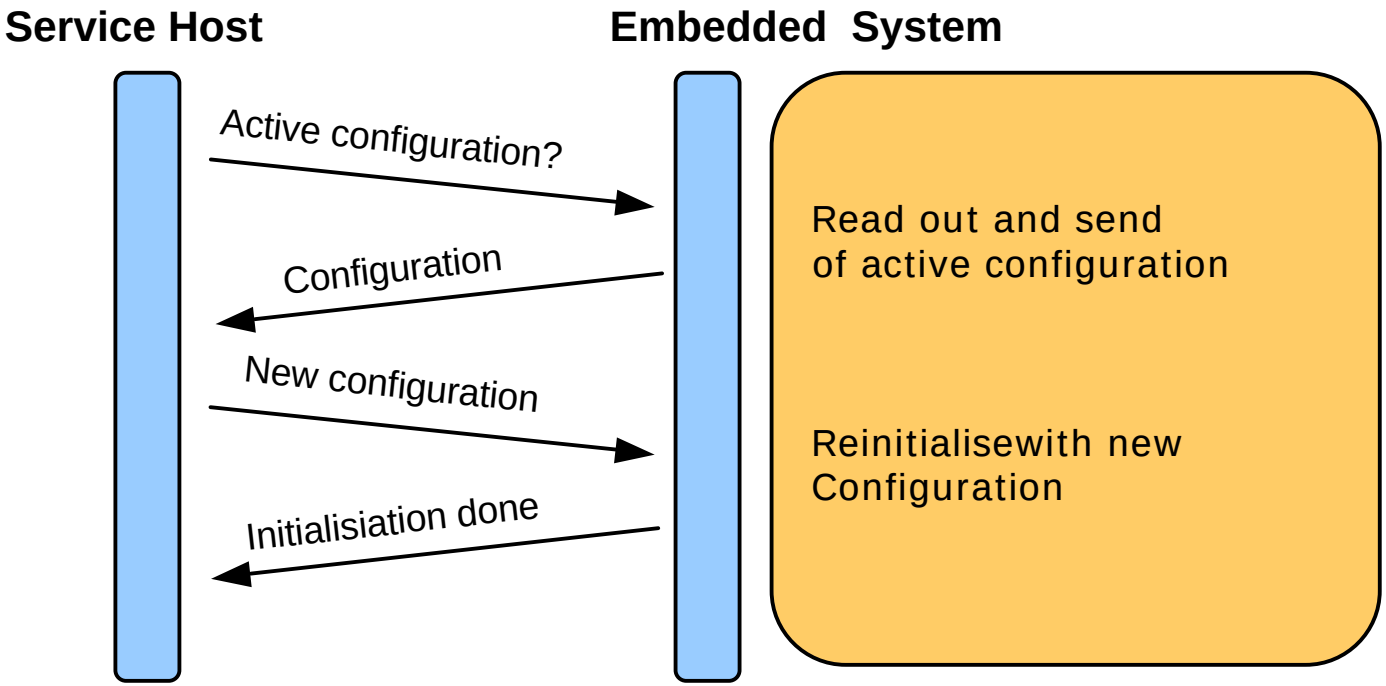


Goal: Flexible data transfer between service host and embedded system

Remote service applications

- Configuration
- Monitoring
- Diagnosis
- Alarm&Event handling

Remote Service - Configuration

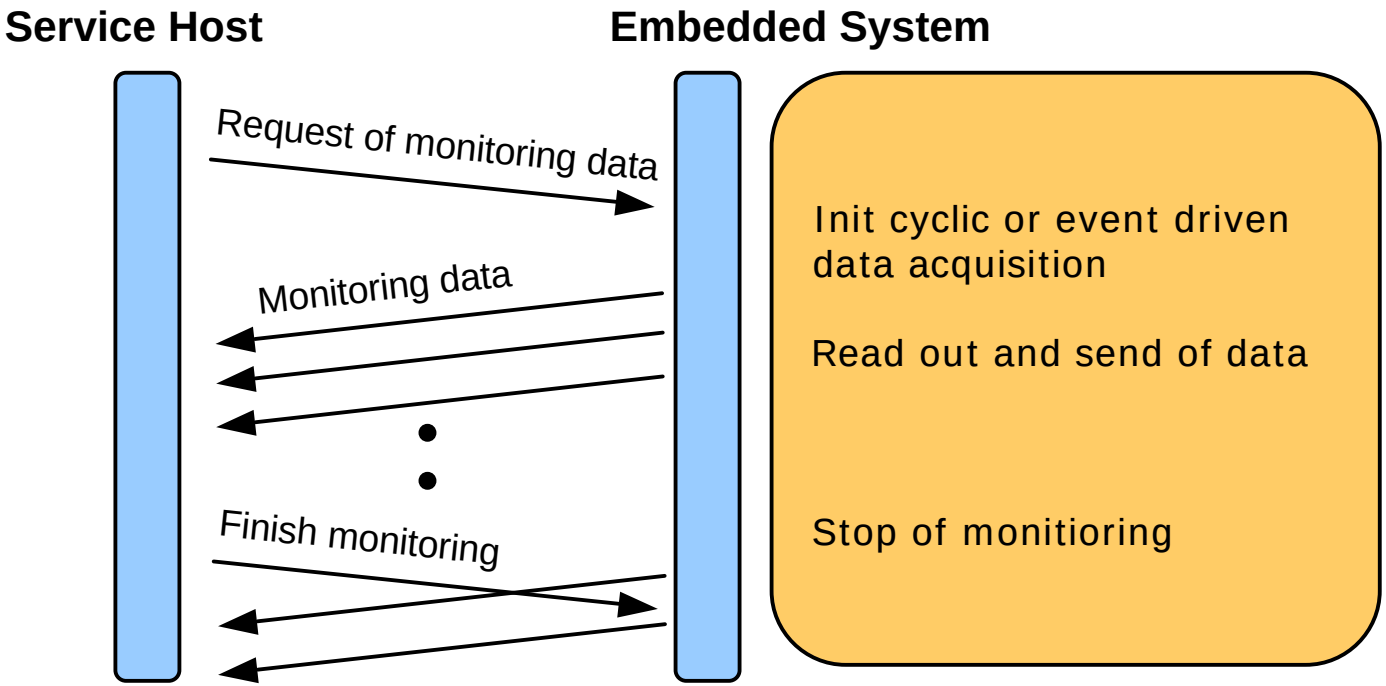


Charakteristika

- Active configuration is requested
- Change of configuration and re-transmit
- Take over of configuration and re-init of embedded system
- Signal succesful operation

- „Request-Response ....“ - Mechanism

Remote Service - Monitoring

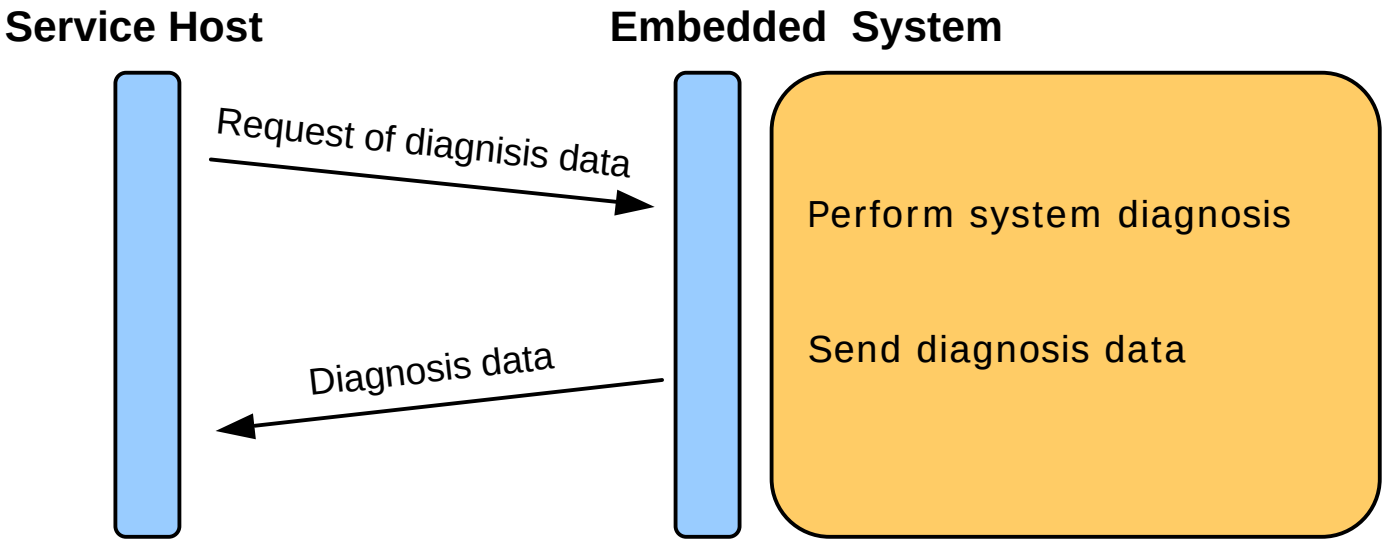


Charakteristika

- Request with detailed specification (What? How often? To Whom? ...)
- Cyclic data acquisition and sending
- End bysignal of Service Host

- „Request-Response-Response-Response ....“ - Mechanism

Remote Service - Diagnosis

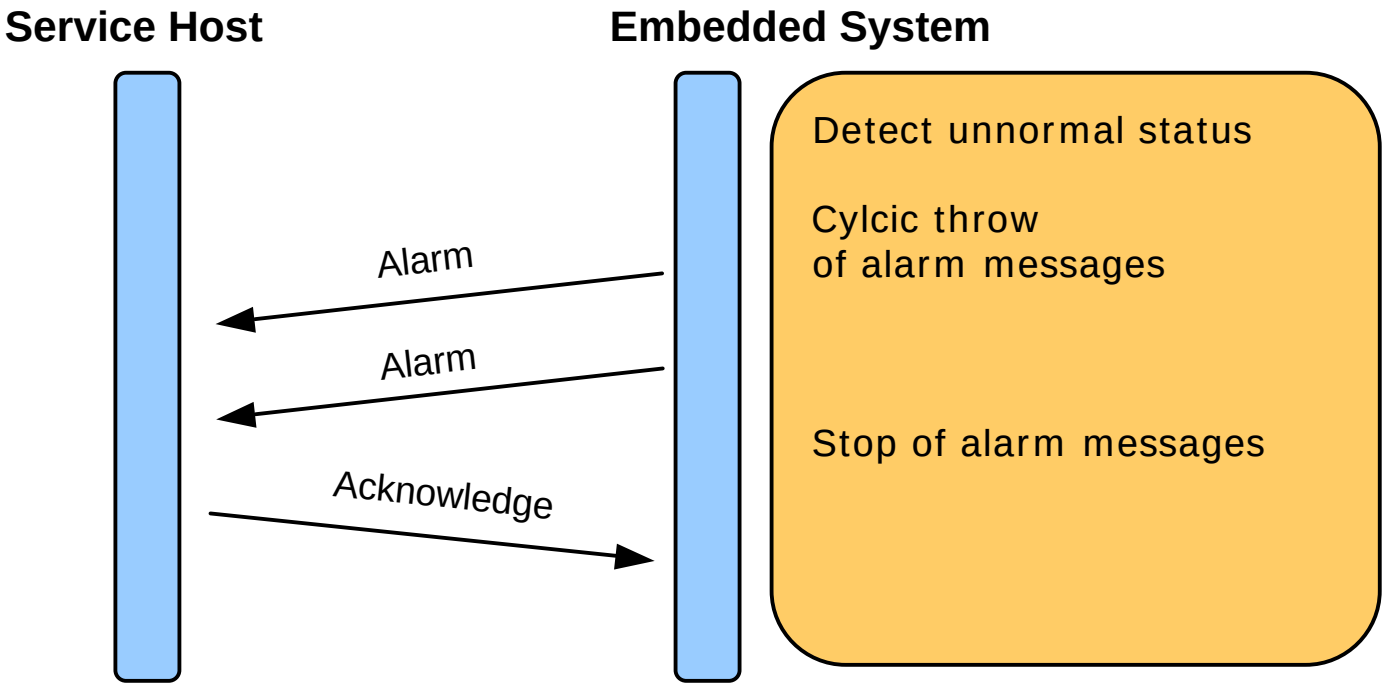


Charakteristika

- Request with detailed specification (What? To Whom? ...)
- Perform internal system diagnosis
- Send diagnosis data

- „Request-Response“ - Mechanism

Remote Service – Alarm & Event handling

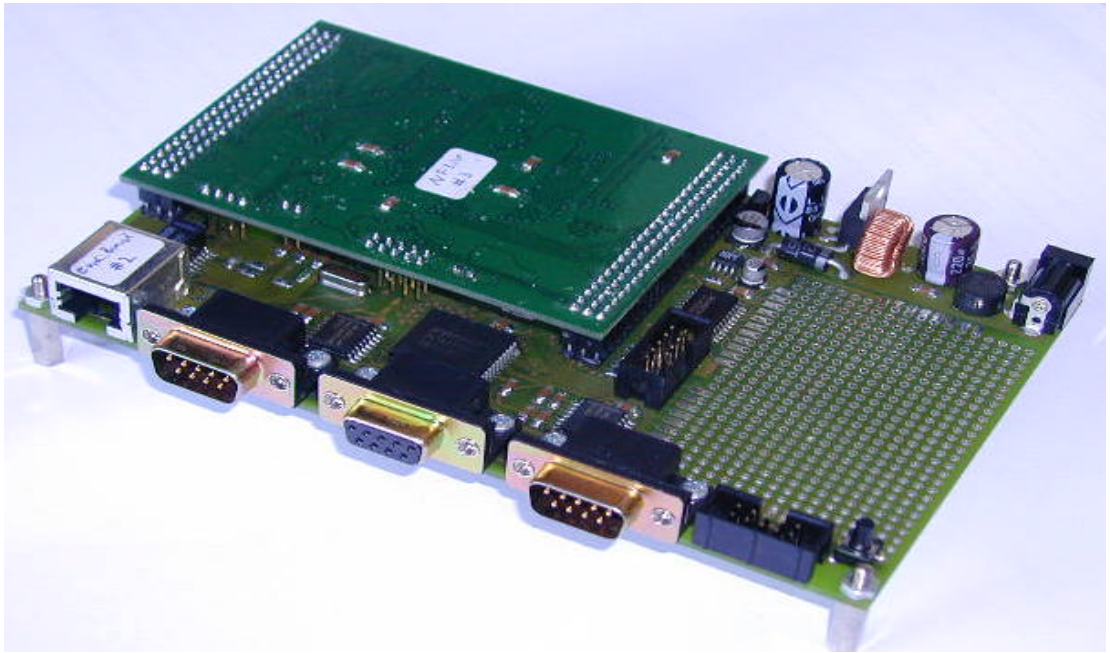


Charakteristika

- Detect exceptional system status
- Init of cyclic throwing of alarm messages
- Stop alarm messages after acknowledge

- Alarm reciever must be configured at initialistion time

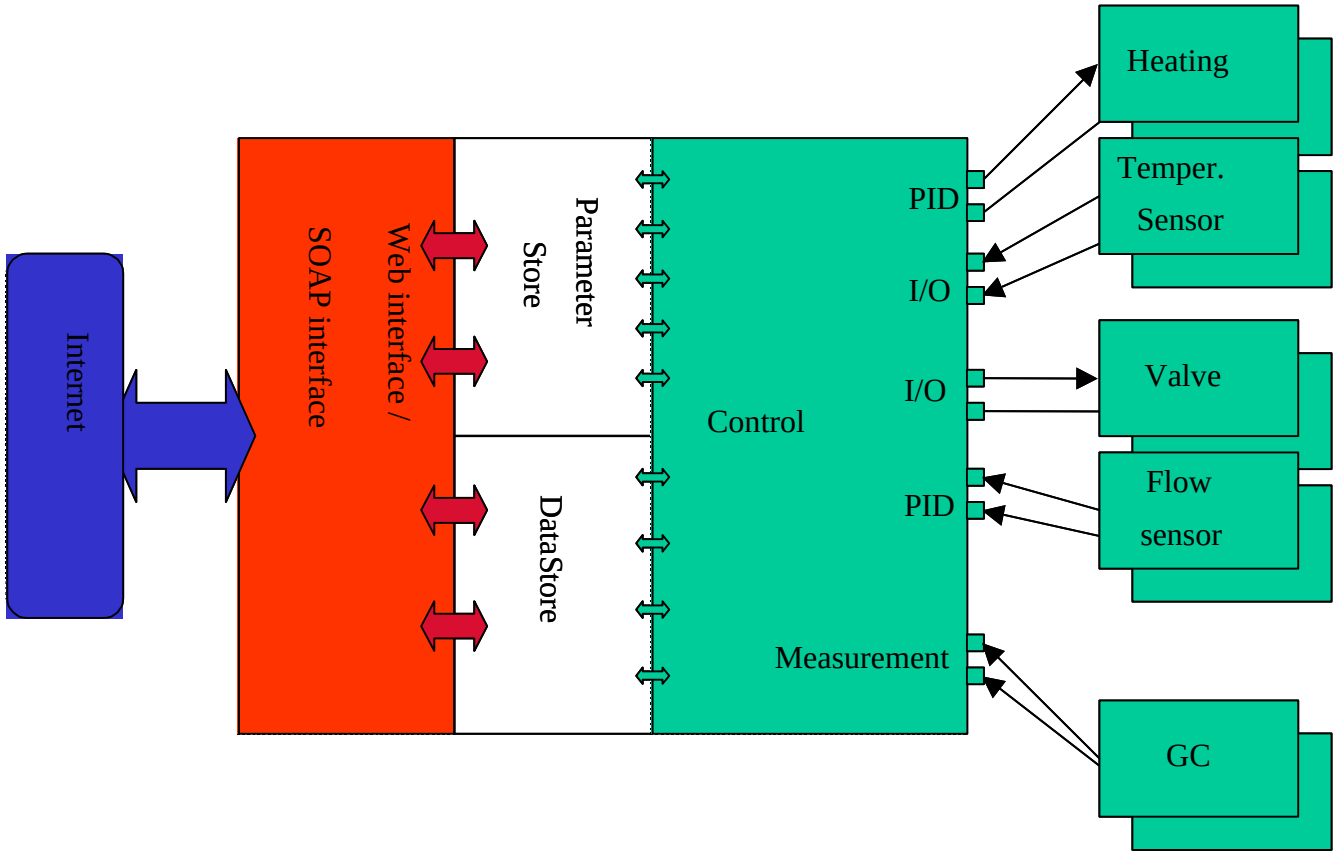
Prototype hardware



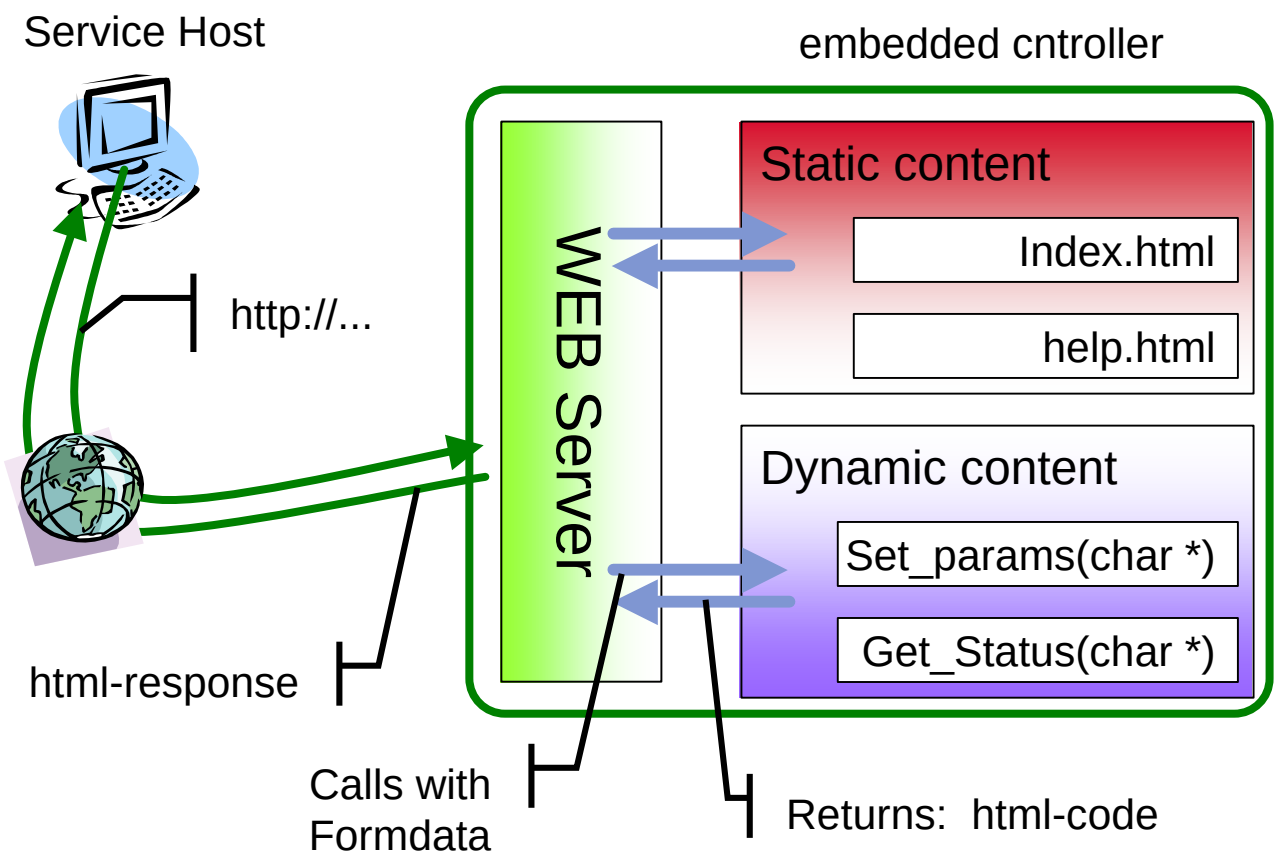
- Motorola 68332 CPU
- 1.25 MB RAM
- 1.25 MB FlashROM
- OS: RTEMS
  (RealTime for embedded multiprocessor systems)

- Connections
  - •Ethernet
  - •Serial
  - •Profibus
- Onboard IO
  - • 5x Analog IN
  - • Analog OUT via PWM
  - • Digital IO

General Architecture



Approach 1: Embedded Webserver with CGI



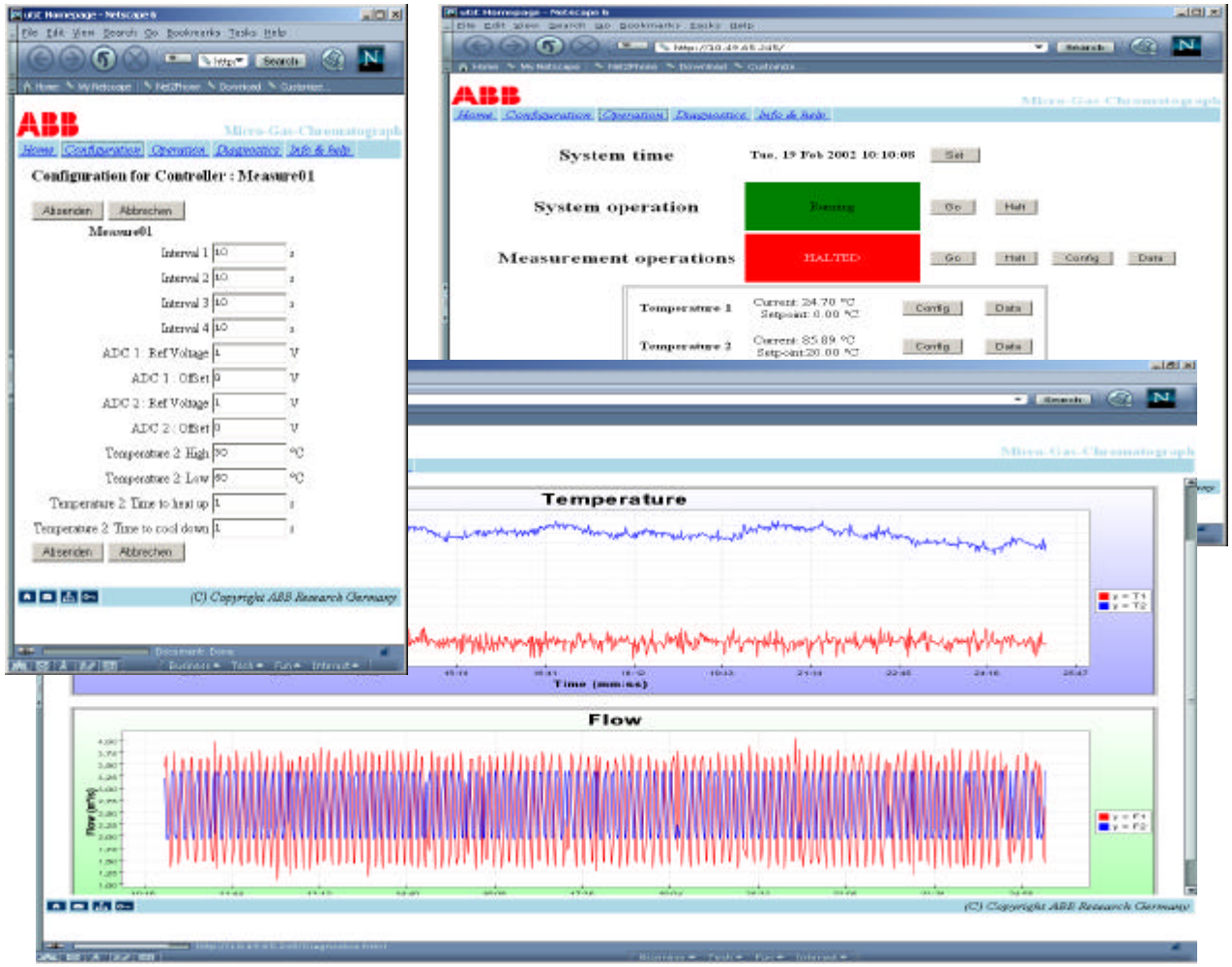
Webservice with CGI (II)

Static content

- „Homepage“ as portal page to the system
- Frame based layout
- Help texts
- Site index
- Link to mail recipient

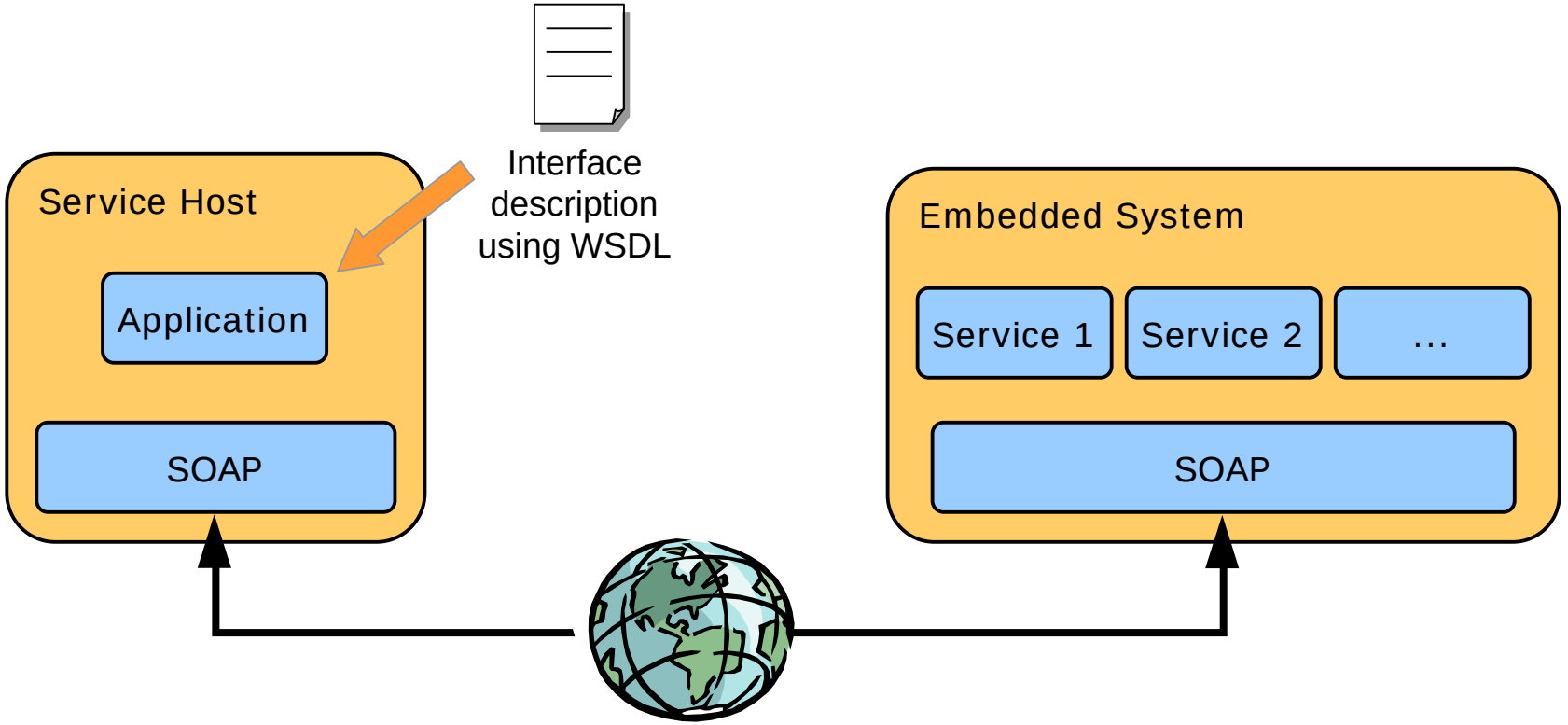
Webservice with CGI (III)

Dynamic content



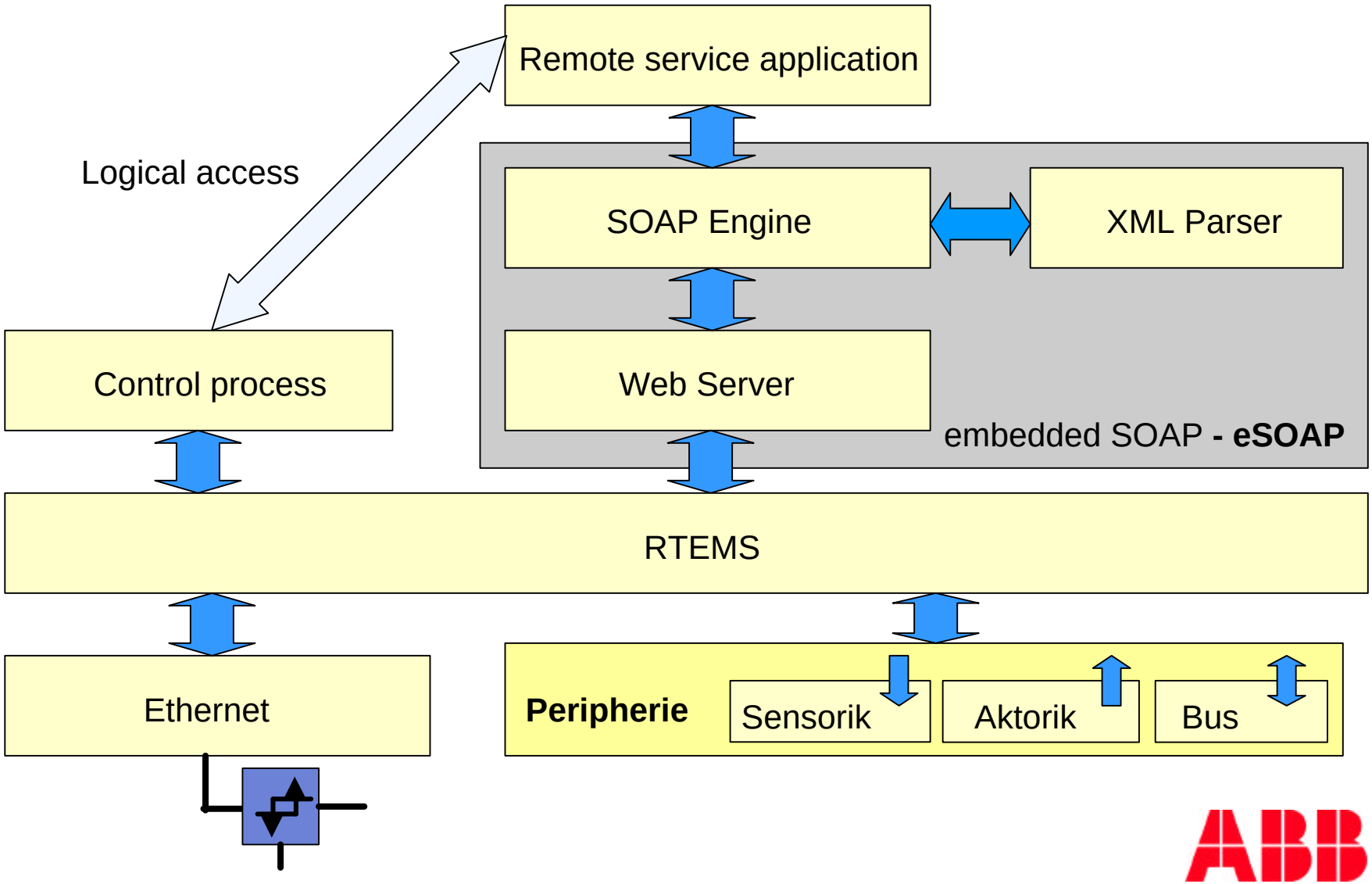
- Device status
- Configuration page(s)
- Graphical display of historical data

Approach 2: Embedded SOAP



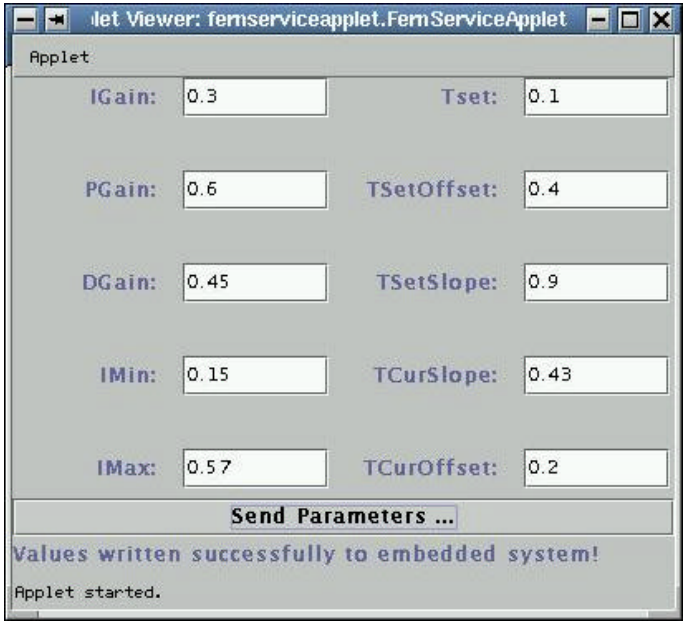
- Common commuication base: SOAP : guarantees interoperability
- Application on Service Host gets interface from WSDL specification of the embedded system
- Generic application possible

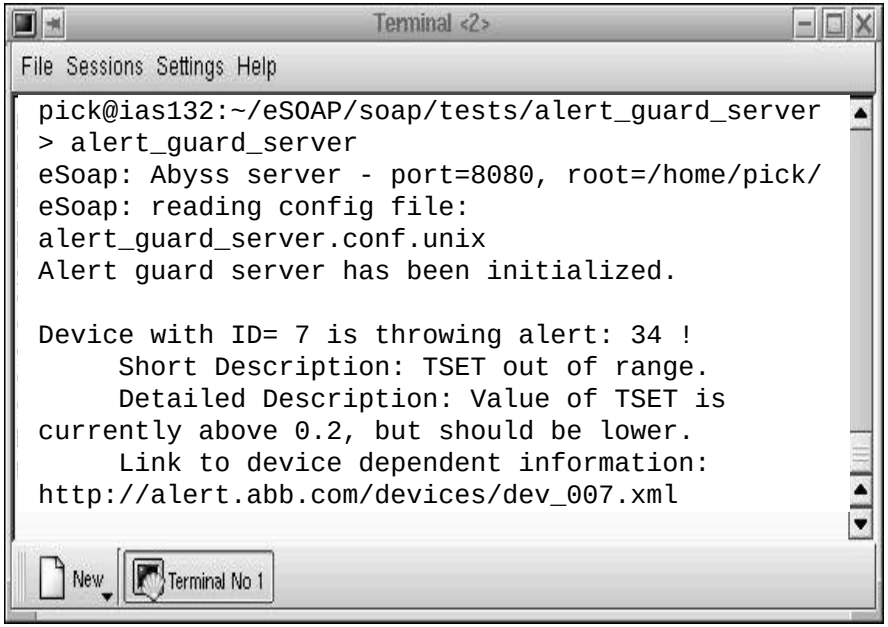
Embedded SOAP (II) Architecture



Embedded SOAP (III)

Implemented scenarios





- Configuration:
  - Java program generated with tool support out of WSDL description
- Alarm handling
  - PC hosted Alarm server

SO WHAT?

Comparision of different approaches

	Conv., propr. protocol	webserver w/ dynamic html	Webserver w/ SOAP services
Inter-operability	—	+	+
User Interface	— (generated by hand)	+ (web browser)	O (generated with tool support)
Resources	+	O (ca. 45kB)	— (ca. 400kB)
Reuse of components	—	O	+

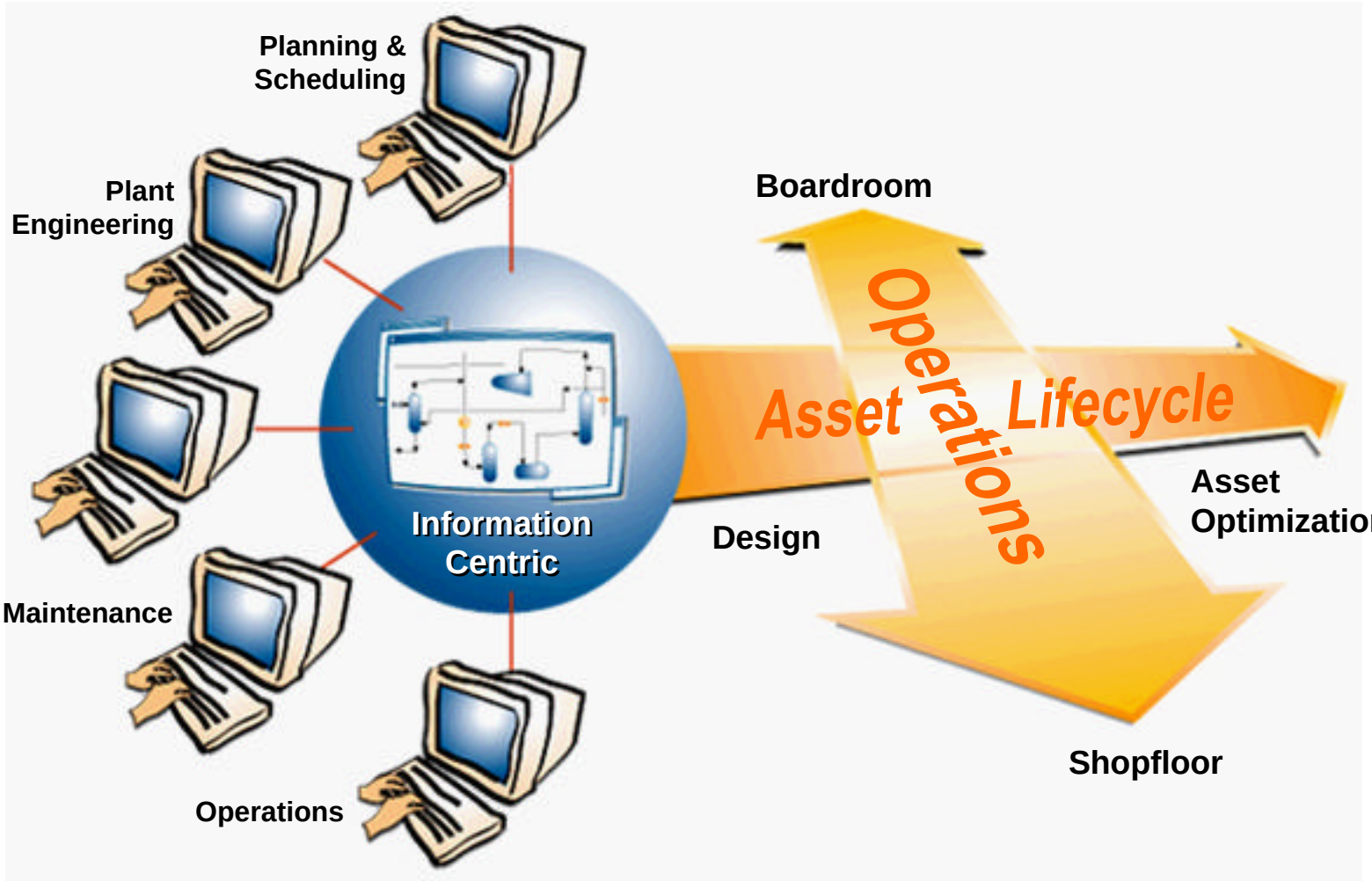
Outlook

- Open issues
  - Security
    - Standardized Internet and Web Technology opens the door for hackers (proprietary protocols are known only by insiders)
  - QoS over internet connection
    - Alarm and Event handling might be time critical
    - Back up strategies must be in place in case an alarm gets no response
- Next steps
  - Identify other potential devices and systems for the remote service capabilities
  - Thorough implementation of the different scenarios
  - Further employment of the SOAP strategy

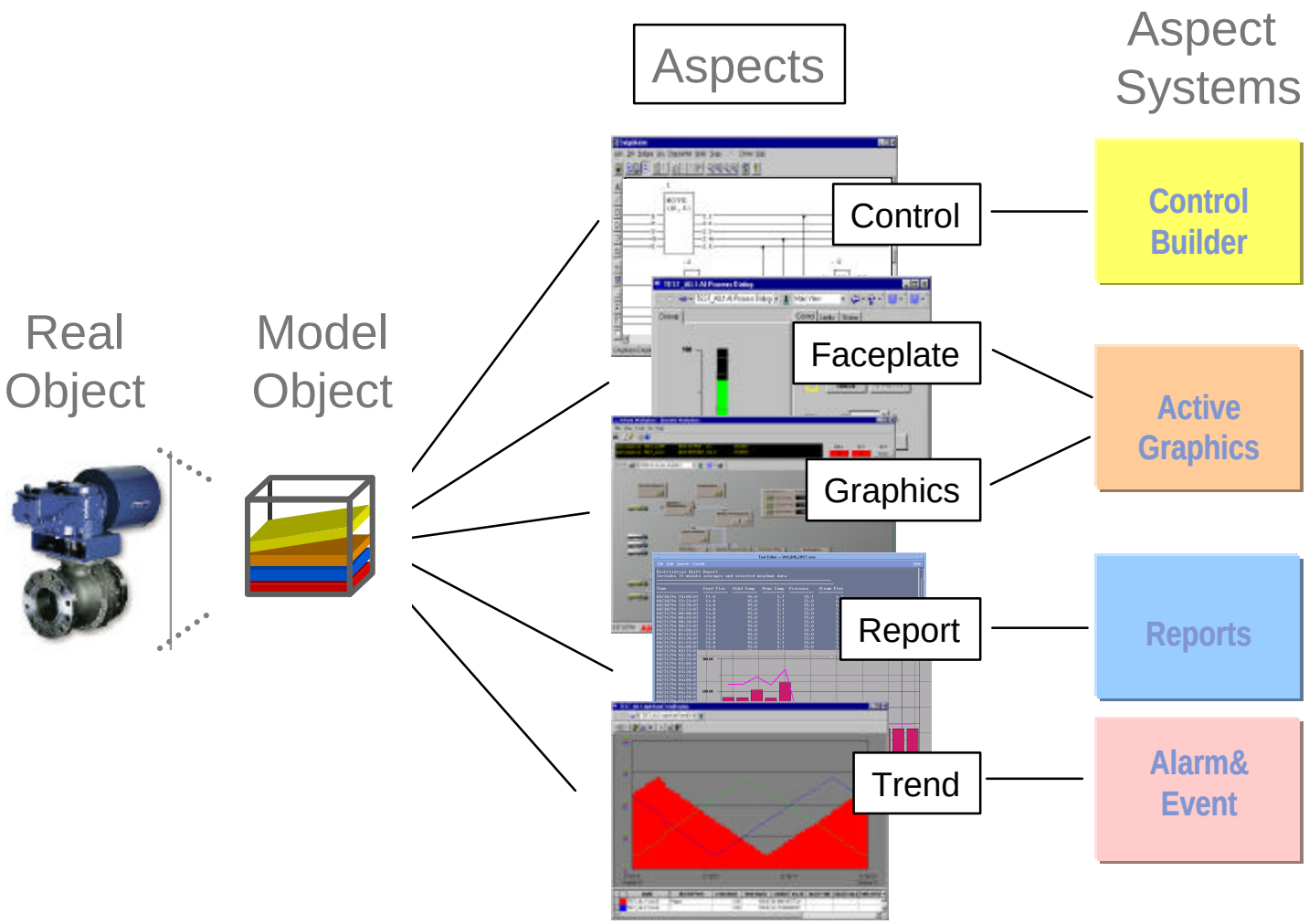
&

Universität Stuttgart
Institut für Automatisierungs- und Softwaretechnik
Prof. Dr.-Ing. Dr. h. c. P. Göhner

Industrial-IT (I)



Industrial-IT (II)



Industrial-IT Integration