OSADL Fieldbus Framework Meeting

SPS/IPC/Drives 2008
2008-11-26, 9:30 - 13:00 MEZ

Robert Schwebel <r.schwebel@pengutronix.de>

Agenda

- 10:00 - 11:00 Presentations I: Technology
  - Aims of the Fieldbus Framework Project (R. Schwebel, M. Kremer)
  - Current State of Prototypes: 3S, Pengutronix (R. Schwebel)
  - User Expectiations (Plenum Discussion)
- 11:00 - 11:30 Coffee Break
- 11:30 - 12:00 Presentations II: Administrative Aspects
  - Legal Status of Userspace Drivers (C. Emde)
  - Business Plan Draft (R. Schwebel)
- 12:00 - 13:00 Panel Discussion
  - Commitments
  - Definition of a Roadmap
  - Assignment of Tasks

Agenda

- 10:00 - 11:00 Presentations I: Technology
  - Aims of the Fieldbus Framework Project (R. Schwebel, M. Kremer)
  - Current State of Prototypes: 3S, Pengutronix (R. Schwebel)
  - User Expectiations (Plenum Discussion)
- 11:00 - 11:30 Coffee Break
- 11:30 - 12:00 Presentations II: Administrative Aspects
  - Legal Status of Userspace Drivers (C. Emde)
  - Business Plan Draft (R. Schwebel)
- 12:00 - 13:00 Panel Discussion
  - Commitments
  - Definition of a Roadmap
  - Assignment of Tasks

Aims of the Fieldbus Framework Project

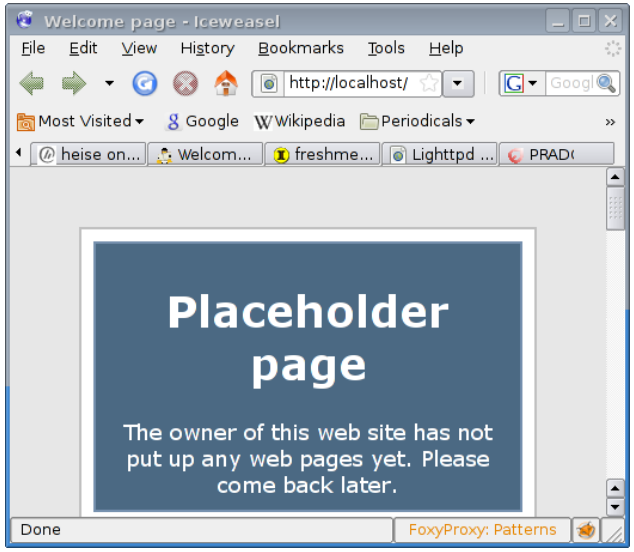
- How does non-Industrial networking work these days?

```
root@thebe:~# ifup eth0
root@thebe:~# apt-get install lighttpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  lighttpd
0 upgraded, 1 newly installed, 0 to remove and 552 not
  upgraded.
Need to get 360kB of archives.
After this operation, 979kB of additional disk space will be
  used.
Do you want to continue [Y/n]? y
Get:1 http://debian.tu-bs.de testing/main lighttpd 1.4.19-5
  [308kB]
Fetched 360kB in 0s (431kB/s)
Selecting previously deselected package lighttpd.
Unpacking lighttpd (from .../lighttpd_1.4.19-5_i386.deb) ...
Setting up lighttpd (1.4.19-5) ...
Starting web server: lighttpd.
root@thebe:~#
```

Aims of the Fieldbus Framework Project

- How does non-Industrial networking work like these days?

```
root@thebe:~# ifup eth0
root@thebe:~# apt-get install lighttpd
```

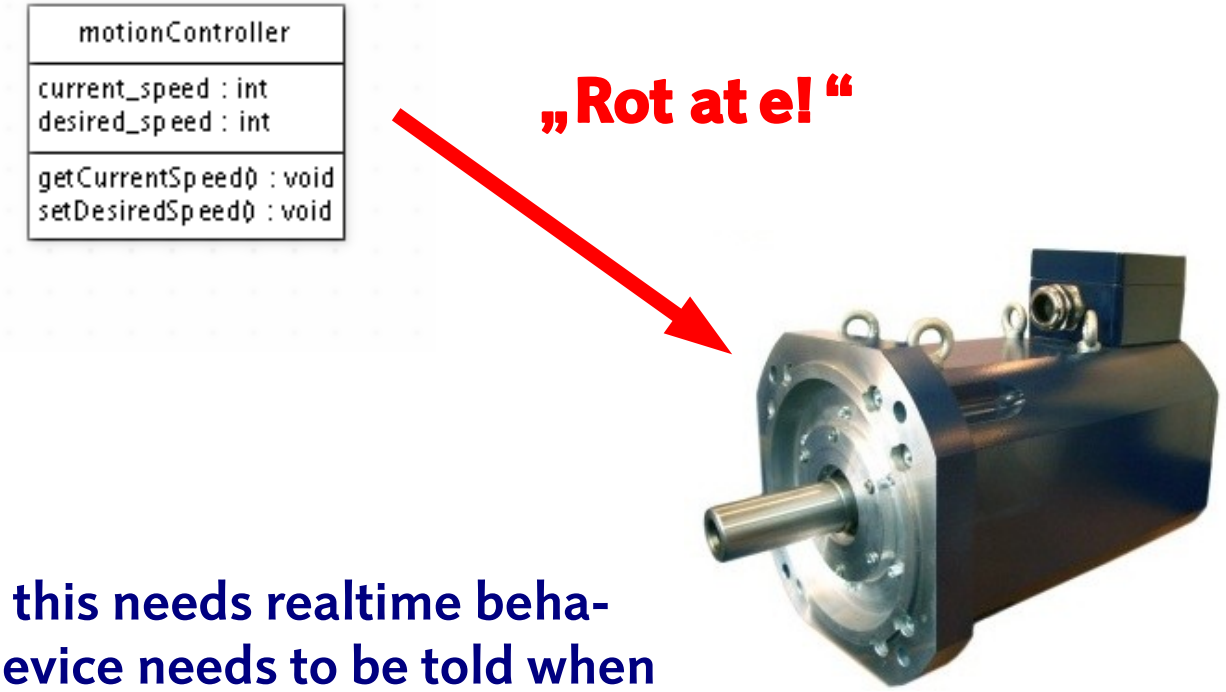


- **Industrial Networking should be That Simple (TM)!**

Pengutronix.

Which Problem are we Trying to Solve?

- Tell hardware what to do:



(Obviously, this needs realtime behaviour, the device needs to be told when and how, but we leave these boring details out for now ...)

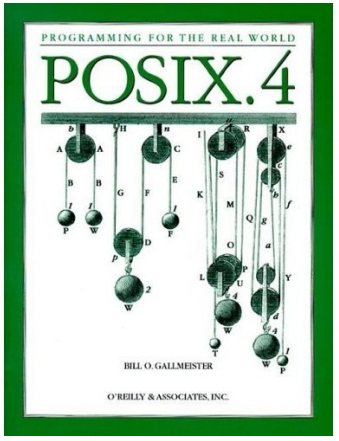
Pengutronix.

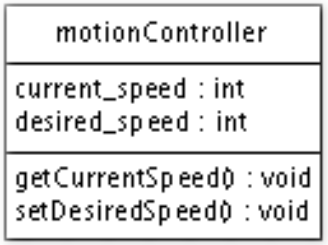
What we have these days instead is ...

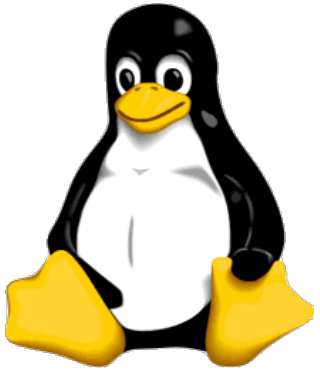
1. ... no modern APIs (object dictionary is no IT interface!)
2. ... no language independence
3. ... no clear layering (in code, not in marketing slides)
4. ... no public review of stack code
5. ... no vendor independence
6. ... no bus independence
7. ... no market independence

Design Goal 1: Modern APIs

- Think POSIX
- Respect the last 15 years of Linux Development
- Object Oriented Design
- Multi Instance







Pengutronix.

Design Goal 2: Language Independence

- No constraints to programming language:

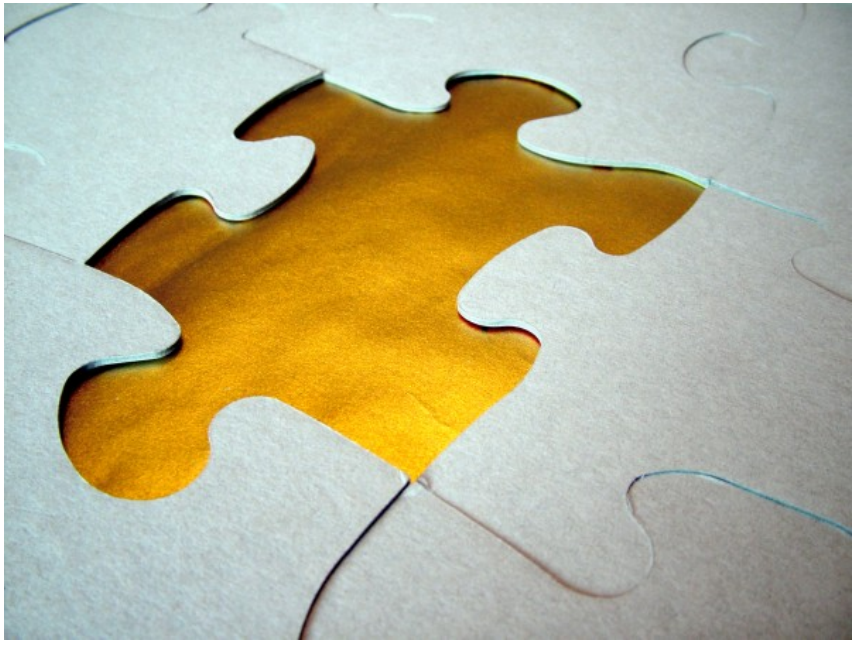
  C, C++, Java, C#, IEC-61131, Python, Ruby, ... ergo: C with Wrappers

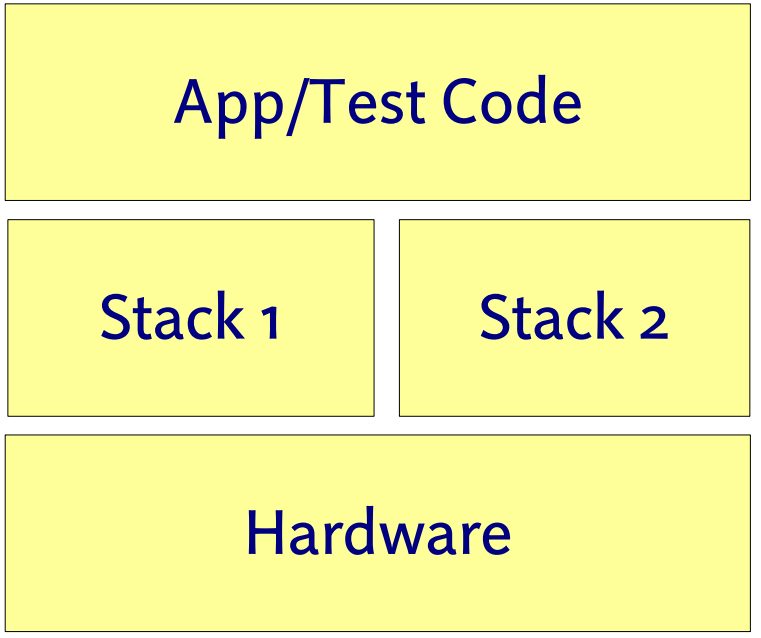
- Must run on low-ressource systems:

  H720x (70 MHz ARM); Blackfin (no mmu)

Design Goal 3: Clear Layering

- Well Defined Interfaces

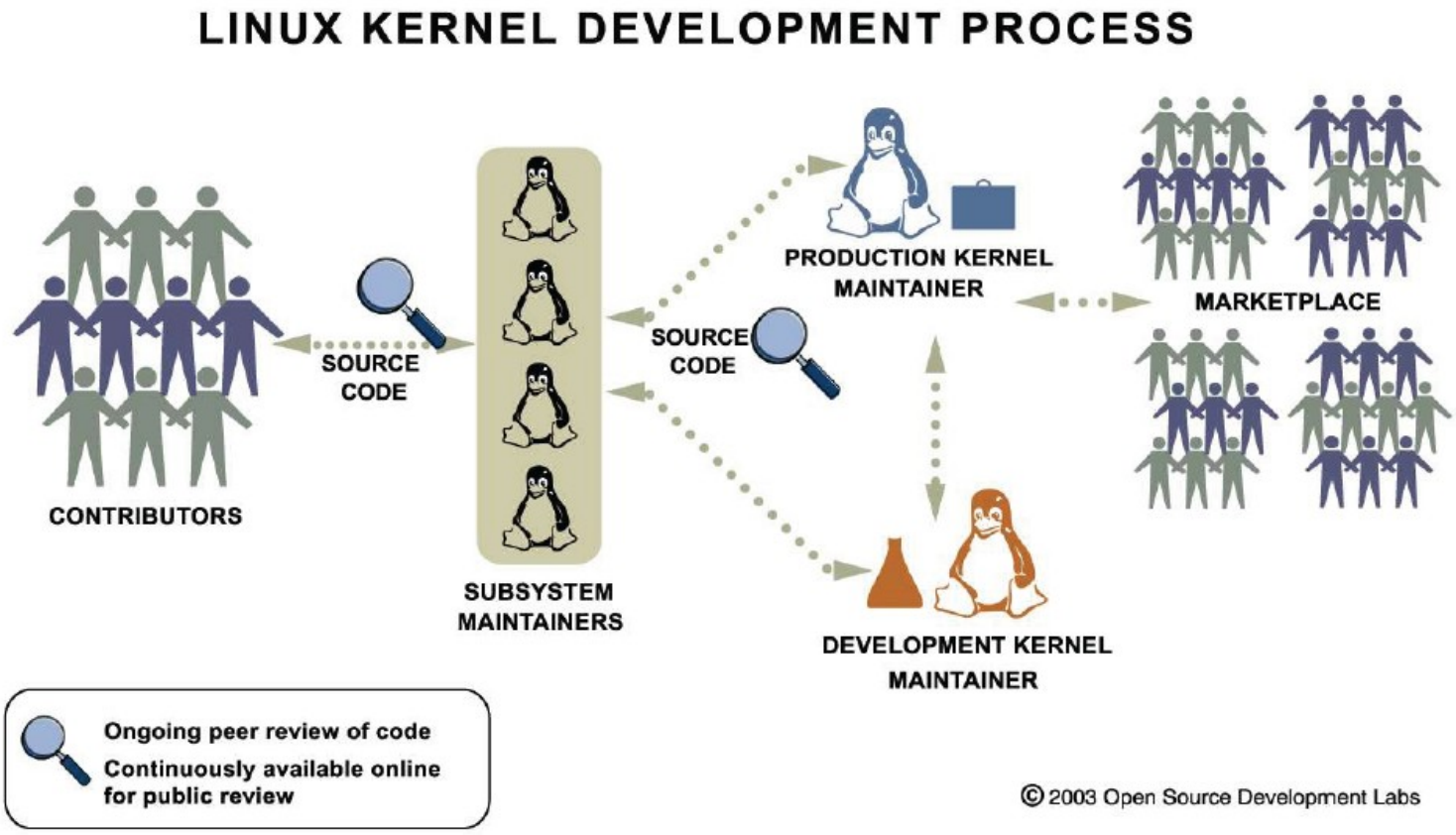




- Testability / Certification
- Clean, Layered Stack Design
- Plugin Concept (Proprietary / OSS) - „Buy & Play“

Pengutronix.

Design Goal 4: Peer Review



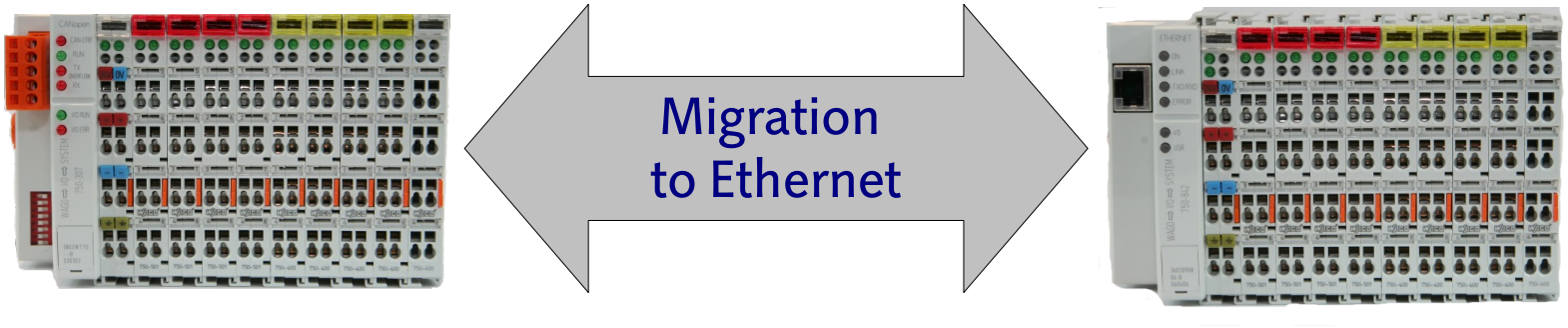
Design Goal 5: Vendor Independence

- Strategic Arguments:
  - Escape one-way vendor lock-ins by migrating to service & support business models
  - „Best Source“ strategy: optimal mix of open and closed source
  - Linux makes independence of a single processor manufacturer possible
  - Linux makes independence of operating system vendors possible

Pengutronix.

Design Goal 6: Bus Independence

- Unified Framework for All Fieldbus Technologies



Or:

- Several Fieldbusses in One System (i.e. local I/O + Fieldbus I/O)
- Box with local I/Os (GPIO, SoC integrated ADC/DAC)

Design Goal 7: Market Independence

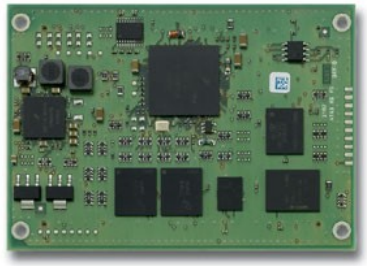
- We must match to as many markets' demands as possible:
  - Automation (PLC: read inputs - calculate - write outputs - cycle)
  - Measurement (bus sends data packets, stream away; sync reqs)
  - Automotive (self defined protocols)
  - Embedded (local I/Os, no fieldbus at all)

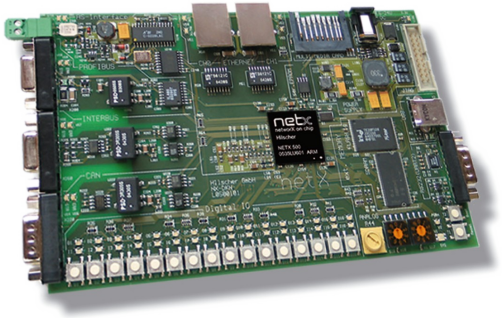
Agenda

- 10:00 - 11:00 Presentations I: Technology
  - Aims of the Fieldbus Framework Project (R. Schwebel, M. Kremer)
  - Current State of Prototypes: 3S, Pengutronix (R. Schwebel)
  - User Expectiations (Plenum Discussion)
- 11:00 - 11:30 Coffee Break
- 11:30 - 12:00 Presentations II: Administrative Aspects
  - Legal Status of Userspace Drivers (C. Emde)
  - Business Plan Draft (R. Schwebel)
- 12:00 - 13:00 Panel Discussion
  - Commitments
  - Definition of a Roadmap
  - Assignment of Tasks

Current State of Prototype at Pengutronix

- Industrial I/O has some history in our projects (see next slide ...)
- We are „Embedded People“, not „Automation People“
  (but some of our customers are)
- Our customers have Embedded Systems which need to interact with their environment

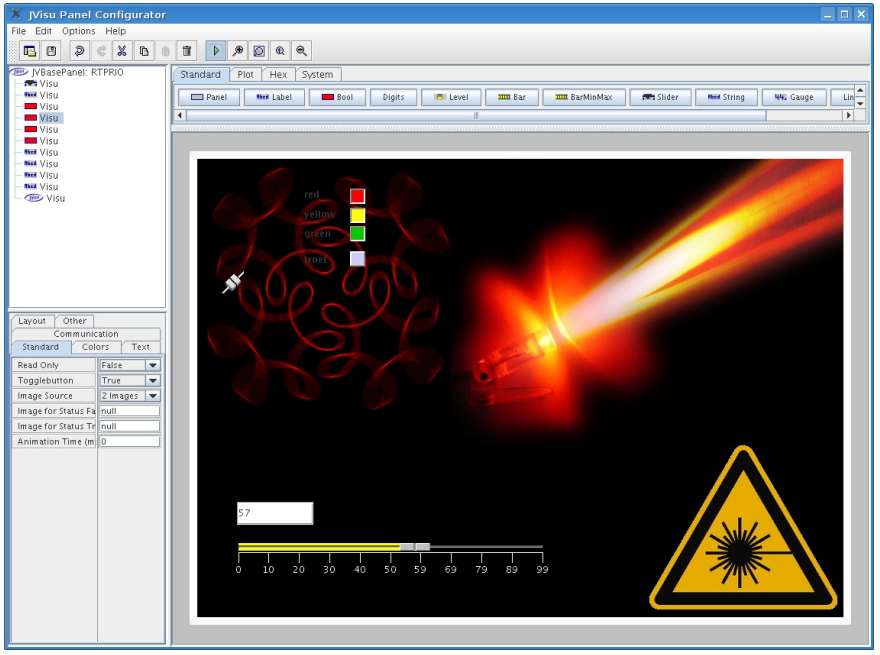




Pengutronix.

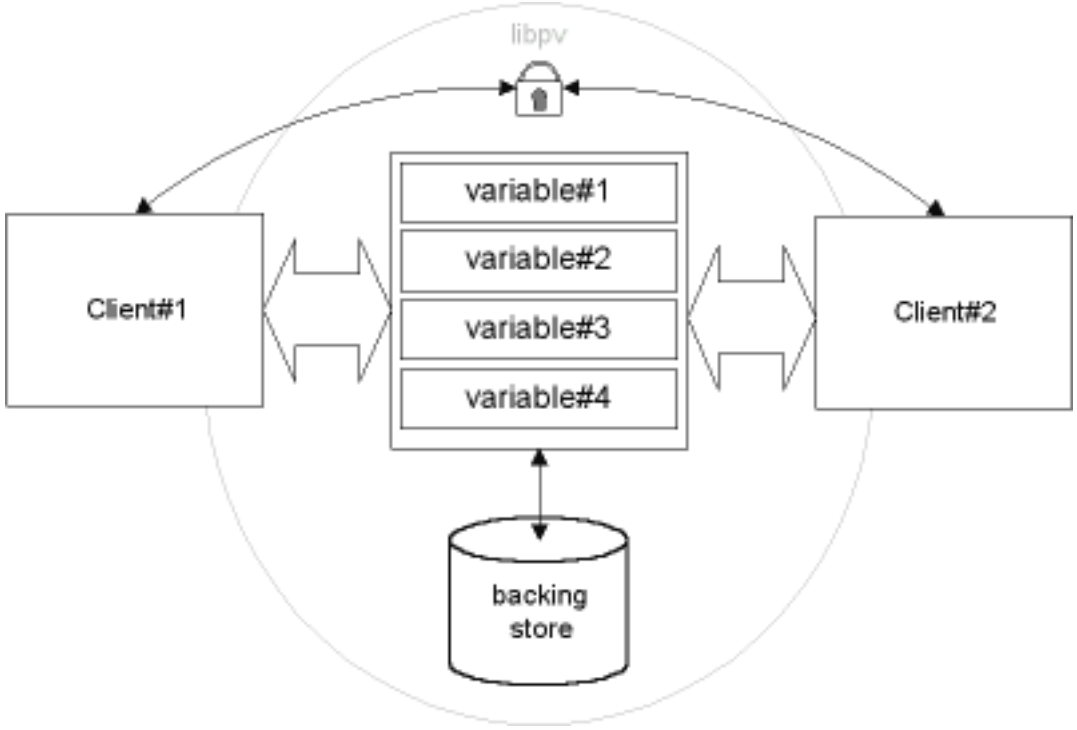
History @PTX: JVisu

- Process Data first came up in our project around 2003
- No fieldbus involved - first project was about ultrasound measurement
- First experiments with JVisu (Gleixner), but it was necessary to compile software on the backend side for process data integration



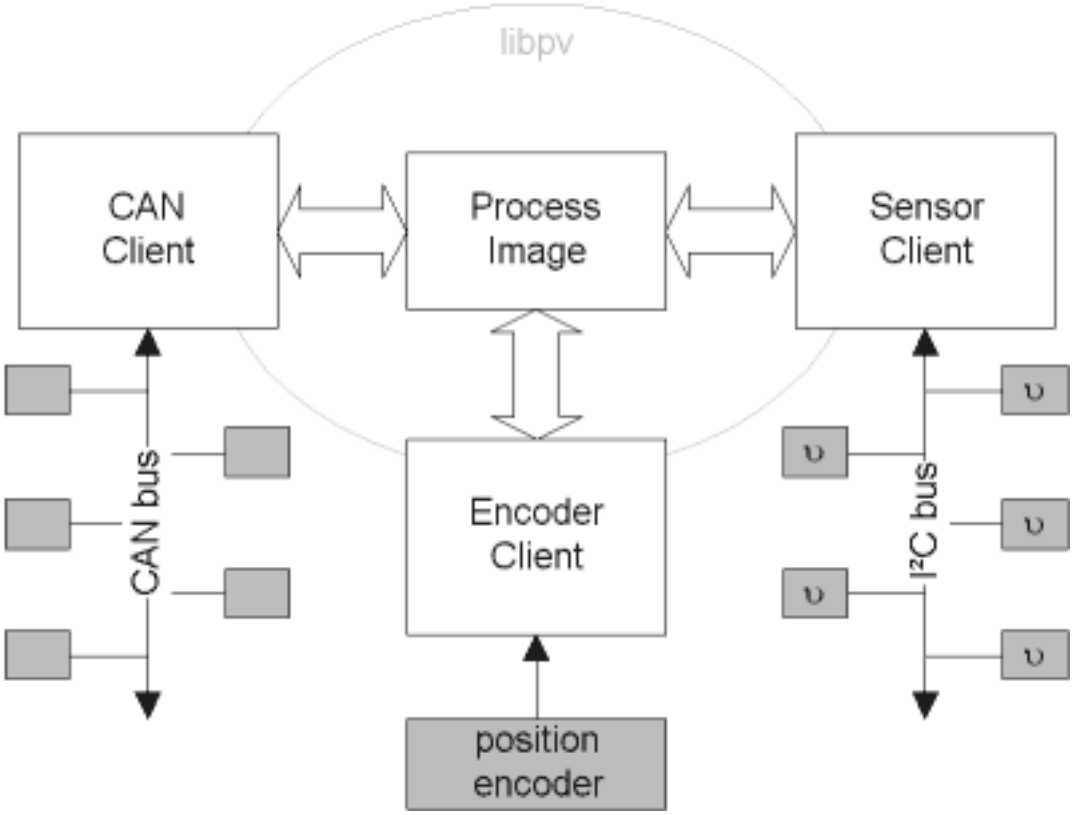
History @PTX: libpv-1.x

- During our projects, an Open Source library has grown up to access shared process data:



History @PTX: libpv-1.x

- libpv-1.x is used in more complex scenarios today
  (autonomous measurement robots, data loggers, heat pumps ...)



Problems with libpv-1.x

- We didn't care about the „where do the data come from“ question
- Centralized locking: race free, but slow
- No mapping layer, variables have to be referenced by name
- We started searching for alternatives in 2006...

Ptolemy II

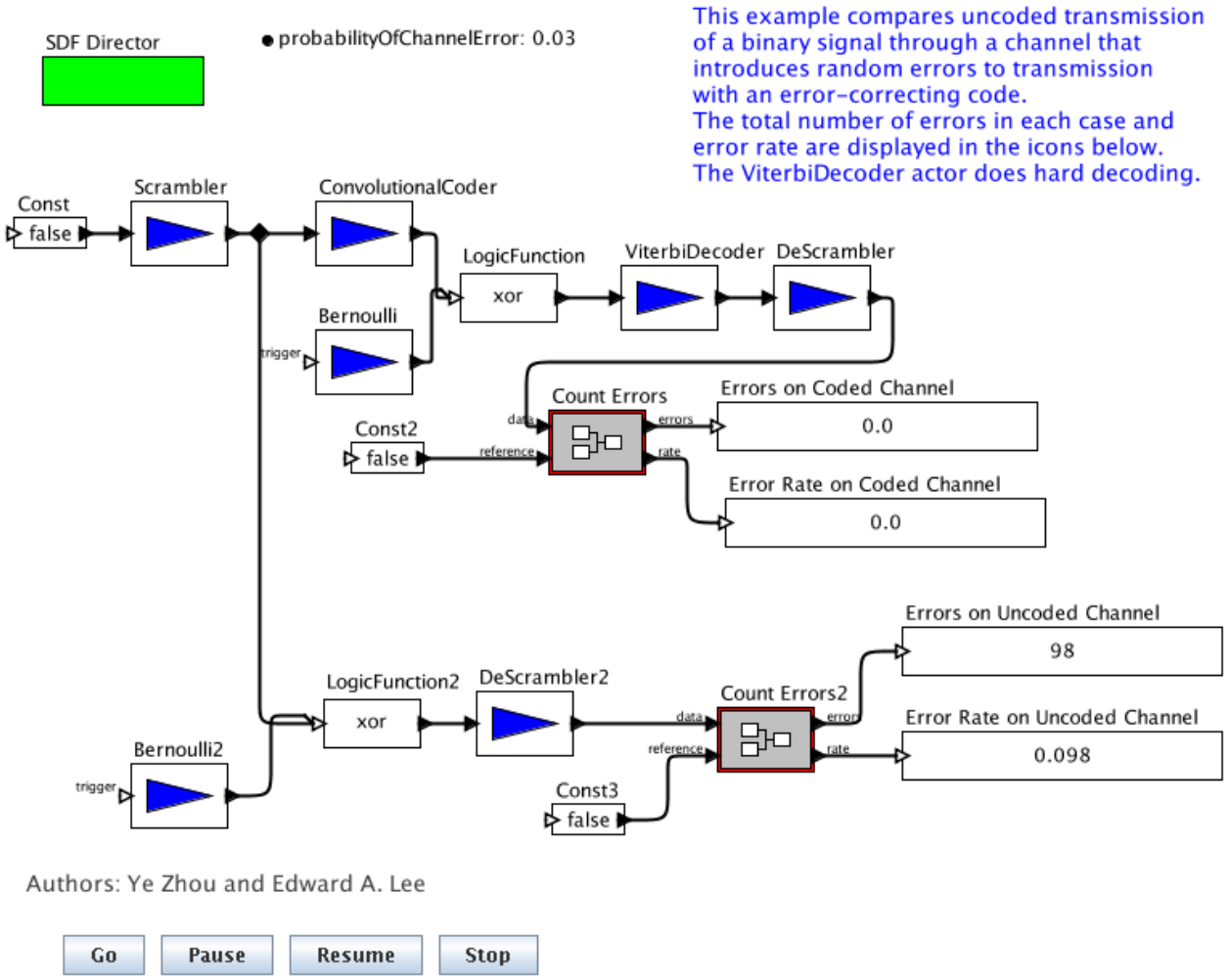
Pengutronix.

Ptolemy II:

- „Heterogeneous Hierarchical Modeling“
- „The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems“

  (http://ptolemy.eecs.berkeley.edu)
- Model Based Design:
  - Hierarchical Topology Models
  - Several „Domains of Execution“ - Semantics
- The Ptolemy II project has provided the theoretical background for almost all modern realtime control systems & modeling tools

Ptolemy II: „Actor Based Design“



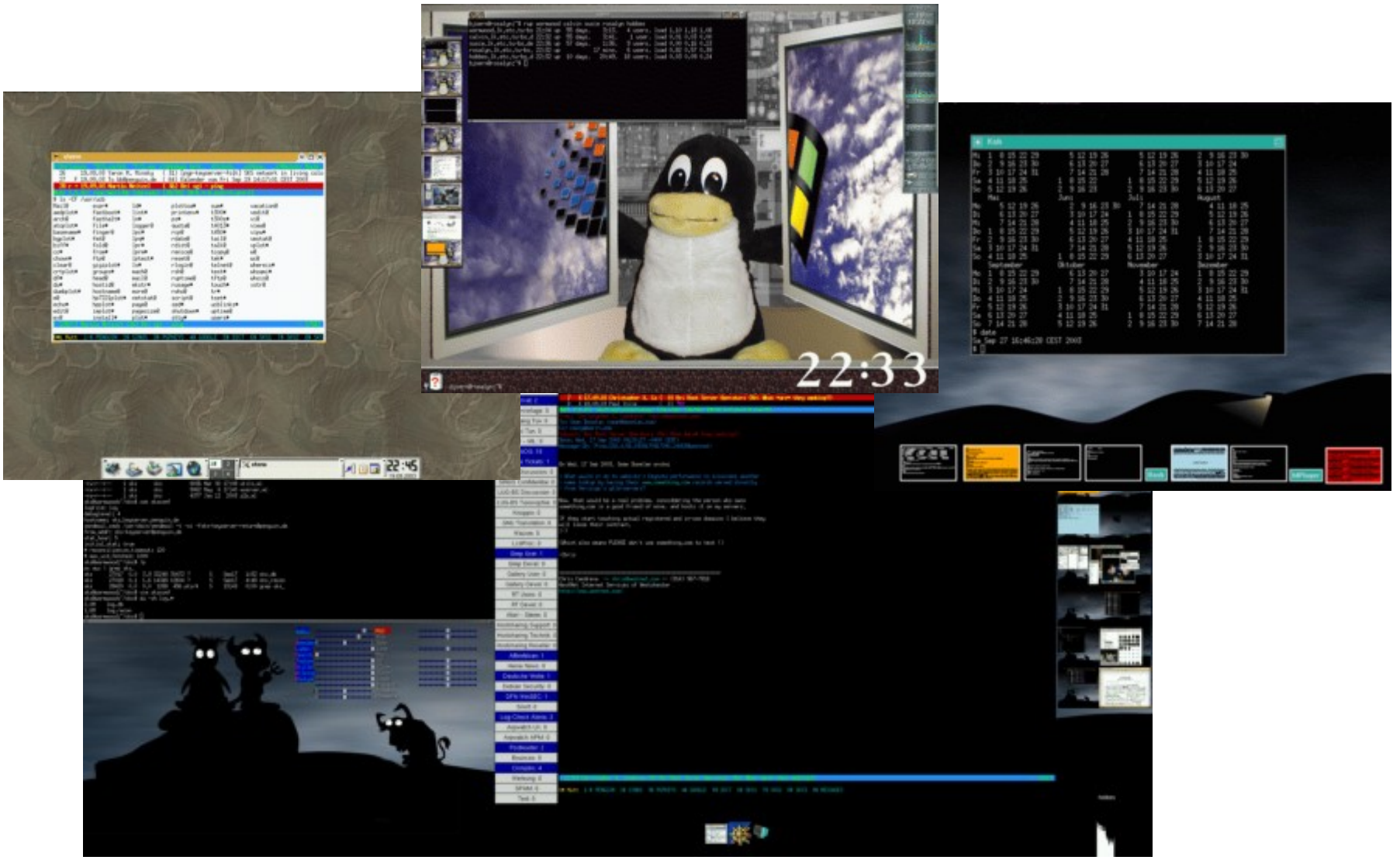
History @PTX: libpv-2.x

- Based on Ptolemy II, we started a prototype implementation
- Actor Based, no execution semantics in the actors
- Fully object oriented, but written in C
- The „network“ is the process image (with persistence)
- Connection == Mapping (actor code only knows it's local variables)
- Actor code is without semantics:
  - „PLC Director“ possible (static schedule: read/process/write)
  - Fully asynchronous event system possible
  - Lockless / Locking variants (independend of actor implementation)
  - Bridging to other modeling tools (LabVIEW, Matlab/Simulink)

Roadmap (Draft)

- Phase I:
  - Definition & Implementation of an Infrastructure Library
  - Merger of the 3S / Homag / Pengutronix Prototypes
  - Proof of Concept, based on selected & simple Use Cases
  - Milestone: Library Release 1.0 (for testing purposes)
- Phase II:
  - Call for Stack Implementation
  - Refactoring of the Infrastructure based on Feedback
- Phase III:
  - Graphical Configuration Tool for Mapping Layer

Thank you for listening! Questions ... ?



Pengutronix.