TINE Video System

Undergoing A Redesign

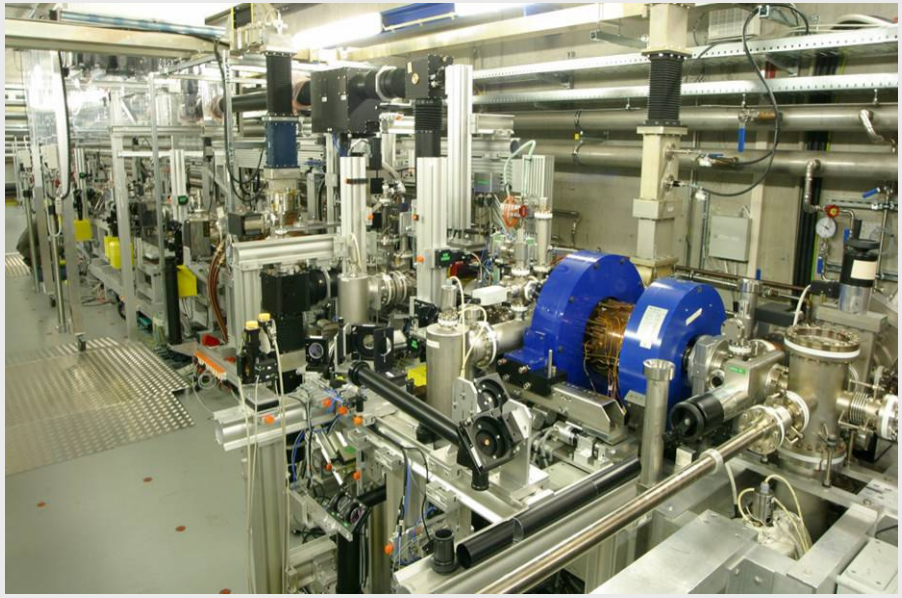
A Modular, Well-Defined, Component-Based and Interoperable TV System

Stefan Weisse DESY, Germany

PITZ Overview

Photo **I**njector **T**est Facility **Z**euthen (2002 on)

- test, condition and optimize sources of high brightness electron beams for future free electron lasers and linear colliders
- goal: intense electron-beam with very small transverse emittance and reasonably small longitudinal emittance
- goal is requirement for FEL operation



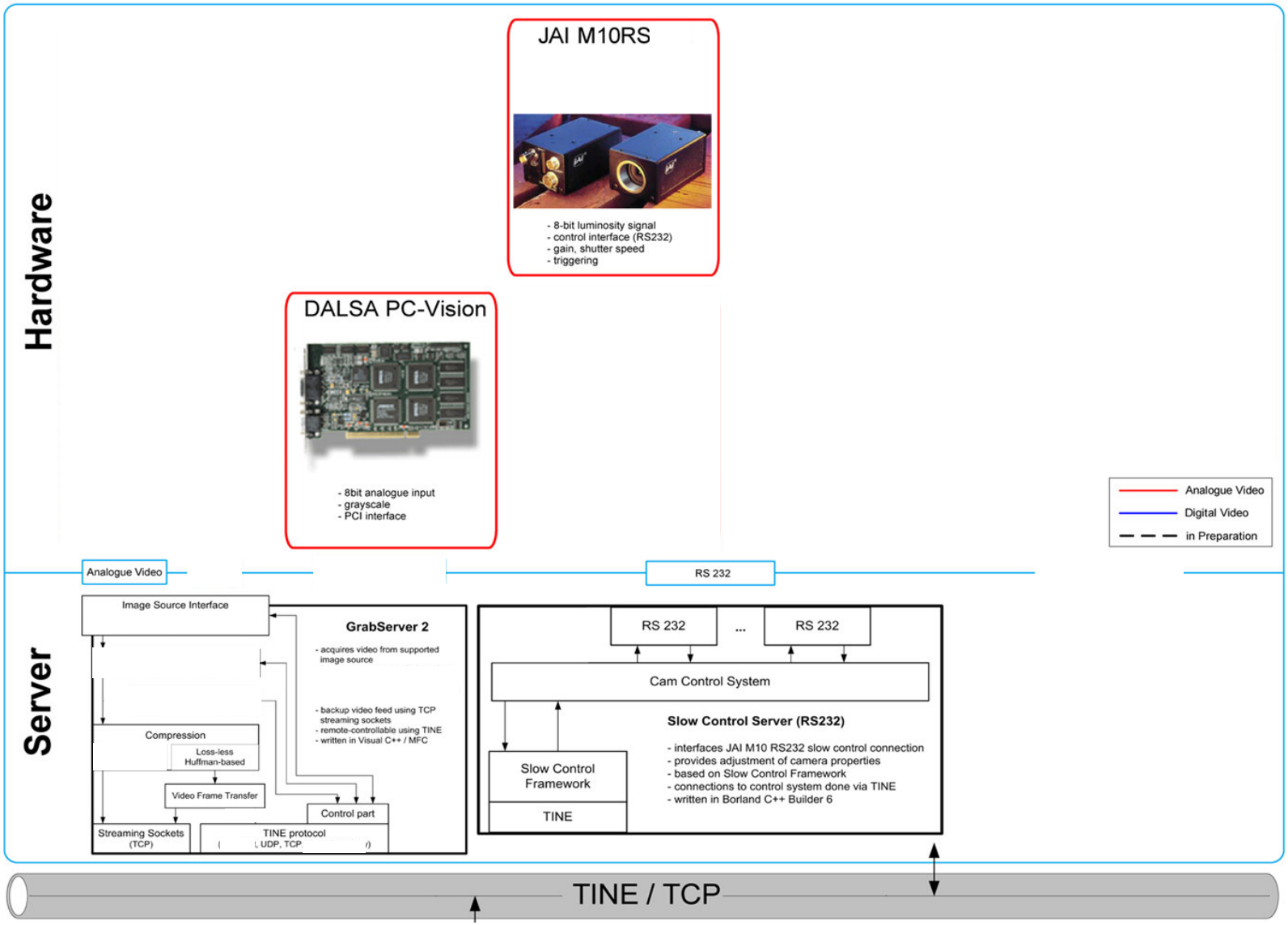
"The challenge of PITZ is the production of such beams with very high quality by applying the most advanced techniques in combination with key parameters of projects based on TESLA technology like the FLASH, the European XFEL and the proposed BESSY-FEL."

Video At PITZ
Field of Activity / Challenges

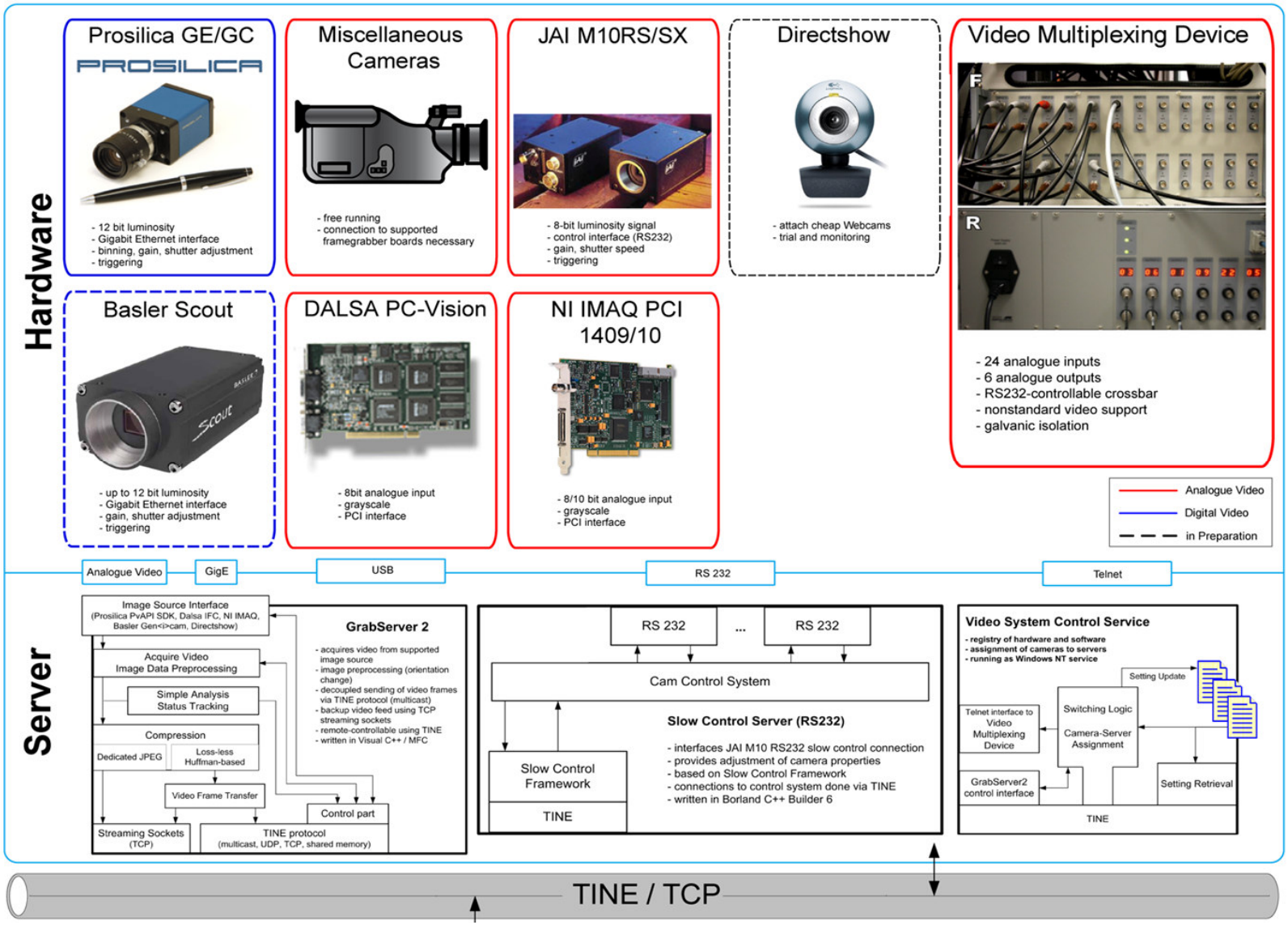
- **Field of Activity**
  - Acquire image sequences from hardware and provide a **lossless** stream via controls network
  - **Live** view of electron and laser beam position, size and shape
  - Support for semiautomatic and automatic measurements
  - Remote inspection
  - Optimize **image quality** at image acquisition level

- **Challenges**
  - test facility also **test bed**: experimental setups, a lot of **refactoring**
  - evaluation of optics, optical readout constructions, new devices
  - changes at [goals of] PITZ force changes in TV system and software
  - improvements in image readout generate **new physics demands** (things that were not possible before)
  - **special, nonstandard usage** at PITZ requires rare, custom, unique solutions
  - **hardware defects** because of radiation and tweaked use

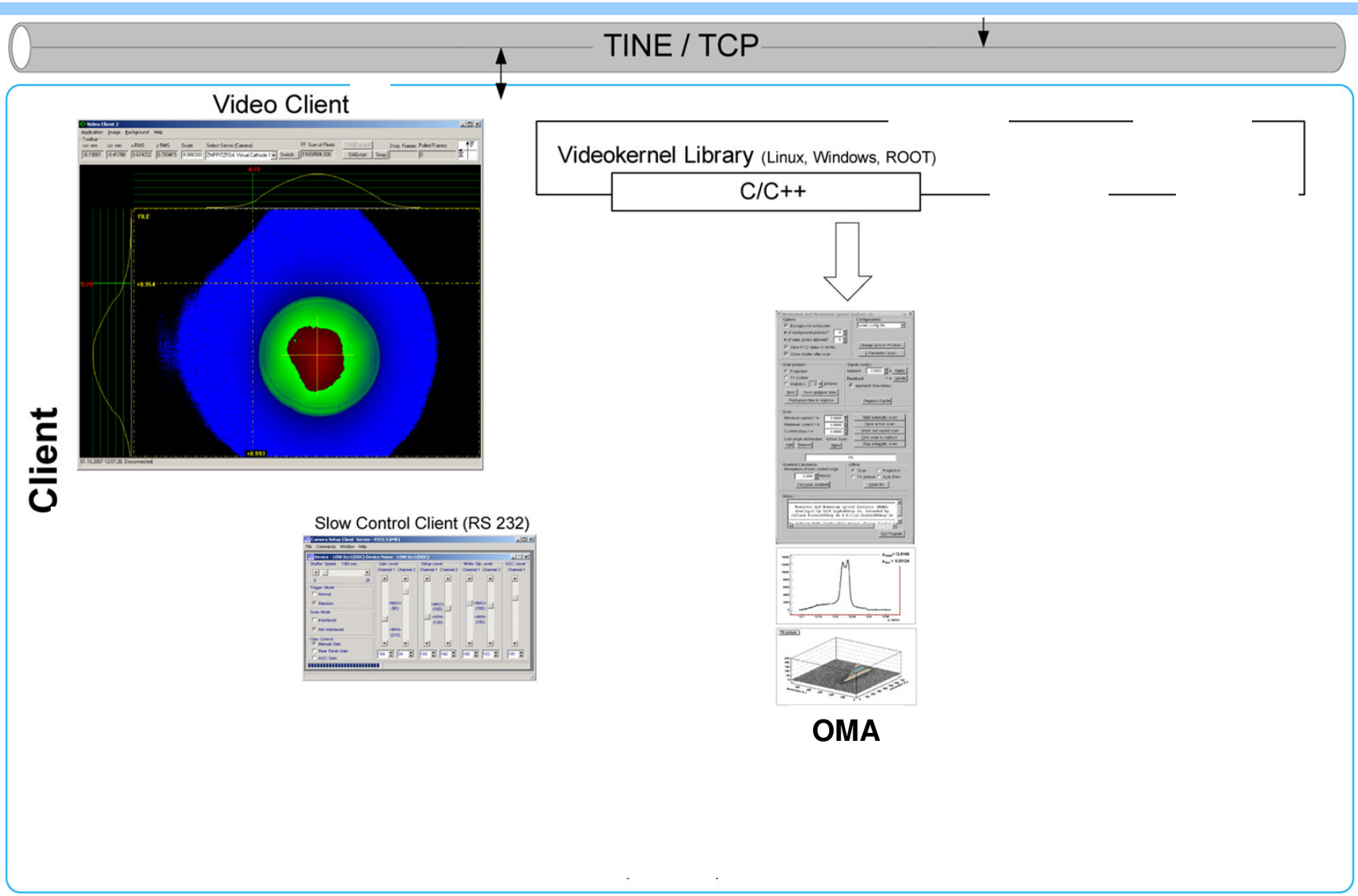
Hardware, Server 2002



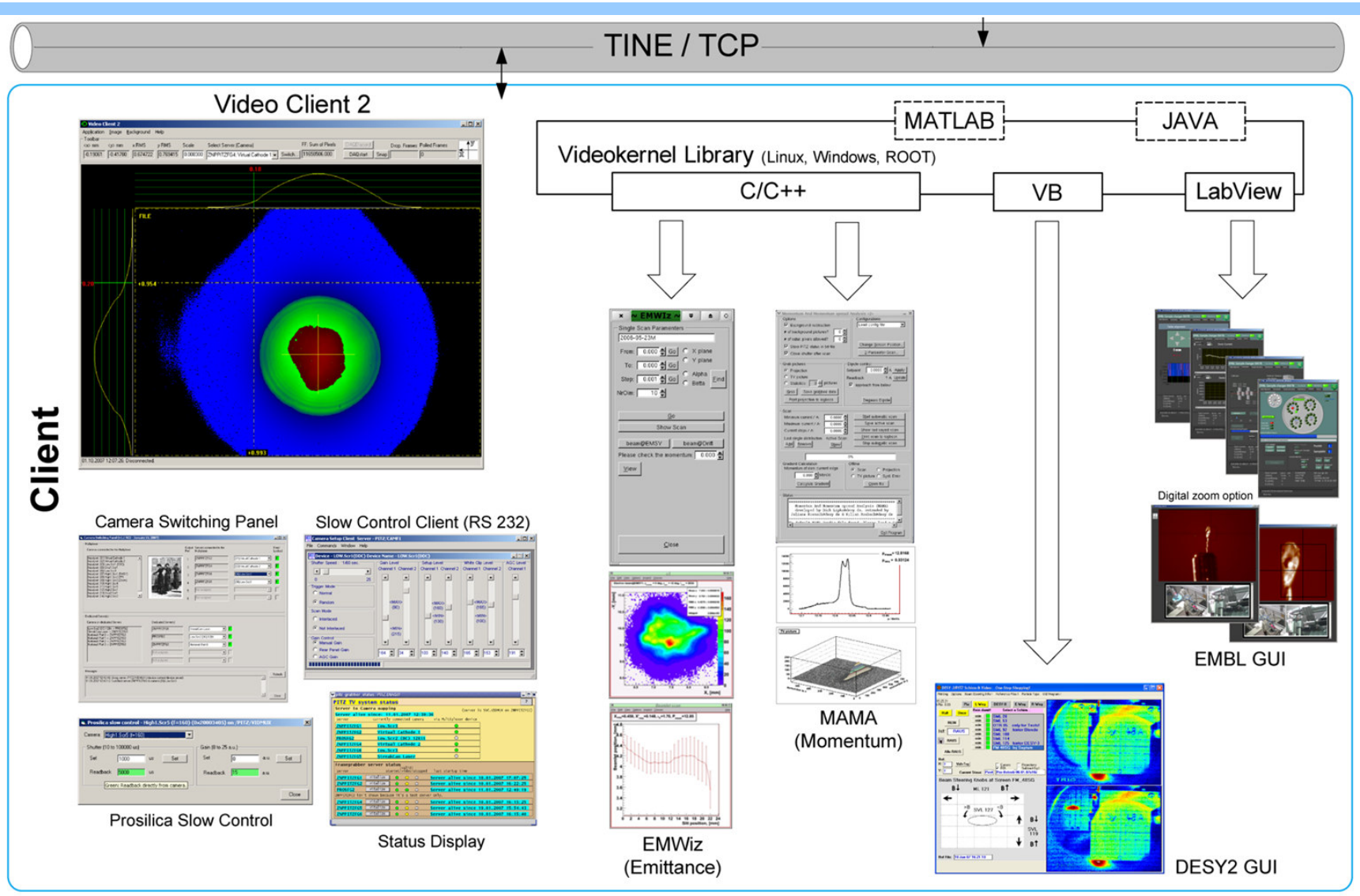
Hardware, Server 2007



Client side 2002



Client side 2007



Why Redesign?

- software was not designed for current tasks and usage
- simplistic, limited way of data transport
  - no colour data
  - no lossy transport
  - insufficient space for metadata
- no Java support
- inadequate use of standards for permanent data storage to file(s)
- integration of new imaging hardware more difficult than necessary
- core software tightly bound to Windows API, MFC and Microsoft development environment
- ‘reuse of software work’-guideline not followed from the very beginning

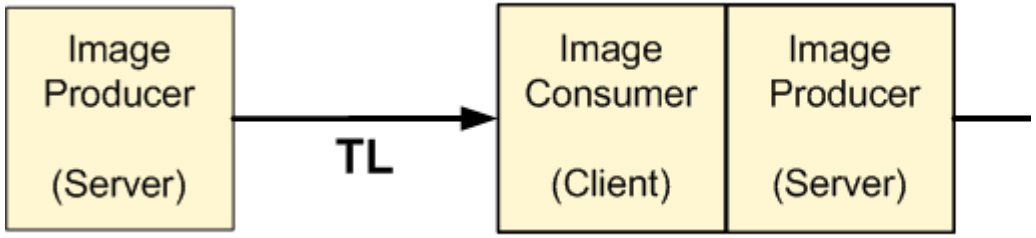
Redesign Focus

- easy software reuse
- flexible, open documented transport layer
- native Java support for client side
- abstract, modular image acquisition interface
- component-based design
- use of standard image formats for permanent data storage

Changes Already Performed
Software Reuse

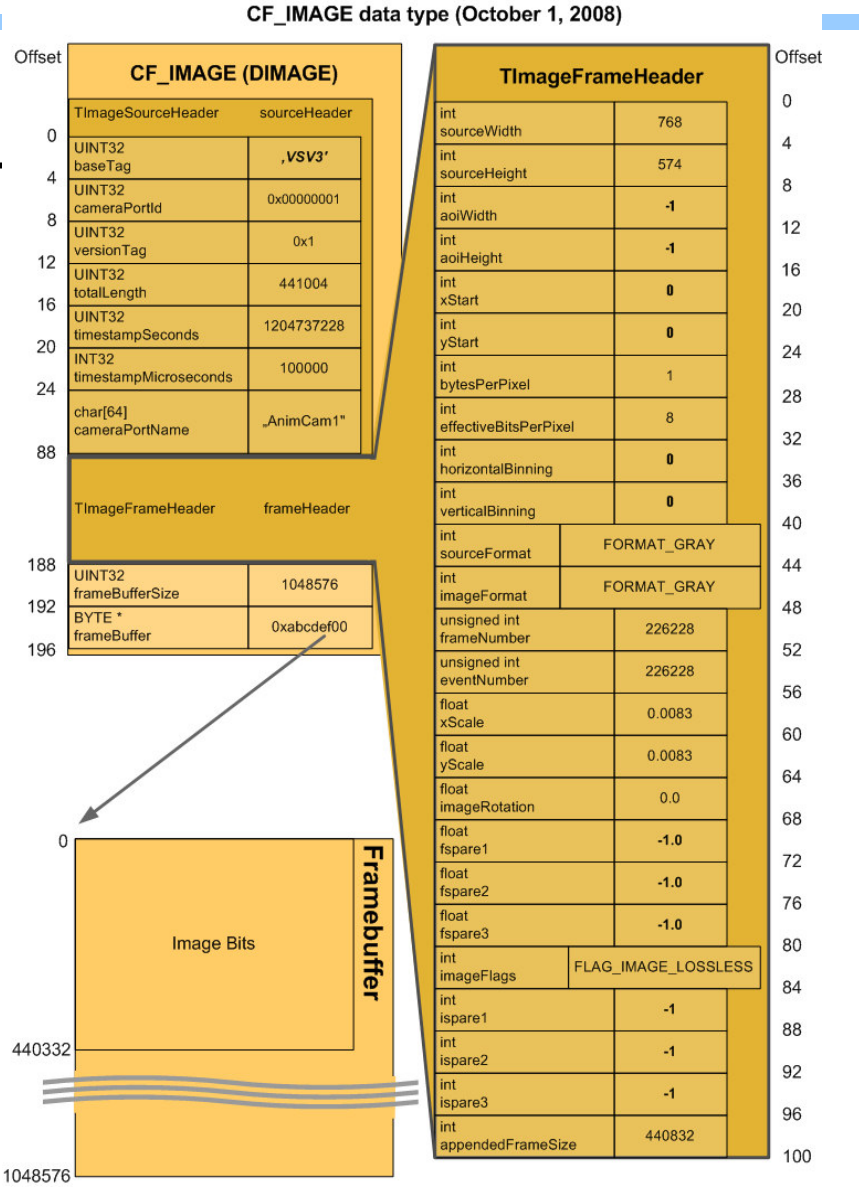
- philosophy of having most sourcecode platform- and compiler independent
  - new software components, parts and algorithms that use C/C++ as language are written in a platform and compiler-independent fashion
  - use ANSI C and ISO C++ where possible
  - platform-specific parts are wrapped, most modern platforms provide the same functionality, just a little bit different each time (threads, semaphores, sockets, …)
  - successfully tested at least on Linux 32 and 64 bits gcc and on windows 32 bit msvc6 and msvc8 (visual studio 2005)

Changes Already Performed
Video Transport Layer

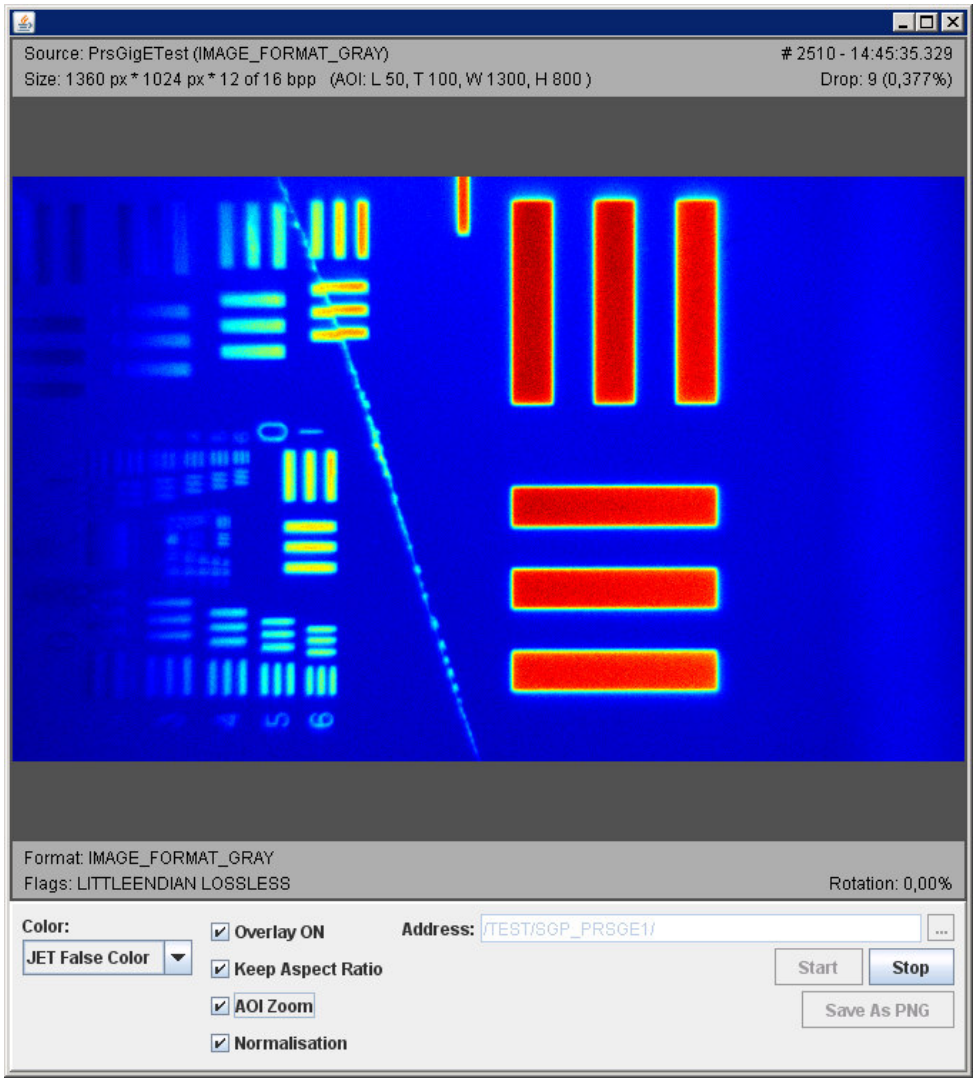


"How does the video data come from place A to place B?"

- transmission via TINE
- TINE datatype CF_IMAGE (DIMAGE) introduced to TINE API (C and Java)
- special mechanisms in TINE enhance efficiency and response time of 'image sequence' data transfer



Changes Already Performed
Java support / TINE Video Bean



- basic Live image consumer
- display enhancement options
- on-screen display
- RGB colour,greyscale,JPEG
- supports old Video System data feeds

- 100% native Java
- performance satisfactory

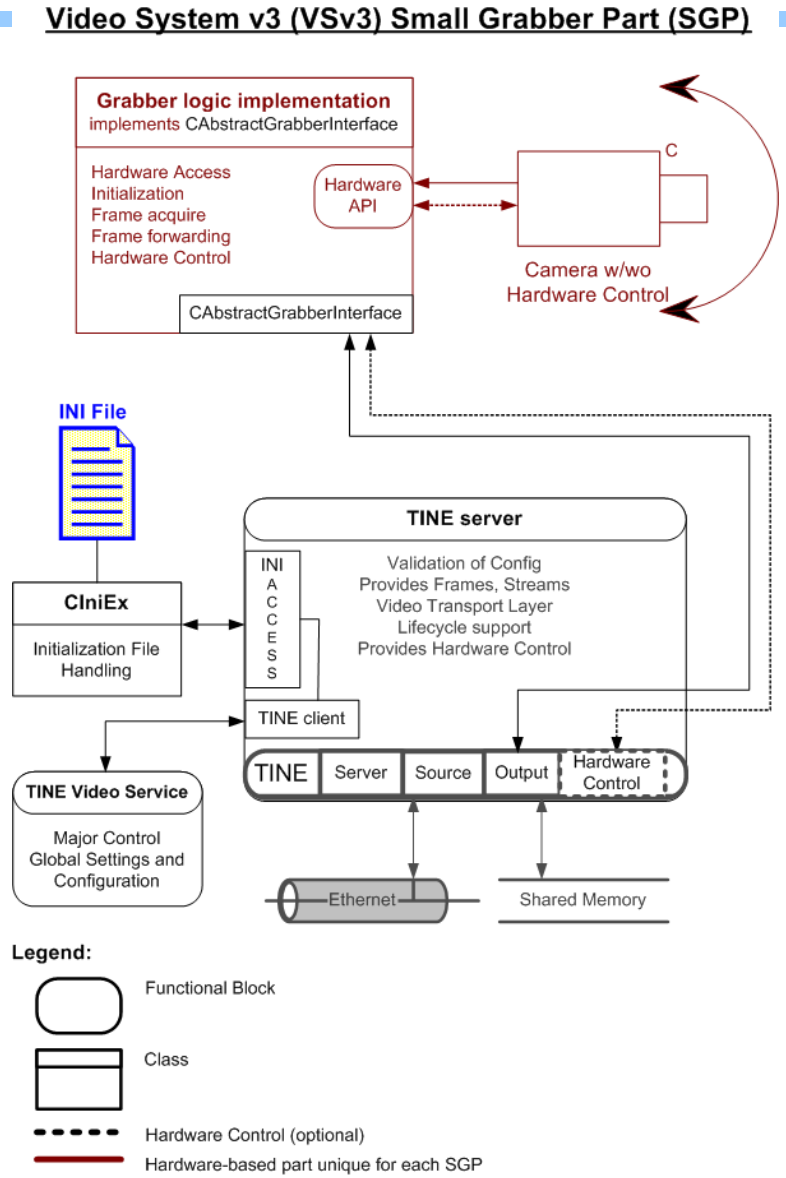
- integrated in TINE Acop Bean framework

Changes Already Performed
Small Grabber Parts

SGP: Hardware-independent Image Producer Framework

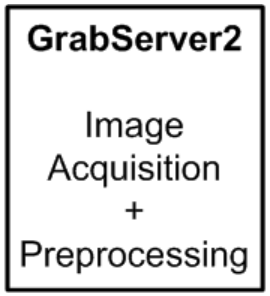
- acquires images from hardware
- provides control of hardware
- defined interfaces to hardware-dependent part and rest of Video System

- ready: 5+ unique interfaces



Changes in the Queue
CoreProvider Component

- past: image acquisition and image preprocessing was done in GrabServer2



- **but** image preprocessing is hardware independent pure software work

- functionality does not need to be implemented in each SGP but extracted and put to CoreProvider

Changes in the Queue
VideoService Component

- VideoService: Based on Video and Muxer Service (VIDMUX)
  - interface between outside (control world) and inside (video system)
  - abstraction level for access from outside world
  - make available, rearrange and lock image sources to providers/consumers
  - central control server
  - repository of cameras, camera ports, servers, settings

- Renovation
  - ‘facelift’ dated VIDMUX
  - properly represent new component-based nature
  - improve mechanisms of ‘image source requests’ by clients or components
  - rewrite parts of business logic (historically limited consistency)
  - extend possibilities of repository

Changes in the Queue
Standard Image Formats

- Former Formats
  - IMM, BKG, IMC, BKC
  - BMP (Windows BMP format)

- New Formats

PNG

(grayscale and colour data, lossless, supports lossless compression)

JPEG

(grayscale data as 8 bits per pixel, colour data, lossy, near-lossless possible)

  - both support metainformation tags for keeping image metainformation
  - both image formats are widely used and supported
  - for both formats open-source libraries are available

Near-Future Prospects

- Finish Main Changes
  - CoreProvider
  - VideoService
  - standard file formats
  - migration of interface library, creation of core shared library, extend user support by introduction of API

- further tests installations at PITZ
- rollout of new core setup at PITZ
- test installations at Petra III, LINAC2, EMBL

Far-future prospects

- Java support framework?
- Consider advanced installations at Linac2, Petra III, EMBL
- Intermediate components
  - DAQ integration
  - Save MPEG1/2 movies to disk
  - Gain of functionality by parts and clients written by users

Thank you for listening.

Questions? Comments?