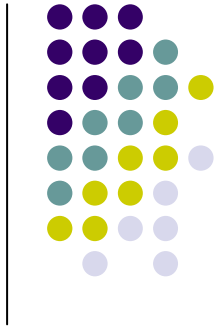
TINE Control System Overview and Status

P. Bartkiewicz, P. Duval, S. Herb, H. Wu (DESY/ Hamburg)

and

S. Weisse (DESY/ Zeuthen)

TINE: A Quick Tour



- Three-fold Integrated Networking Environment
  (->Keep your eye on the word "*Integrated*")
- Mature Control System
  - CERN Isolde Spin-off (~1991)
  - All the usual central services :
    - Archive, Alarm, Naming, Security, Logging, etc.
- Small Footprint
  - TINE Kernel written in C (just like your Operating System) or Java
  - Berkeley Sockets (NO SunRPC, CORBA, or other 3rd Party dependencies !)
  - Either Single-threaded or Multi-threaded mode
- Easy to Install
- High Performance
- Plug and Play
- Scalable to very large machines !

TINE @ DESY

HERA

- Large machine!
- 6.3 Km Proton-electron storage ring, collider
- Experiments at ZEUS, H1, Hermes, HeraB
- superconducting proton ring, RF cavities => QPS
- > 100000 control points

Principal Devices :

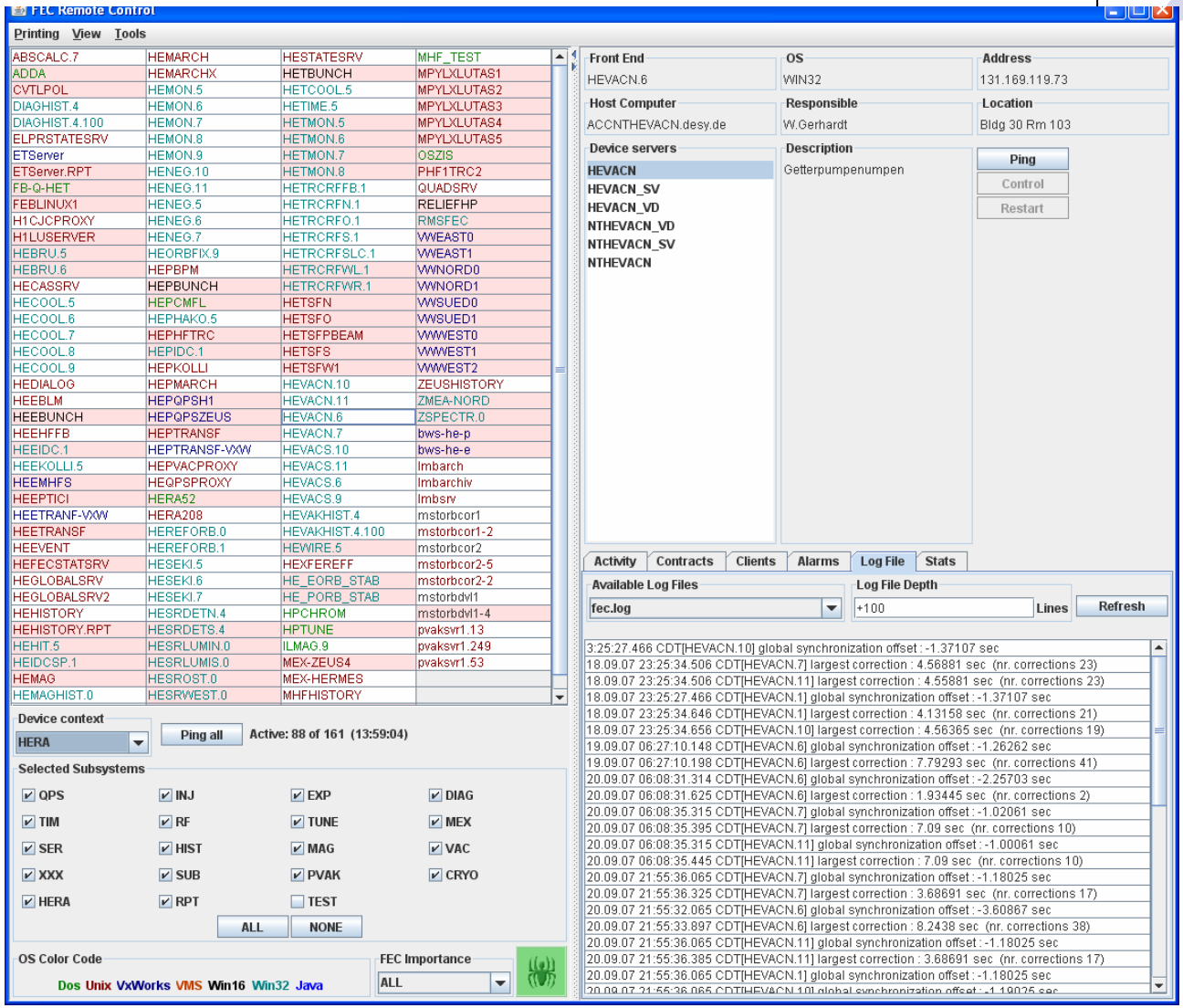
Device Type	Nr. of Units
Magnet PSCs	2000
RF Systems	230
Vacuum	3000
BPMs	800
Other beam measurement instrumentation	2000
Quench Protection System	2000
Other diagnostic instrumentation	1000
Air conditioning, water cooling	500

HERA FECS:

Shutdown on June 30, 2007

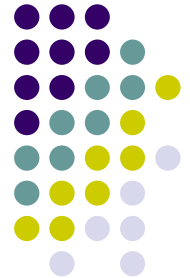
~160 Front-End Controllers (FECs)

Many Operating Systems !



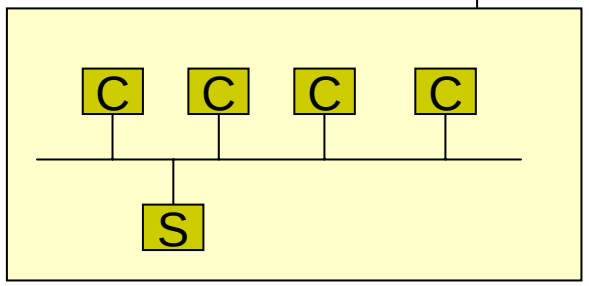
Control Systems101

Control Systems (one way or another) have to deal with …



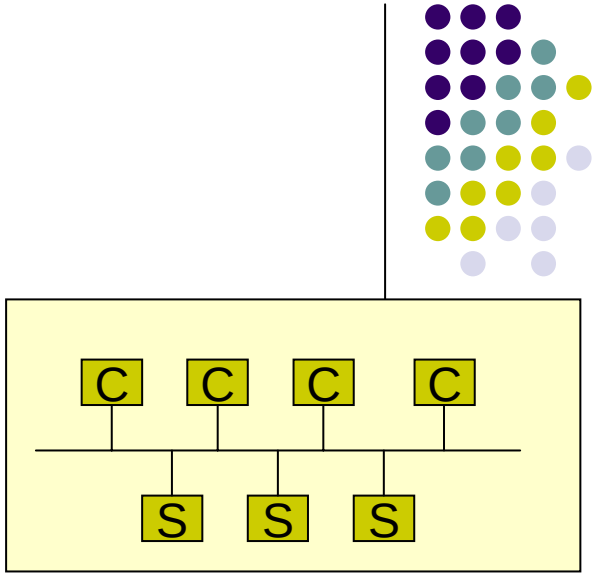
- **Distributed** end points and processes
- **Data Acquisition** (front end hardware)
- **Real-time** needs (where necessary)
- **Process control** (automation, feedback)
- **Central Services** (Archive, Alarm, Name Resolution, …)
- **Security** (who's allowed to do what from where?)
- **States** (Finite State Machines, sequencing, automation…)
- **Time synchronization** (time stamps, cycle ids, etc.)
- **Databases** (configuration, machine data, post-mortem data, …)
- **Statistics** (control system itself, operation, …)
- **Logging** (central, local, application, …)
- **Data transport** (data flow, *control system protocol*, scalability)

TINE Protocol (1)



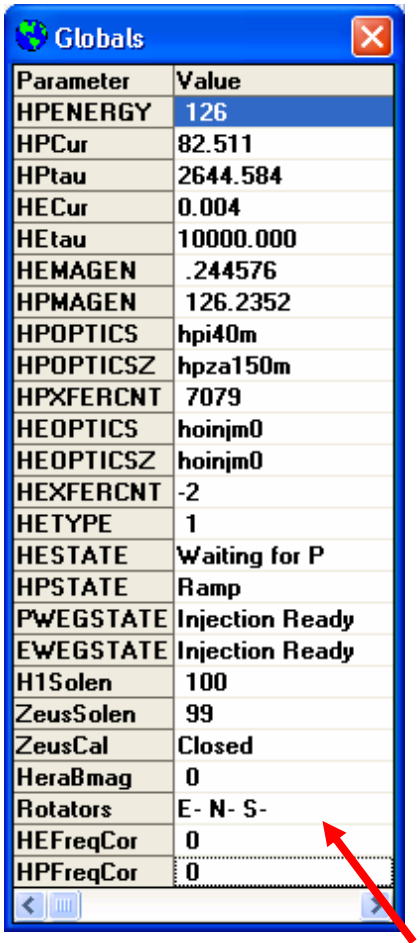
- Client-Server (classic)
  - Transaction based
  - Synchronous data access only
  - The “**N-Client**” Problem ? (do threads help?)
  - ExecLink(“/HERA/BPM/WL167MX”,”ORBIT.X”, …)
- Publisher-Subscriber (nearly classic)
  - **Connection Tables !**
  - Synchronous/Asynchronous data access
  - The “**10N-Client**” Problem ?
  - AttachLink(…, DATACHANGE, 100, linkCb)
  - Callback events !

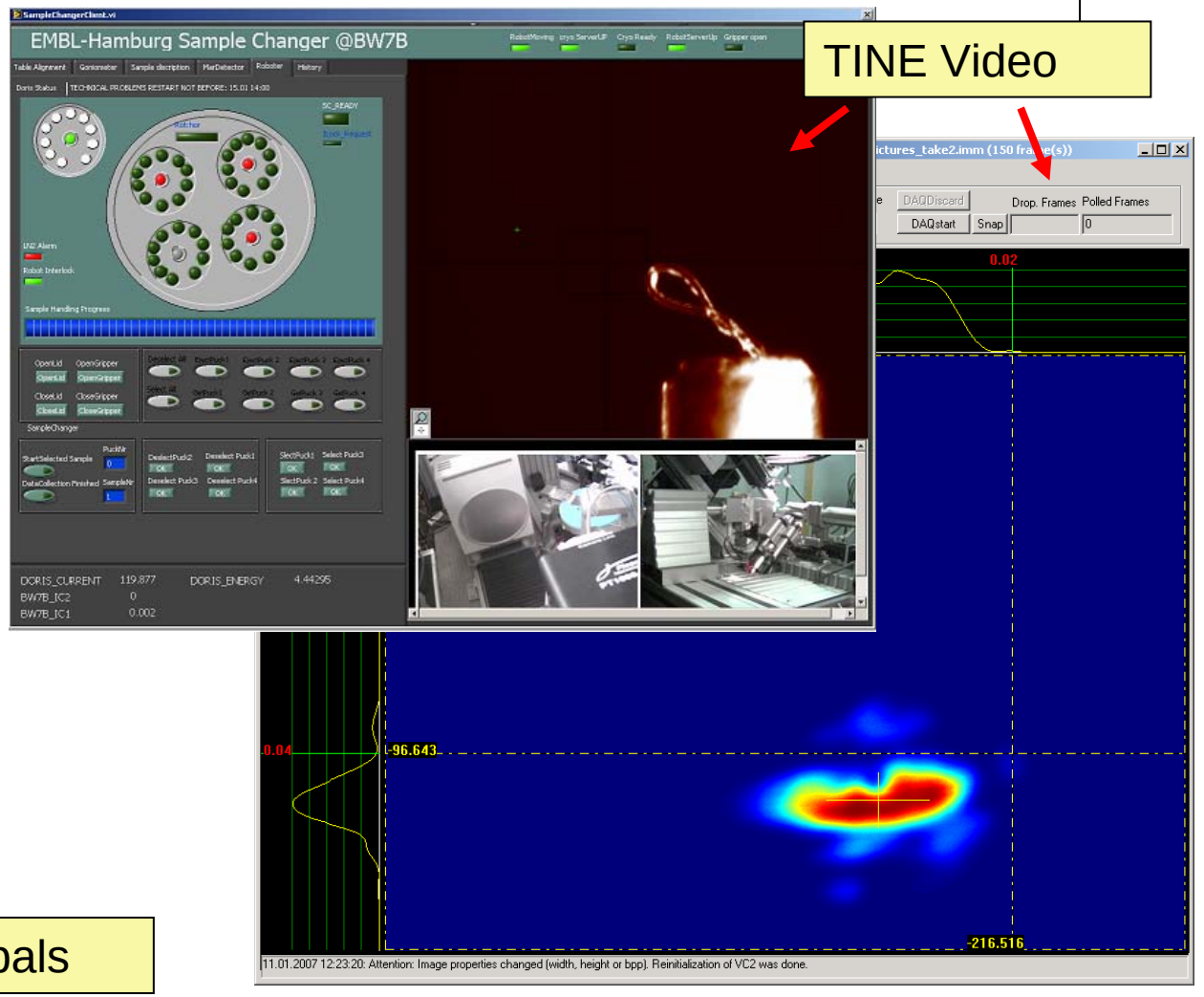
TINE Protocol (2)



- Producer-Consumer
  - Asynchronous data messages (Multicast)
  - The “**N-Producer**” Problem ?
  - AttachLink(“HPENERGY”,…, RECEIVE, 1000, linkCb)
- Publisher-Consumer (Producer-Subscriber?)
  - Like Publisher-Subscriber but:
  - Multicast group is a single connection Table entry
  - **N = 1 !!!**
  - AttachLink(…,DATACHANGE|NETWORK,100, linkCb)

TINE Protocol (2)





TINE Globals

TINE Data Types

- Primitives (byte, short, int*, long*, float, double, char*)
- Fixed-length Strings (“NAME16”, “NAME32”, “NAME64”, …)
- Doublets (FLTINT, “INTINT”, “DBLDBL”, “NAME32INT”, …)
- Triplets (“FLTINTINT”, “NAME16FLTINT”, …)
- Quadruplets (“FLTINTINTINT”, “INTINTINTINT”, …)
- Specials (“USTRING”, “UNAME”, SPECTRUM, VIDEO…)
- Bitfields (new to Release 4.0):
  - CF_BITFIELD8, CF_BITFIELD16, CF_BITFIELD32, CF_BITFIELD64
    - Data type: DBITFIELD
    - Bitfield Registry

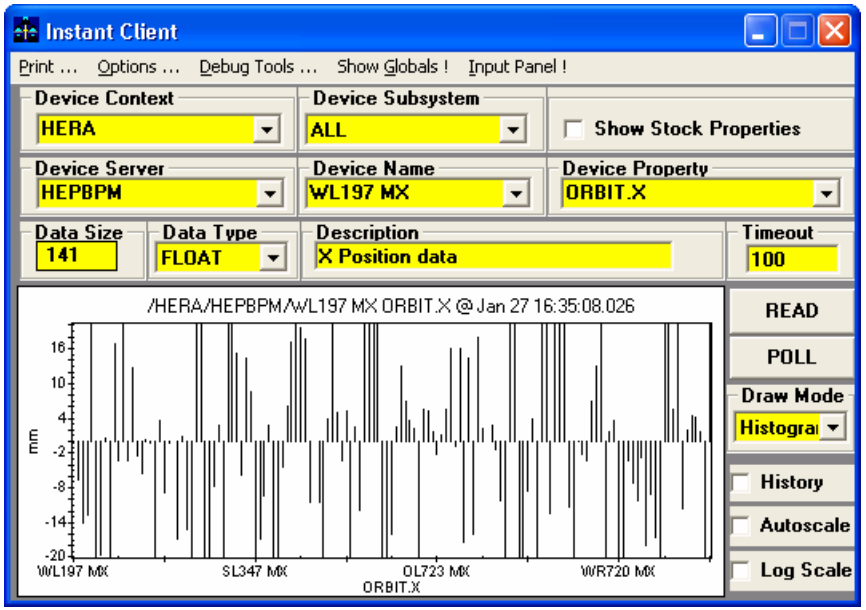
- **User-defined “Tagged Structures” !!!**
  - Structure registered at both ends (client, server)
    - Client can ‘discover’ structure fields !

*Platform dependent

TINE Naming Convention

Engineer's View or Machine Physicist's View

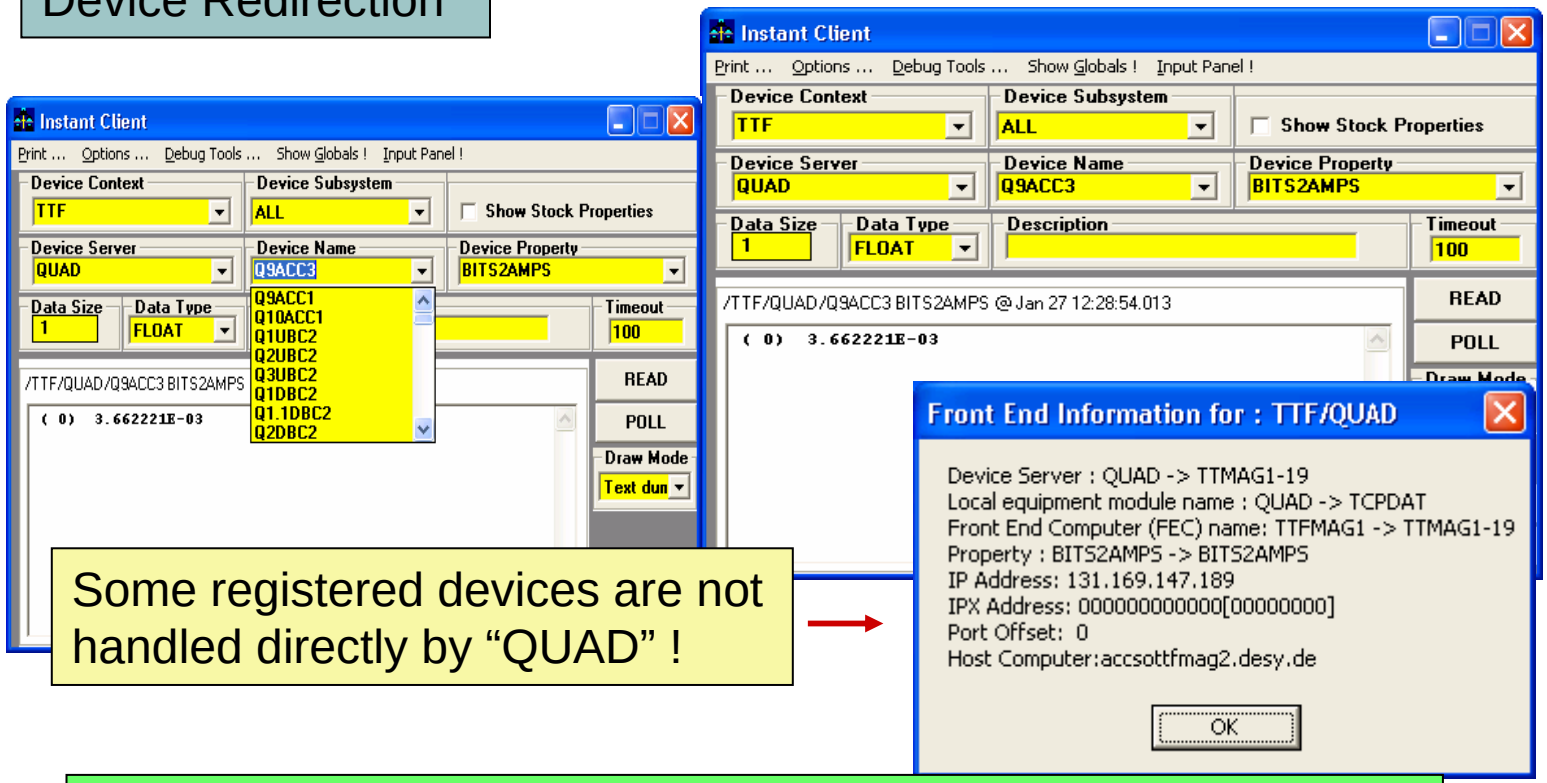
- Hierarchical
- Device is specified by :
  - **Device Context** (*Facility* in DOOCS, *Domain* in TANGO)
  - **Device Server** (or **Group**) (*Device* in DOOCS, *Family* in TANGO)
  - **Device Name** (*Location* in DOOCS, *Member* in TANGO)
- Data endpoint is specified by:
  - **Property** (*Property* in DOOCS, Attribute/Command in TANGO)
    - Are really 'methods' or 'calls'



Note: **Device Subsystem** is not part of the name space, but is a browseable element!

TINE Address Redirection

Device Redirection



Some registered devices are not handled directly by “QUAD” !

Wildcard calls (e.g. /TTF/QUAD/*/BITS2AMPS) still work !

Property Redirection (e.g. “Orbit.X.ARCH” -> central archive server)

TINE ENS: Plug and Play

I want to be known to the system as "**BPM.P**" in the context "**HERA**"

Does BPM.P already exist for HERA?

-> **Yes** : Is same address as already registered?

-> **Yes**: **Accept** and increment reboot count

-> **No**:

Is the currently listed BPM.P for HERA running ?

Yes : **Refuse** and send "in-use" message

No : **Accept** and update database

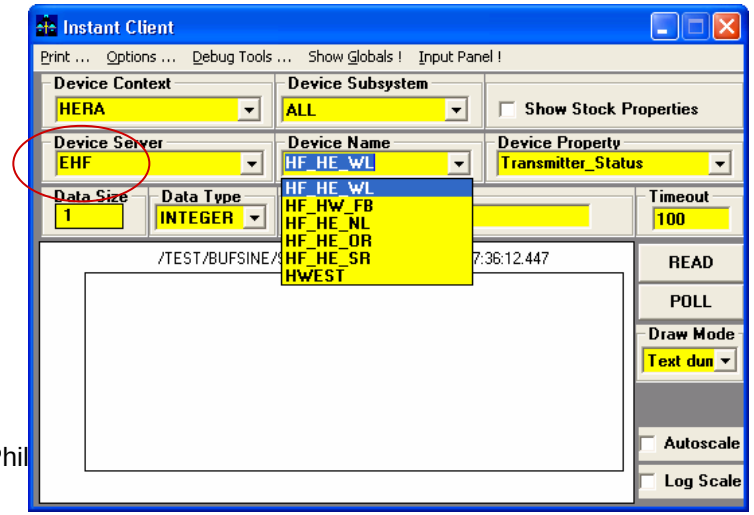
-> **No** : **Accept** and update database

Equipment Name Server (ENS)

- Forward accepted requests to secondary name servers

Plug and Play (joining a group)

I want to be known to the system as “**ERF.WL**” in the context “**HERA**” and join the group “**EHF**”



- ERF.WL registers with ENS as before
- ERF.WL registers group EHF with GENS

  Does group EHF exist ?

  Yes:

  Is ERF.WL a member?

  Yes: Update device list if different

  No: join group

  No:

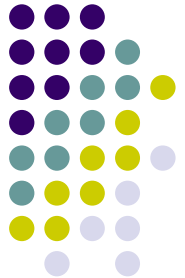
  Create Group and register Group as Server in Context HERA with the ENS

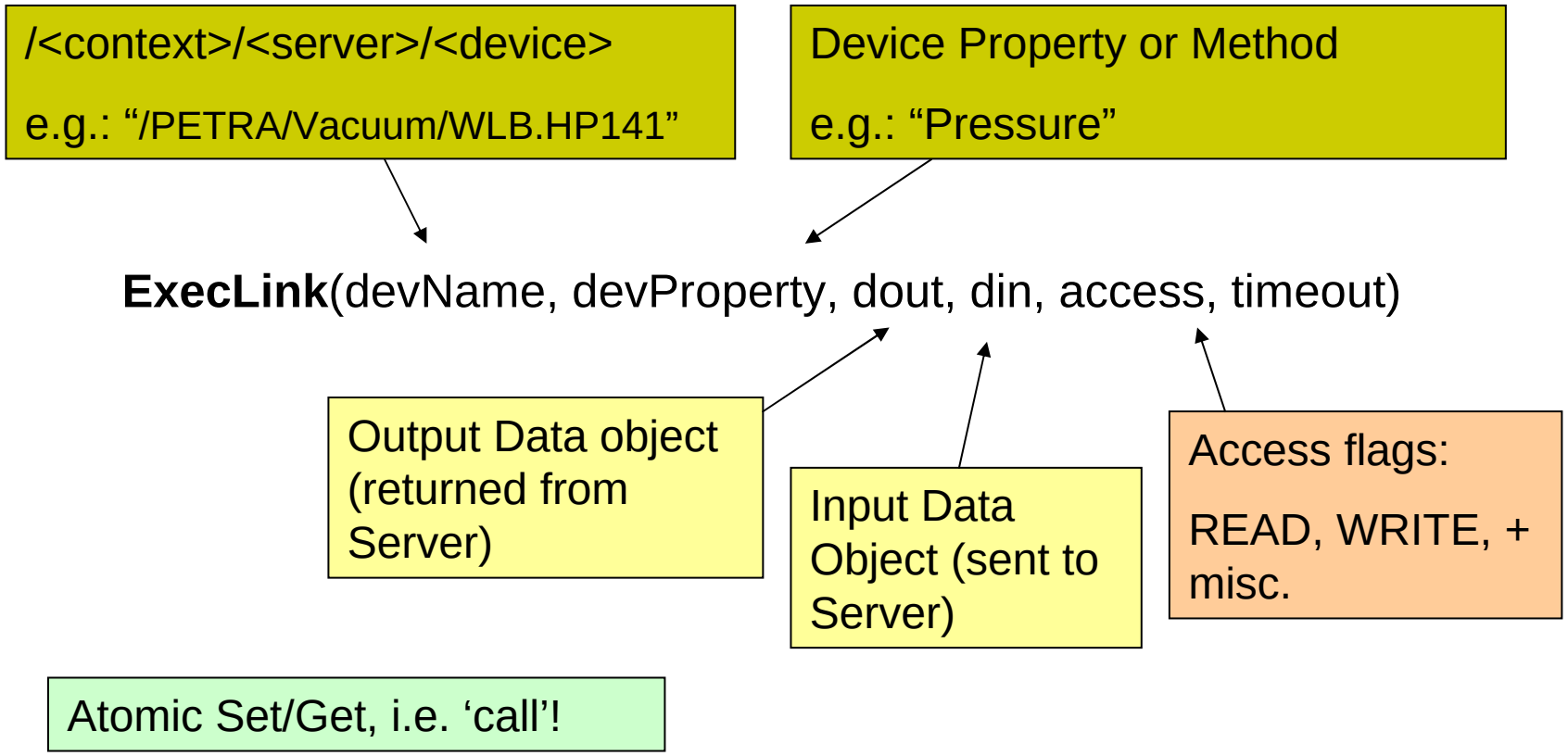
- Clients see a “Server” called “EHF”
- Selected Device is redirected to the appropriate physical server.

TINE Client API …

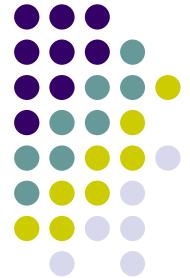
- Fundamental API is :
  - Link based and Not Channel based !
  - Narrow Interface
  - **NOT** ‘get’, ‘set’, and ‘monitor’ !!!
  - Think of ‘calls’ a la RPC or RMI …
  - Synchronous data acquisition
  - Asynchronous data acquisition
    - Callbacks, events

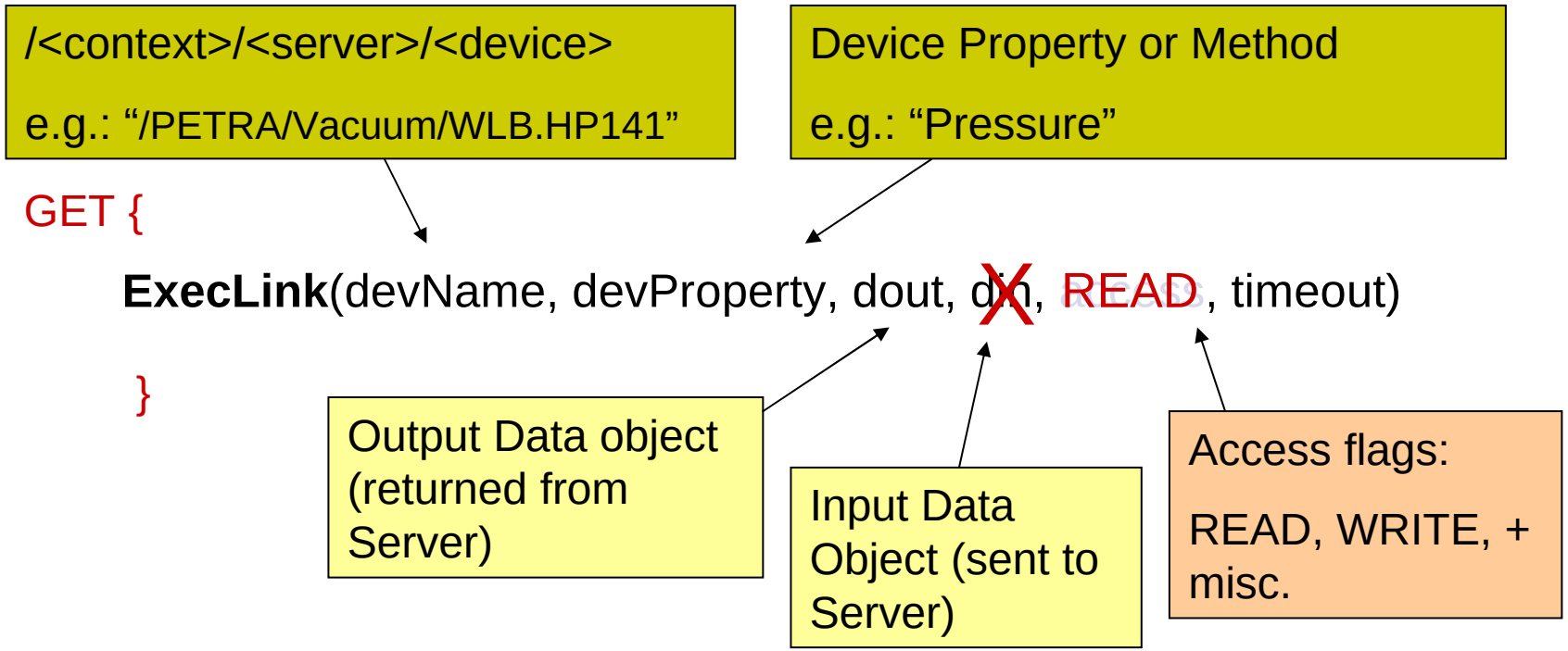
Client API: Synchronous Calls



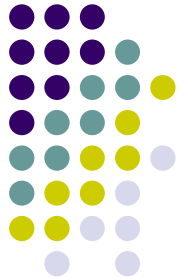


Client API: Synchronous Calls





Client API: Synchronous Calls



e.g. A call to the orbit correction server:

din = tagged struct with optics, current orbit, beam parameters;

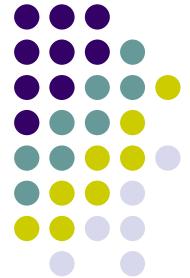
dout = tagged struct with new optics, projected corrected orbit;

ExecLink(“/SERVICE/ORBCOR”, “EFFCOR”, dout, din, READ, 1000)

e.g. “Command” Properties: “RESET”, “INIT”, “START”, …

ExecLink(“/HERA/Transfer/P”, “STOP”, NULL, NULL, WRITE, 1000)

Client API: Asynchronous Calls



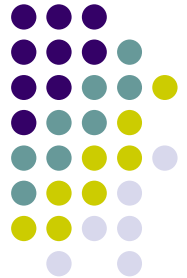
Analogous to synchronous parameters …

AttachLink(devName, devProperty, dout, din, access, pollrate,
void (*callback)(int,int), callbackID, mode)

Callback with callback id
and status code …

DATACHANGE
TIMER
SINGLE
EVENT (scheduled)
NETWORK
GROUPED
WAIT
+ …

TINE API
(Application Programmer's Interface)



- C, C++
- VB
- ActiveX
- Java
- C#, VB.NET (rudimentary, but more coming soon!)
- Command line scripts
- Python Bindings
- Plus …

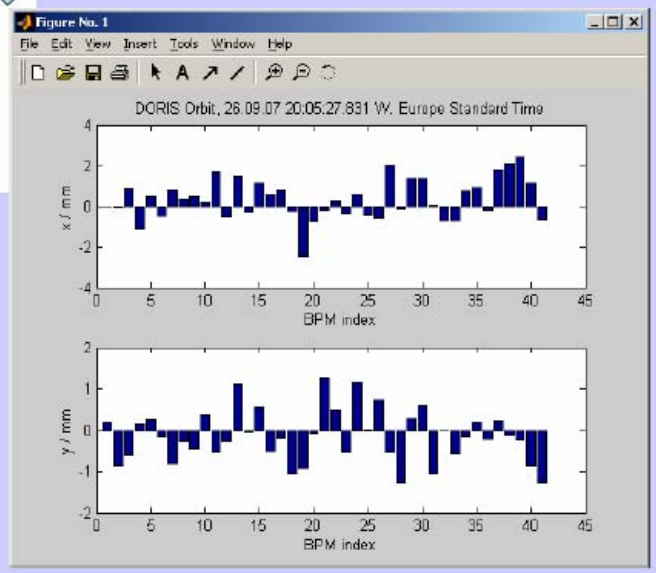
TINE and MatLab …

Example: DORIS Orbit

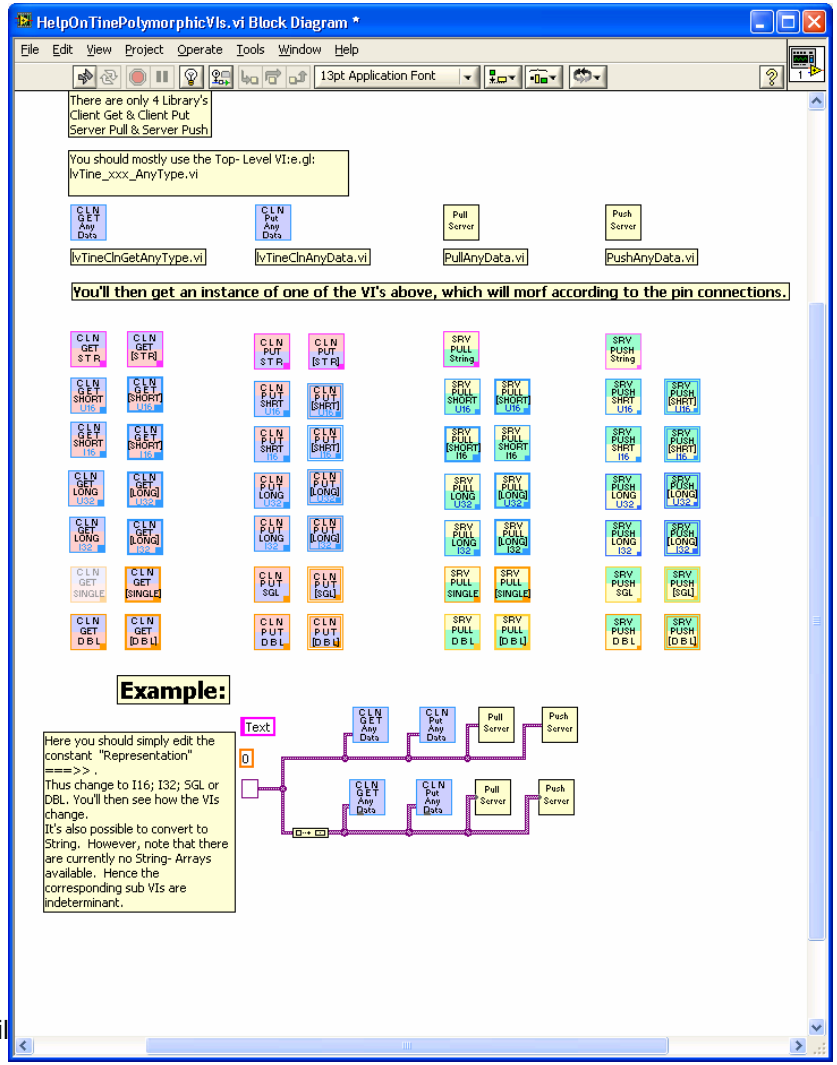
Simple script (“M-file”) to read and plot the DORIS orbit

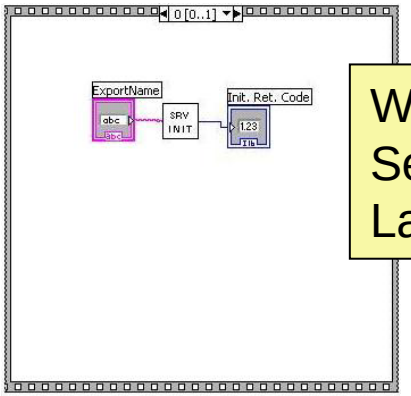
```
val = tineread('/DORIS/DOORBIT/#0      [ORBIT]');

figure(1)
subplot(2,1,1)
bar(1:41, val.ORBIT(2:42))
xlabel('BPM index')
ylabel('x / mm')
title(['DORIS Orbit, ' val.timestamp])
subplot(2,1,2)
bar(1:41, val.ORBIT(44:84))
xlabel('BPM index')
ylabel('y / mm')
```

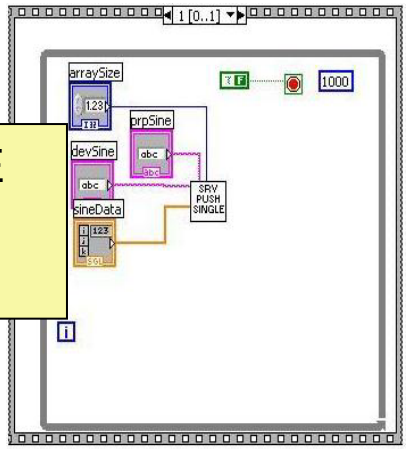


TINE and LabView ...



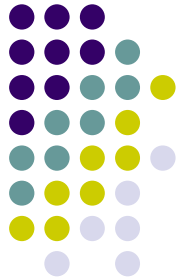


Write TINE Servers in LabView !



Write TINE Clients in LabView !

TINE Device Layer



- “Do it yourself” + your hardware API
  - Use those Windows drivers your hardware comes with!
- EPICS IOCs (asyn drivers) + Epics2Tine
- LabView VIs + TINE LabView
- DOOCS + DOOCS API

- **CDI** (Common Device Interface) !!!
  - Bus plugs for CANOpen, SEDAC, RS232, SiemensPLC, TwinCatADS, Libera, … (asyn?)
- **TICOM** (TINE CanOpen Manager) WPPB21

TINE Archive System

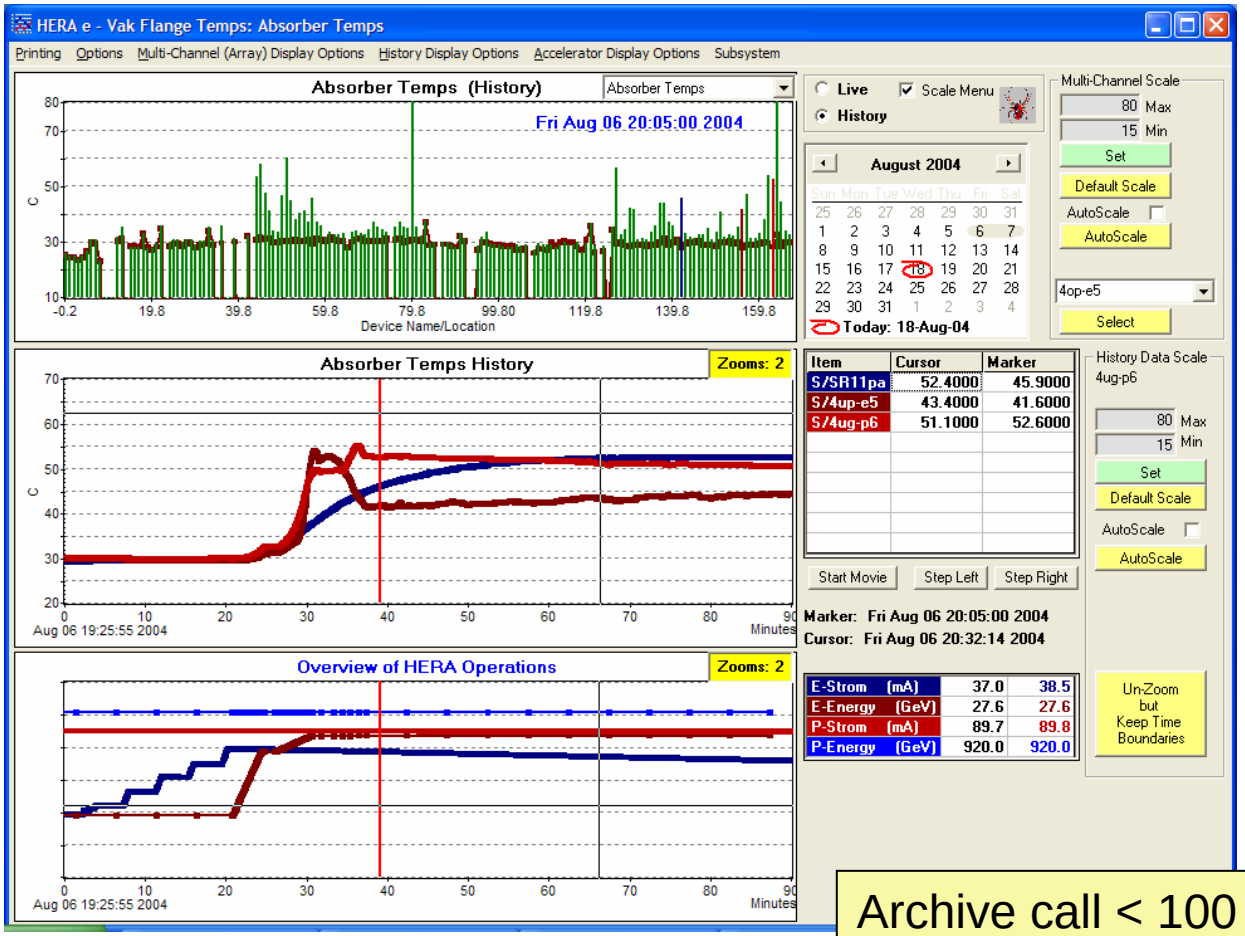
(lickity split data retrieval !)

Example: Multi-Channel Analyzer:

All temperatures as “snapshot” (vs. selected reference)

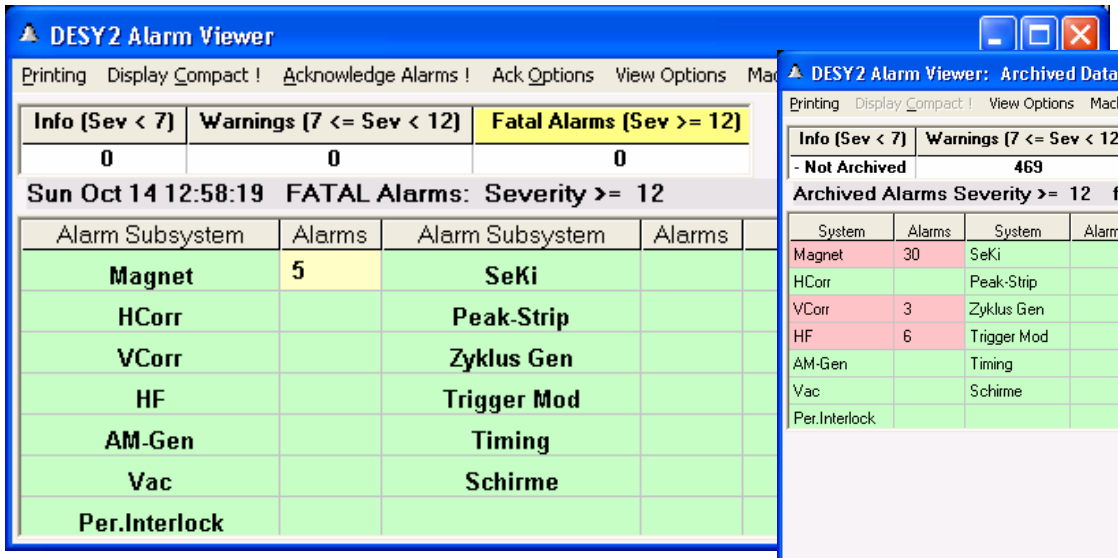
Histories of selected sensors:

Histories of machine operation parameters:



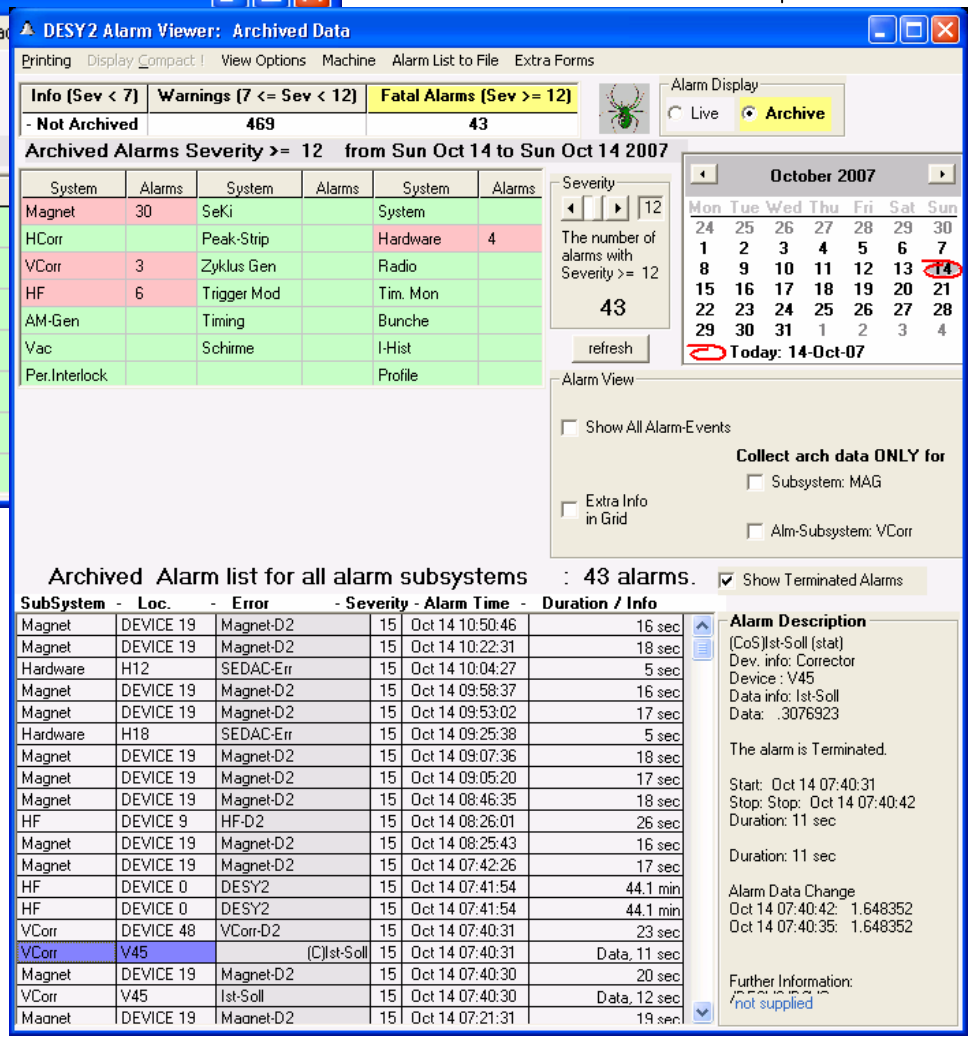
Archive call < 100 msec

TINE Alarm System



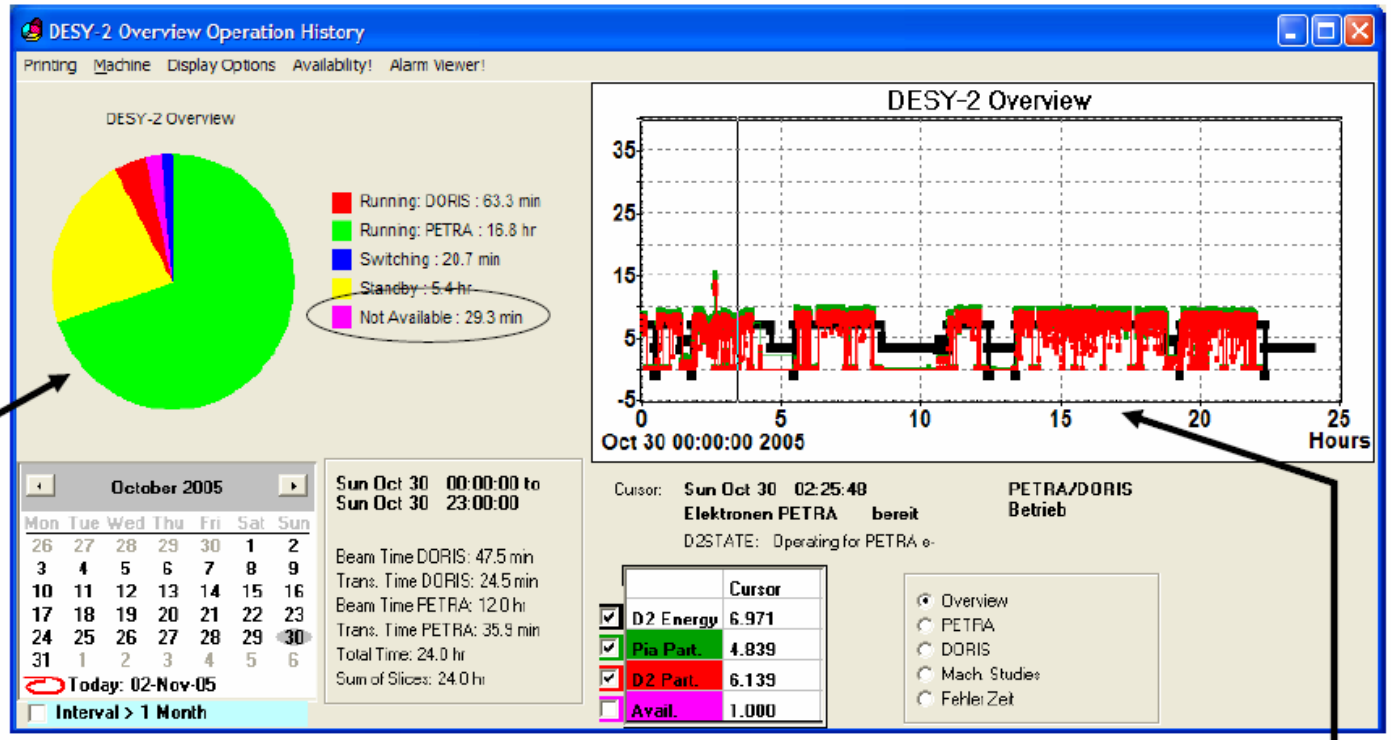
Compact Overview

Detailed view



Operations + Availability

Operations Overview: a Typical Day at DESY-2



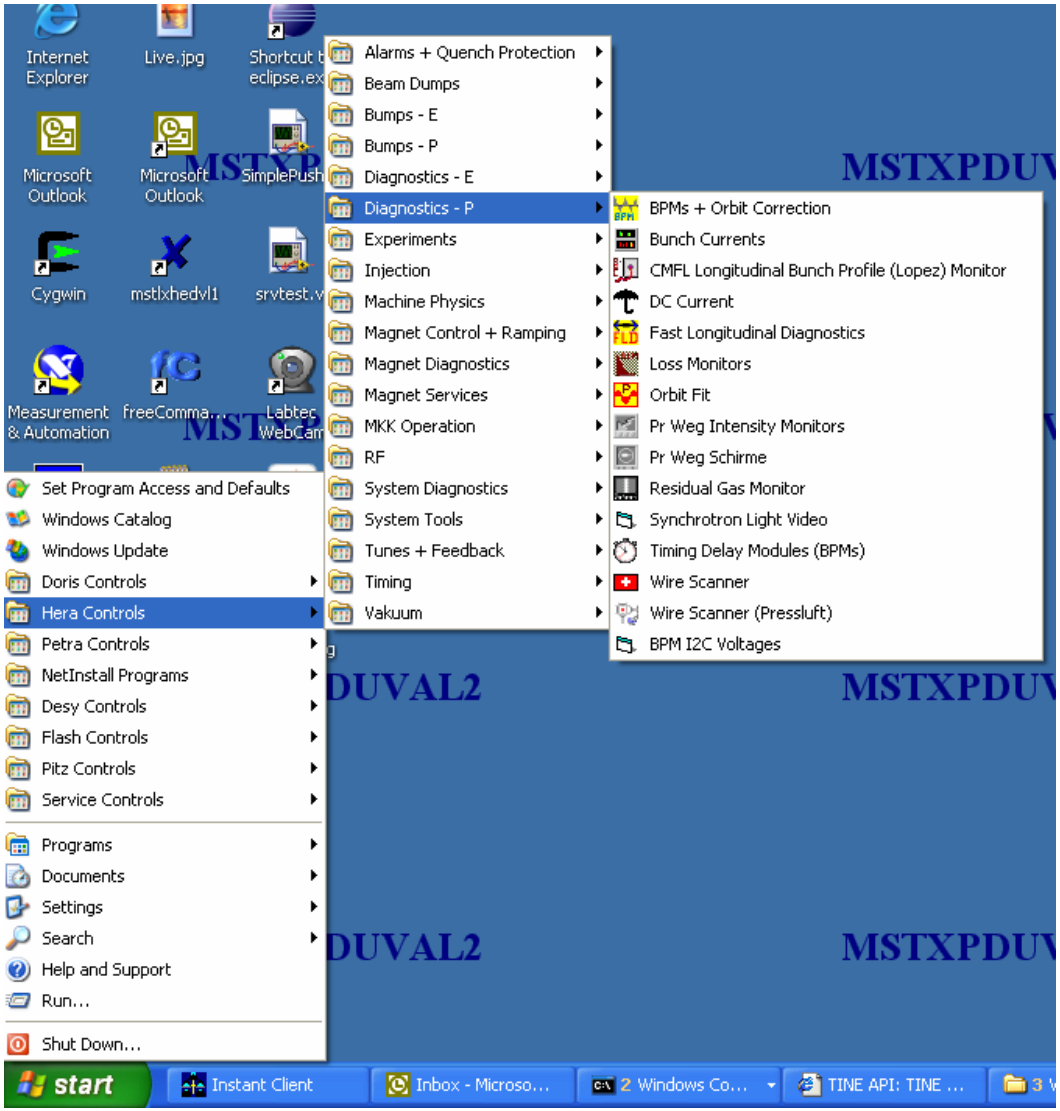
State Information (running mode) during 24 hours

Overview of beam current and energy during 24 hours

F

HERA Console Applications

(rich clients)

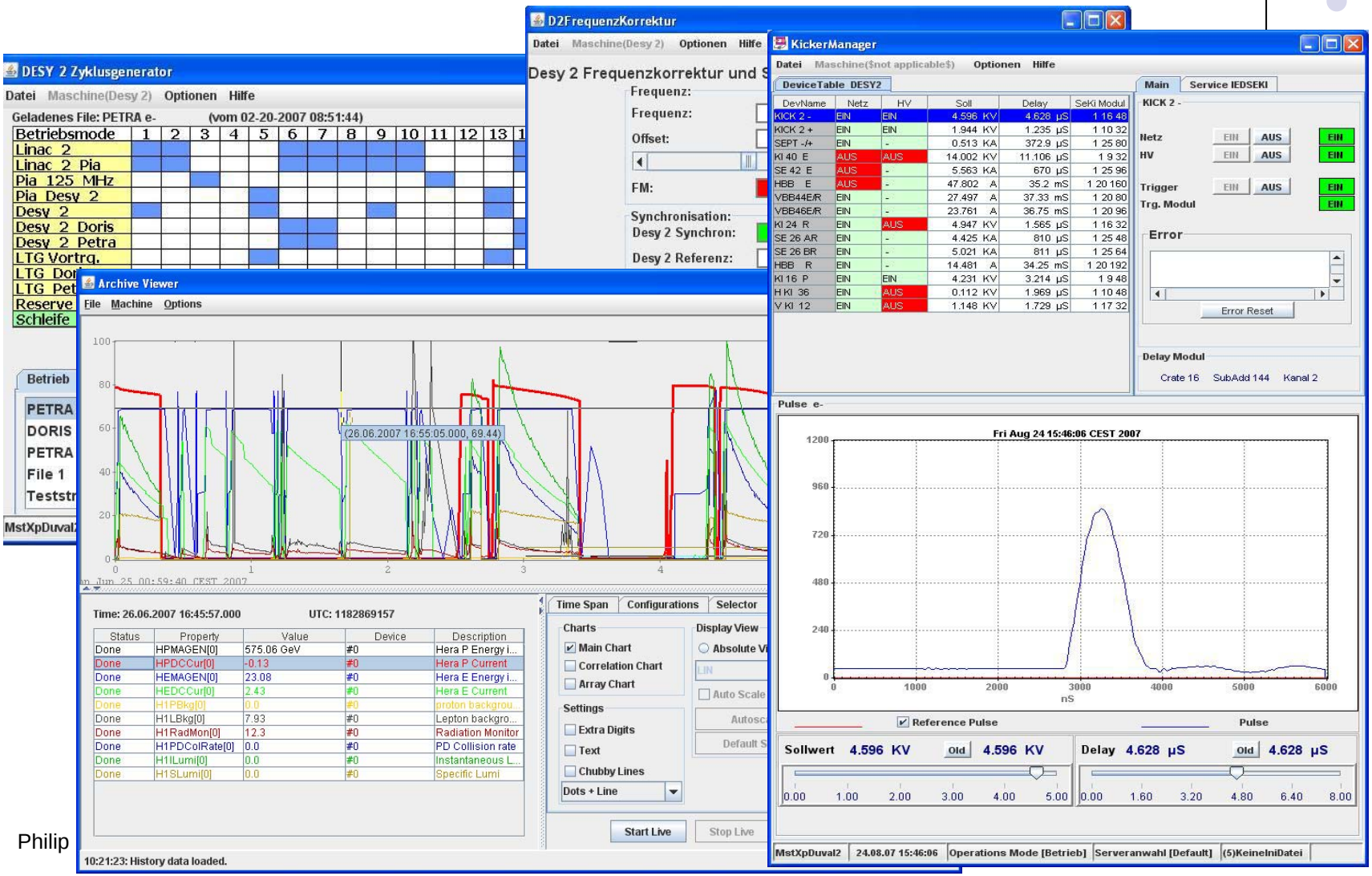


< 200 Console Applications available

~ 20 Console Applications needed for Normal Operation

Console Applications generally “Rich Clients”

TINE Clients : Rich Clients



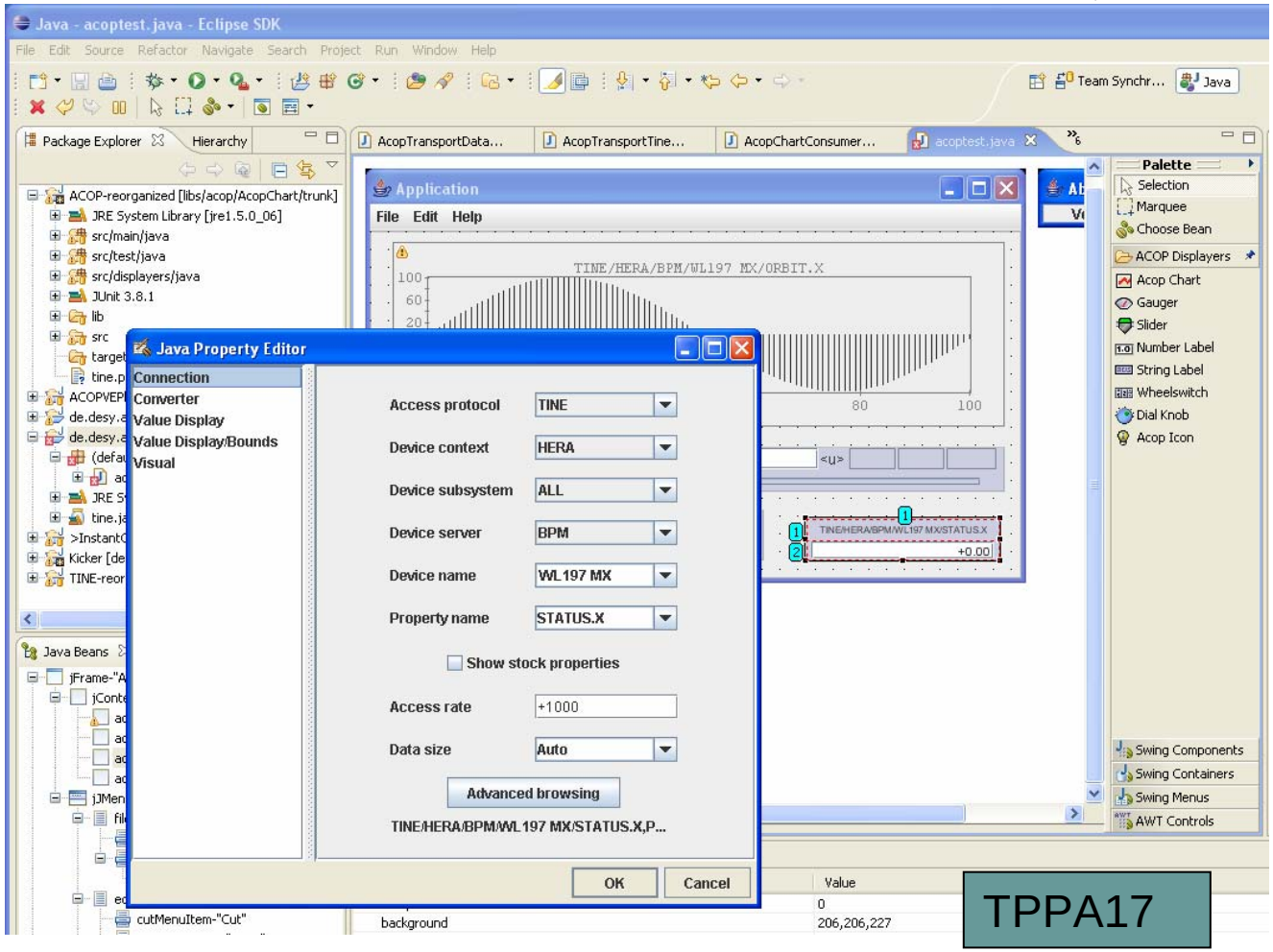
Philip

TINE and Java ACOP
(for rich clients)

No Frameworks !

Use Eclipse, NetBeans, or whatever …

Browse Control system at **design-time** with property panels or *customizer* …



TPPA17

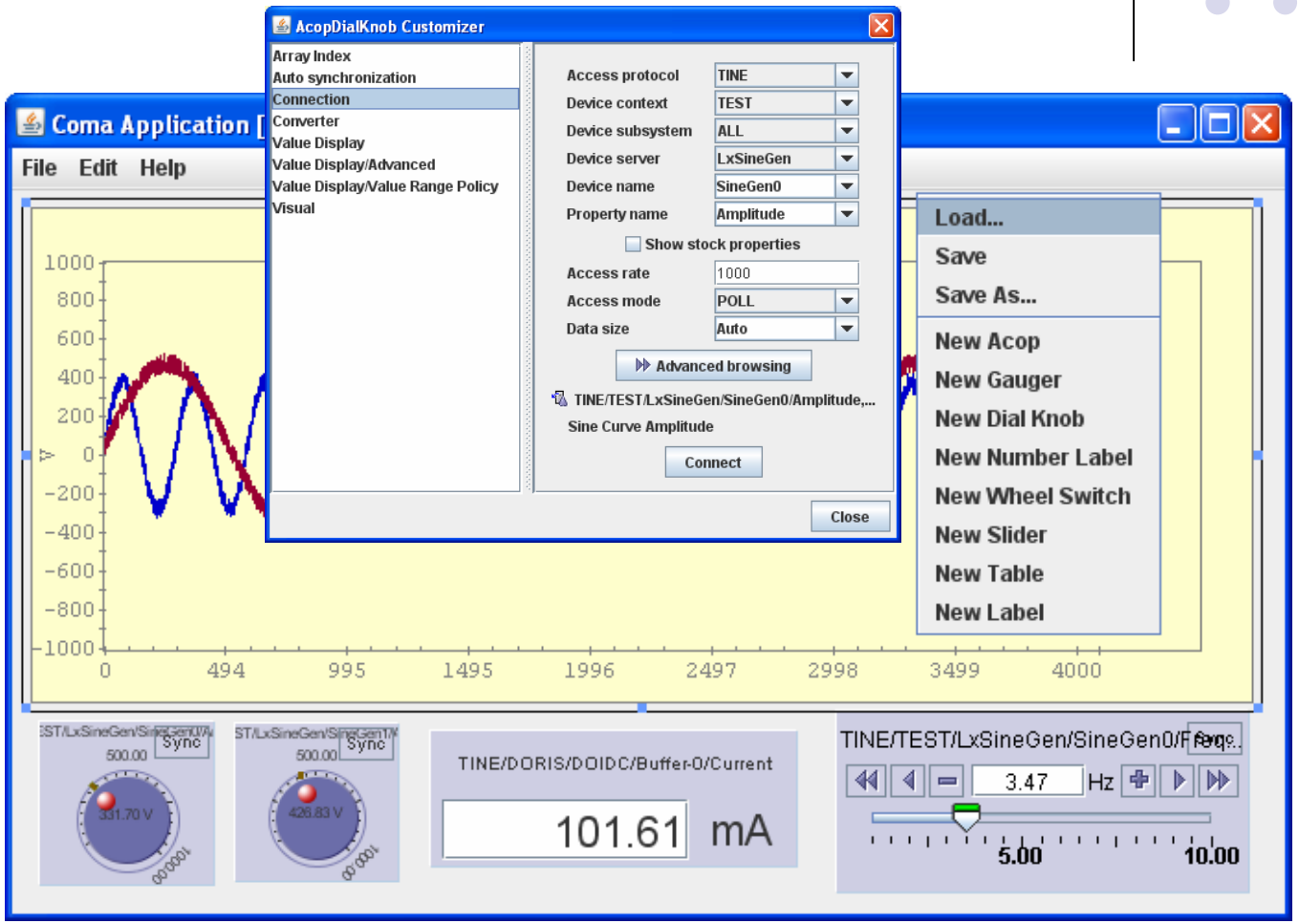
TINE and Java ACOP + COMA
(for simple clients)

No Frameworks !

Lightweight!

Just start a coma application (e.g. an empty coma application)

Browse Control system at **run-time** with *customizer* …

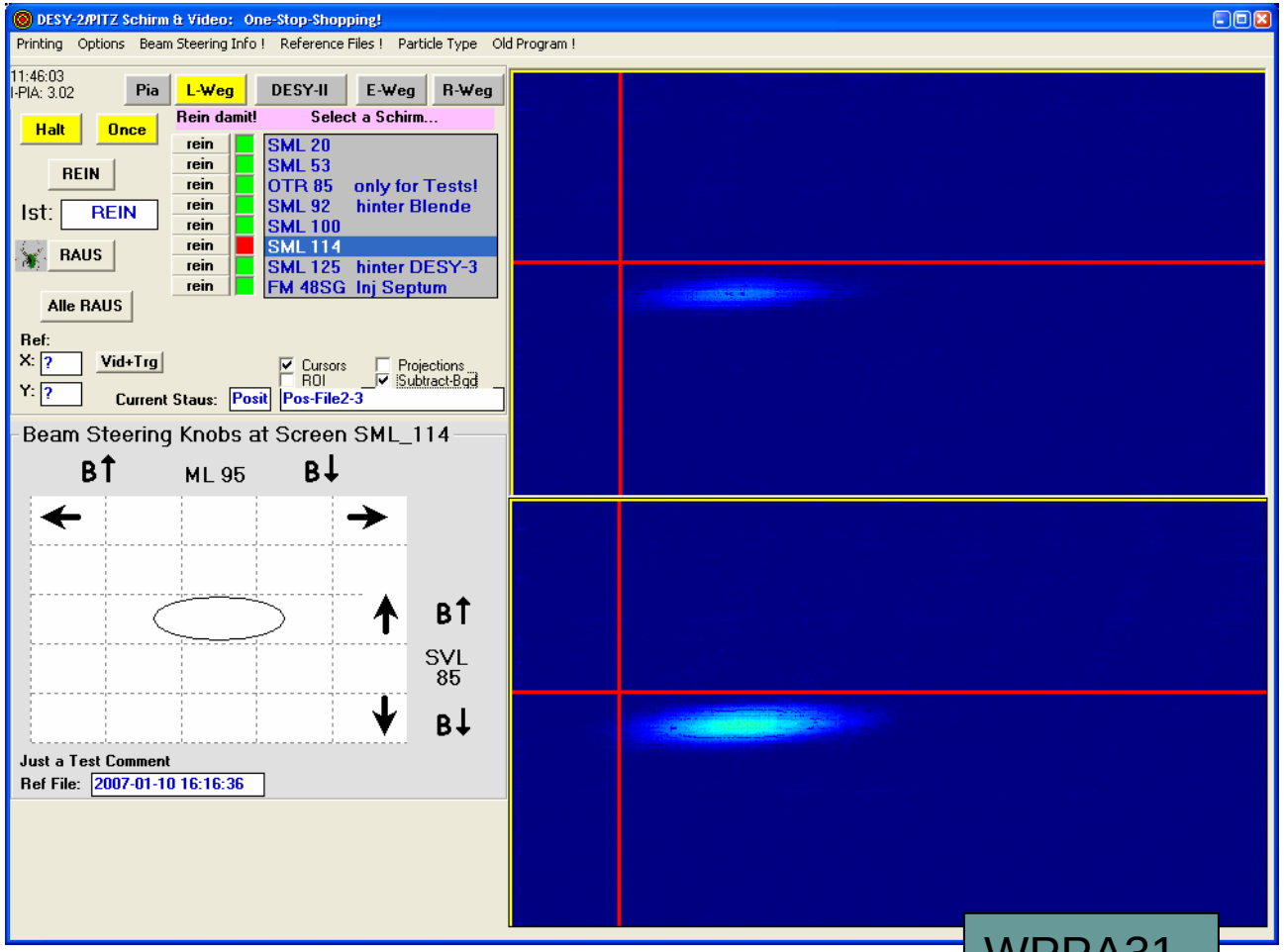


TINE Video
(Multicasting + Scheduling)

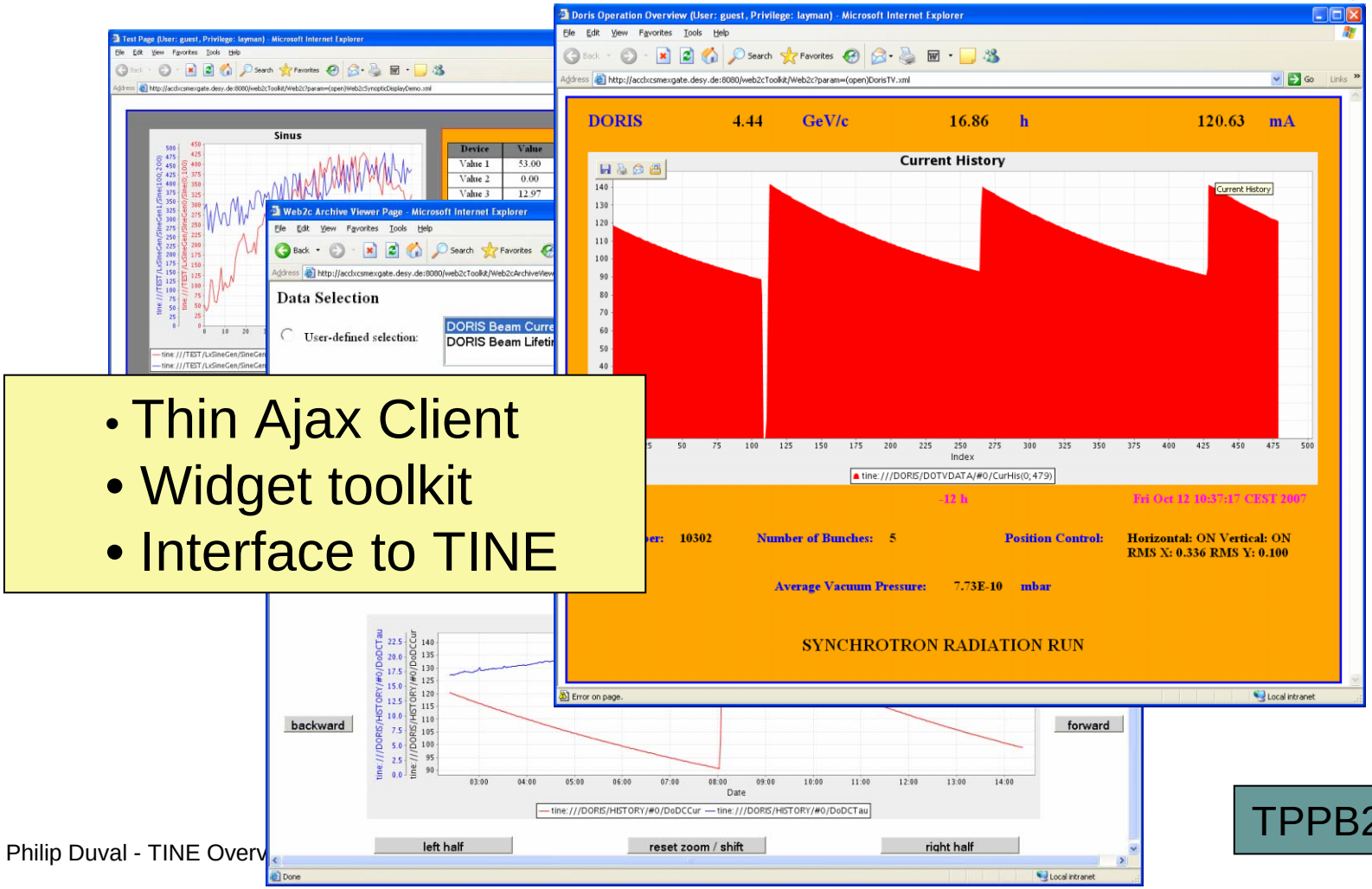
.5 Mbyte Video Frames @ 10 Hz multicast (100 Mb ethernet). (also runs fine @ 20 Hz)

Uses the **NETWORK** switch => as many clients as you want!

Server calls the **Scheduler** when a new frame is grabbed => as real-time as it gets !!!



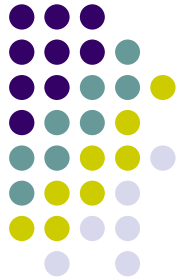
TINE and Web Tools
(Web2C: Web-based Controls Client)



- Thin Ajax Client
- Widget toolkit
- Interface to TINE

TPPB27

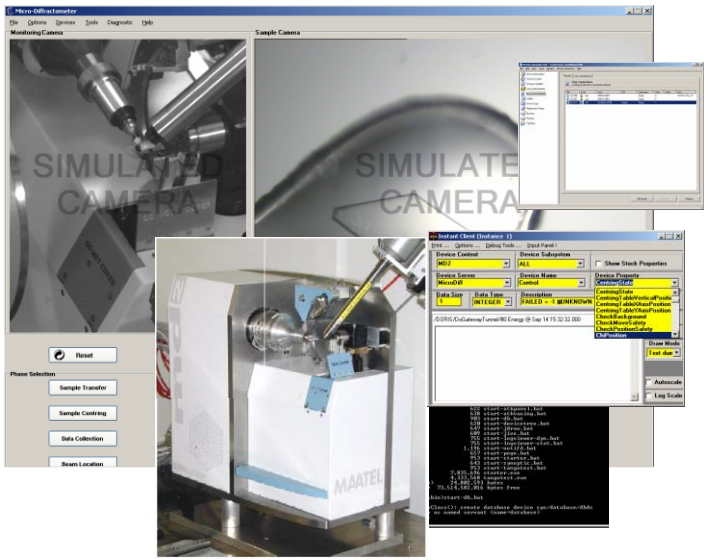
TINE Platforms …



- DOS
- Win16, Win32 (9x, NT, 2K, XP, …), Win64 ?
- Win CE (in progress)
- Unix (Solaris, HP, OSF, SGI, Ultrix, …)
- Linux, FreeBSD (32 bit, 64 bit)
- ELINOS
- MAC OS X
- VxWorks
- VMS (Vax, Alpha)
- LynxOS
- NIOS (plugs, single-threaded LWIP, …)
- Java

TINE and Connectivity to other systems

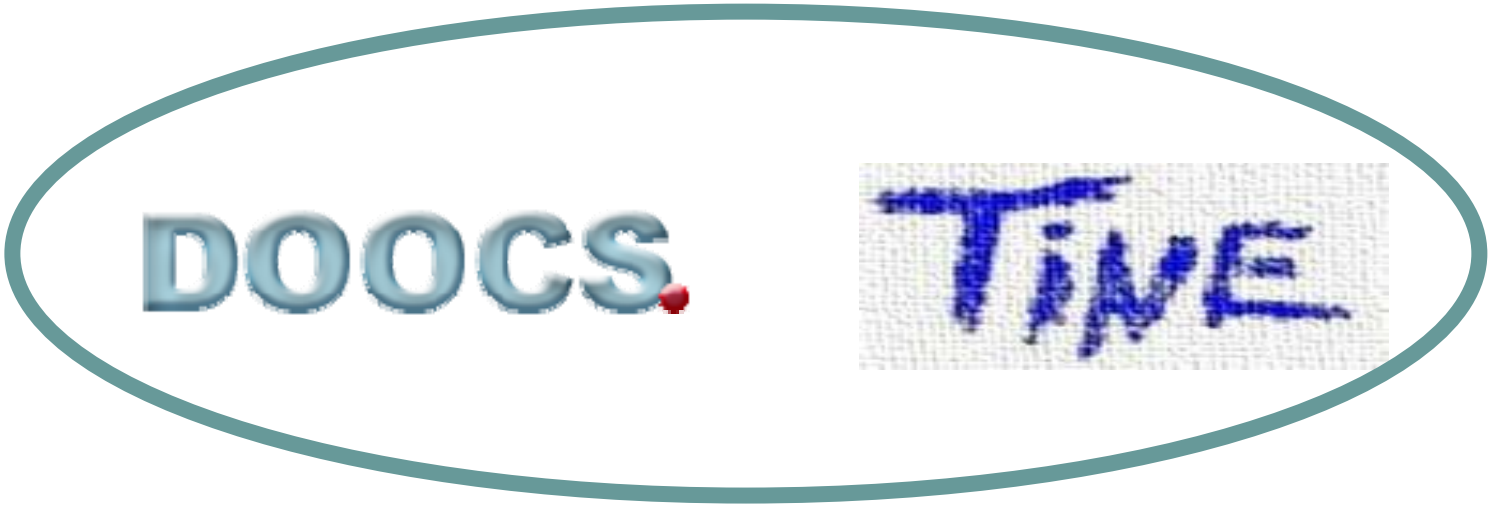
- Already embedded in DOOCS
- Epics2Tine runs on any EPICS ioc
- Connect to STARS/COACK via STARS bridge
- Tango2Tine



TANGO2TINE example MD2 Goniometer

TINE and DOOCS

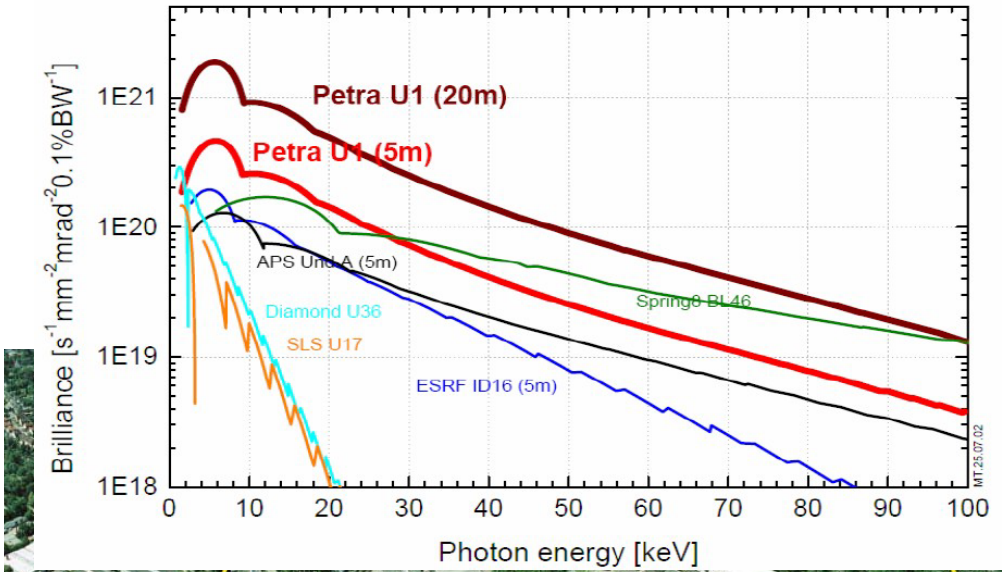
- Always a close relationship between the two.

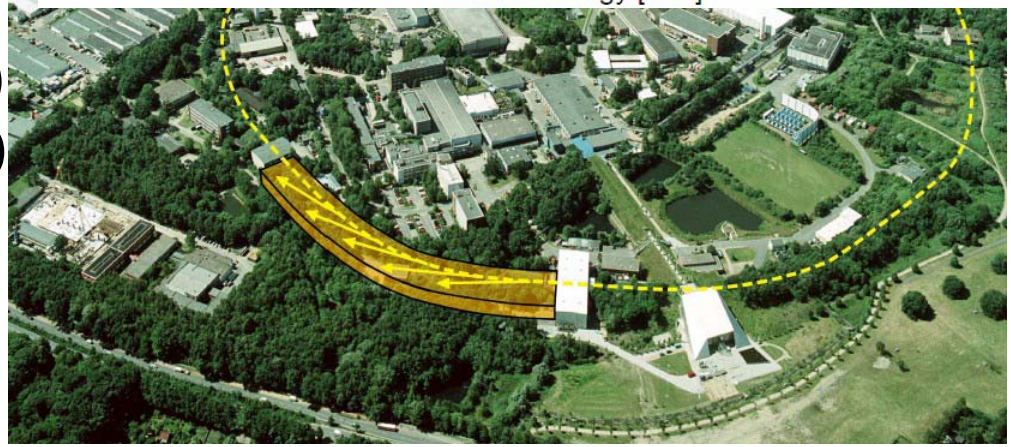


- Imagine (we’re not there yet, but imagine anyway):
  - Download anything from doocs.desy.de; Install it and use it.
  - Download anything from tine.desy.de; Install it and use it.
  - And it all fits together seemlessly (no tweaking)!

TINE/DOOCS Facilities

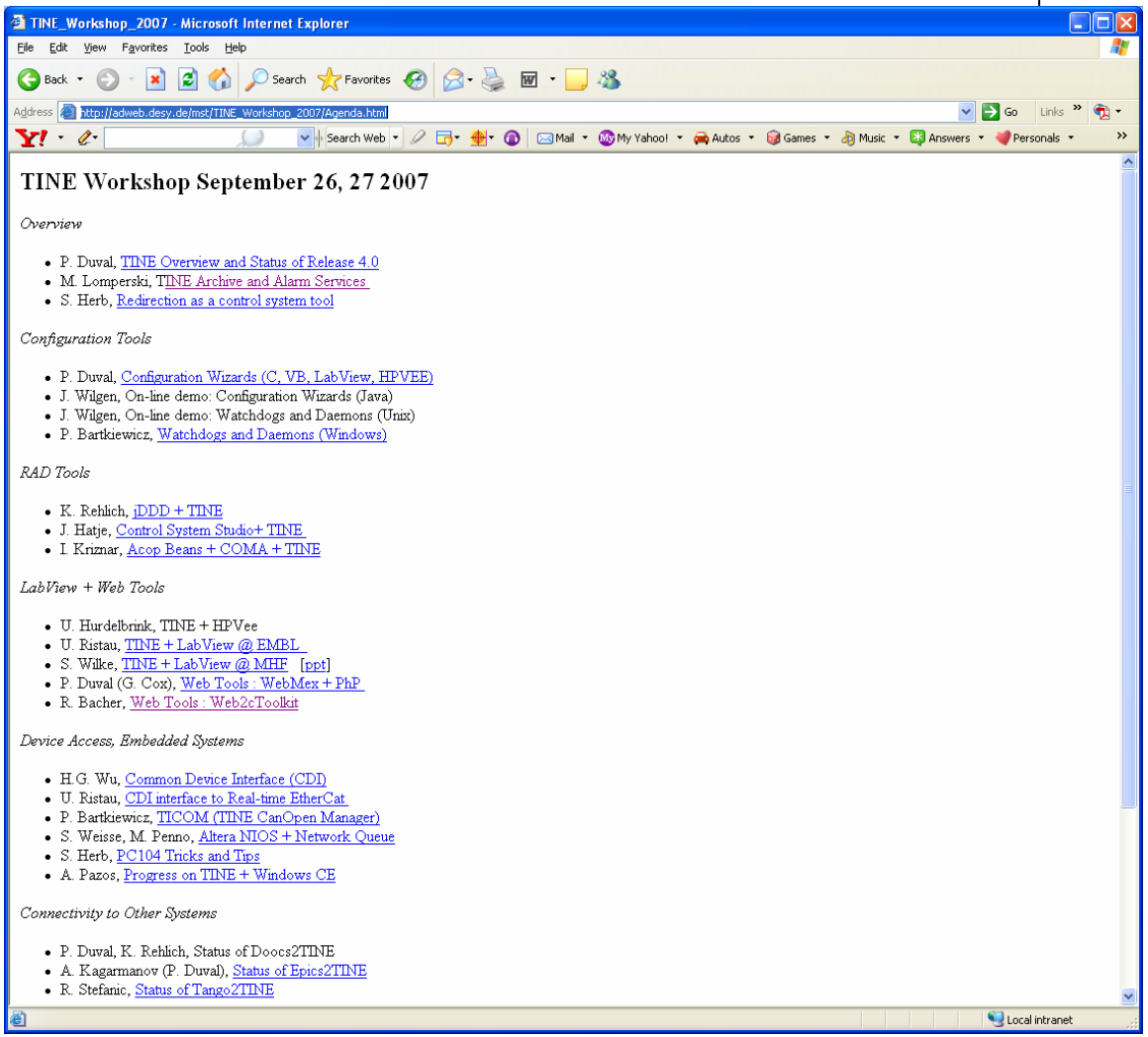
- Petra 3
  - 2009: High-brilliance 3rd generation synchrotron radiation source
- DORIS
- Linac2/Desy2
- PIA
- FLASH
- PITZ (Zeuthen)
- EMBL Hamburg (Beamlines)
- GKSS Hamburg (Beamlines)
- PF Beamlines (KEK)
- LLRF at FermiLab





TINE Workshop

- http://tine.desy.de
  -> workshop 2007



TINE : Where to get it …

- http://tine.desy.de
  - Visit the *download* section and chose your platform.
  - Use setup tools available.
  - Installation takes a few minutes
  - Don’t expect too many miracles (you might have to read a README.txt or two).
- Email to tine@desy.de