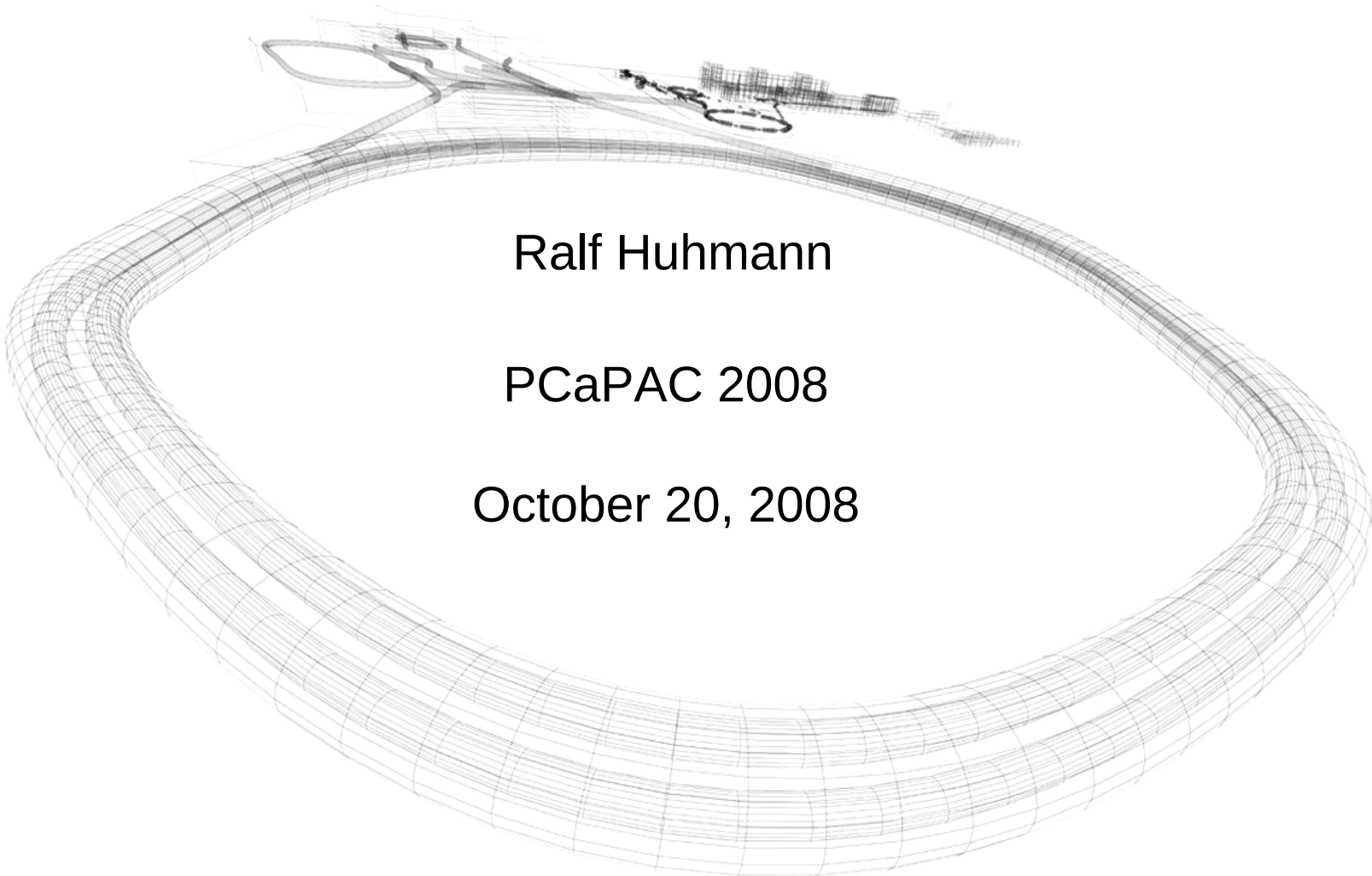
GSI Operating Software – Migration OpenVMS to Linux



Ralf Huhmann

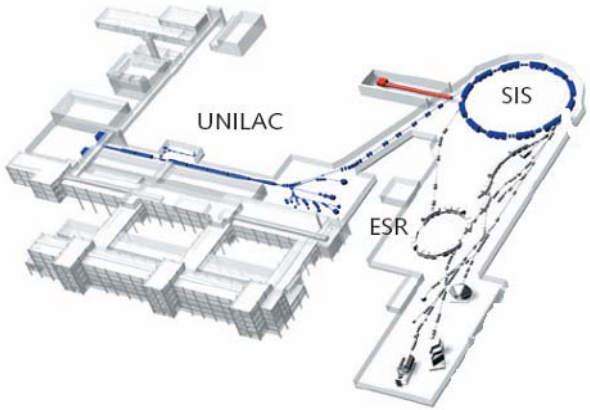
PCaPAC 2008

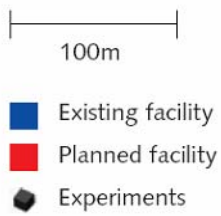
October 20, 2008

Migration OpenVMS to Linux

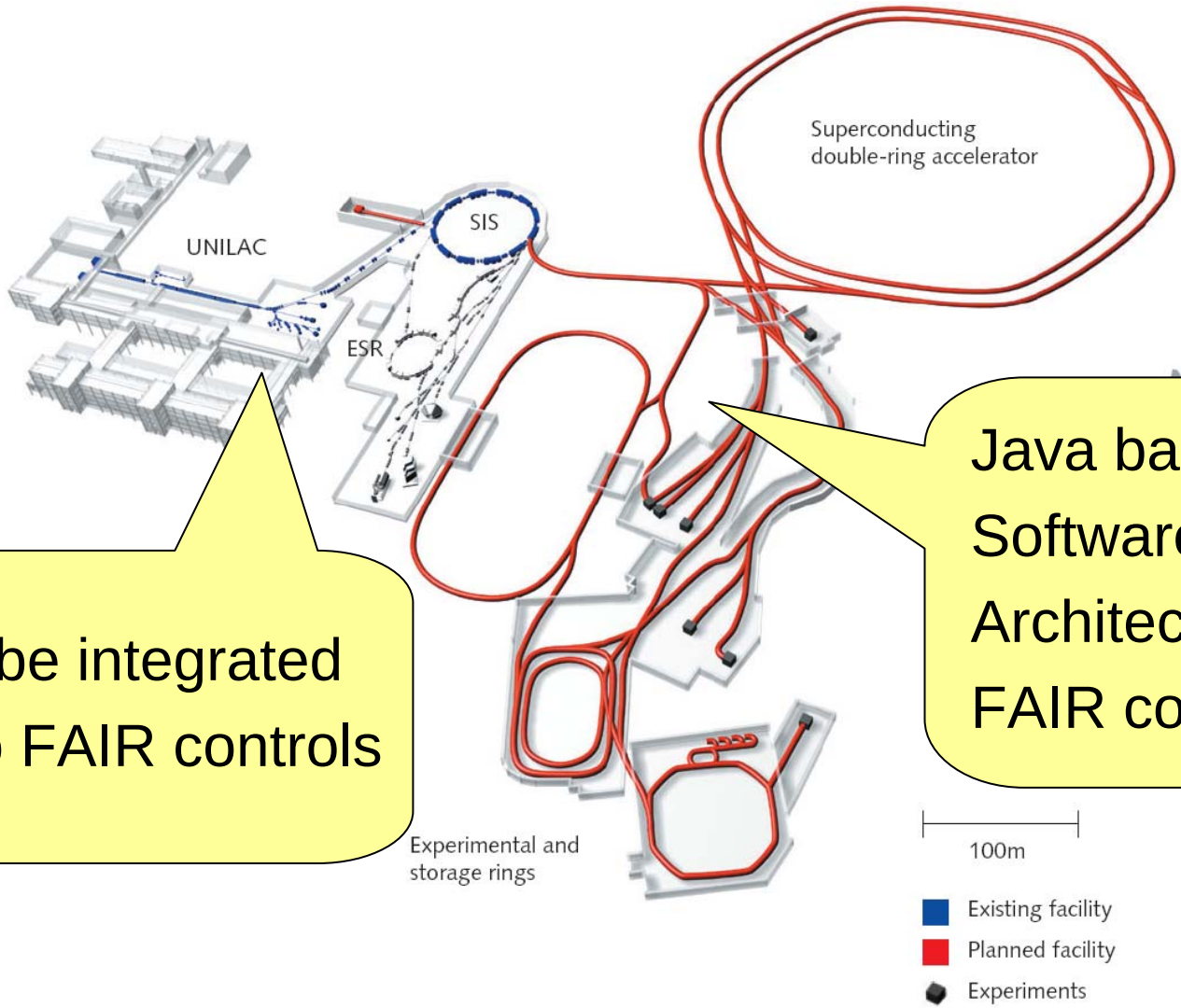
- Motivation
- Present Situation
- Migration Aims
- Details of Porting
- Conclusions

GSI Accelerator Facilities





GSI Accelerator Facilities + FAIR



To be integrated into FAIR controls

Java based Software Architecture for FAIR controls

Migration OpenVMS to Linux

- Motivation
- Present Situation
- Migration Aims
- Details of Porting
- Conclusions

Present Situation

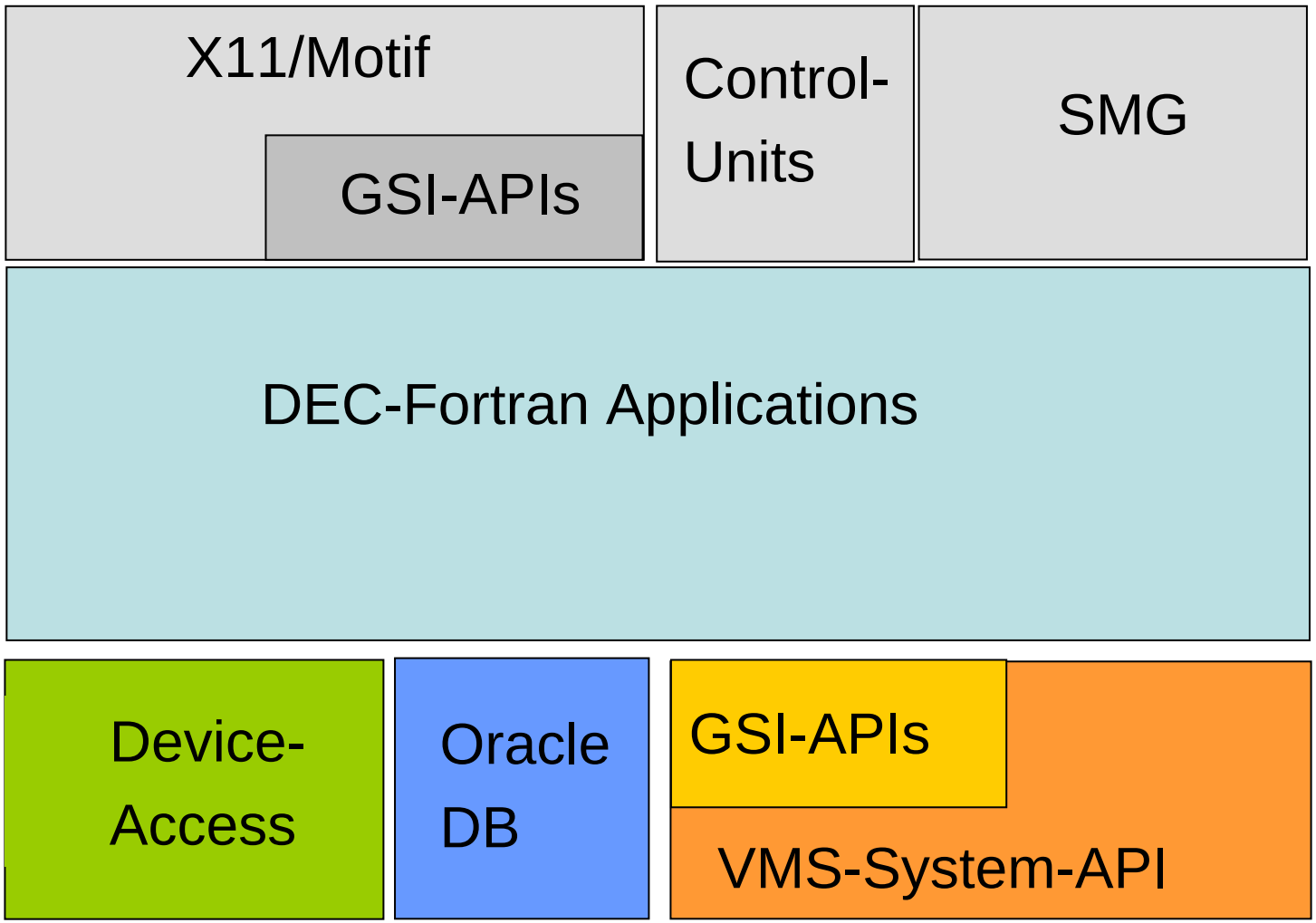
- Hardware: Cluster of DEC-Alphas (Hewlett-Packard)
- OS: OpenVMS 8.3
- Distributed Client-Server Architecture
- User-Interface: X11/Motif-based, Hardware Display and Control Units (Knobs, Key, LED)
- Database: Oracle 10g on Linux-Cluster (migrated from Oracle RDB on OpenVMS)
- Code base: DEC-F77 (F90, F95, 2million lines of code)

GSI Main Control Room

Migration to Linux
Operating Software on VMS/X11

- to enable integration into FAIR controls
- to protect investment (source code)

VMS based Application Software



Migration OpenVMS to Linux

- Motivation
- Present Situation
- Migration Aims
- Details of Porting
- Conclusions

Future Design of FAIR Controls

- Java Software Architecture (JSE, J2EE, Spring)
- New User Interface
- New generic data supply mechanisms (LSA)
- New Device Interfaces (FESA, CMW)
- ...

Migration Aims

- Integration to FAIR Controls (Java Environment)
- Native Unix/Linux Run-Time Environment
- Porting of current VMS-Applications with minimal source-modification

Migration OpenVMS to Linux

- Motivation
- Present Situation
- Migration Aims
- Details of Porting
- Conclusions

Details of Porting

- DEC-Fortran Extensions
- VMS System API
- X11 / Motif
- GSI specific APIs
- Connectivity to new Java Applications

DEC Fortran77 Extensions

- Type definitions: STRUCTURE
- built-in-functions: %LOC(), %REF(), %VAL()
- Compiler Directives: cDEC
- ...

Covered either by

- Intel Fortran Compiler

or

- to be replaced by suitable Fortran90 constructs

Details of Porting

- DEC-Fortran Extensions 
- VMS System API
- X11 / Motif
- GSI specific APIs
- Connectivity to new Java Applications

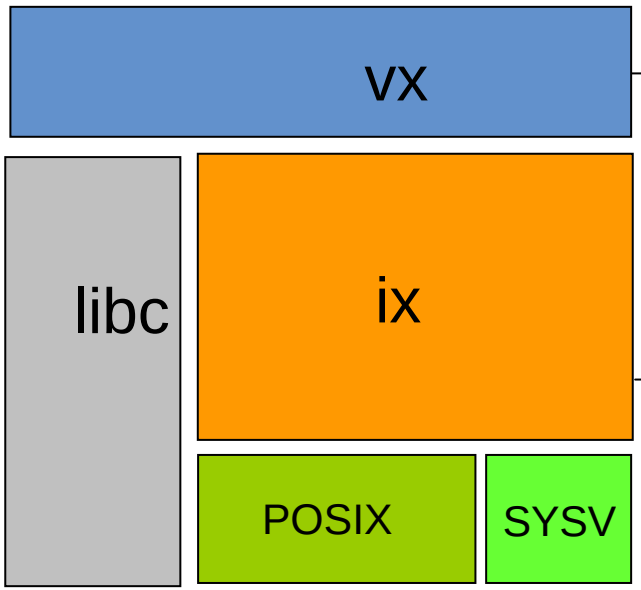
VMS-System API and -Runtime Library

- Non POSIX
- System API to support event driven architectures
- Fortran Utility Libraries
- Commercial Products: inadequate implementation in detail or lacking support

GSI In-house-library and Tools support

- Events
- Timer
- Global Section
- Message Compiler
- ...

“VMS-API” Architecture on Linux



Fortran Interface:
Subset of VMS’
SYS$, LIB$, STR$, ..

C Library: Abstraction Layer for
Events, Timer, Multiplexed I/O,
IPC, SHM, Threads, Logging, ...

Subset of VMS-API implemented by an emulation library

Details of Porting

- DEC-Fortran Extensions 
- VMS System API 
- X11 / Motif
- GSI specific APIs
- Connectivity to new Java Applications

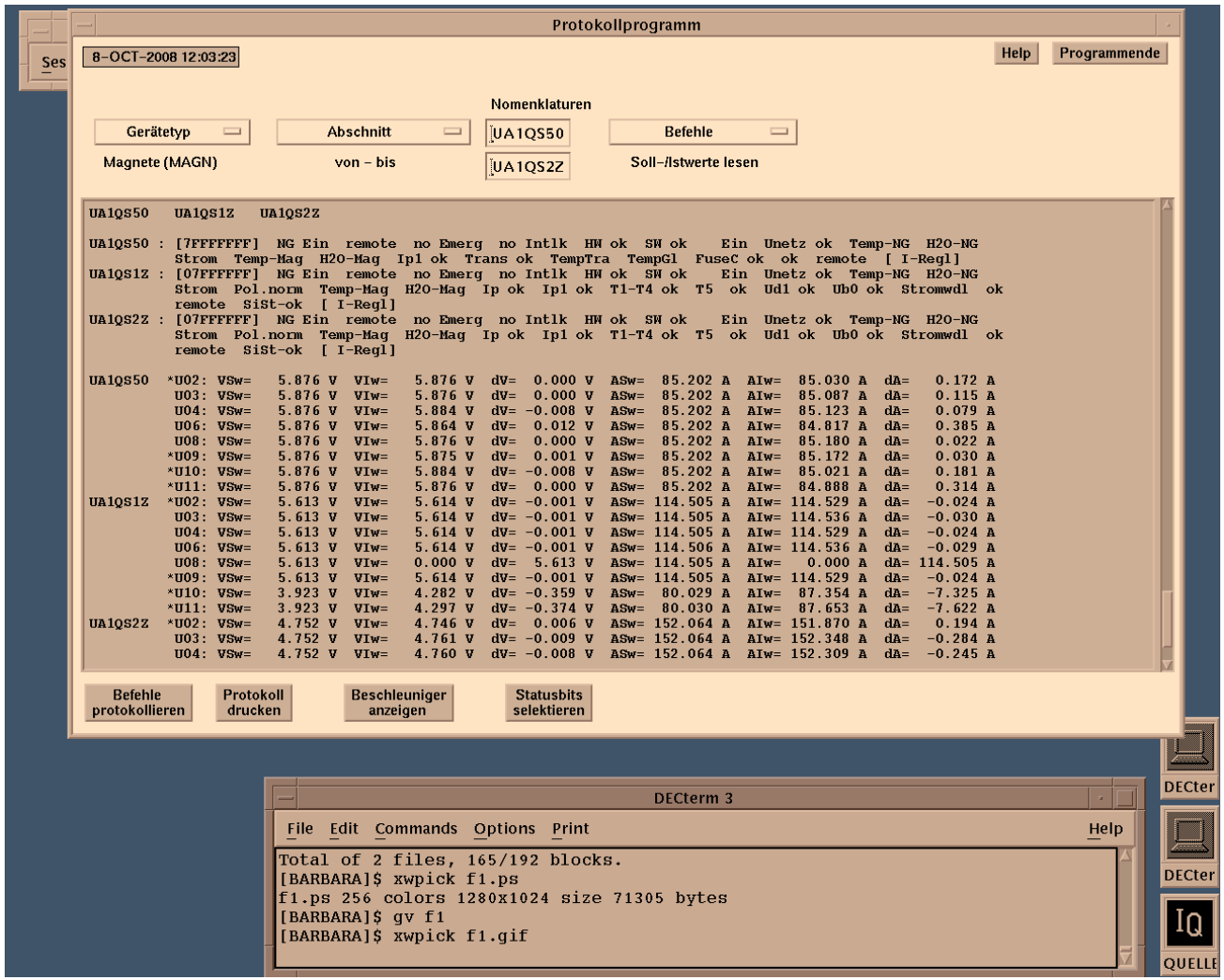
X11 / Motif

- Porting Fortran-X11-Programs using VMS-Fortran-X-lib: Wrapper functions to Linux X11 Library

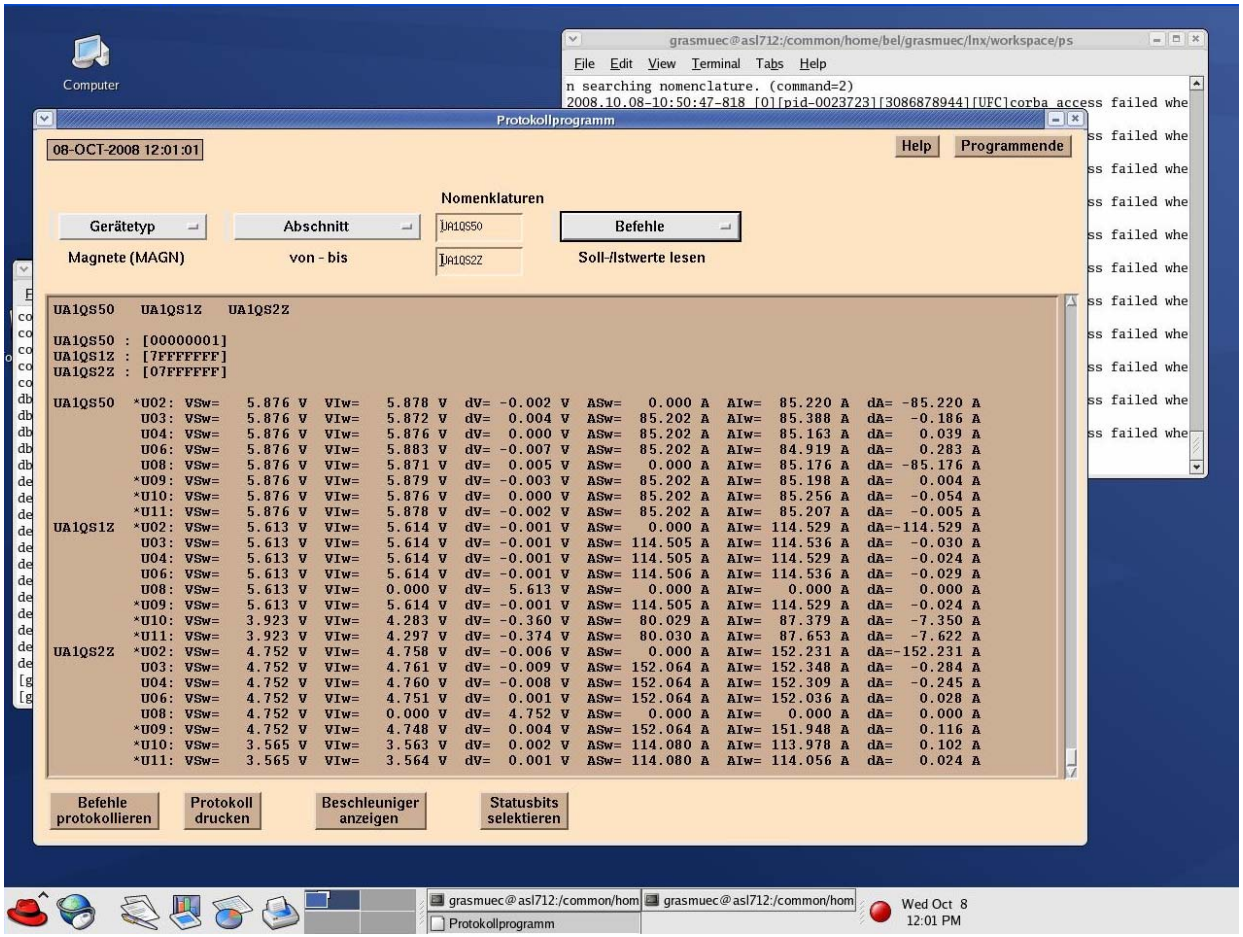
- Porting Fortran-Motif-Programs using VMS-Motif-Library: Same C-API

- But porting software using Motif-XtAppMainLoop()/XtAppAddInput() framework: VMS-specific feature handled in a special GSI-in-house-library

Example: Motif-Program on VMS/X11/DECwindows



Same Program on Linux/X11/KDE



Done with minor code changes

Details of Porting

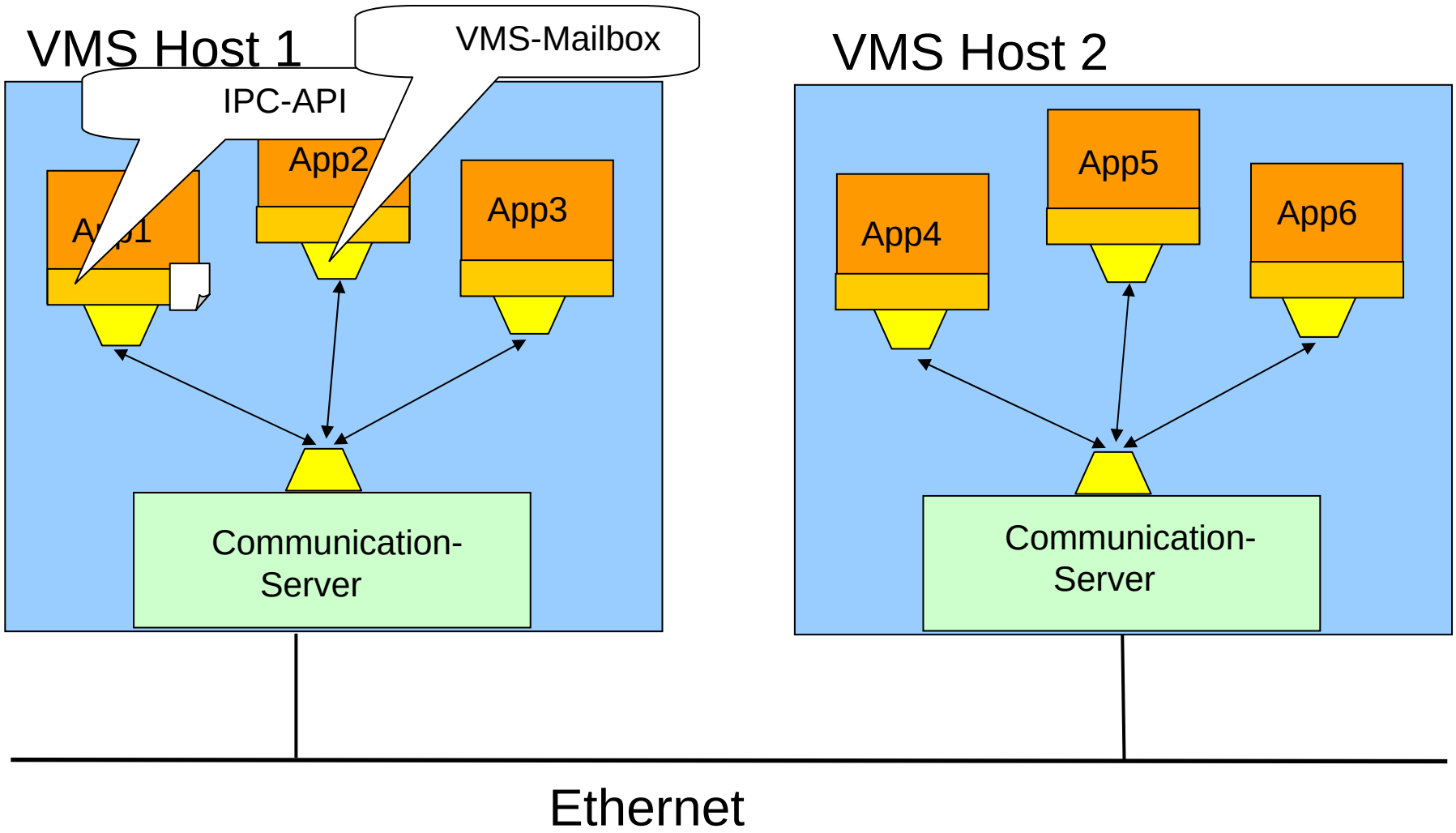
- DEC-Fortran Extensions 
- VMS System API 
- X11 / Motif 
- GSI specific APIs
- Connectivity to new Java Applications

GSI specific APIs

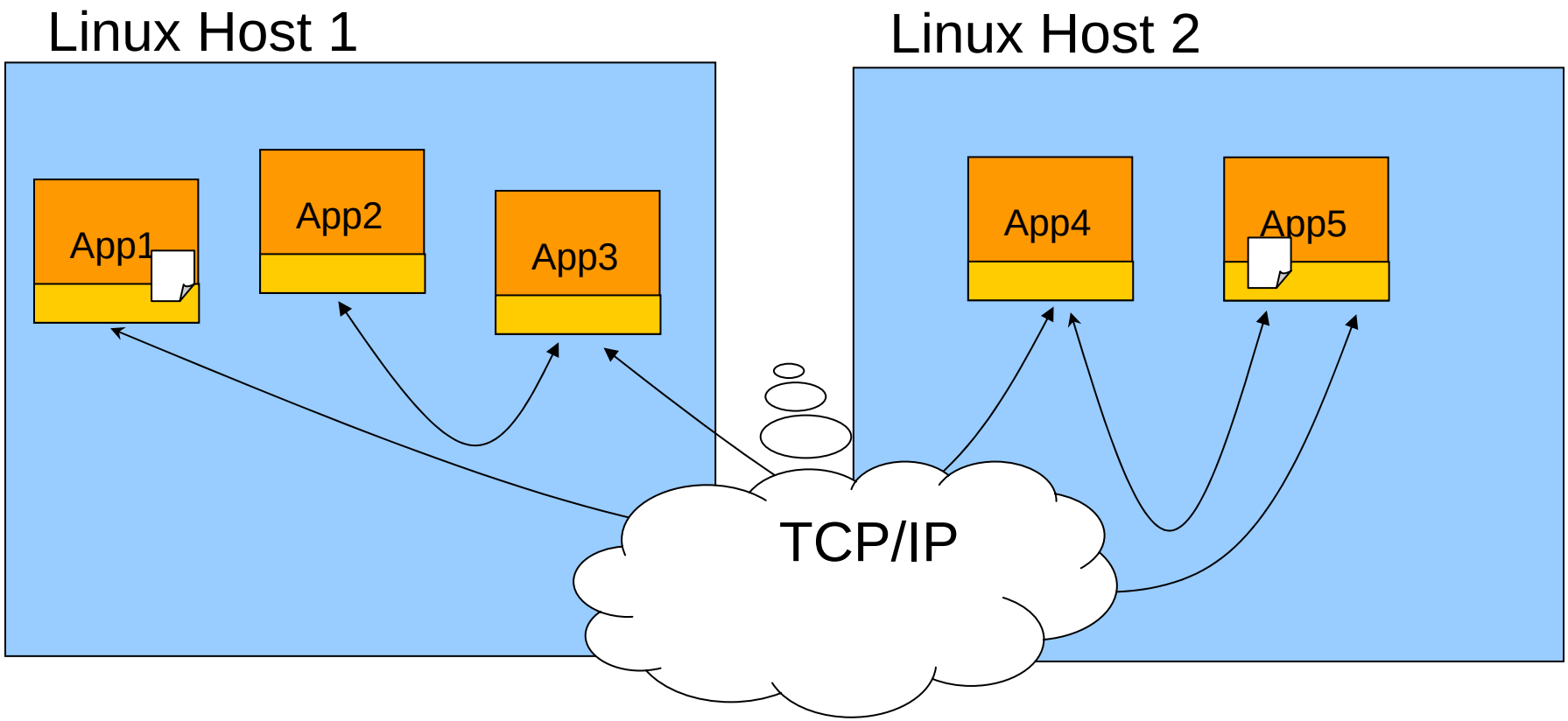
Example: GSI-API for Inter Process Communication

- Written in Modula-2
- Transfers binary data between programs
- VMS-Events
- VMS-Mailboxes
- Communication Server
- Raw Ethernet Protocol

IPC Architecture (VMS)



IPC by TCP/IP peer-to-peer network

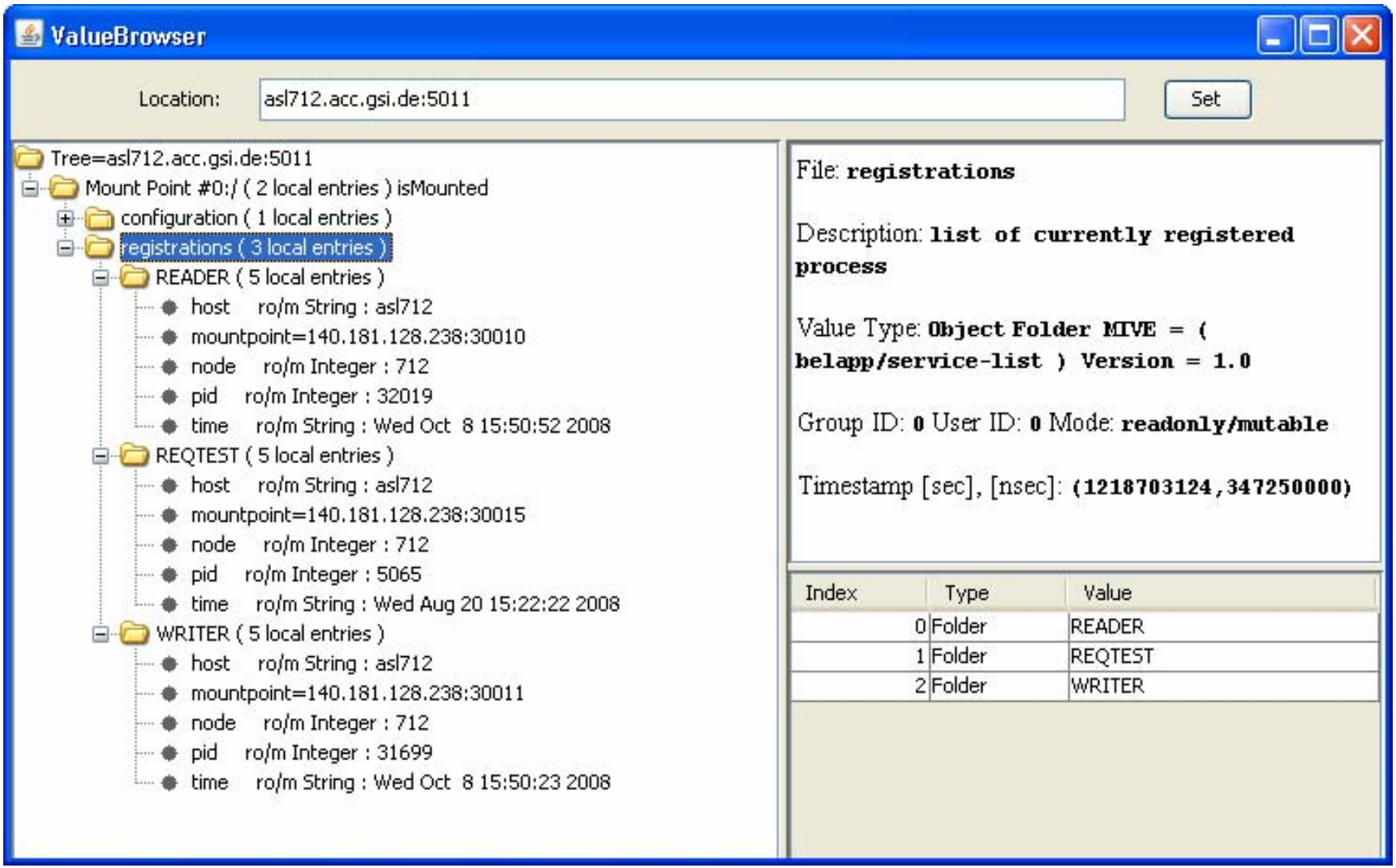


Registration-Service (Broker) to map Process-Name to Host/Port

IPC by TCP/IP peer-to-peer

- Same API
- Peer-to-peer network through TCP/IP sockets, addressed by host:port
- Registration service (name service, broker) for mapping application-name to TCP/IP host/port information

IPC Registration Service (Broker)



GSI specific APIs

- Another example: device access

Part of our front end renovation project, see

“Integration of a Renovated Networking Middleware into a Running Control System Environment”

by Udo Krause, PCaPAC 20.10.2008

Details of Porting

- DEC-Fortran Extensions 
- VMS System API 
- X11 / Motif 
- GSI specific APIs 
- Connectivity to new Java Applications

Connectivity to Java Applications

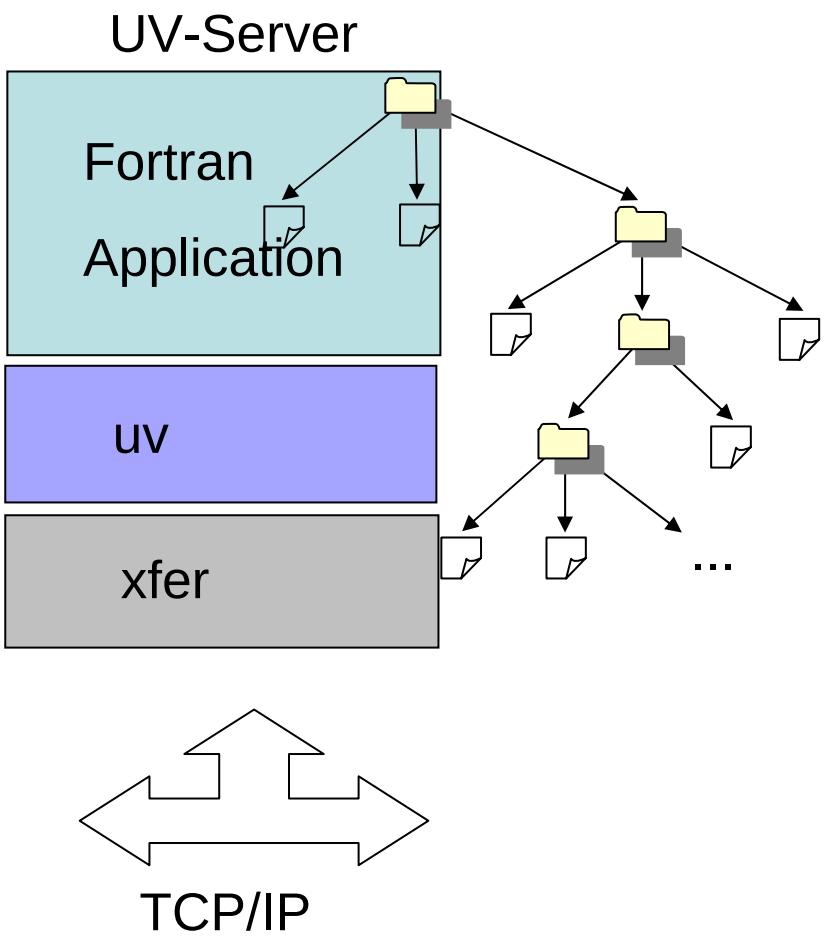
Constraints:

- Minimal source code modification for existing Fortran Applications
- Flexible data structures
- Simple and generic interface to avoid API adaptations

Solution:

Universal Value Architecture (UV)

UV (Universal Value Architecture)



File Properties:
- Name
- Value
- Timestamp
- ....

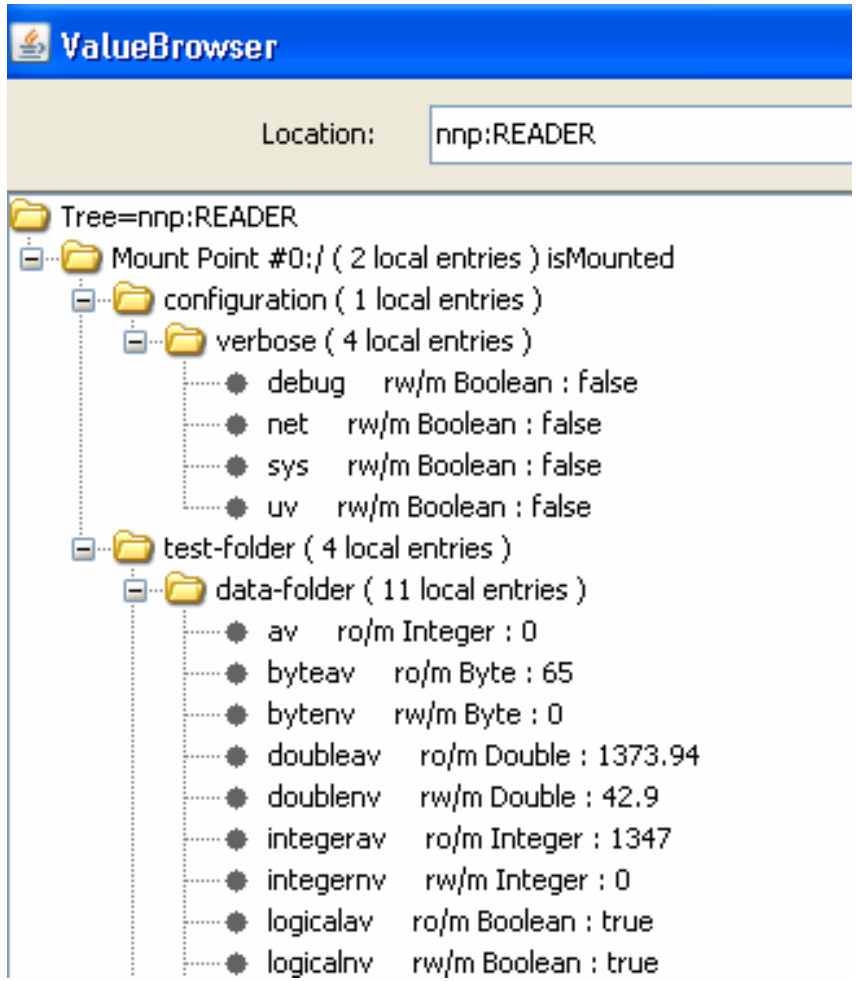
UV-Client

Java
- Read Tree
- Nodes addressed by path
- Subscribe for Changes
- Set Values

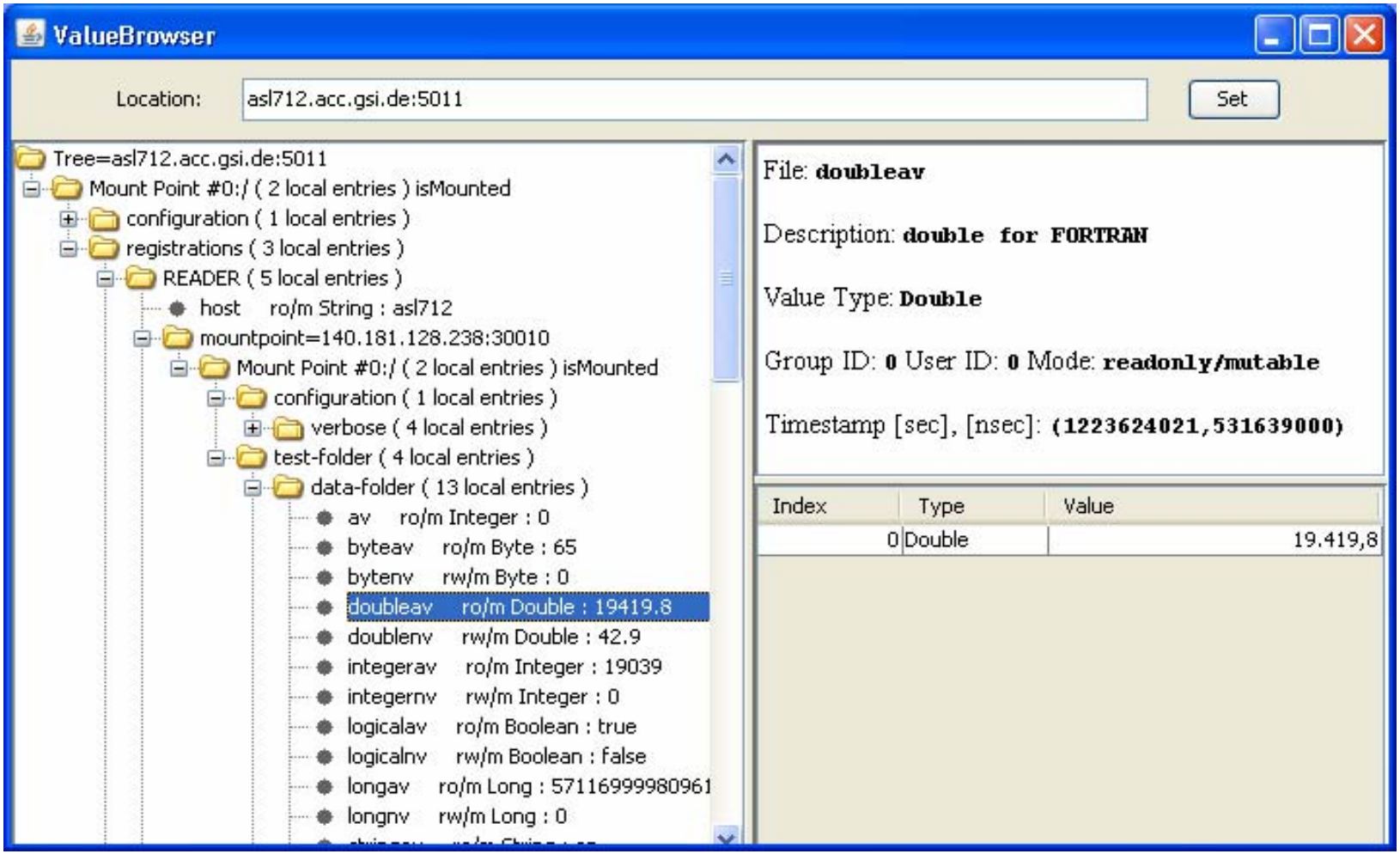
Example: Browser as UV-Java Client

Generic Browser:

- Java Application
- Value Tree
- Finds application by name service lookup



Example: Browser as UV-Java Client



UV-Java Client

- A Java Client can survey and modify any set of values presented by the ported VMS software
- Values can be handled in a single tree
- Tree structure can be changed at run-time

Details of Porting

- DEC-Fortran Extensions 
- VMS System API 
- X11 / Motif 
- GSI specific APIs 
- Connectivity to new Java Applications 

Migration OpenVMS to Linux

- Motivation
- Present Situation
- Migration Aims
- Details of Porting
- Conclusions

Conclusions and Outlook

- Proof of concept (minimal source modification, Java integration)
- Libraries and Tools build solid basis for migration
- Further efforts will be made to integrate into FAIR controls

Thank you!

GSI Operating Software – Migration OpenVMS to Linux

FINE