High Level Applications and Software Components integration using a SCADA Tool

Majid OUNSY

Synchrotron SOLEIL, Saint Aubin, France,
http://www.synchrotron-soleil.fr

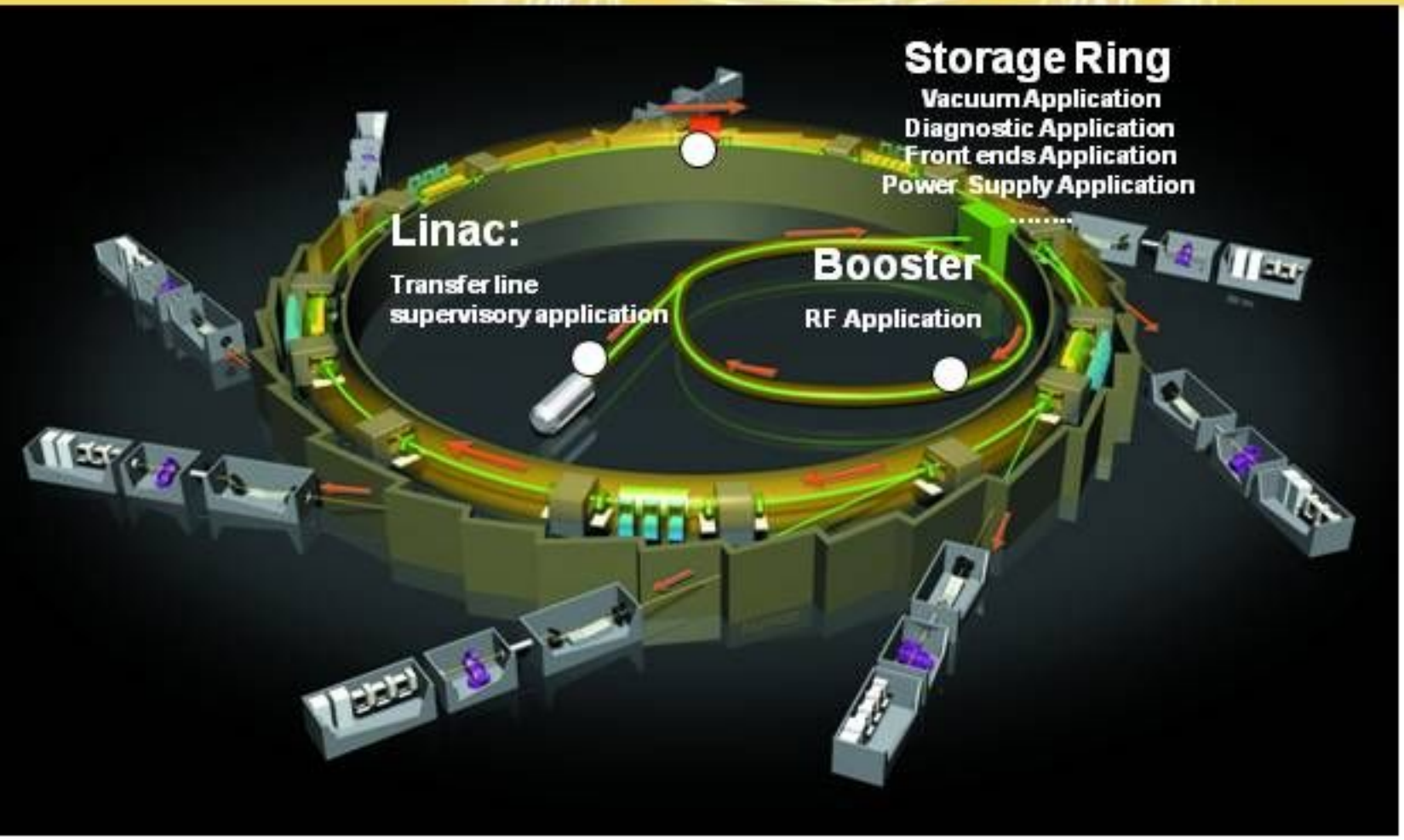
Content

- Context and issues
- SOLEIL Strategy
  - Governing ideas
  - The architecture
  - The governing rules
- Current status and future

Content

- **Context and issues**
- *SOLEIL Strategy*
  - *Governing ideas*
  - *The architecture*
  - *The governing rules*
- *Current status and future*

SOLEIL: HMI development Needs



Machine applications
+
43 possible Beamlines:

One Application per beamline
+
Other specific applications:
Archiving
Scanning
Data storage

- Environment:
  - ✓TANGO for the Control System Bus
  - ✓JAVA and the ATK framework
- Ressources:
  - ✓13 members in the Control software team (5 Java developers),
  - ✓~ 2 Java developers' subcontractors

One way of doing

⇨ The software team takes in charge development of everything from scratch

- Start the development of a new Java standalone application for each new need
- Dedicate a developer to each application
- Subcontract the development if needed

- No experience sharing between developers
- Heterogeneous look and feel
- Maintenance issues
- Lack of perenity (very developer dependant)
- Do we have enough resources ?

Another way of doing

⇨ Find a set of tools (e.g Labview,..) that fit in our users' software development knowledge

- Give them support for tool maintenance
- Let them choose their own GUI look and feel
- Give them support for functional developments

✓**No** way to share **common software** control services
✓**No** way to guarantee software maintenance rules (**versioning**)
✓**No** way to guarantee application **interoperability**
✓**Unmanageable** for the software team in duty support
✓**No** way to guarantee effective users' **autonomy**

Content

- *Context and issues*
- **SOLEIL Strategy**
  - *Governing ideas*
  - *The architecture*
  - *The governing rules*
- *Current status and future*

Our way of doing : Help users to be autonomous

⇨ Use a commercial SCADA for ready to use high level components in a user friendly drag and drop software development environment

- Provide our users with a rich library of components
- Let them develop their applications themselves
- Concentrate on developing and maintaining the underlying frameworks and giving support on them

✓**Perenity**: Fits in modern software development standards
✓Ease of duty support : **Homogeneous** look and feel
✓**Manageable** by a small team : a set of co-developed frameworks
✓Ease application **interoperability** : service oriented

Content

- *Context and issues*
- *SOLEIL Strategy*
  - **Governing ideas**
  - *The architecture*
  - *The governing rules*
- *Current status and future*

Keep things simple

⇨ Identify the core software control functionalities and provide them to users as centralized services

- **Process control and workflow management** **(Sequencing)**
- **Facility abnormal behavior diagnosis** **(Data logging)**
- **Collecting sensors data on moving actuators** **(Scanning)**
- **Equipment operation and monitoring** **(Tango supervision)**
- **Physical and experimental data processing** **(Data storage)**

THEN :

- **Choose the right solution for each need (could it be commercial)**
- **Be sure it conforms to the software environment constraints (Java)**
- **Build GUI components or applications on top of this**
- **Use GLOBALSCREEN as the final container for integration**

Content

- *Context and issues*
- *SOLEIL Strategy*
  - *Governing ideas*
  - **The architecture**
  - *The governing rules*
- *Current status and future*

S ... A Service Oriented Multilayered ... ure

GUI integration : GlobalSCREEN

Passerelle GUI components

Archiving GUI components

Scan GUI components

Supervision GUI components

NeXus Storage GUI components

Passerelle Widget Library

ATK Widget Library

Comete Widget Library

Passerelle Core API

ATK Core API

NeXus Core API

Sequencing service

Data logging service

Scanning service

Tango supervision service

Data File Storage service

Process & calculation devices

TANGO Devices

Device

Device

Device

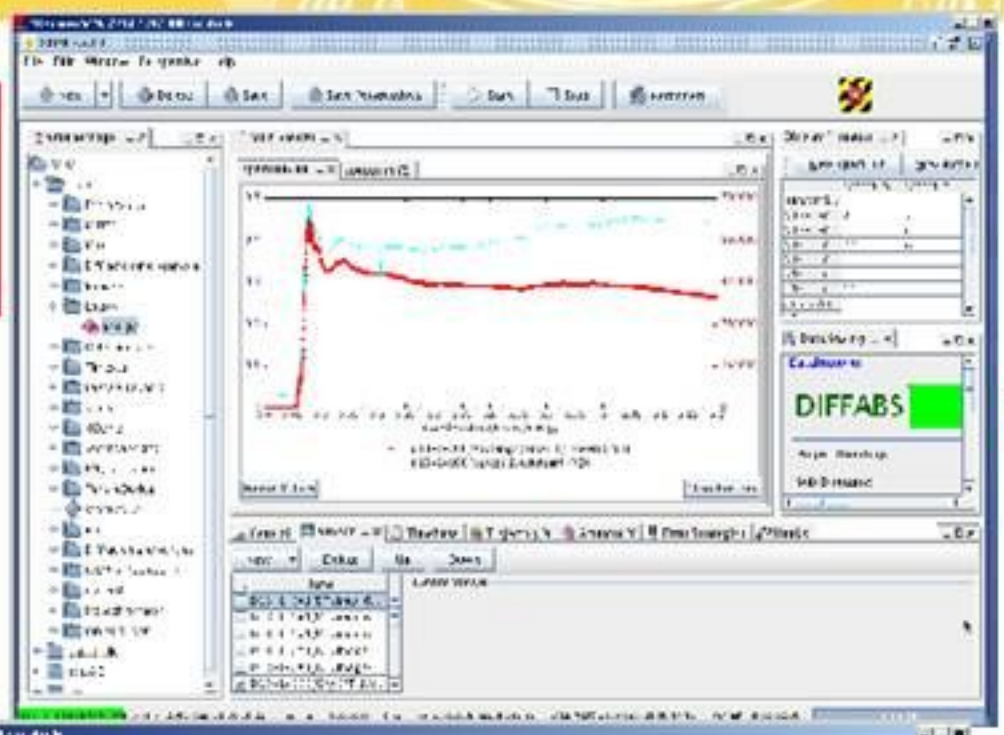
Equipment and subsystem devices

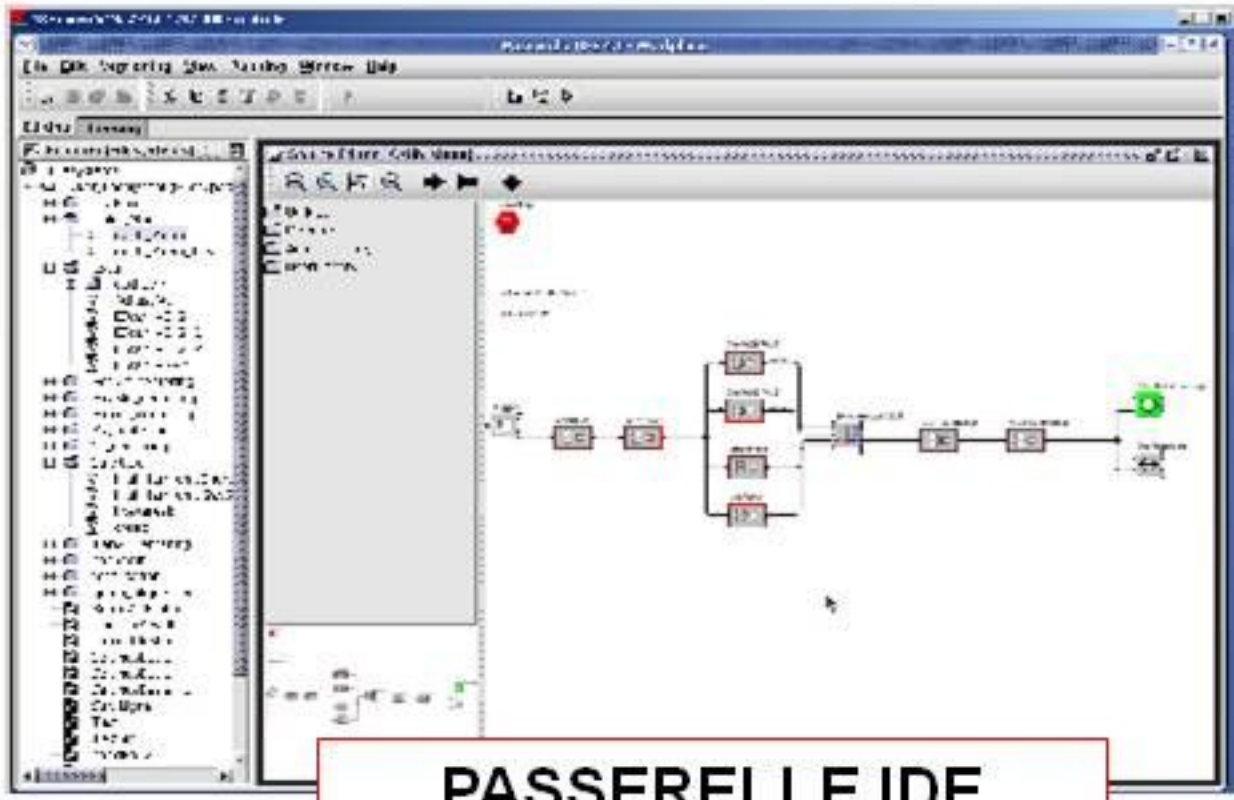
Hardware access devices

Functional development

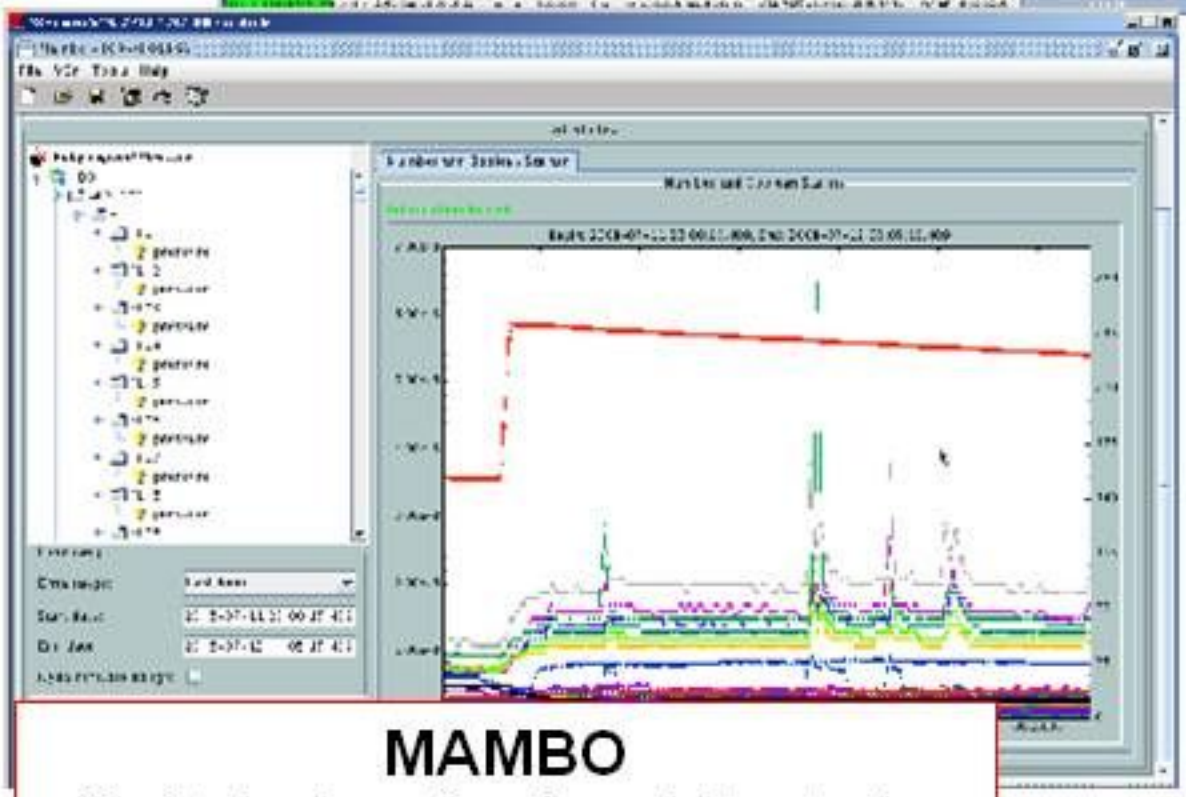
Example of GUI components giving access to all features of a service

SALSA
The scan service front end





PASSERELLE IDE
A graphical GUI for sequencing



MAMBO
The data logging configuration and data extraction visualisation

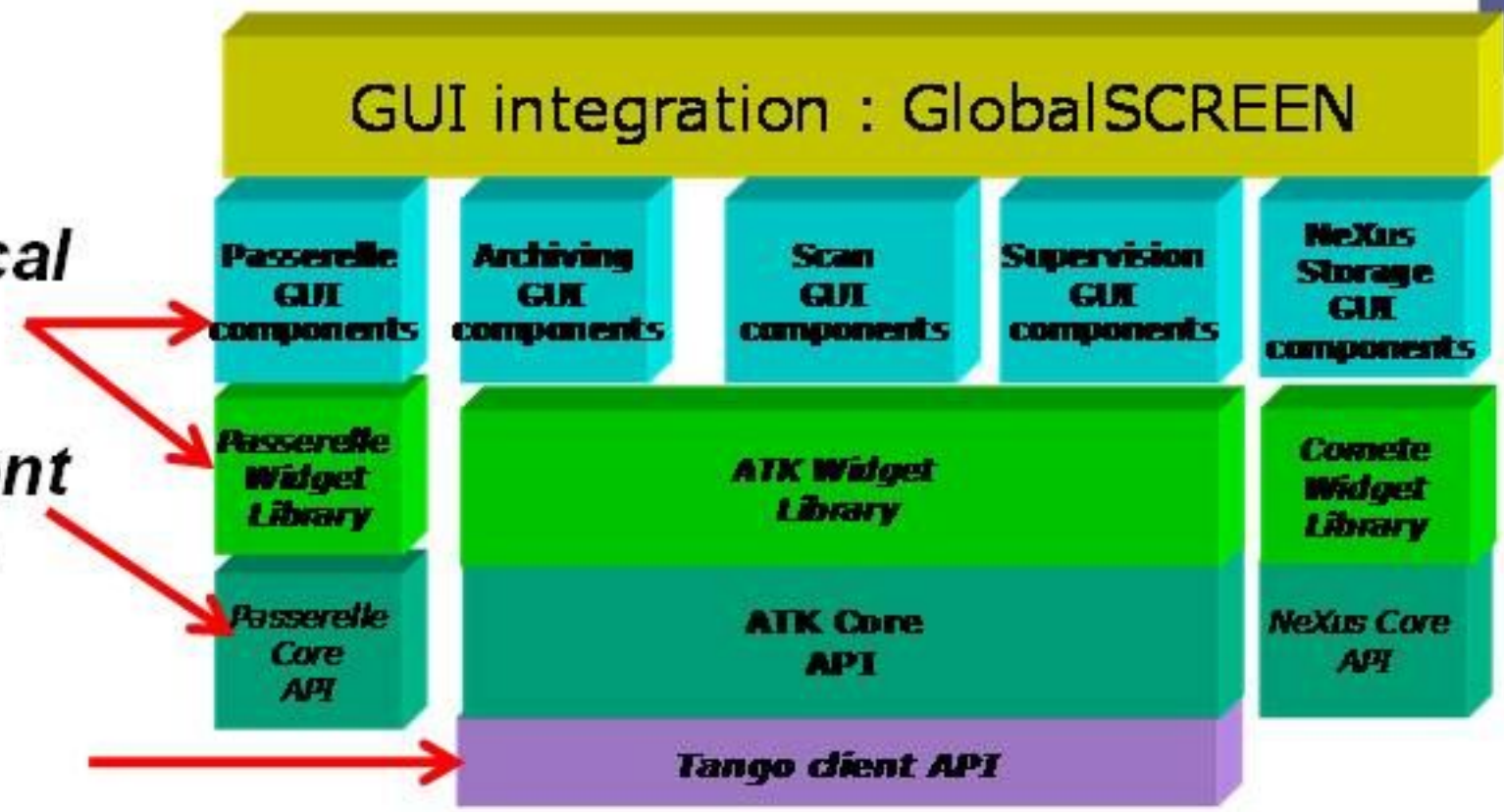
Content

- *Context and issues*
- *SOLEIL Strategy*
  - *Governing ideas*
  - *The architecture*
  - **The governing rules**
- *Current status and future*

How to keep the whole system coherent

Technical Requirements for each service:

- *"Ready to use" graphical components*
- *Java API library for client access to each service*
- *Usage of Common frameworks when needed*

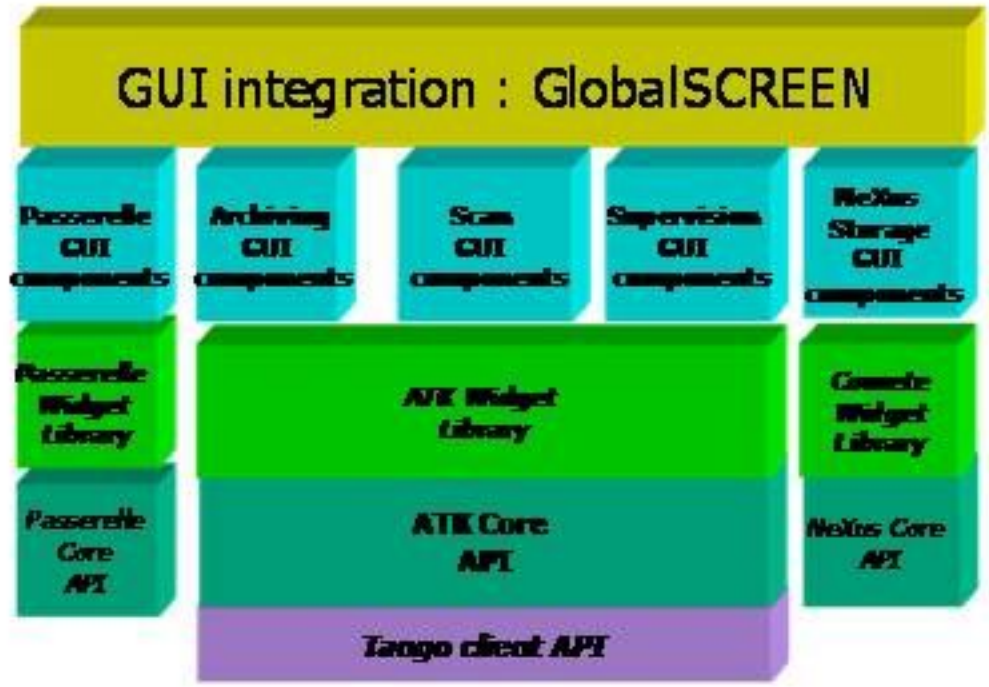


Must Conform to Java Standards (javabean, jdbc,...)

How to keep the whole system coherent

Examples of how there requirements are fulfilled

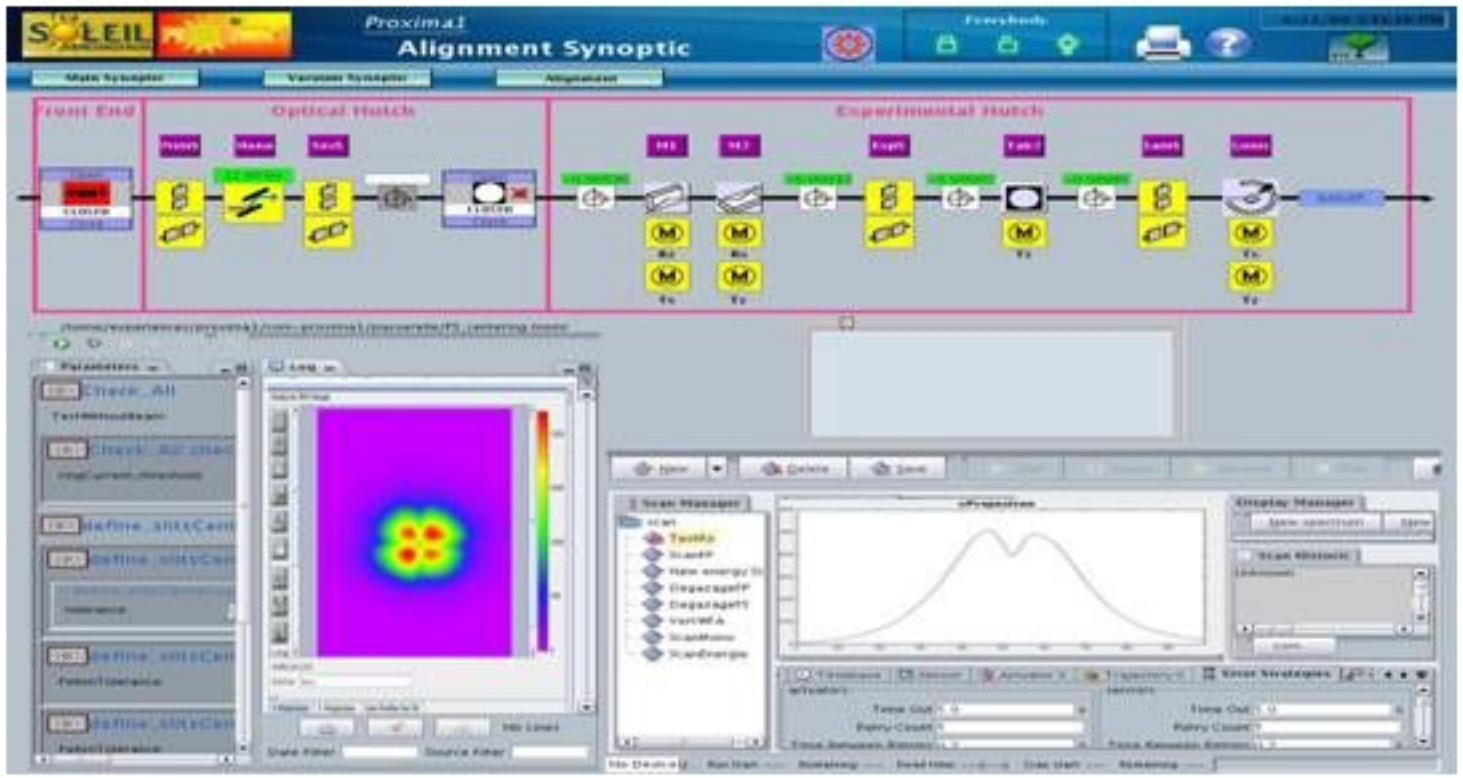
- Tango Supervision :
  - *ATKCore + ATKWidget (Java Bean standard)*
- Data Logging :
  - *ArchivingApi + ATKWidget (JDBC standard)*
- Sequencing
  - *PasserelleApi + PasserelleWidgets (Webservices)*
- File DataStorage
  - *NexusApi + COMETE framework (ImageJ)*



Content

- Context and issues
- SOLEIL Strategy
  - Governing ideas
  - The architecture
  - The governing rules
- Current status and future

Current Status and Future



- ~200 components provided by Software group
- ~50 GlobalSCREEN applications in operation
- System adoption by users *(25 GlobalSCREEN applications developers in Accelerators and Beamlines teams)*

THE FUTURE

- Web deployment of these applications thanks to :
  - Jboss Application Server
  - Java Web Start

Thanks for your attention