Agenda

- Which databases for **which data** ?
- Some reminders on **services** required on top of these databases
- Some reminders on project organization
- A few figures on operational usage of these databases at SOLEIL and their evolution
- Software evolutions since the last Tango meeting
- Foreseen evolutions

Some reminders on the different databases and the associated software services

Which databases for which data ?

- **The Tango static database**
  - *It contains Tango **control system configuration** data: properties of devices , properties of attributes*
  - *It can also be used by developers as a central persistency database to store parameters useful for a particular device, set of devices or GUI application : this is usually done through free properties*
  - *At SOLEIL we have about 30 Tango dabases deployed*
  - *Tomorrow there will be a presentation of Gwenaëlle Abeillé giving our feedback after 10 years of operation on operating these databases*
- **The Historical Database : also called HDB**
  - *It **stores the values of a predefined set of Tango attributes** which have been stored following a set of criteria (periodic , on change, etc ..)*
  - *These data should be stored for the **lifetime** of the institute*
  - *as they represent the “memory” of all equipments and of the facility itself*
- **The Temporary Database : also called TDB**
  - *It **stores the values of a predefined set of Tango attributes** which have been stored following a set of criteria (periodic , on change, etc ..)*
  - *These data should be stored for a **limited period of time** (in SOLEIL case a beam run)*
  - *These data are used for the daily operation of the facility (accelerator or beamline)*

Which databases for which data ?

- **The Snapshots database**
  - *It contains values of Tango devices attributes setpoints*
  - *The list of Tango devices attributes are regrouped in so called “contexts”*
  - *For a given context , values are acquired when executing a Snap (“a picture”)*
  - *These values acquired during a snap can be reapplied later to the Tango devices*
  - *The purpose is to reconfigure a set of equipments to a working configuration : for example put the Accelerator in “single bunch” configuration*
- **The Alarm Database :**
  - *It contains “rules” which are evaluated*
    - *A rule is an expression based on the values of Tango attributes*
    - *When a rule is evaluated to true it produces an “Alarm”*
  - *Data related to the Alarm are then stored in the AlarmDataBase*

- *The Data containers can be either MYSQL or Oracle*

Which software service have to be delivered on top of these databases ?

For each database the following services must be provided

- ***1 Configuration GUI tool*** *must be available to allow the people in charge of operation of the facility to configure the "database service" according to its needs*
- ***1 Data extraction and visualization GUI tool*** *to allow people in charge of operation of the facility to extract data and visualize them*
  - *The HDB/TDB visualization tool must be available from a* ***WEB interface*** *outside from the controls network*
- ***Data collection mechanisms*** *must be embedded within the Tango control system to benefit from standard Tango administration tools for their deployment (like jive and astor)*
- ***Administration and monitoring*** *of the archiving system itself to give to operational people the possibility to monitor they are working correctly*
- *The possibility to* ***extract data from other tools*** *(MATLAB, Labview, etc ..) :*
  - *once again Tango devices are a very open way to do it*
  - *as many clients applications can then access extraction mechanisms through Tango commands and attributs*

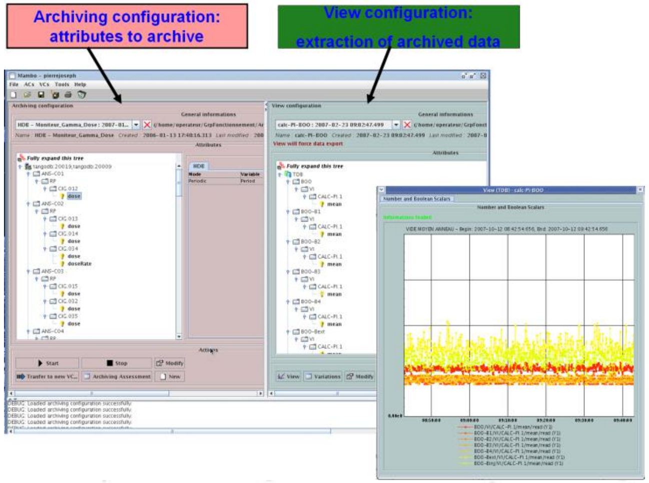
The requirements for these services lead to the following Software architecture for all different kind of databases

On the functional part

- **Archivers Tango devices :**
  - *They are in charge of data collection*
  - *They are based on dedicated Java APIs that make the interface with the databases*
- **Extraction Tango devices**
  - *They are in charge of data extraction*
  - *They are based on dedicated Java APIs that encapsulates the connection to the databases*
- **Archiving watcher (or managers)**
  - *They are in charge of periodically checking the insertion in the database are coherent to the archiving configuration itself*
  - *They make some diagnosis on archiving problems to help operator to correct it*

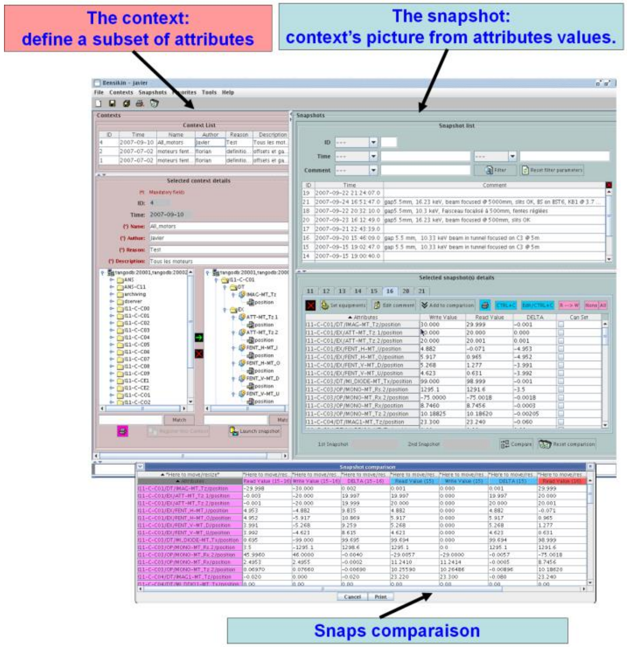
The requirements for these services lead to the following Software architecture for all different kind of databases

- **From the end user point of view the GUI shares :**
  - *The same philosophy for configuration and visualization*
  - *The same set of data visualization components (by internally using the Java COMETE library to share GUI components*



Archiving configuration: attributes to archive

View configuration: extraction of archived data



The context: define a subset of attributes

The snapshot: context's picture from attributes values.

Snaps comparaison

Mambo application

on top of HDB/TDB database

Bensikin application

on top of SNAP database

Some reminders on project organization

SOLEIL ressources in charge of the project

- SOLEIL contact
  - raphael.girardot(at)synchrotron-soleil.fr
- All source code is publicly available on tango-cs SVN Sourceforge project
- Reminder : Last Archiving packages are available on SOLEIL external MAVEN repository:

http://www-controle.synchrotron-soleil.fr:8001/packages/

- Features requests or bug tracking
  - Please use the SourceForge tracker
- Your feedback or contribution of any kind (like enhancing installation procedure, source code patch , etc..) is very appreciated

Operational figures at SOLEIL on the different archiving services

Databases figures

- **Accelerator** (ORACLE):
  - **HDB**: 16868 attributes, (+3% since last Tango meeting)
  - **TDB**: 8900 attributes, (+0% since last Tango meeting)
  - **SNAP**: About 2000 snapshots per Run
    - Critical for Accelerator operation
  - **ALARM** : Is now in official production state
- **Beamlines**:
  - **HDB** (ORACLE):
    - 10 up to 450 attributes / beamline,
    - used on all beamlines
  - **SNAP** (MYSQL):
    - Seldom use on most lines
    - Critical for PX1/PX2, SIRIUS
  - **ALARM** : Not deployed

A first analysis of the current situation

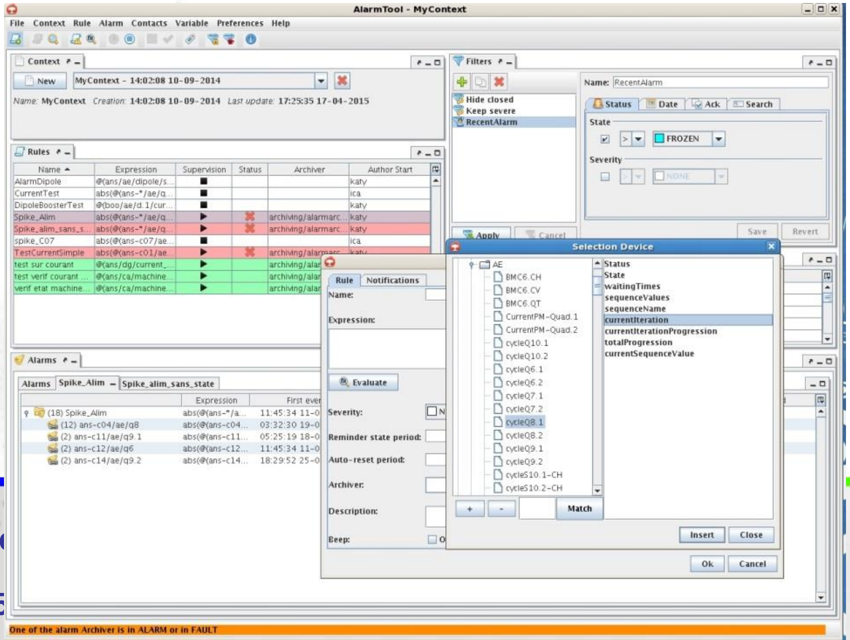
- **These different archiving system are very stable from the computing groups in charge of their daily operation (less than 10 incidents per year)**
  - *This is the result of about 10 years of work of SOLEIL software team to iteratively enhance the current software*
- **For SOLEIL operational people these databases services are :**
  - *Mandatory for Machine operation*
  - *Mandatory for automated beamlines operation (like MX ones)*
  - *Useful for the majority of the beamlines*
- **We observe that the number of data archived is quite stable**
  - *On explanation is the lack of data mining applications that could help Machine or Beamlines experts better understanding accelerator/beamline behavior*
  - *We are also lacking "accelerator" data analysts that could use data mining techniques to find correlations thanks to "massive " data storage*

What are the news for the Tango archiving services

since last Tango meeting

Alarm Tool is a new database service officialy in production at SOLEIL

- Improvement of the database to use a common scheme for PANIC and AlarmTool
- Finalization the AlarmDatabaseAccess device to insert rules and events without AlarmTool GUI (MaxIV requirement).
- Added a Notification service mail and textalker device
- Improvement of the GUI, to sort the alarms and select device and attributes with a Tango Tree.
- **Start snapshot integration to launch a snapshot on alarm event**.



MyContext - 14:02:08 10-09-2014

Name: MyContext *Creation:* 14:02:08 10-09-2014 *Last update:* 17:25:35 17-04-2015

Name	Expression	Supervision	Status	Archiver	Author Start
AlarmDipole	@(ans/ae/dipole/s...	■			katy
CurrentTest	abs(@(ans-*/ae/q...	■			ica
DipoleBoosterTest	@(boo/ae/d.1/cur...	■			katy
Spike_Alim	abs(@(ans-*/ae/q...	▶	✖	archiving/alarmarc...	katy
Spike_alim_sans_s...	abs(@(ans-*/ae/q...	▶	✖	archiving/alarmarc...	katy
spike_C07	abs(@(ans-c07/ae...	■			ica
TestCurrentSimple	abs(@(ans-c01/ae...	▶	✖	archiving/alarmarc...	katy
test sur courant	@(ans/dg/current_...	▶		archiving/alar	
test verif courant ...	@(ans/ca/machine...	▶		archiving/alar	
verif etat machine...	@(ans/ca/machine...	▶		archiving/alar	

Hide closed
Keep severe
RecentAlarm

Name: RecentAlarm

State
☑ > FROZEN

Severity
☐ > NONE

Alarms | Spike_Alim | Spike_alim_sans_state

	Expression	First ever
(18) Spike_Alim	abs(@(ans-*/a...	11:45:34 11-0
(12) ans-c04/ae/q8	abs(@(ans-c04...	03:32:30 19-0
(2) ans-c11/ae/q9.1	abs(@(ans-c11...	05:25:19 18-0
(2) ans-c12/ae/q6	abs(@(ans-c12...	11:45:34 11-0
(2) ans-c14/ae/q9.2	abs(@(ans-c14...	18:29:52 25-0

Name:

Expression:

Severity: ☐ N

Reminder state period:

Auto-reset period:

Archiver:

Description:

Beep: ☐ O

Selection Device

AE
BMC6.CH
BMC6.CV
BMC6.QT
CurrentPM-Quad.1
CurrentPM-Quad.2
cycleQ10.1
cycleQ10.2
cycleQ6.1
cycleQ6.2
cycleQ7.1
cycleQ7.2
cycleQ8.1
cycleQ8.2
cycleQ9.1
cycleQ9.2
cycleS10.1-CH
cycleS10.2-CH

Status
State
waitingTimes
sequenceValues
sequenceName
currentIteration
currentIterationProgression
totalProgression
currentSequenceValue

One of the alarm Archiver is in ALARM or in FAULT

What SOLEIL delivers to the Tango community ?

- *SOLEIL delivered 5 ARCHIVING_ROOT packages on our maven web site containing the copy of the software deployed in SOLEIL facility*
- *The ARCHIVING_ROOT package (a single zip file) contains all the software components required to provide the databases services on a Tango Control System:*
  - *Java API*
  - *Tango devices*
  - *GUI applications*
  - *A Release Note file gives the modifications since last package version*
- *Since last Tango meeting the "Tango Alarm Tool database service " is packaged within the ARCHIVING_ROOT package*
- *Tens of software modifications have been done to the various software components mostly for bug fixes, and "small evolutions" and performances improvement for data visualization*
  - *Look at the release notes for details*

Modifications of the last 3 ARCHIVING_ROOT packages

- ================================
- release 15.2.1 (February 2015):
- ================================
- - SnapManager/Bensikin: setEquipments works even if value is not formatted as an int (TANGOARCH-399)
- - SnapManager: Extract the correct data type from database (TANGOARCH-401)
- - SnapManager: Database case sensitivity bug correction (TANGOARCH-409)
- - Bensikin: NullPointerException avoided (TANGOARCH-409)

- ================================
- release 15.1.3 (January 2015):
- ================================
- - Former watchers restored for external uses
- - All linux scripts: start with empty CLASSPATH

- ================================
- release 15.1.2 (January 2015):
- ================================
- - AlarmTool: Fix AlarmArchiver template (TANGOARCH-388)
- - AlarmTool: Add AlarmDataBaseAccess device (TANGOARCH-362)
- - All linux scripts: Better CLASSPATH management (CI-414)
- - Former watchers removed

Foreseen evolutions

SOLEIL vision on the next evolutions of these database services

- From our end user point of view most of the enhancements they are asking for are related to GUI applications (mambo , bensikin and alarmtool applications)
- From the SOLEIL Software development team point of view the current software implementation suffers from :
  - *Many copy/past codes between API and DeviceServers*
  - *Data collection mechanisms is based on an old ATK core version*
- Storing larger amount of data by using NoSQL databases is not required by end users
  - *But appears as a driver to push for more “accelerator data analyst” activities*
  - *Which will required usage of “data mining” tools outside of our development scope*

SOLEIL vision on the future steps for Tango databases development

- SOLEIL has a **12 years** experience in developing and maintaining the “Tango databases services” for the community
- Our resources for development are nowadays limited and are focused on our end user’s request (mostly GUI features)
- We can **participate** in any initiative of modifying internal software mechanisms
  - *Such as data collection mechanisms*
  - *Or data storage and extraction in NoSQL databases*
- We can participate in specifications, design and tests phases
- In all cases , the services delivered to the end users must **be at least the same**
  - *Same GUI they are familiar with*
  - *Same Tango DeviceServers interfaces to keep integration with other control system software compatible*

The HDB++ initiative is a very promising one

Further discussions during this meeting should be fruitful to keep the databases services provided with TANGO coherent for the community