Modernizing TANGO Archiving at Synchrotron SOLEIL

J.Guyot: database administrator

R.Girardot: developer

A.Tison: engineer apprentice, developer

G.Abeillé: developer, product owner

SOLEIL II
La science éclaire l'avenir
Science lights up the future

Context

A critical system for SOLEIL operations

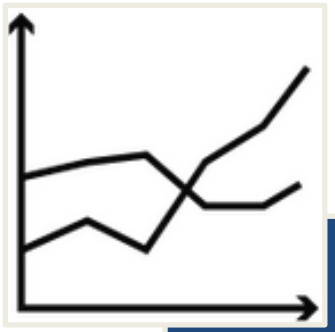
Handles **high-frequency time-series data** & **on-demand snapshots**

In production since 2006,
Used on accelerators, 29 beamlines and 5 labs

+40k attributes collected
Originally based mostly on **Oracle DB** (and MySQL)

Migration to **PostGreSQL/Timescale** in progress

Legacy System Overview

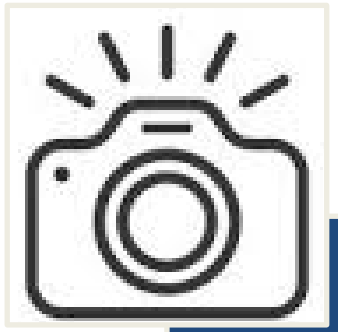


Historical (HDB) and Temporary (TDB)

- Collect attributes w/ TANGO polling or events
- TDB reset at each machine shutdown
- HDB persistent data since 2006

ORACLE DATABASE

MySQL



Snap

- On-demand collect of a set of attributes
- Compare & restore system states

Legacy HDB/TDB architecture

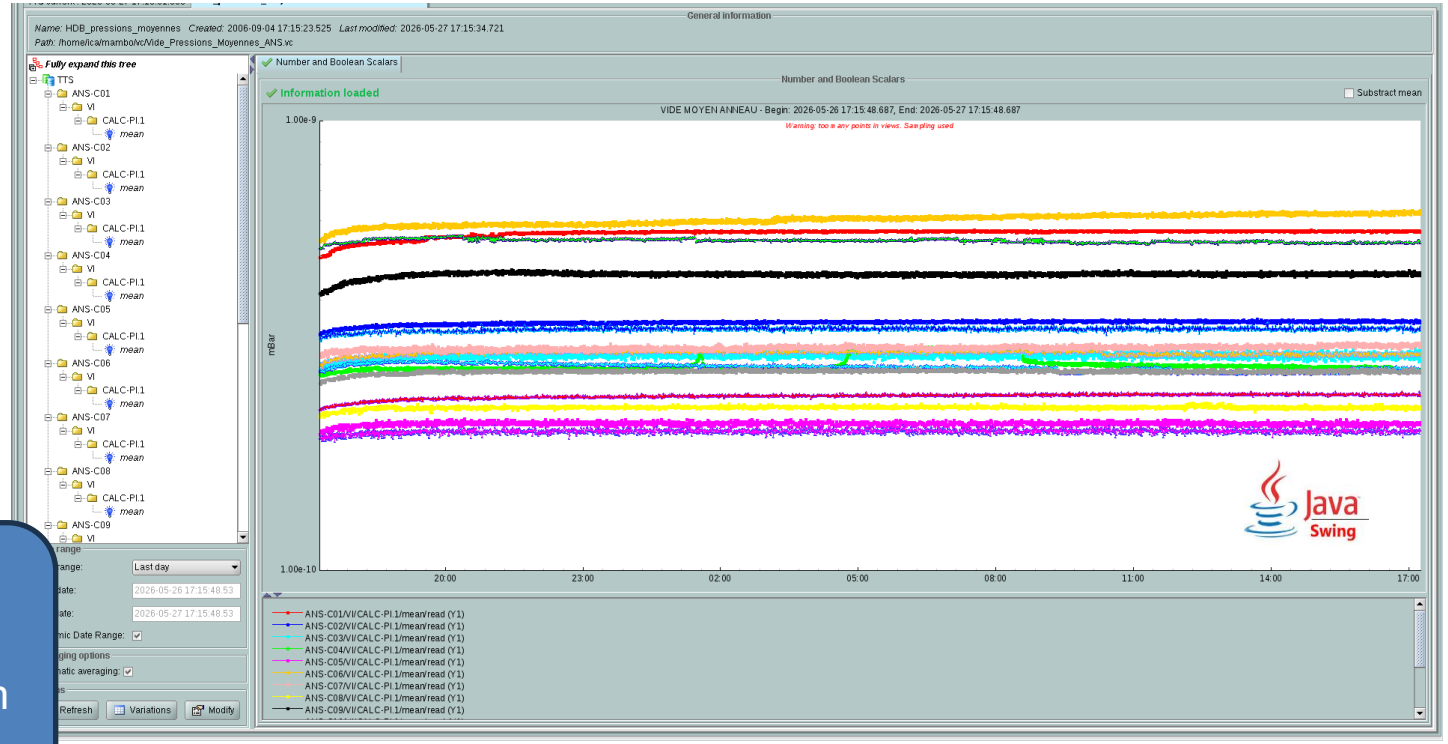
Manager device
start/stop attributes

Watcher device
monitor archiver & check last inserted values

Extractor device
fetch values from db

Archiver device
collect & insert

Mambo
start/stop attributes
fetch values from db



Java

HDB/TDB API
Insert / Fetch

GraphQL extractor API
fetch values from db

Database

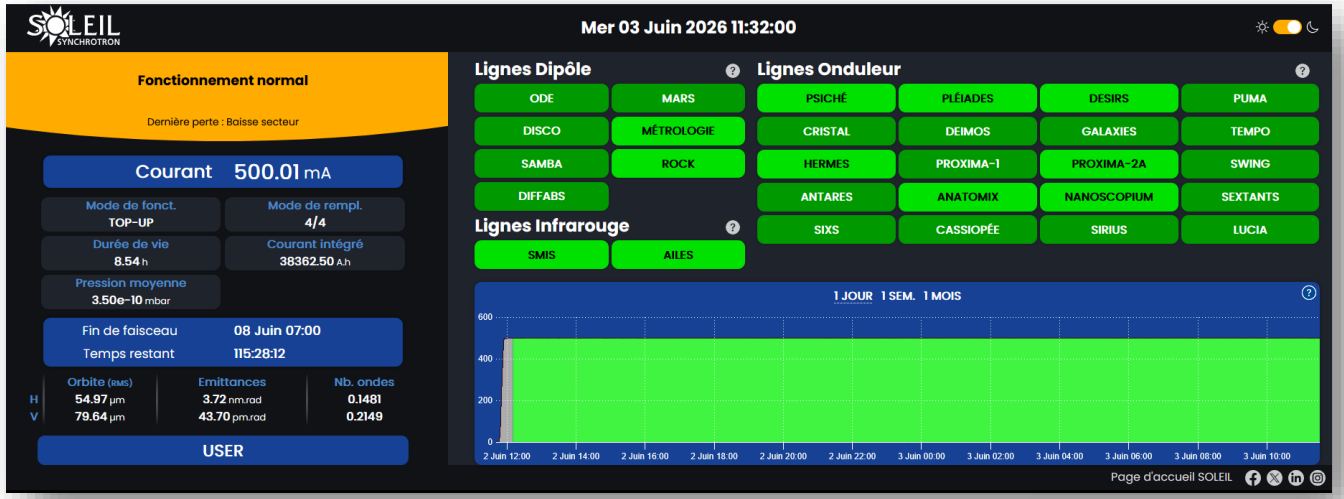
MySQL

ORACLE DATABASE

-> details on next slide

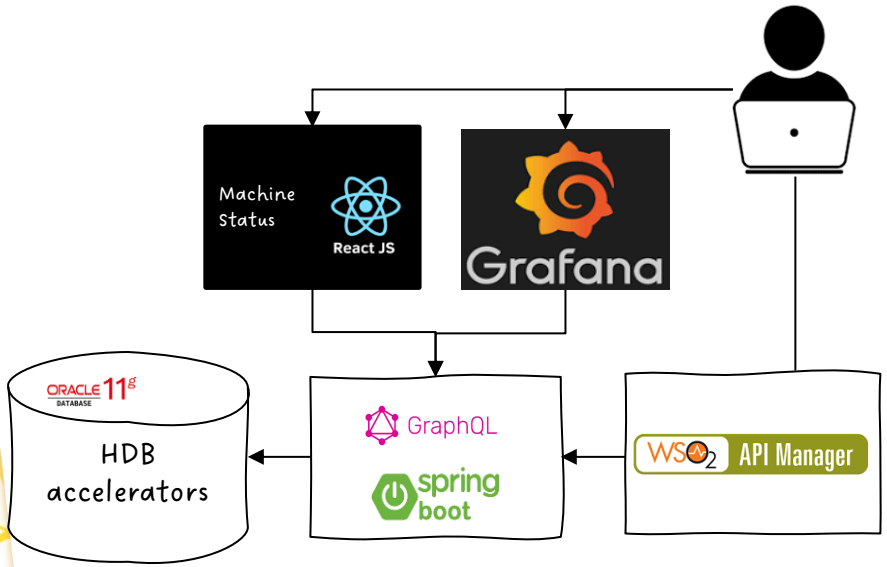
SOLEIL II
La science éclaire l'avenir
Science lights up the future

Web apps upon HDB



Machine status

Grafana dashboards



GraphQL API to archiving

Legacy Snap architecture

Manager device
configuration

Bensikin
configuration
fetch values

Archiver device
collect & insert
Manage contexts

Extractor device
fetch values from db

Snap API
Insert / Fetch in PG

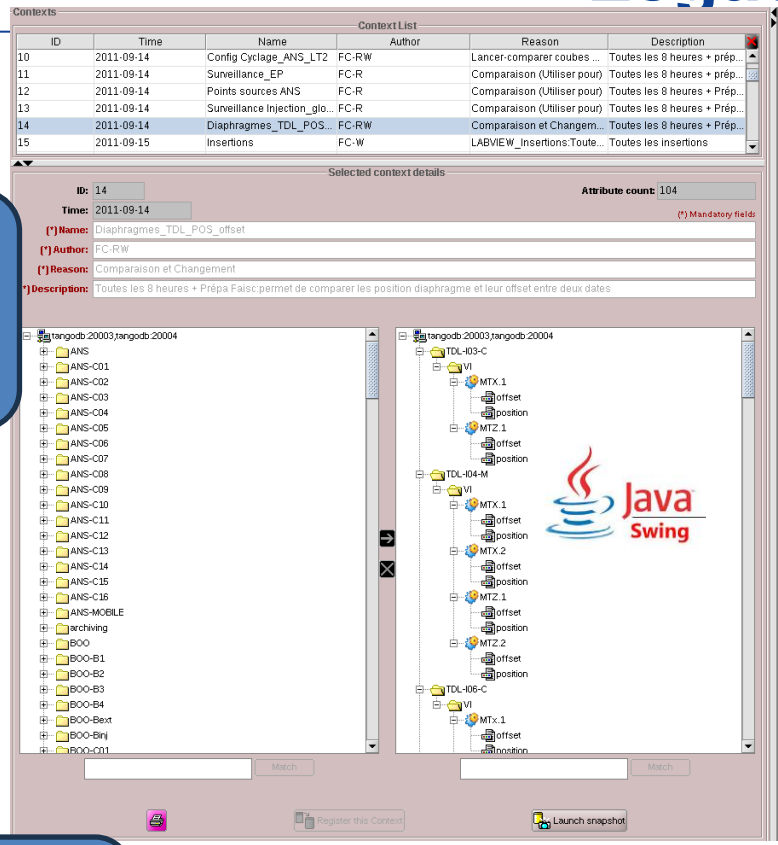
Python GUIs
Developed by operators

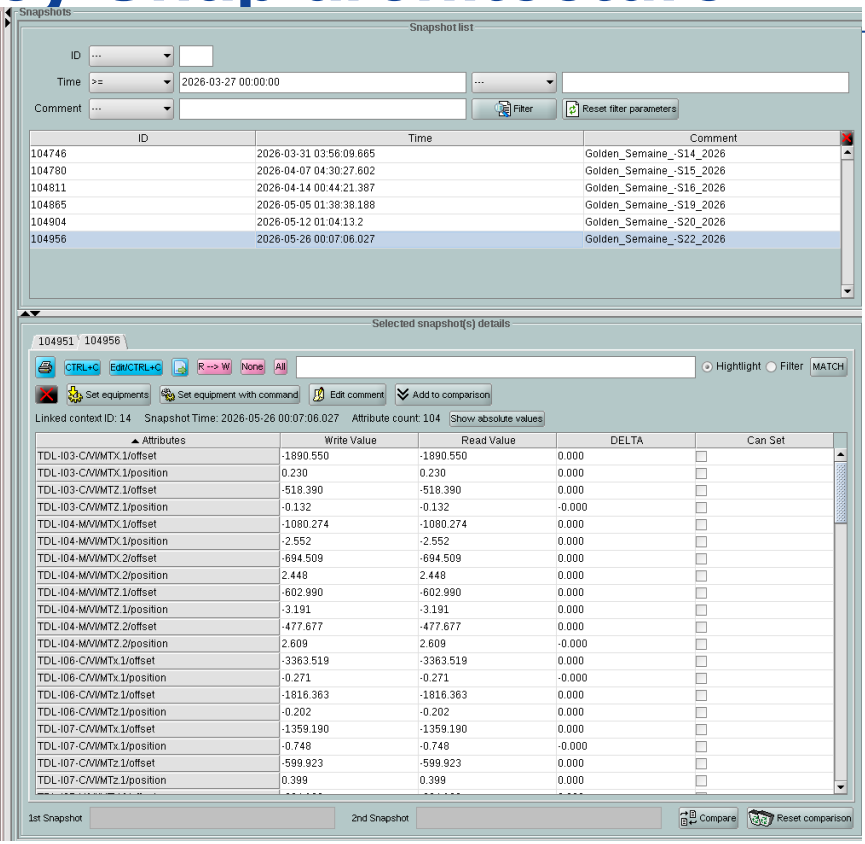
Java

MySQL

ORACLE DATABASE

Database





Modernizing HDB/TDB

SOLEIL II
La science éclaire l'avenir
Science lights up the future

Key Challenges & Constraints

Why was change needed?

- 

High **Oracle licensing costs**
- Aging and complex codebase (especially legacy DB APIs)
- Manage increasing data volume for SOLEIL II
- 👉 Need for **modern, scalable, maintainable solution**

Migration constraints:

- No operational disruption demand
- Maintain existing user interfaces
- Migrate existing TDB/HDB configuration with polling and events.
- Leverage internal expertise (SQL / DBA / Java)

Target Solution

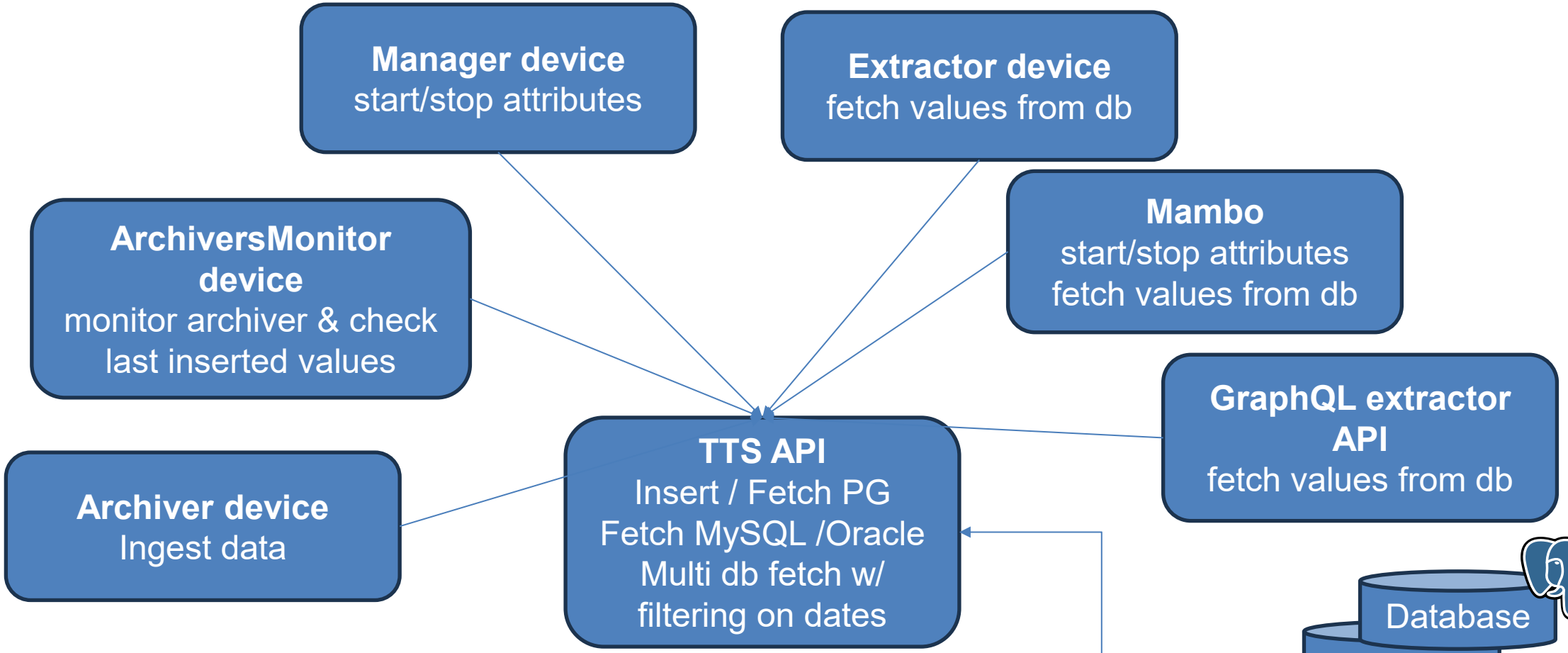
Introducing Tango TimeSeries (TTS):

- In place of legacy HDB and TDB → Keep interfaces, modernize backend

- PostgreSQL/TimescaleDB w/ HDB++ DB schema
- But with an added table « modes » for archiving configuration :
  - Maintain existing HDB/TDB modes: periodic, difference, relative, theshold, event
  - Allow to migrate smoothly existing conf. from TDB/HDB

- New Java API layer for DB access
- Add transparent access to multiple databases :
  - to legacy HDB
  - and different cold TTS DBs (one per year)

TTS software architecture



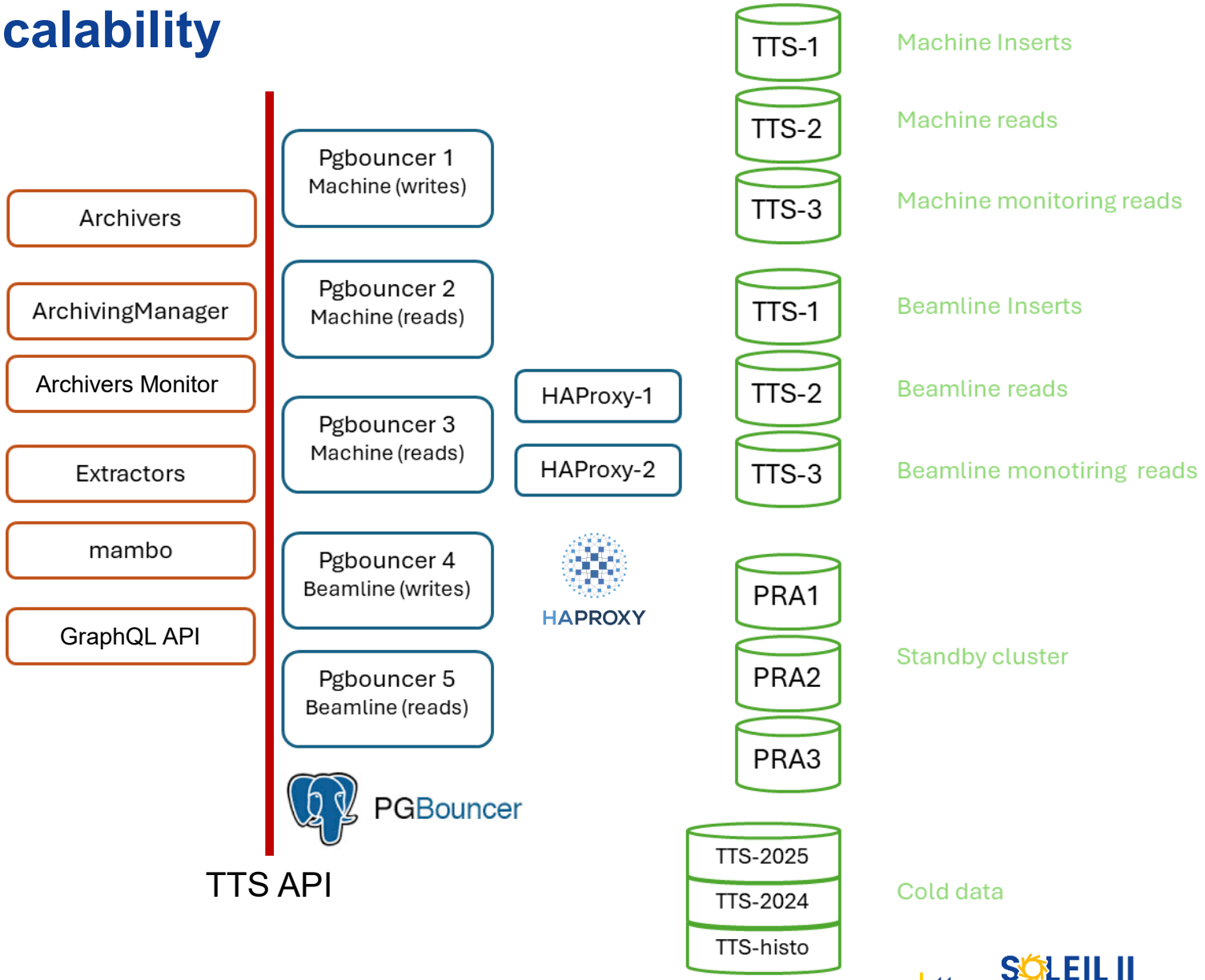
Still in Java 8 / migration to Java 21 in progress

jdbi 光 HikariCP Testcontainers Guice

TTS infrastructure

Designed for **high availability & scalability**

- 3 clusters (VM servers):
  - Machine
  - Beamlines
  - Standby (disaster recovery)
- 3-node cluster setup:
  - 1 write node
  - 2 read nodes
  - Dedicated databases for cold data
- Key components:
  - HAProxy (failover)
  - PgBouncer (connection pooling)
- Backups:
  - Full: weekly (300 GB, 1h45)
  - Incremental: daily (10 min)



TTS migration strategy

- **Progressive and safe transition**
- Coexistence of:
  - Legacy systems (HDB/TDB)
  - New system (TTS)
- Multi-DB access layer: → transparent for users
- Gradual migration: Accelerators, Labs, Beamlines
- 👉 Zero disruption for operations

- **Current situation:**
  - ✅ Accelerators:
    - TTS in production since 2024
    - HDB insertion stopped (2026)
  - ✅ Labs: Fully migrated (2025–2026)
  - 🔄 Beamlines: 7 / 29 migrated. Migration of all beamlines expected for end of summer 2026.
  - 🔄 Preparing Grafana infrastructure for beamlines

Data Strategy: Managing data growth

- Accelerator workload:
  - 35,700 attributes
  - ~2,500 inserts/sec
  - 1.2 TB of data per year

- Retention policies depends on each use case:
  - For beamlines: one year is sufficient for some of them, some want to keep everything
  - For accelerators: keep some data at vitam eternam, some for one year
  - Data decimation may be used, decimation strategy has to be defined per attribute

- Optimization techniques:
  -  Timescale compression (×7 gain)
  -  Pause some attributes during technical shutdowns for accelerators
  -  Delete some attributes older than a period of time, in accordance to user needs
  - Upcoming Decimation: to be implemented (avg, lttb, asap_smooth…)

Modernizing Snap

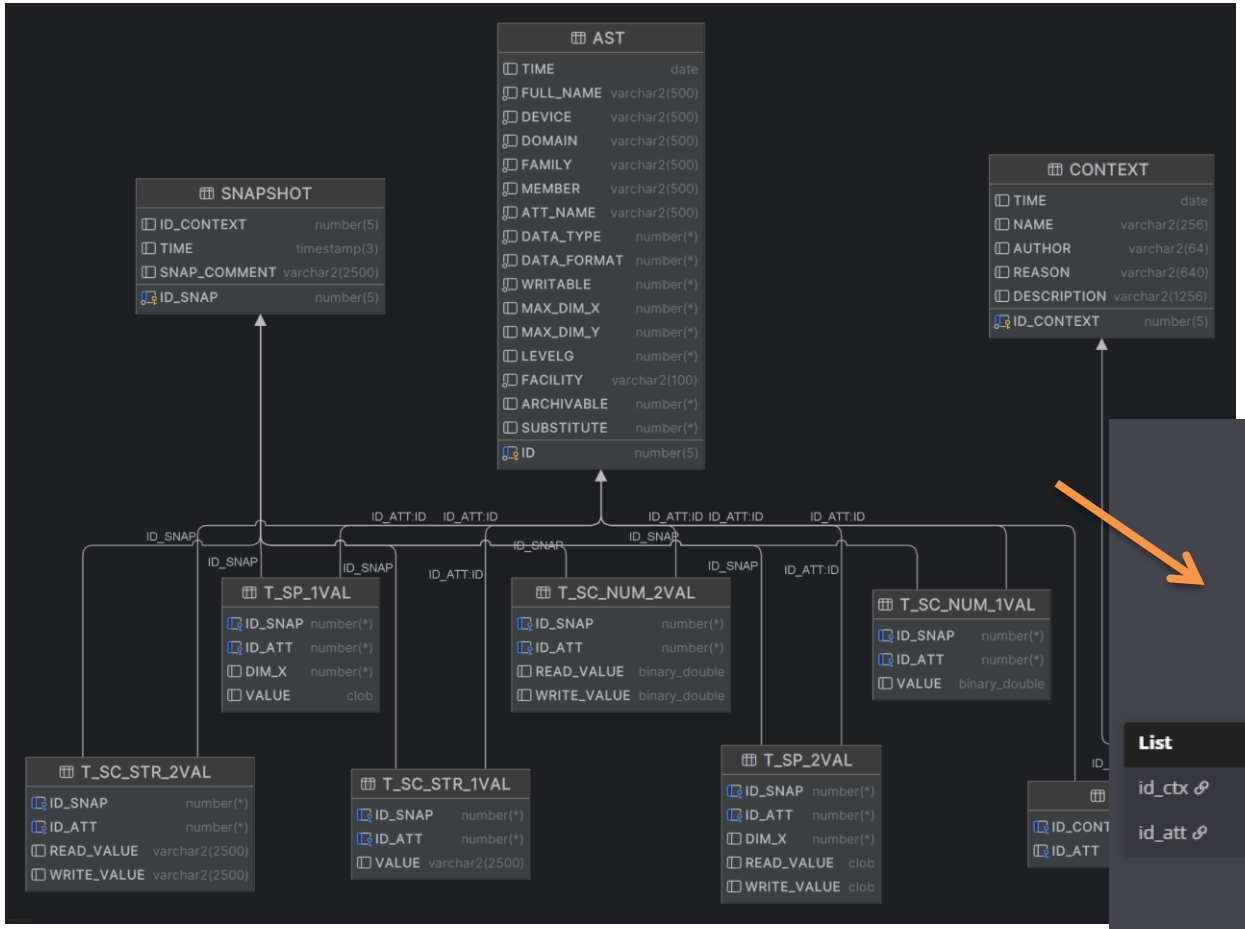
SOLEIL II
La science éclaire l'avenir
Science lights up the future

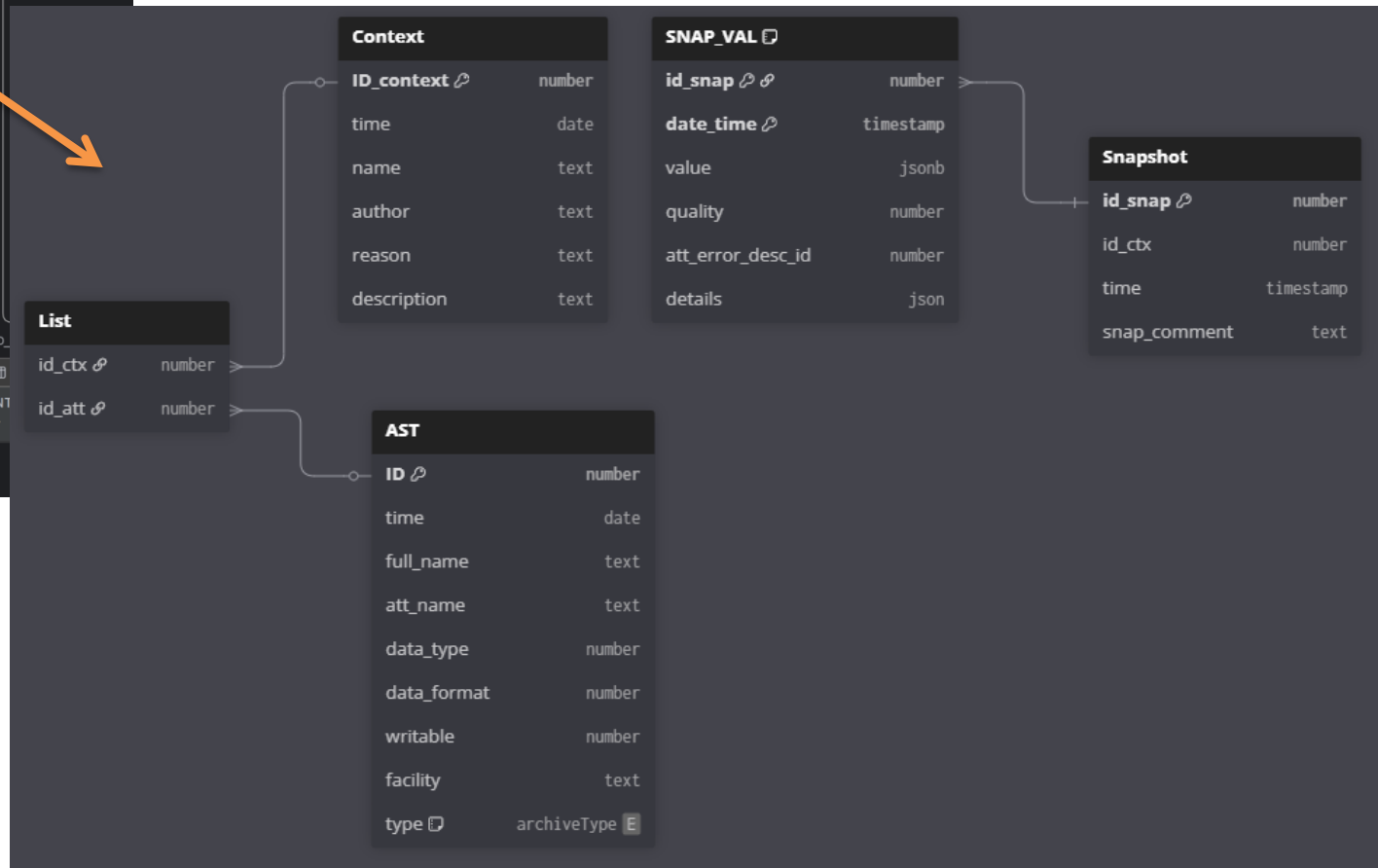
SNAP Modernization

- We have applied the same strategy as for TTS:
  - Keep existing interfaces (devices and GUI)
  - Exit from Oracle, and migrate to PostGreSQL DB
- Improvements :
  - Simplified DB schema
  - Merge 2 devices (Archiver/Manager w/ code duplication)
  - Replace old API and devices, keep Bensikin
  - Prepared for future extensions
    - Like snapshotting properties in addition of attributes values

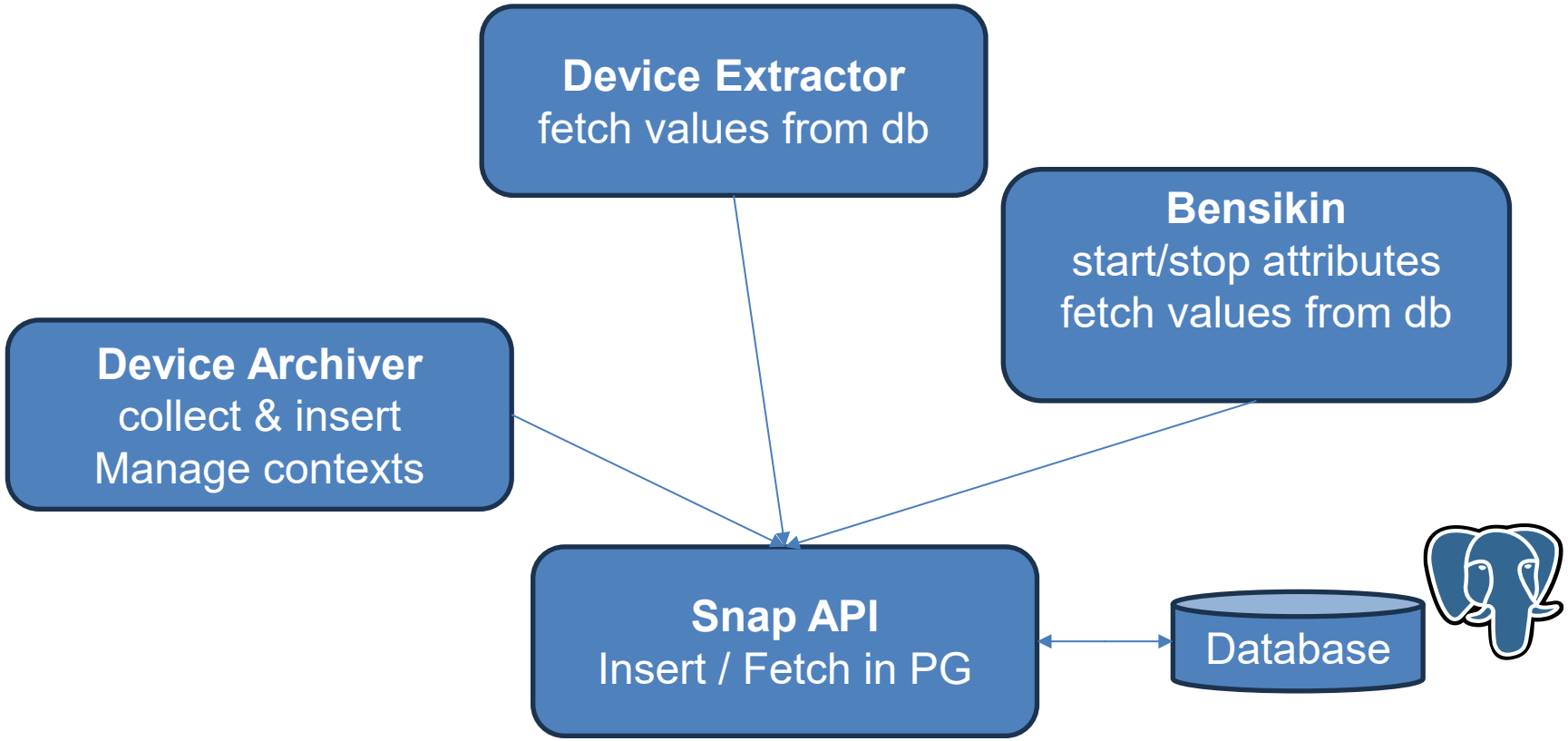
SNAP database revamping

- The old DB schema was containing several tables for data
- In PostGreSQL, only one table for values, using JSONB column type





Snap Software architecture



Java 21 jdbi 光 HikariCP Testcontainers Guice

SNAP Status

- **Current progress**
- ✅ Development complete
  🔄 Testing on accelerators
- Next steps:
  - Fix remaining issues
  - User validation
  - Production rollout on accelerators & beamlines

Conclusion

SOLEIL II
La science éclaire l'avenir
Science lights up the future

Next Steps

- Complete TTS beamline migration
- Implement TTS data decimation
- Validate SNAP in accelerators, and migrate beamlines afterwards
- Improve CI/CD pipelines
  - Jenkins → GitLab CI
  - Java 21 upgrade in progress
  - But binaries are still on an internal repository (artifactory)
  - Meanwhile, manual publication of a zip package containing all binaries and scripts
- Code:
  - https://gitlab.synchrotron-soleil.fr/software-control-system/tango-controls-archiving
- Package:
  - https://gitlab.synchrotron-soleil.fr/software-control-system/packaging/archivingroot/-/releases

Acknowledgments

- SOLEIL:
  - Acceleratos: Thomas Marion, Xavier Delétoille, Laurent Nadolski
  - Experiments: Alina Vlad, Sebastian Schoader, Denis Menut
  - Controls: Patrick Madela, Hiba Kendi, S.Lê, Yves-Marie Abiven
- And the TANGO community