**GENERAL PROBLEM**
- Diverse experiment control apps across different facilities
- Multiple tools for different DAQ modes within the same lab
- Users lack of time to understand DAQ architecture

**SOLUTION**
- Flexible GUI and CLI interfaces to the CS
- Active synoptic as navigation tool to instrument panels
- Python based macro execution environment
- Turn-key step and continuous scan macros
- Scan framework for development of custom scans
- HDF5, SPEC, FIO or any custom data format
- 1D and 2D data references accessible via HDF5 VDS
- Generic DAQ synchronization (SW or HW)
- Single, grouped and autonomous DAQ modes
- Generic interface of experimental channels

**TECHNICAL PROBLEM**
- Rich in features DAQ HW are hard to abstract in SW
- Detectors HW produce larger volumes of data at higher rate

**BENEFITS**
- Improved UX at different sites within Sardana and Taurus collaboration
- Gentle learning curve to the CS:
  - Consistent look-and-feel
  - SPEC-like macro execution syntax
- Well isolated layers of DAQ SW
- Easier to understand DAQ architecture
- Promote code reusability
- Facilitate new developments

**NEXT**
- Reduce dead times in scans
- Finish implementation of autonomous DAQ
- Refactor plugin system
- Generalize capabilities
- Support non-linear motion trajectories

**MORE INFO**
- [https://sardana-controls.org](https://sardana-controls.org)
- [https://taurus-scada.org](https://taurus-scada.org)