About Me

- Marcin Nowak, Software Engineer, Polish
- Joined BNL PAS group in 2006 to work on the ATLAS Offline Software
- Stationed full time at CERN (building 32-S-C10, PAS office shared now with Johannes and sometimes with group member visiting CERN)
- I now work full time on the ATLAS Core Software in the Event Store group
  - with Peter Van Gemmeren (ANL) and recently Serhan Mete (ANL)

ATLAS Offline Software Stack

- Based on the LCG project common software (LHC Computing Grid)
  - includes ROOT
- ATLAS Externals
  - Packages not developed by ATLAS and not common enough to be in LCG (some in external repositories, downloaded during release build process)
- ATLAS Online Software TDAQ layer
- Gaudi Framework (the LHCb software framework)
  - ATLAS maintain their own clone repository
- **Athena**
  - The offline production software framework for ATLAS
  - Open source since the beginning of this year https://gitlab.cern.ch/atlas/athena
  - Cite as:
    - ATLAS Collaboration. (2019, April 9). Athena (Version 22.0.1). Zenodo http://doi.org/10.5281/zenodo.2641997

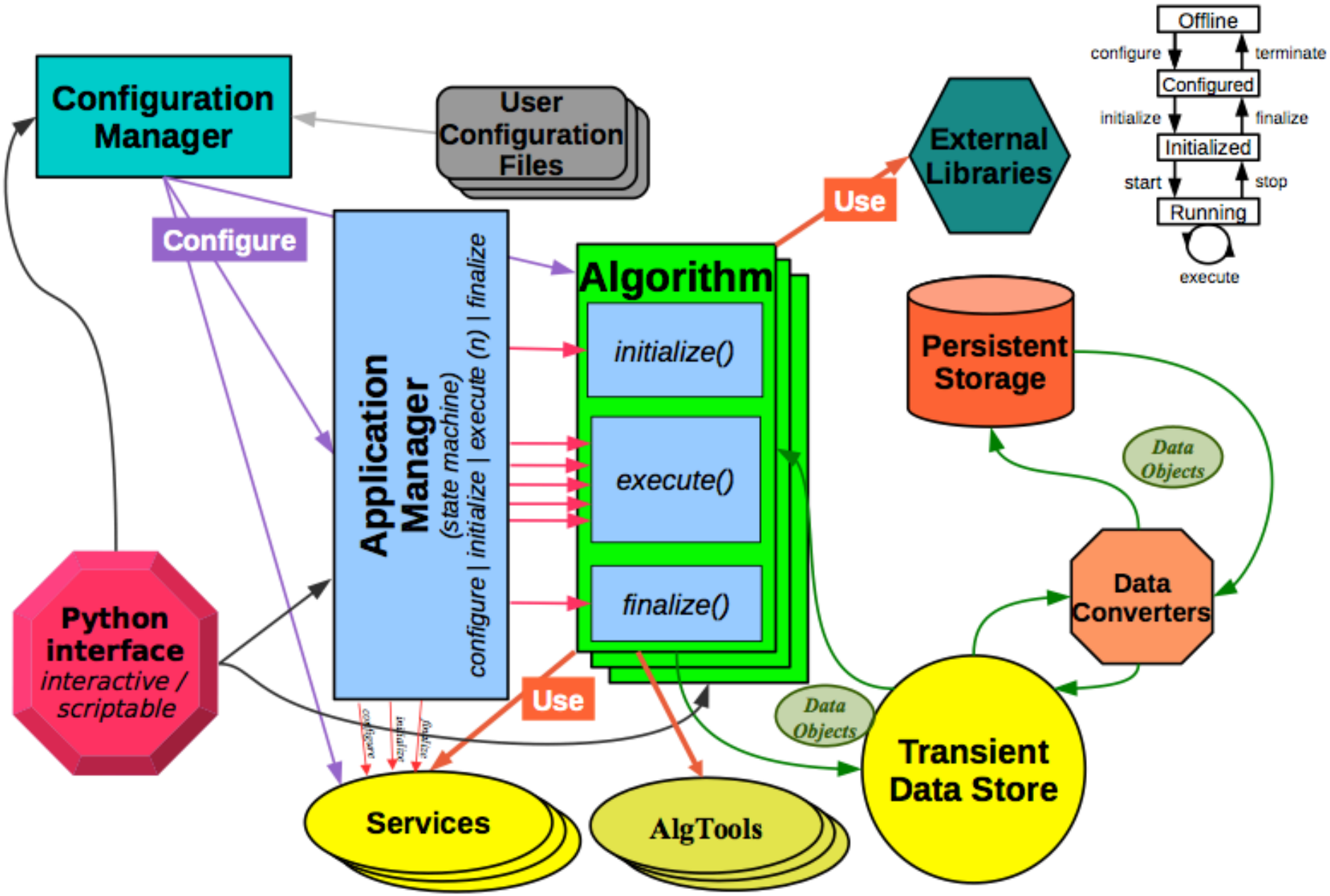
Athena Software Infrastructure

- Athena source code is now managed in CERN hosted GitLab (before in CVS and SVN + custom Release Management tools e.g. TagCollector)
- The code is divided into over 2125 packages, but they are all in one big project
- 10 active branches - some diverging a lot, built nightly
  - Automatic migration of commits between branches, but also manual management
- Every developer has a private repository fork in GitLab where they can commit changes
  - Changes are submitted to the main repository branch as Merge Requests (via web interface)
  - MRs are reviewed by shifters and experts for compliance with ATLAS coding rules and for desirability of proposed changes
  - Continuous Integration (Jenkins) automatically compiles the entire project + MR and runs tests
  - Changes accepted by the release coordinator if everything OK
- This development model is relatively new in ATLAS and dramatically improved code quality in the master branch (even though introducing changes takes now longer and requires more manpower)

Athena Software Development

- Athena: ~5 million lines of code - 80% C++, rest Python (mostly configuration)
  - 20 years old, hundreds of developers
- Compiled with gcc 6/8 – also Clang/LLVM (ROOT) on Scientific Linux 6/7
  - building takes 5-8h on 16 cores system
  - most branches compiled nightly, possibly in several flavors
  - one debug build takes over 100GB disk space
- Nightly builds are installed on CVMFS and kept for a month
  - developers can work against any of these builds (plus official numbered releases)
  - only need to compile selected packages (and dependencies!)
  - but the local git clone has to have the right version checked out
- JIRA - issue and task tracking system
  - integrated with GitLab - tickets are updated when MRs addressing a given problem are merged into branches

Gaudi/Athena Components and States



What Do We Do in the Event Storage Group

- General mandate is to provide mechanisms to store transient Event Data that can be found in the application memory and to retrieve it later.
  - Support for really complex C++ types (~3000 types)
  - Support for data model changes over time (schema evolution)
  - Ensuring storage size efficiency
  - and I/O performance
- Navigation (identification and retrieval of single events)
- Storage technology independence (mostly)
- Recently: parallel processing (AthenaMP output merging)
- New: multi-threading support

My Past Projects (for ATLAS)

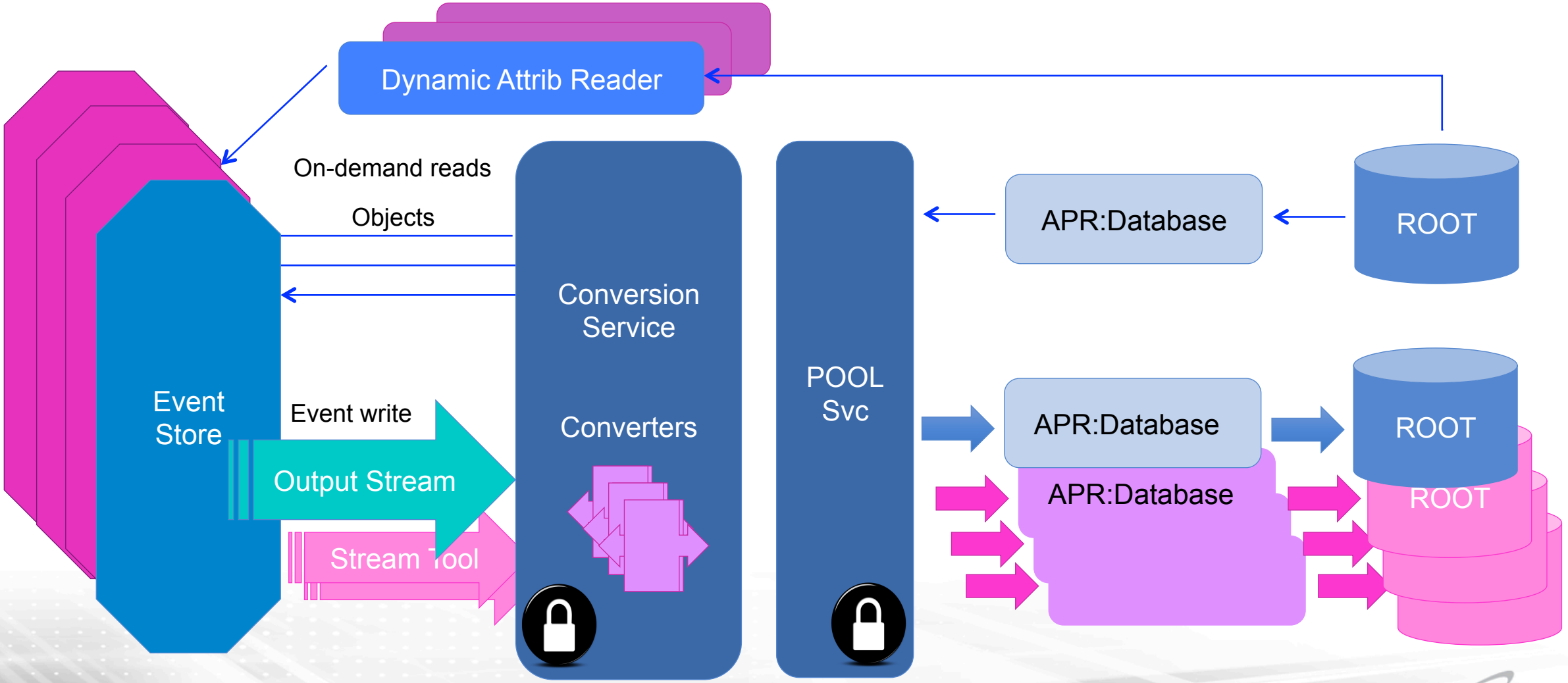
- Schema Evolution support for Run1 Event Data (T/P separation/conversion layer)
  - to make sure we can “always” read old data
  - implemented by introducing separate independent persistent EDM with versioned classes and converters
  - mostly replaced now by the new Run2 EDM (xAOD) but still existing for some classes. Allows Athena to read Run1 data still.
  - good mechanism for reducing storage size and increasing I/O performance
    - Data selection, compression and reordering
- Event TAG Database
  - first implementation of an information system containing summary information of all Events for different processing stages RAW/ESD/AOD
  - Now replaced by Event Index
- I/O implementation for ATLAS Run2 EDM (xAOD types)
  - xAOD (C++ objects) can be extended with arbitrary new data members at runtime and ATLAS had no I/O infrastructure to handle such dynamically defined objects
- Athena migration from ROOT5 to ROOT6
  - Athena uses selection.xml files to tell ROOT which C++ types need dictionaries for I/O
  - CINT, Reflex and Cintex gone with ROOT5, code parsing with Clang in ROOT6

My Current Activities

- AthenaMT - multi-threading Athena that is able to process multiple Events concurrently
  - Big and important project for the entire Core Software and other groups
    - all components must be MT-safe
  - Well advanced – simplified production jobs can be executed
    - Physics validation planned for beginning of 2020
  - I/O components are MT-safe, working on increasing concurrency

- Athena I/O layer review
  - Activity initiated by Torre 2 years ago to identify steps required to prepare for Run3 and Run4 increased data rates
  - Ongoing work to improve I/O code quality and performance

AthenaMT I/O Layer Overview



Current Activities (2)

- EventInfo – related changes
  - Migration to xAOD::EventInfo
  - Introduction of mini-EventInfo
    - Low level ROOT attributes independent of Event Data for fast file content scanning
- Event indexing (modified version of TTree Storage Technology)
  - Indirection layer for navigation (TTreeIndex)
  - Needed for fast (pure ROOT) file merging and concurrent writing to the same TFile
- SharedWriter – output merging for AthenaMP
    - Helping Peter van Gemmeren

New Projects

- Event Service
  - Adapting AthenaMT to continuous processing of short event ranges
    - Main challenge – handling multiple overlapping output streams
    - Implemented for Event Data, tested with mock examples
    - Working on a solution for in-file Metadata
      - Starting analysis of Simulation jobs metadata processing
- Event Streaming Service
  - Data serialization and streaming
  - Event I/O and persistency that are long range or are directed at exascale platforms