ATLAS Software Build System
Part 2 - Infrastructure

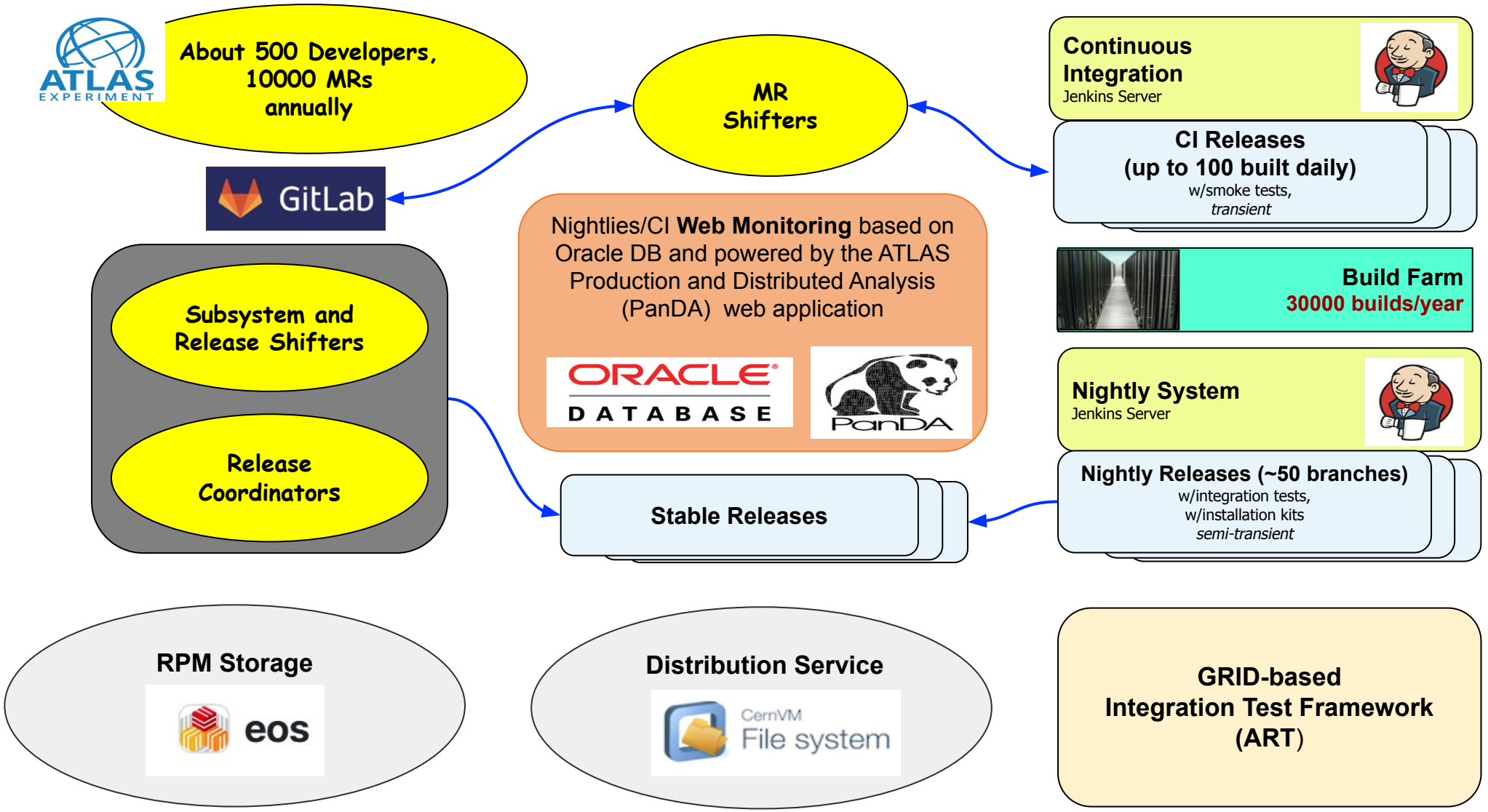
Alexander Undrus for the ATLAS ASCIG team

Joint experiment meeting
February 6, 2025

This Talk

- Outlines the infrastructure of the CI and Nightly systems
- Details installation procedures for ATLAS offline software releases
- Describes tools and techniques for release testing
- Provides insights into dynamic monitoring

Offline Software Development Workflow at a Glance



ATLAS CI System

- **Key component of ATLAS offline software workflow since 2017**
- **Jenkins based build and testing system interconnected with GitLab**
- **CI build for each GitLab Merge Request (MR) creation/update**
  - Up to 100 CI jobs daily
  - ~16500 CI jobs completed in 2024
- **Rapid unit and integration testing**
- **Efficient pipelines with dynamic optimization of build and test scale**
- **Comprehensive feedback to developers**
  - Dynamic monitoring is based on the Oracle DB technology and integrated with the ATLAS BigPanDA web service
  - Job results are posted directly to GitLab MR views
- **ATLAS teams use CERN GitLab CI for smaller projects**

ATLAS Nightly System

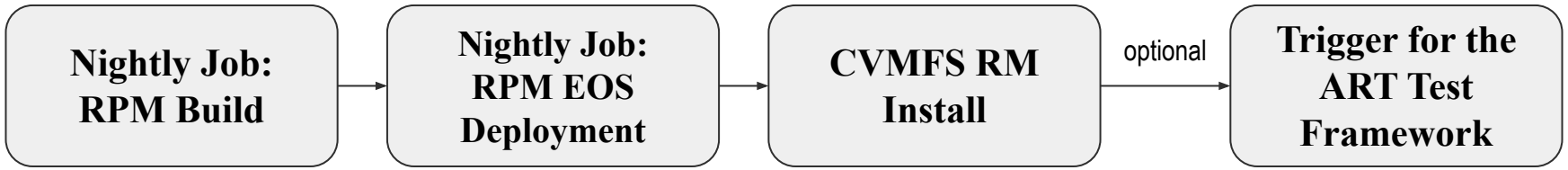
- **Validates accepted code changes every night**
- **Dedicated Jenkins automation server (separate from the CI system)**
- **~ 50 branches -** production, experimental, testing new externals, legacy
  - Optimized scheduling - some branches do not run daily
- **Support for Alma9 and ARM platforms**
  - Legacy builds within CentOS7 containers
- **~ 13200 Nightly jobs completed in 2024**
- **Employs the same dynamic monitoring as the CI system**
- **Release installation on the CVMFS file system**
  - Worldwide accessibility
  - See details in the next slides
- **Rapid fast unit testing, executed locally on build nodes**
- **Comprehensive GRID-based integration testing in the ART framework**
  - See details in the next slides

ATLAS Software Build Farm

- **18 powerful 64-core Alma9 BM nodes**
  - 251 GB RAM, 1.7 TB SSD
- **10 16-core Alma9 VM nodes**
  - 114 GB RAM, 160 GB SSD, 490 GB eph SSD, 490 GB Ceph
- **4 20-core ARM VM nodes**
  - 57 GB RAM, 200 GB SSD, 490 GB Ceph
- **Efficient use through sharing nodes between the CI and Nightly systems**
  - Priority for CI jobs at day time, nightly jobs at night time

Release Installation

- **Nightly installations**
  - Executed on **three CVMFS release managers**
  - Created from **RPM files** using **dnf5**
  - Procedure based on **Bash shell scripts**, featuring:
    - **Parallelization mechanisms**, including lock-based synchronization
    - Support for **all CVMFS transaction types**, with results analysis and notifications
    - **Storage space management**
      - Standard nightly release retention policy: 30 days



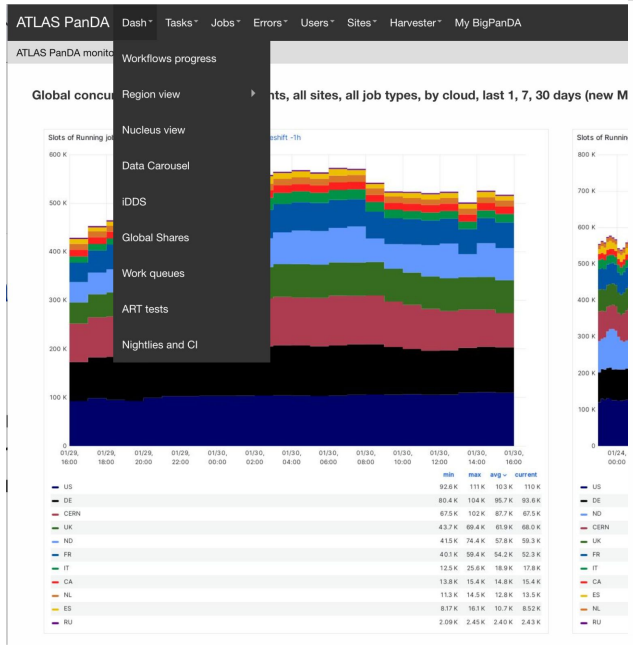
- **Stable releases installations** performed on a separate CVMFS server
- **Additional installations maintained**
  - Data files for test frameworks
  - External packages and tools
- **Key statistics**
  - **Release installation time:** 10 to 100 minutes, depending on system load and release size
  - **Peak Activity:** About 40 release installations on peak days
  - **Storage utilization:** 7.2 TB out of 15 TB available

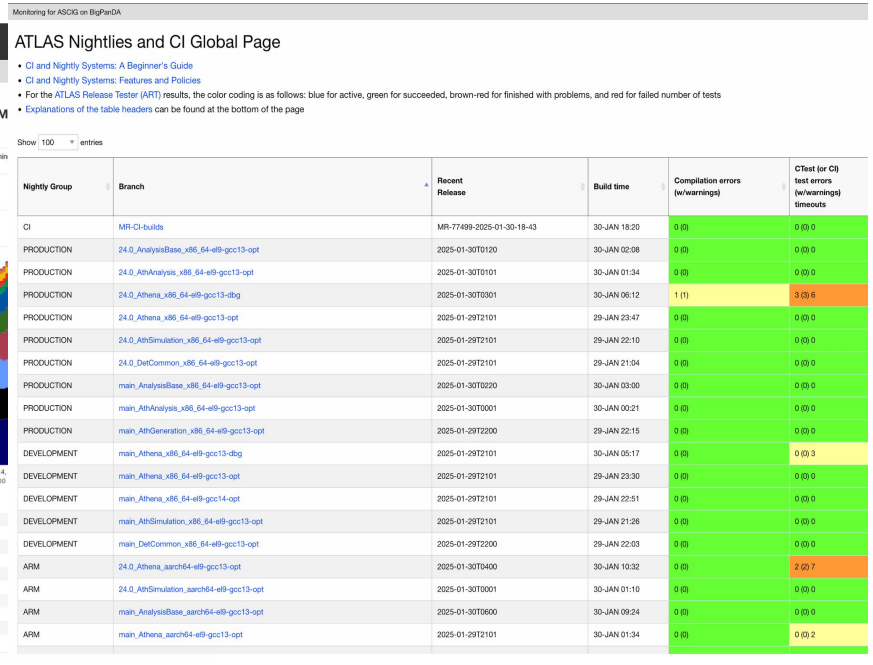
ART: ATLAS Release Testing Framework

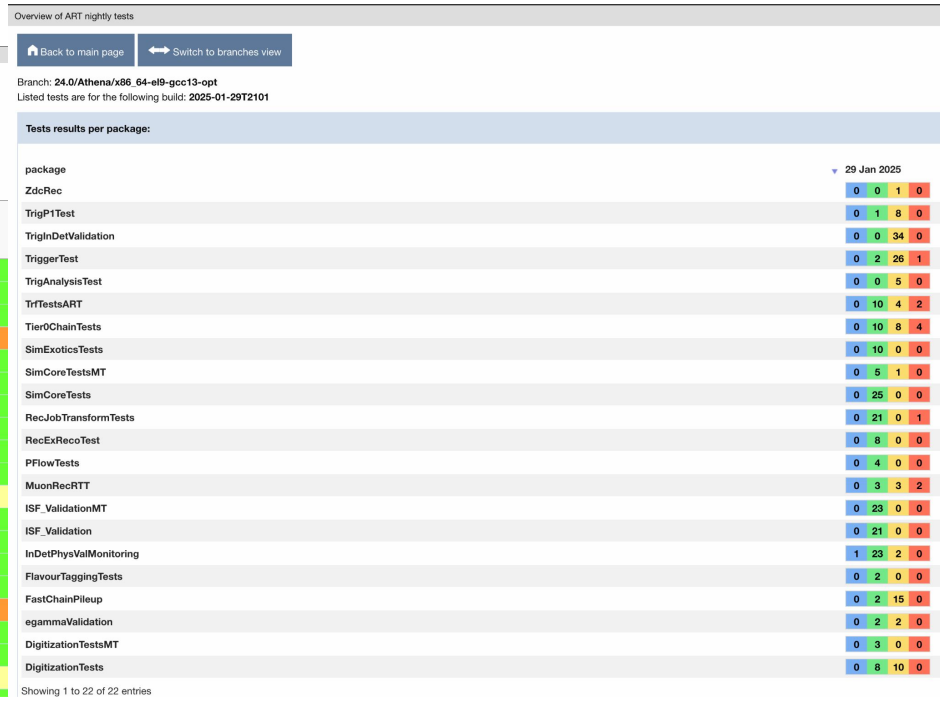
- **Python testing framework** designed to detect defects and bugs in nightly builds
- Executes tests locally on the **dedicated VMs or on the GRID**
- **Triggered by Jenkins Nightly Server** and runs on nightly builds installed on CVMFS
- **Jenkins initiates the GitLab-CI API** to start a pipeline
- **Jobs can run for up to 24 hours**, processing a high number of events, which helps uncover rarer bugs
- **Current operations:**
  - Runs for approximately 20 nightly releases each day
  - Executes thousands of grid jobs and hundreds of local node jobs daily
  - GRID job results are published to the BigPanDA web service
  - Local job results are hosted on EOS local web but will soon be integrated into BigPanDA

ATLAS BigPanDA Monitoring System

- **Django-based web application** that aggregates data from **Oracle DB** and other sources
- Provides a **wide range of dashboards**, from high-level summaries to detailed views of individual computational jobs and their logs
- **Supports dashboards for ART, CI, Nightly Systems, and many other workflows**







Conclusion

ATLAS Offline Software Build System Overview – Part 2 provided insights into:

- The infrastructure supporting the CI and Nightly systems
- Key aspects of testing and monitoring
- For details on build tools, external dependencies, platforms, and compilers, refer to **Part 1**

Acknowledgments:

Thanks to Aleksandr Alekseev, Dario Barberis, Oana Vickey Boeriu, James Catmore, John Chapman, Johannes Elmsheuser, Tatiana Korchuganova, Attila Krasznahorkay, Chris Lee, Edward Moyse, Asoka De Silva, Brinick Simmons, and Frank Winklmeier, Shuwei Ye