Traceability in ATLAS Distributed Computing

disclaimer: this is just a partial view

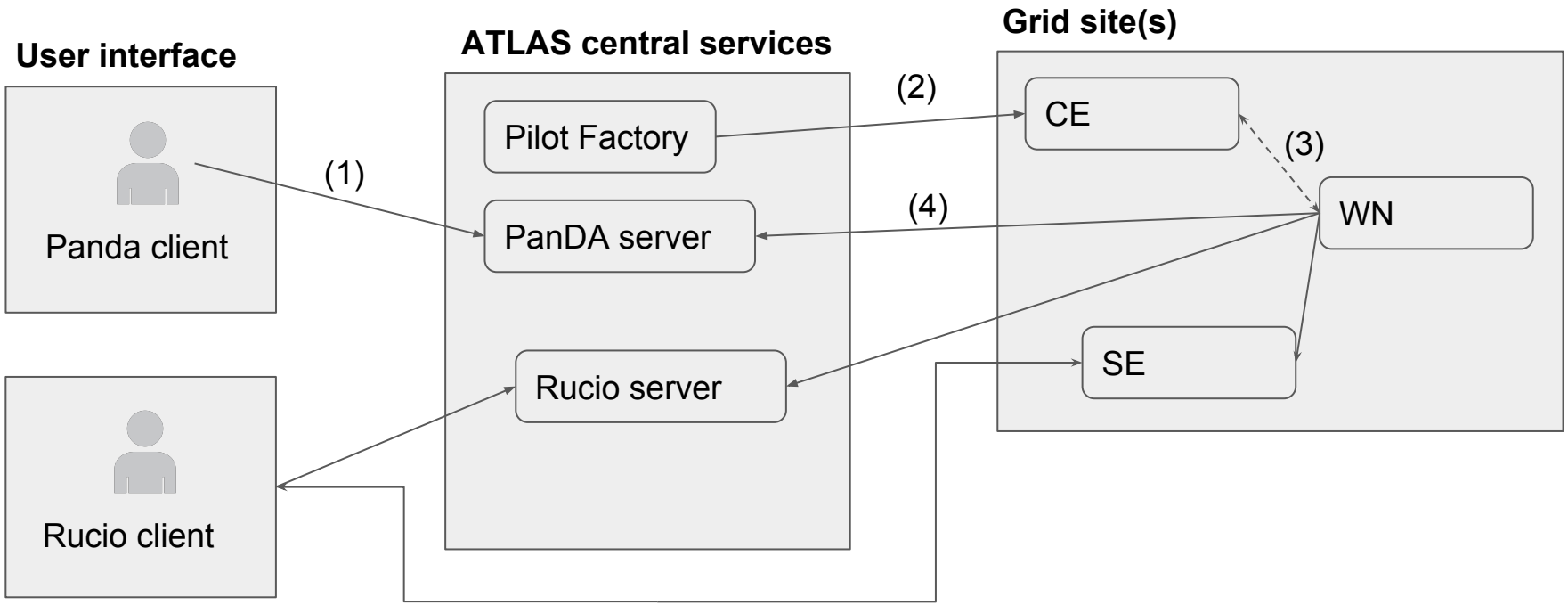
All material provided by ATLAS Central Service Team (in particular Tomas Kouba and Shaojun Sun)

Why and what

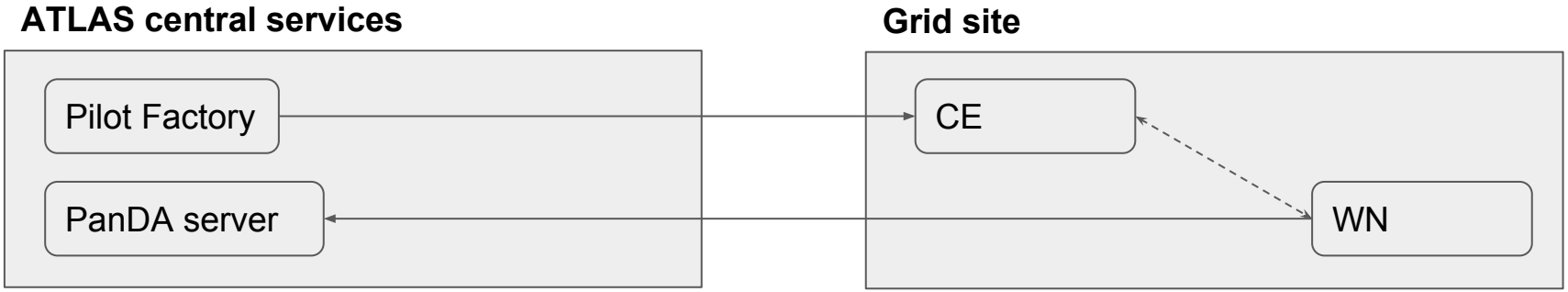
- Follow up of WLCG Workshop in Lisbon
- ATLAS demonstrated to be efficient and effective in case of security incidents
  - this is not a reason to relax: ATLAS is always alert and working in improving infrastructure and procedures
  - interesting chat with Ian Collier, Romain Wartel, Hanna Short: let's organize few "role game" to solidify and structure even better ATLAS reaction to security incidents

- disclaimer (again):
  - this is going to be a very rough most probably not complete view,
    - not enough time to discuss this with all the relevant experts.
  - … but it can start giving an idea

Overview: users interaction

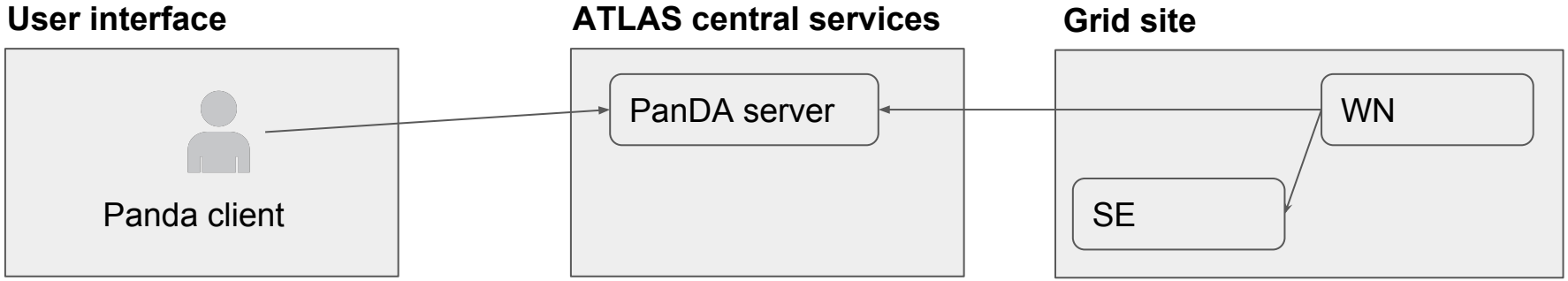


Interaction with sites: Central point of view



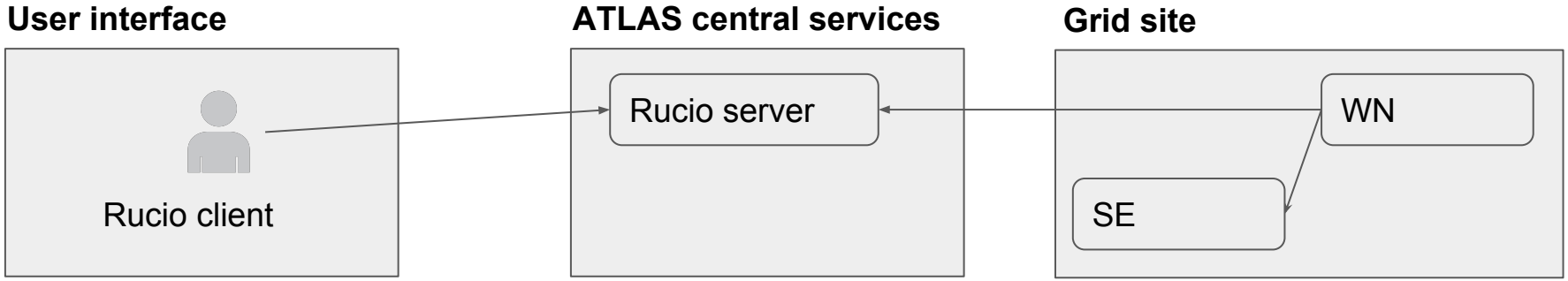
- APF sends pilot wrappers to CE (or directly to WN in Cloud resources)
  - Uses two proxies
    - Each with two roles (Pilot, Production)
  - APF is spread over 18 machines (located in several place, CERN, BNL, some sites)
  - log: **1 line per job** (other lines in /var/log/apf/apf.log are not important)
  - **40k** of logs from pilot wrapper **per job**
- Pilot wrapper runs a pilot on a worker node
  - The pilot code is downloaded by the wrapper from the PanDA server
  - Logs of the pilot and the logs of the payload are stored in the grid
- This workflow is fully automated, no user interaction

Workload management: User point of view



- Job management
  - Job actions (submit, query, cancel) are done via HTTP queries to PanDA server:
    - nodes*(apache daily log + squid daily log) = 8 * (0.8 + 0.0358) = **6.78 GB per day**
  - The mapping of job → user is stored in the logs and PanDA DB (for monitoring purposes)
- Pilot asks PanDA for payload job
  - The pilot+payload is executed with the proxy mentioned above
    - Pilot role for user analysis
    - Production role for production jobs
  - The log of the job is stored in the grid (order of MBs per job)
  - To identify the submitter, one needs to look up the job in bigpanda monitoring

Data management

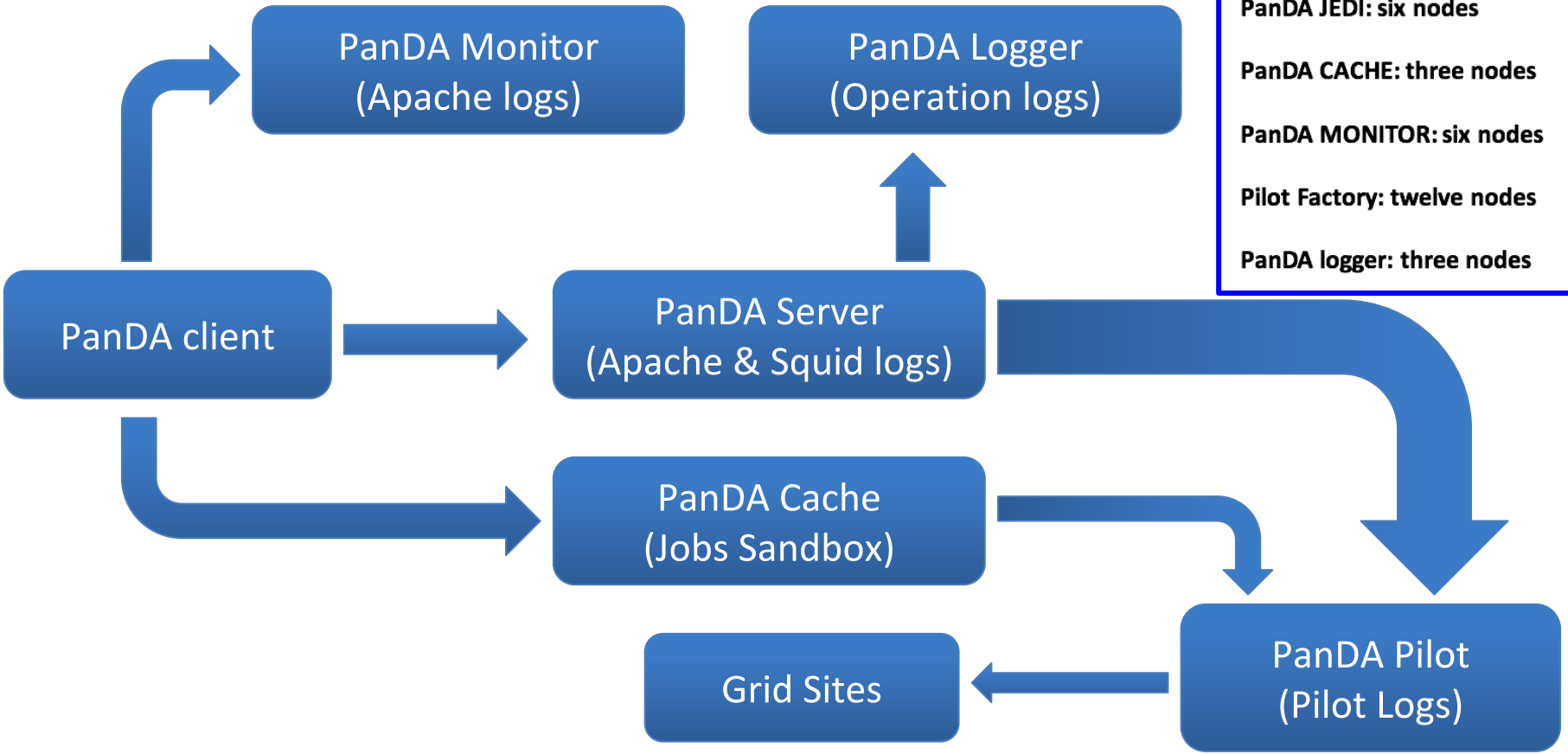


- Data transfers
  - Data actions (queries, download, upload, transfer requests) are done via HTTP to Rucio frontend
    - This includes user queries and actions performed by a running pilot on a WN
  - nodes*(apache daily log) = 14*~250 MB = 3.5 GB per day
  - The logs store directly DN of the client

Access to central services

- All services are configured with puppet ⇒ rsyslog sends important log records to the monitoring team via Flume
  - Is this used for security checks? (I see sshd logs there, but not pam etc.//AB: It should be sssd related, but not pam directly)

PanDA Logs Working Process



Related services:

PanDA SERVER: eight nodes

PanDA JEDI: six nodes

PanDA CACHE: three nodes

PanDA MONITOR: six nodes

Pilot Factory: twelve nodes

PanDA logger: three nodes

Panda

Traceability Related services:

- PanDA SERVER: 8 nodes - 7 GB/day
- PanDA JEDI: 6 nodes -
- PanDA CACHE: 3 nodes - 123 GB/day
- PanDA MONITOR: 6 nodes - 35 GB/day
- Pilot Factory: 12 nodes - 296 GB/day
- PanDA logger: 3 nodes - 8 GB/day

Summary

- This was just meant to be an overview to explain the workflow for what concern traceability
- still more thinking is needed, more ATLAS experts need to be involved to be sure to have the complete view