



Security Issues in Panda

Torre Wenaus (BNL)

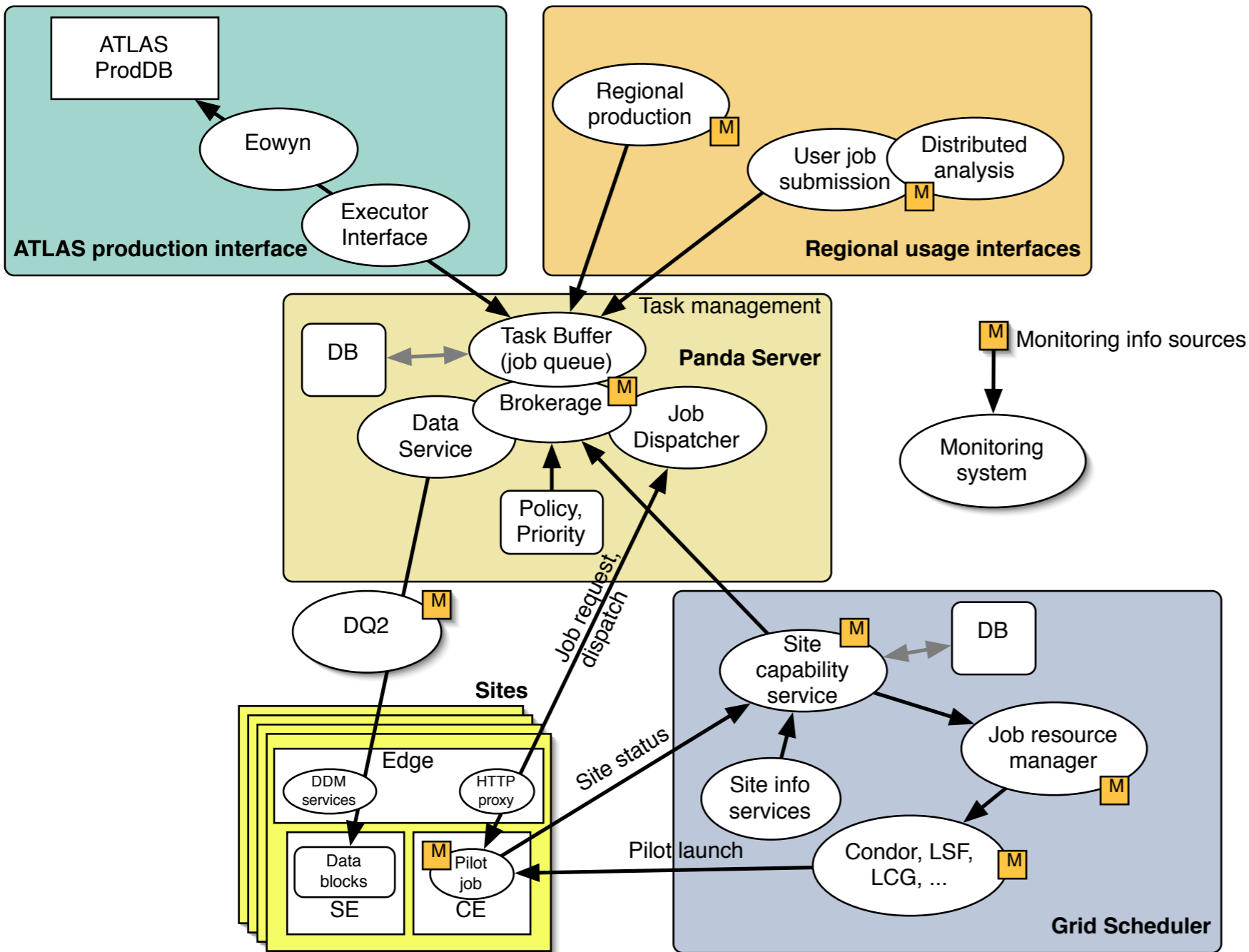
Middleware Security Group Meeting
UCSD

March 2, 2007



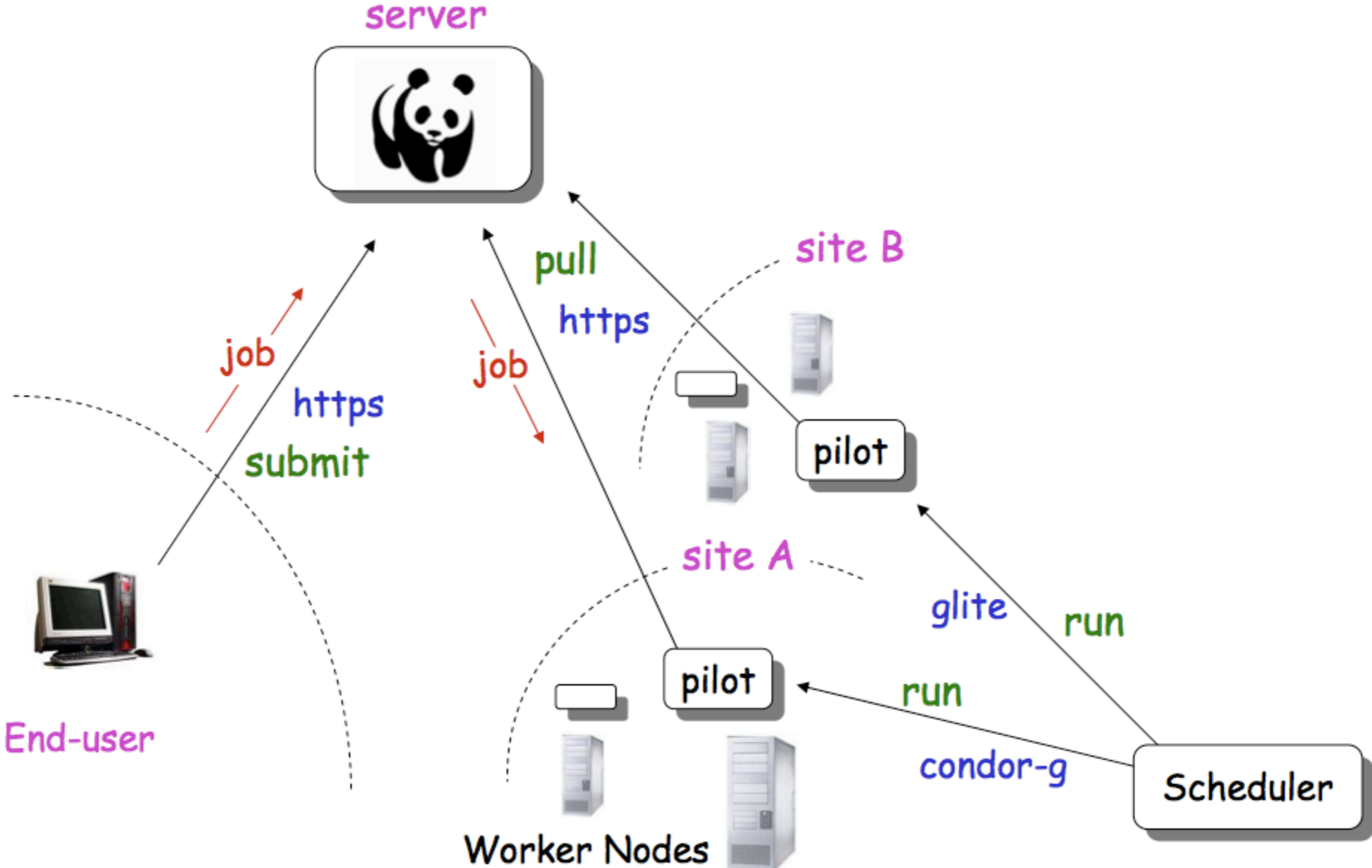
Panda Basics

Workload management system for Production AND Distributed Analysis



- Launched 8/05 to achieve scalable data-driven workload management
 - Prototype 9/05
 - Production 12/05
- Tightly integrated with DDM
- Pilot-based 'CPU harvesting'
- Designed for analysis as well as production
- Designed for high automation, comprehensive monitoring, low ops manpower

Panda Operation





Panda Status

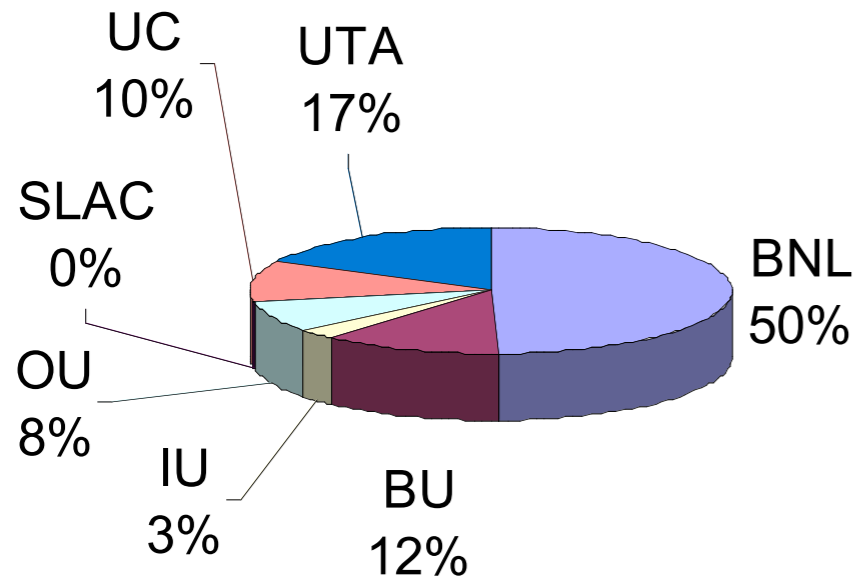
- All US ATLAS production, ~28% of overall ATLAS production with Panda (~50% more than official share)
 - Single shifter, spends <10% of time operating Panda
- Distributed analysis for US ATLAS, also used by Int'l ATLAS, about 50 users
- Recently extended to full OSG, LCG
 - 250 queues at 186 sites, ~200 queues successfully handling test Panda jobs
 - Working on deploying ATLAS production, analysis to these
- OSG Extensions effort on 'just-in-time' workload mgmt
 - ATLAS Panda, CMS glide-in factory, **Condor**
 - First non-ATLAS OSG Panda user starting prod: CHARMM



Completed ATLAS Production Jobs 2006

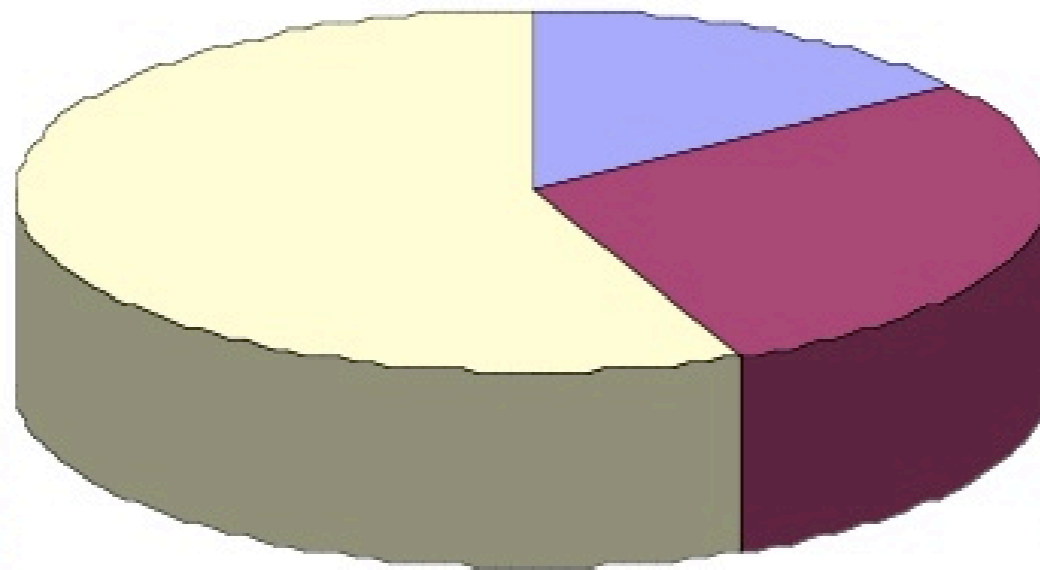


Jobs Finished in 2006



NorduGrid,
135331, 15%

LCG/EGEE,
507243, 57%



OSG/Panda,
252109, 28%

Panda production – 50% of the jobs done on Tier 1 facility at BNL
50% done at U.S. ATLAS Tier 2 sites

K. De

VEN

SCIENCE LABORATORY

Panda Plans



- Main deployment/ops efforts in 2007:
 - OSG-wide WM service
 - ATLAS-wide production/analysis capability
- Main development efforts in 2007:
 - System scaling
 - Via partitioning; architecturally and technically 'easy' thanks to use of standard web software stack & protocols
 - Integration in OSG WM program
 - **Security**

Security Issues in Panda



- Client/server communications
- Client validation, server validation
- Job payload validation
- User identity, tracking/accounting, controls
- Data integrity, security, ownership

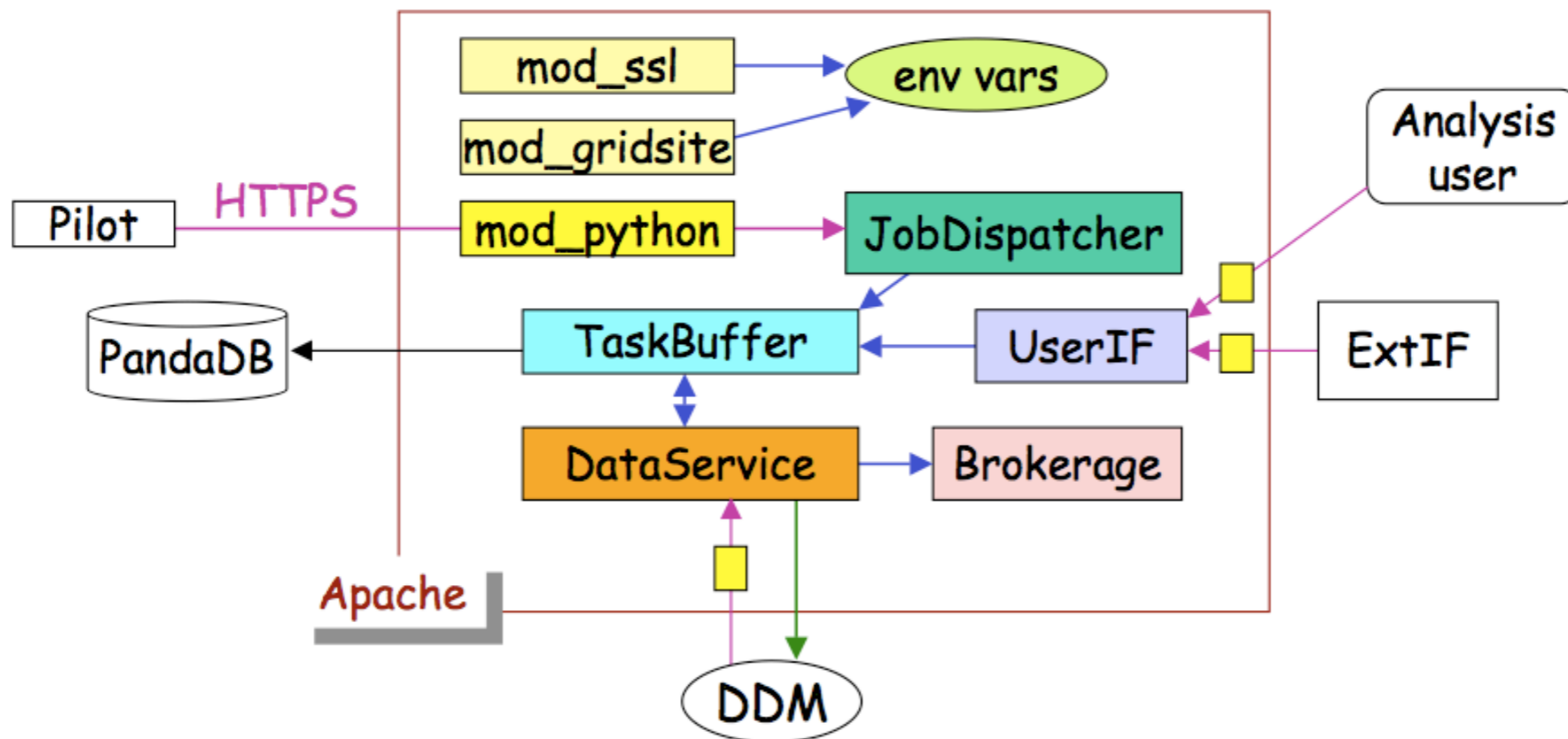


Panda Server

➤ LAMP

- RHEL3 / SLC4
- Apache 2.0.59
- MySQL 5.0.27 - InnoDB
- Python 2.4.4

➤ Multi-process (Apache child-processes) and multi-threading (Python threading)



Client/Server Security



- **Client to server:** Server certifies client by requiring valid grid proxy: all client requests to Panda are via https, GSI authentication via mod_gridsite at Apache server
- **Server to DB:** Panda Server-DB communication is all internal to BNL (or CERN) IT, uses standard MySQL user/pass. Adequate, and best performance
 - Pilot scheduling subsystem uses DB connections over WAN -- currently user/pass, will migrate to **GSI authentication to MySQL**, currently being deployed
- **Server to client:** Server contacted by client is specified by its URL in the setup configuration of the client, which is obtained from subversion repository
 - How do we authenticate the validity of the server? *Authenticate the payload it offers the pilot*

Job Payloads



- If a proxy is stolen, unless job payloads are validated the hacker has an open door to using WN resources
 - Rely on tracing the delegation chain to clean up afterwards
- If GSI authentication was accompanied with secure validation of the job payload, hacker cannot put the WN to nefarious purposes unless this is cracked too
 - Prevent the hacker from causing serious trouble in the first place
- Presently we have an unused hook in the system to validate payload script against a key pair encrypted checksum
 - Need to incorporate using this in our job definition workflow
 - But needs scheme to manage the keys
 - Lacking manpower and expertise, would rather it was done for us...
- Epensys proposal (FNAL) offers ideal solution
 - Symmetric key providing encryption plus signing, with centrally managed key store service



User Job on WN

- Currently, Panda pilot and payload job run with DN of the *pilot submitter* (can be different from the *job submitter*)
- Panda DB records DN of certificate used by the job submitter to authenticate
 - So real identity of end user always available from Panda itself
 - But at least some facility people want (or require) this info directly -- as the authenticated DN of the WN job
 - So we plan to integrate glExec in the pilot to switch identity from pilot submitter to job submitter
 - Some see this as security requirement, others as a security threat
 - We'll use it where admins want us to use it
 - Expect to get this 'for free' via its integration into Condor
- Panda@OSG users (CHARMM) use new pilot scheduling system that end users can use directly, so CHARMM submits its own pilots, and thus owns its own jobs

Accounting, Usage Controls



- Panda has its own (simple and adequate) user and group level usage tracking/accounting/control systems
 - Basically dormant at the moment; haven't had the need to activate them. Analysis and physics working group usage still low
 - Groups defined for admin superusers, US ATLAS, Int'l ATLAS, physics working groups
- Quotas assigned at user and/or group level can be applied against usage measurements; walltime logged at job level
- Why no VOMS? Was trivial to implement an adequate system for *initial* use until VOMS is fully stable, robust, safe
 - We're under no illusions we can live with our own little system forever. We don't want to be in the business of managing identities of hundreds of ATLAS (let alone OSG...) users.



Data Protection

- We have the usual, enviably simple data security environment of HENP. Write once, read many by 'anyone'.
 - No data read/integrity protection issues of medicine/biology/global warming etc (so don't kill us with performance hits in the middleware from supporting their requirements)
- Ownership of produced files passes to DDM system once they're injected into the system
- Only the DDM system can delete files
 - Except for site operators doing cleanup, who must respect 'archival' (cannot delete) and 'recently used' (please keep if you can) info from DDM in their cleanup
- User ownership rights enter at the *dataset* level. Users can delete datasets, including their contained files if they do not appear in other datasets
- In data movement, DDM system validates transfer integrity via source/destination file size and (currently) md5sum

Summary



- Panda has proven successful as a workload manager, so it'll be around a while; it's time we made it (more) secure
- Basic GSI based security for the server's LAMP software stack and its client communications is in
- User identity, tracking, accounting, controls system is in-house
 - Will move to VOMS etc when it's 100% safe to do so
 - Will integrate glExec for use where needed
- Client \leftrightarrow Server validation, payload validation still to come
 - Expect to draw on outside work
- Data protection handled by DDM system, with which Panda is tightly integrated