Selected PanDA Development Highlights

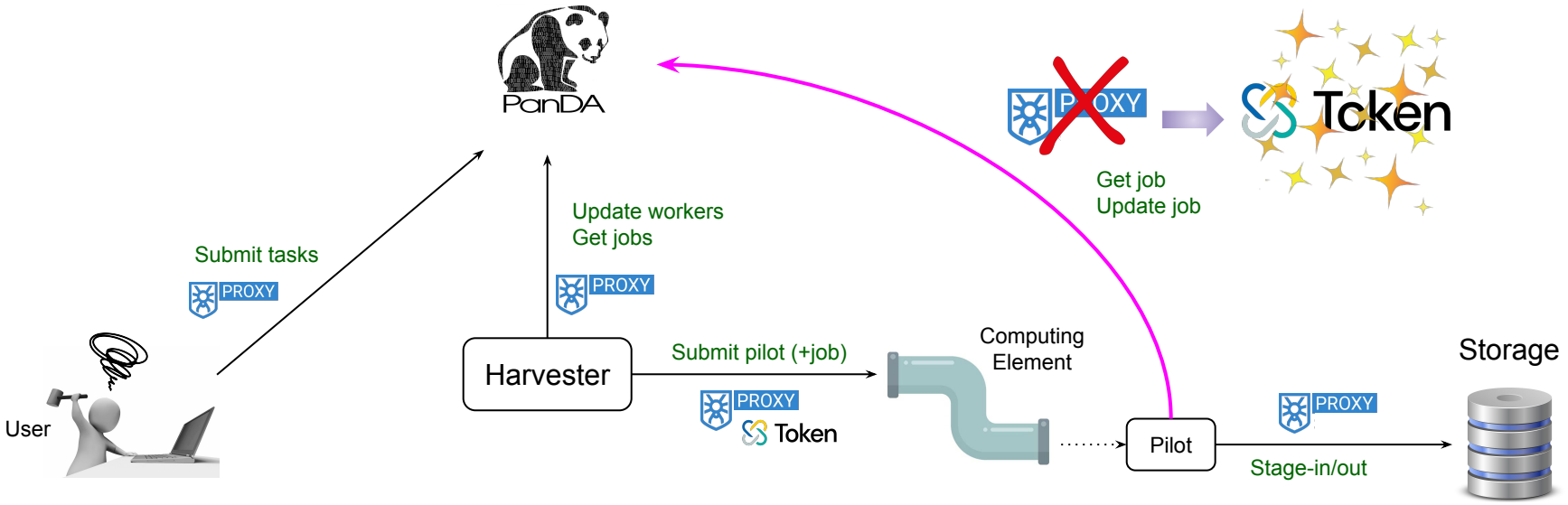
Tadashi Maeno (BNL)
On behalf of PanDA/iDDS/REDWOOD Teams

7 Nov 2024
PanDA Community Forum

Token-based Pilot-PanDA Communication 1/5

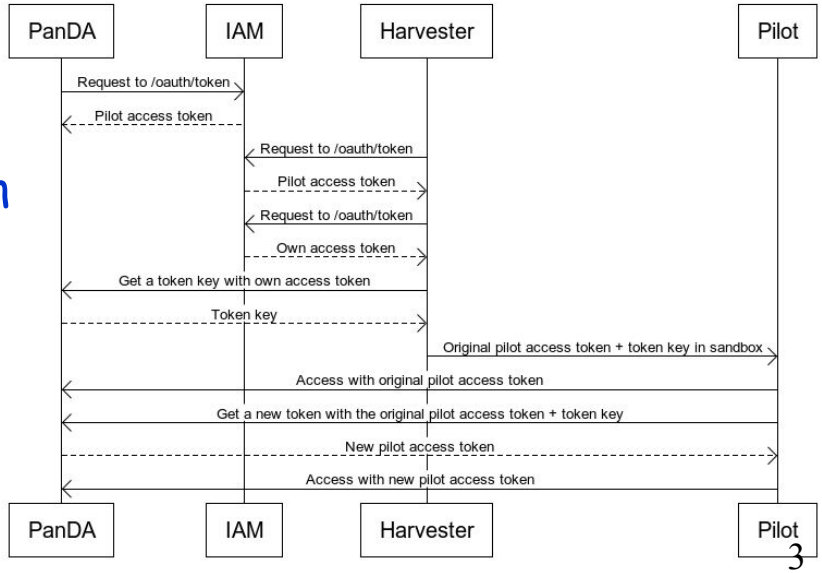
- Legacy x509 proxes still in use among various components (+users)
- Switching to industry standard, OIDC tokens, gradually progressing
  - Access to HTCondor CEs
  - 3rd party transfers between SEs in data challenge
- Migration of Pilot-PanDA communication to OIDC tokens
  - Internal to the PanDA system, no real obligation but trying to respect WLCG security task force recommendations

  - Would mandate usage of short-lived tokens
  - Discouraging token exchange on worker nodes

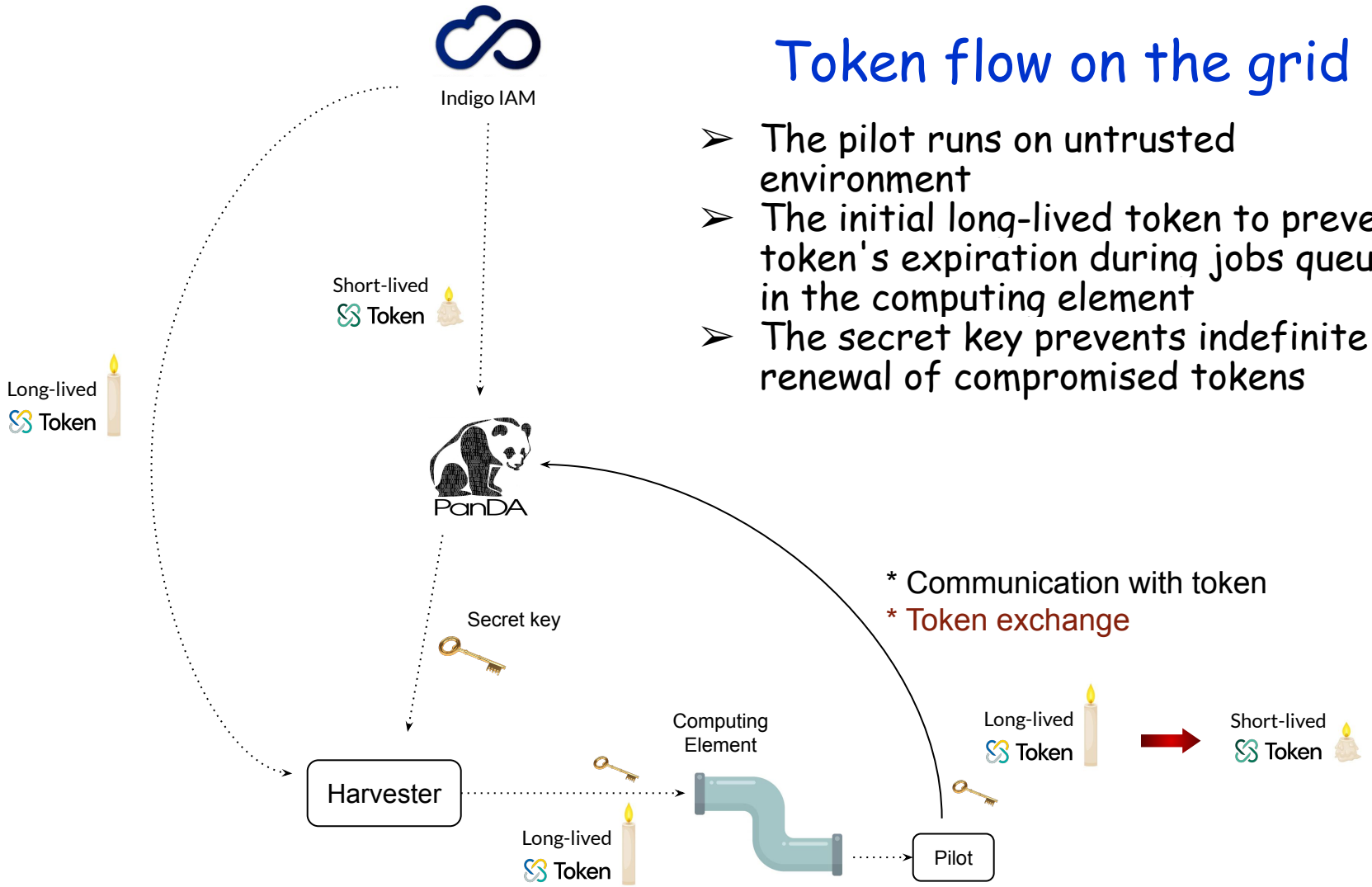


Token-based Pilot-PanDA Communication 2/5

- The recommendations implicitly suggest CEs should provide a mechanism to renew tokens on behalf of pilots and periodically distribute new tokens to pilots while jobs are running on WNs
  - Might be implemented in the future
  - CMS simply uses long-lived HTCondor and HashiCorp Vault tokens that exceed the lifetime of individual jobs
- ATLAS approach
  - Using WLCG tokens
  - Incorporating a dual role into the PanDA server as both a token distributor and a service provider
    - Renewing short-lived tokens periodically with ATLAS IAM and distributing them to the pilot
    - Authorizing pilot access by using tokens
  - Submission of the pilot with an initial long-lived token and a secret key
    - The key prevents indefinite renewal of compromised tokens
  - The pilot retrieves a short-lived token with the key immediately after its activation on a worker node, followed by the deletion of the old long token
    - Subsequent periodic retrieval of short-lived tokens
    - Minimizing presence of long-lived tokens on WNs



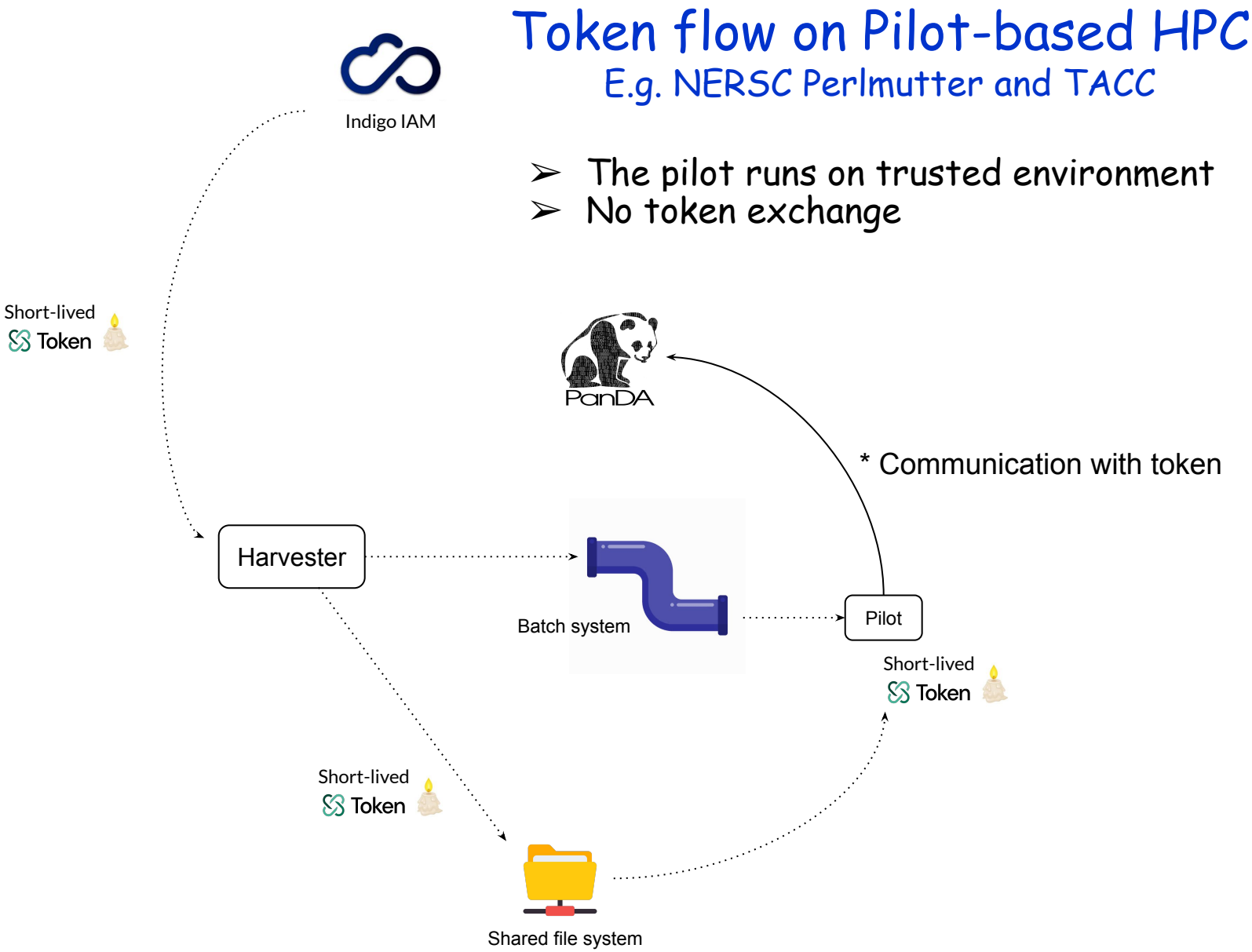
Token-based Pilot-PanDA Communication 3/5



Token flow on the grid

- The pilot runs on untrusted environment
- The initial long-lived token to prevent token's expiration during jobs queuing in the computing element
- The secret key prevents indefinite renewal of compromised tokens

Token-based Pilot-PanDA Communication 4/5

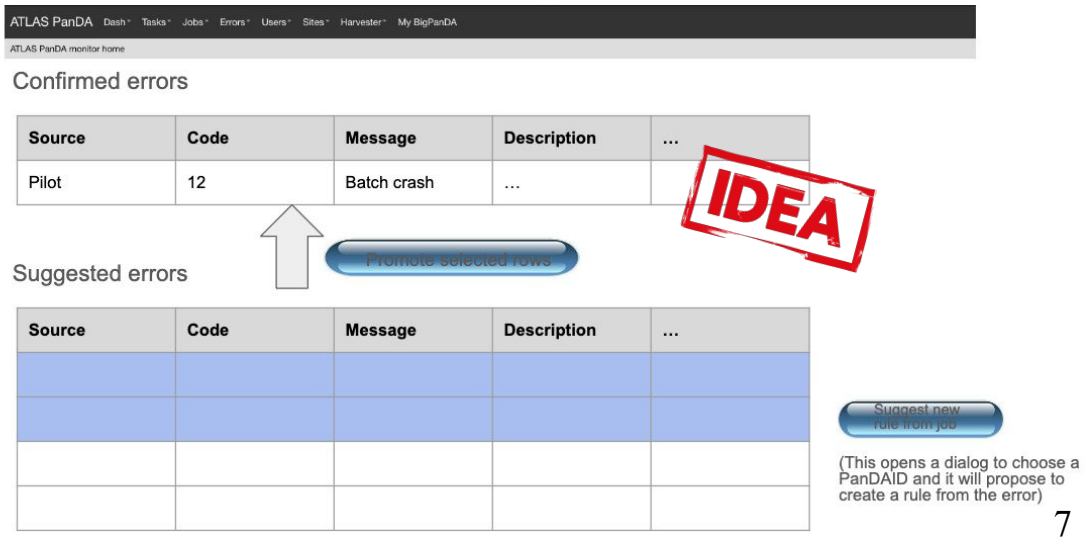


Token-based Pilot-PanDA Communication 5/5

- Re: Token lifetime
  - Long-lived tokens are unavoidable for the standard workflow with the grid computing elements
  - No initiative to standardize automatic token renewal in computing elements
  - WLCG security experts are concerned that some experiments have started implementing their own tokens with long lifetimes, which do not comply with the WLCG token profile
  - Petr Vokac in contact with WLCG security experts to introduce a special scope
    - Only tokens with such scope will be allowed to have long lifetimes in a new version of WLCG JWT profile
    - To ensure that PanDA aligns with WLCG policies
- Harvester-PanDA communication next
  - Straightforward as it only involves a configuration change, and there are fewer accesses from Harvester compared to the pilot
  - By March 2025 for ATLAS

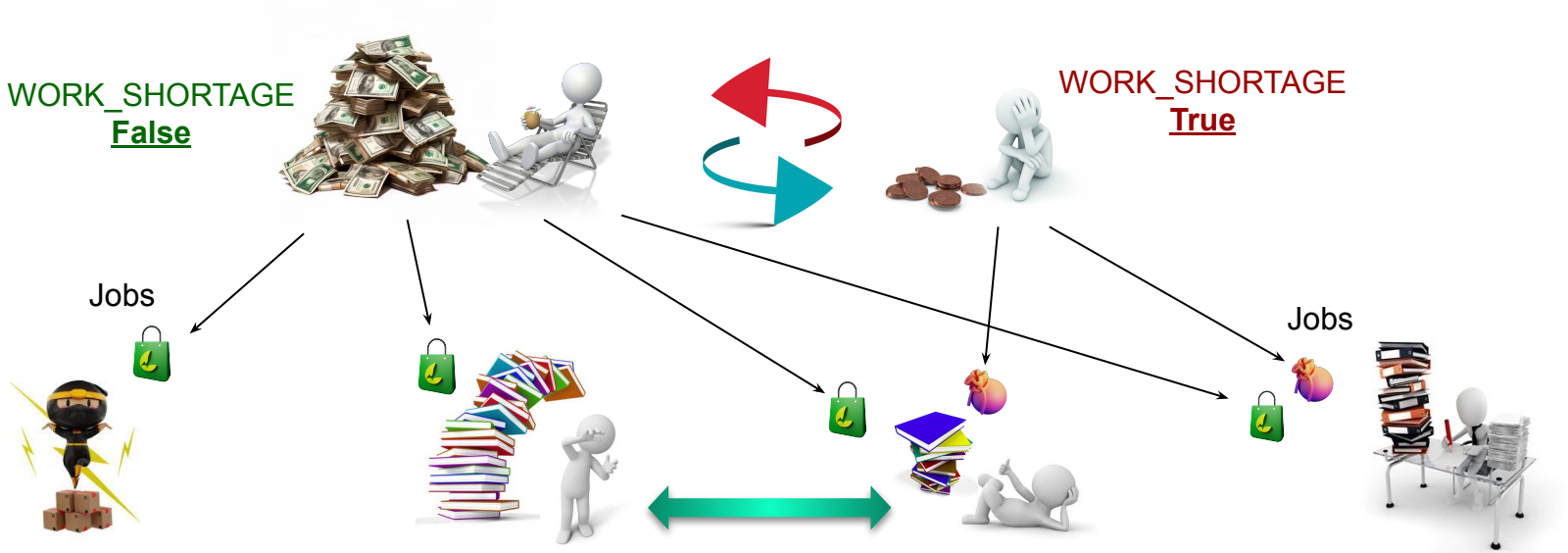
Error Classification

- New ATLAS and REDWOOD joint effort [epic]
  - Tatiana, Fernando, Jammel
- Default three job attempts are not always enough to overcome transient site issues, requiring manual intervention
  - Job's attempt counter increments after each failure, and the job gives up once the limit is reached
  - Desirable if it ignores failures that jobs don't have responsibility
    - Closed jobs don't increment counters
- Goal: To distinguish non-responsible failures from others
- Steps
  - New DB table to suggest/confirm responsibility for each error
  - Logic in JEDI to consider only responsible failures when counting the number of attempts
  - Interface for error management, populating the table above
  - Automation with ML
    - Exploring many possibilities:
      E.g. data collector behind the interface, learning behaviour of input persons



Using Only Pledged Resources

- Sites are unhappy when their pledged resources are not fully utilized, even when the system runs out of work
  - Once or twice a year, between large campaigns
- Actions to be taken in case of work shortage
  - To skip opportunistic resources
  - To keep the number of running cores matching with ATLAS allocation at each shared resource
- Implementation
  - WORK_SHORTAGE in gdp config (False/True)
  - pledgedCPU in CRIC to specify # of pledged cores
    - Doesn't have to be consistent with WLCG allocation



Ongoing/Future Developments

- User feedback via BigPandaMon
  - Empowering users by allowing them to provide direct feedback on their tasks through clicking smiley-face icons on BigPandaMon
- ML-driven task resource requirement prediction
  - Prediction of resource requirements for each task using ML techniques without running actual scout jobs, enabling proactive workload management
- Utilizing semantics of data entities for task/data grouping
  - Physics meanings in data entities, such as dataset names
    - mc23_13p6TeV.545789.MGPy8EG_SMEFT_VBFHyy_cHWtil1p5.merge.AOD.e8417_e8528_s4159_s4114_r15530_r15514 → MadGraph + Pythia8 + Event Generation + Standard Model Effective Field Theory + …
  - To treat "similar" data/task in a similar manner
    - E.g. scaling up replicas for data that exhibit semantic similarity to popular date
  - Embedding string attributes of data into a multidimensional parameter space, where similarity is defined as distance between two data points
  - Possibility to leverage existing LLMs and NLP techniques
- Anomaly detection and auto recovery
  - Experience with ML for error classification automation could facilitate the development of anomaly detection and automated retry rule creation
  - Learning patterns of successful retries through accurate error classification to determine appropriate recovery actions
- System metrics
  - One of the most crucial milestone
  - To quantify the system-wide effects resulting from any changes
    - Averaged TTC, network (mis)usage, fairness across all stakeholders, etc
  - Definition of metrics and implementation of collector