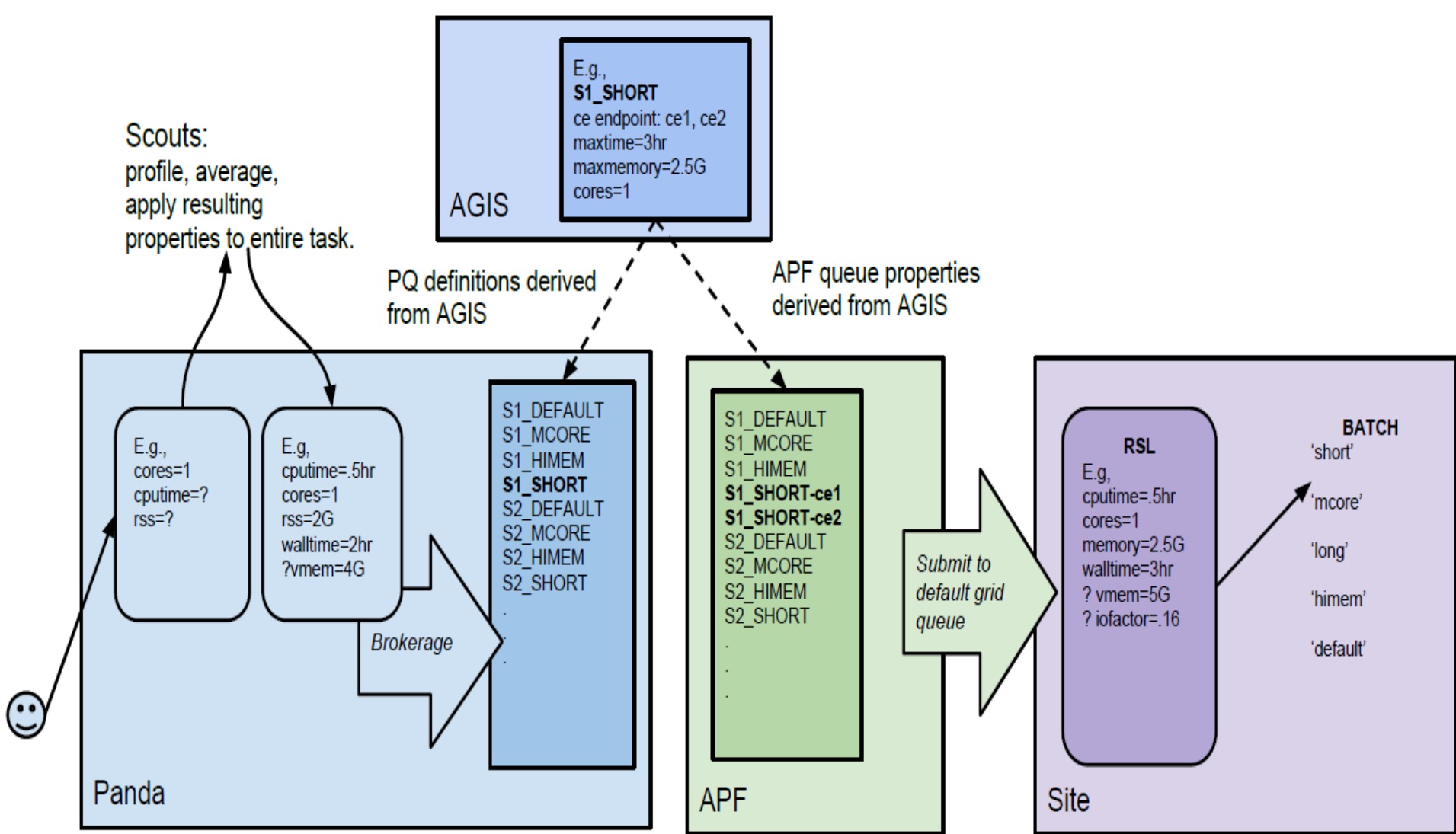


Job-Property-Based Scheduling at Sites

- This is about the way sites present their local resources at the Grid Interface level
 - So far Sites maintained 1:1 correspondence between PanDA queues, Globus RSL queue (and other parameters) and internal Site Batch queues
 - With increasing number of functionally specialized PanDA queues (e.g. MCORE, LMEM, SHORT) this may no longer be desirable
 - Directly effects how pilot submission is handled
 - APF is in the middle between the grid view of the site, and the state of the WMS queues
- We could improve by eliminating grid-level queues and always submit Jobs with their requirements expressed as standard attributes (e.g. cpucount, maxwalltime, maxmemory)
 - E.g. this would allow us to get rid of “PROD” vs. “ANALY”
 - If the job properties are ~the same we could let PanDA broker both types to the same PanDA Pilot
 - PanDA is doing this already with different sub-job types, e.g. evgen vs. reco vs. simul using priorities and desired percentages in AGIS
 - If we don't simplify, PanDA queue and Pilot Factory entities will become more complicated and numerous
 - “R” job requirements and “T” types of work => (“R * T”) PanDA and Factory queues



Note: Panda queue-APF queue relationship must (aside from scaling/load balancing) be 1:1. Each APF queue must handle an *equivalence class*, i.e. a set of jobs with identical properties.

Note: No 'queue' parameter. Submitted job is defined by properties only, not 'prod' or 'analy'. VO derived from VOMS proxy. Site can calculate additional properties, e.g. "iofactor"=1/ (walltime/cputime), to refine local scheduling.

Conclusion (1/2)

- The diagram assumes a static number of bins, represented by PanDA queues
 - Saves needing PROD or ANALY queues (or any queues at all) at the Grid Interface to the Sites
 - Work can be submitted to the Site's Default Queue
- APF/PanDA Interactions
 - APF must be able to query PanDA for ready/activated Jobs within an equivalence class
 - Each Pilot guaranteed to have a Job for dispatch to it
 - Any system where equivalence classes can change, are inconsistent, or overlapping, is unworkable

Conclusion (2/2)

- PROD vs. ANALY
 - Goal is to dispense PROD vs. ANALY concept
 - ANALY vs. PROD used as a proxy for I/O intensity
 - Better would be to let Scout calculate walltime and cputime and calculate I/O intensity factor
- Site Processing
 - Proposed concept works with popular Batch Systems
 - E.g. HTCondorCE job router provides hooks for scripts to transform incoming Class Ad to local Batch ClassAd