The Instant Glidein

A generic approach for the late-binding of jobs to various resource types

- High-Throughput Computing
  - Focuses on the throughput over time of many jobs
    - Rather than performance of an individual jobs
  - Batch systems enables jobs to be executed on computing resources
    - The algorithms implemented optimises the utilisation of those resources
  - Metaschedulers optimise the utilization of geographically-separated computing resources based on state information provided by the local batch system managing those resources

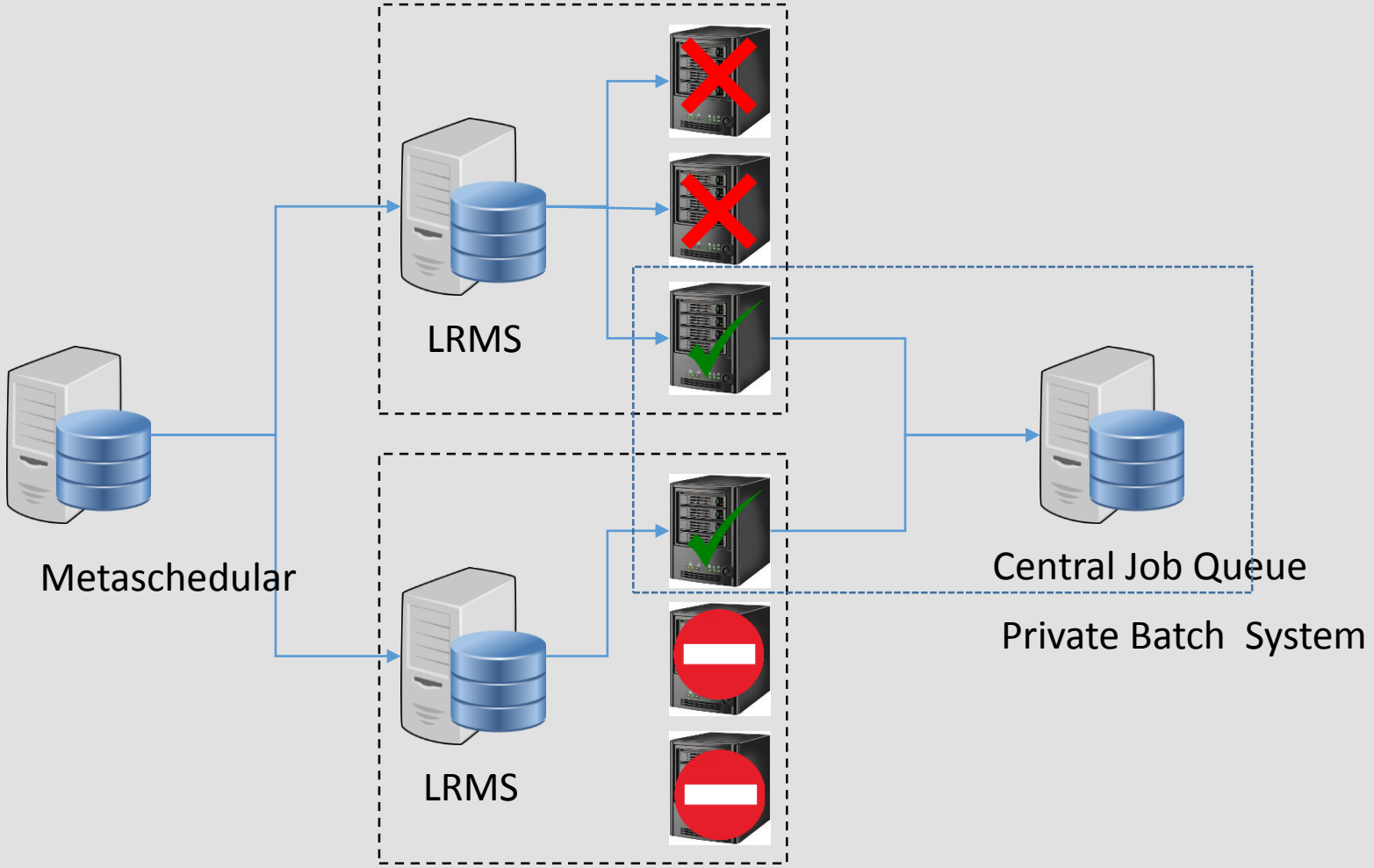
The Instant Glidein

A generic approach for the late-binding of jobs to various resource types

- With the metascheduling, the decision of where the job is to be executed is made quite in advance
- Late-binding uses a place-holder job that when executes requests the most appropriate real job from a central job queue.
  - Allows scheduling decisions to be delayed until the last possible
  - Can also perform basic sanity checks of the environment
  - The result is that a private or application specific batch system is overlaid on the original resources
  - A lesson learned from the experience of metascheduling is that resource allocation should be decoupled from job execution and the order respected
- The HTCondor Glidein is an example implementation
  - The placeholder job starts a HTCondor daemon process
  - The result is a private HTCondor pool created out of dispersed computing resources by gliding-in HTCondor daemons

The Instant Glidein

A generic approach for the late-binding of jobs to various resource types



The Instant Glidein

A generic approach for the late-binding of jobs to various resource types

- Virtual Machines delivered by Infrastructure as a Service (IaaS) has brought a new dimension to HTC.
  - Computing resources can now be dynamically provisioned on demand using *resource leases* which are typically defined in terms of wall time when using a pay-as-you-go model.
  - The late-binding approach is appropriate in this scenario to ensure that job execution occurs after the resource has been allocated.
  - The solution is therefore to overlay a batch system upon those dynamically provisioned computing resources.
  - The alternative to the VAC model is where the lifecycle management is done via the user or application themselves.
    - We will define this model as Throughput Amplification by the Stimulated Emission of Resources (TASER)

The Instant Glidein

A generic approach for the late-binding of jobs to various resource types