# Resource Descriptions

Don Lipari

January 13, 2014

# Premise

A Scheduler Plugin's function starts with

- Resources
- Job Allocation requests

and matches jobs to resources.

# Goals

- Scheduler plugin API that allows for the simplest and most complicated scheduler plugin functionality.
- This API will depend on the ways resources and jobs are represented.
- We will look for ways to represent resources that can be used in the job request.
  - Facilitates matching apples to apples.
- Create a core structure that provides for extensions where and when needed.

# From the Vision Doc…

“The [resource description] language will be structured, extensible, human-readable, and hierarchical, while being capable of representing resources and their relationships in a generic and flexible fashion.”

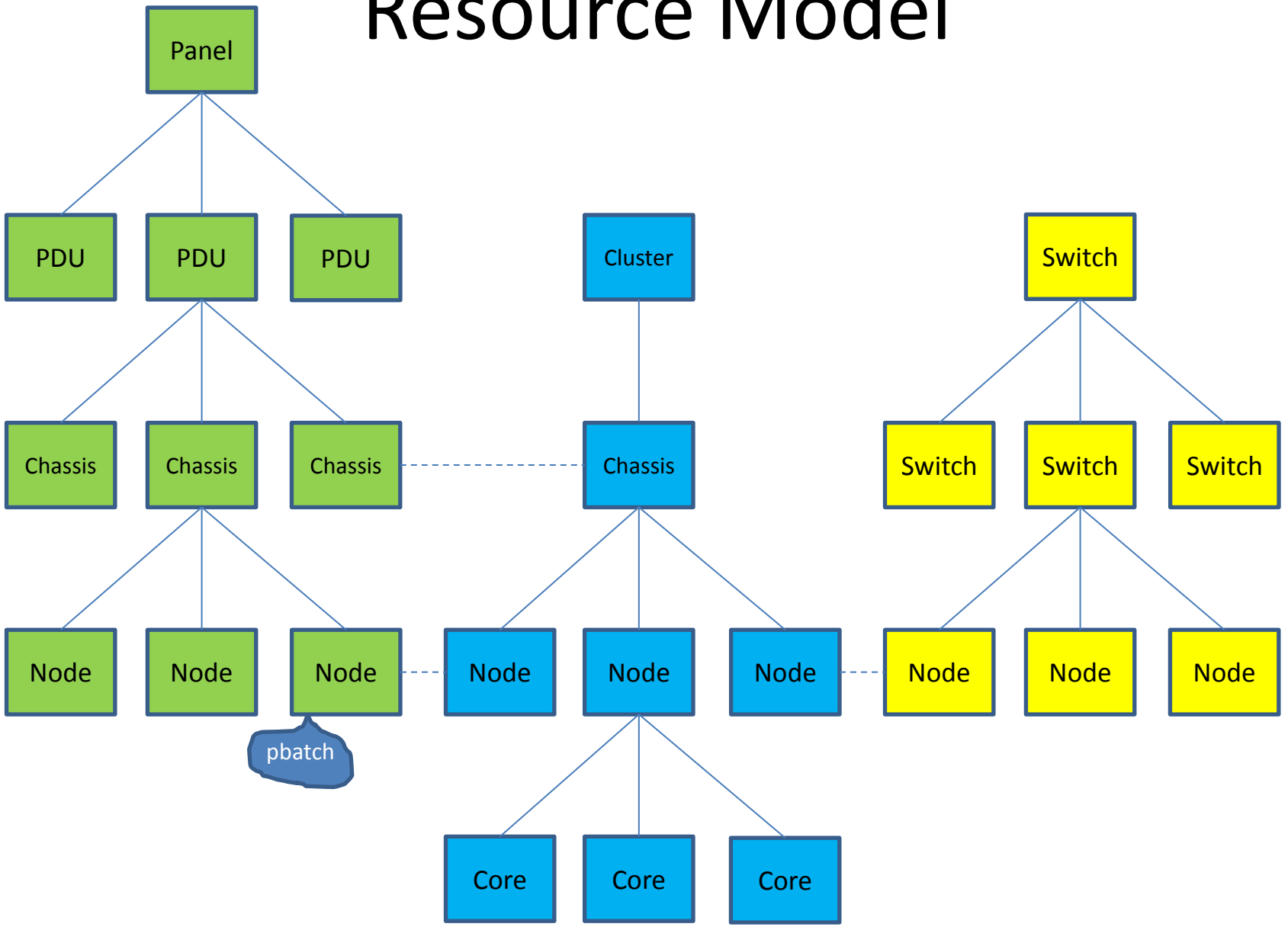
# Existing Frameworks

- Globus Resource Specification Language (RSL)
  - http://toolkit.globus.org/toolkit/docs/5.2/5.2.5/gram5/developer/#gram5-rsl
- ClassAds
  - http://research.cs.wisc.edu/htcondor/classad/refman/
- CEA's Layouts Framework
  - http://slurm.schedmd.com/SUG13/layouts_framework.pdf
- W3C's Resource Description Framework (RDF)
  - http://www.w3.org/TR/rdf-concepts/
- W3C's OWL 2 Web Ontology Language
  - http://www.w3.org/TR/owl2-overview/
- Protégé Ontology Editor
  - http://protege.stanford.edu/

# Resource Taxonomy

- Hierarchic
  - Hardware
    - Rack,chassis,board,socket,core
  - Power
    - Panel,PDU,chassis,board
  - (some) network topology
- Mesh/Torus
  - (some) network topology
- Sets
  - Tags (e.g., queues, processor family, BigMem, etc.)
  - Licenses

# Resource Model



# Attributes of Each Resource

- Permitted users or groups
- Permitted accounts
- Level of sharing
- Containment
- Limits
  - Time
  - Count
- QoS
  - Preemption
  - Additional constraints
  - Exemption from limits