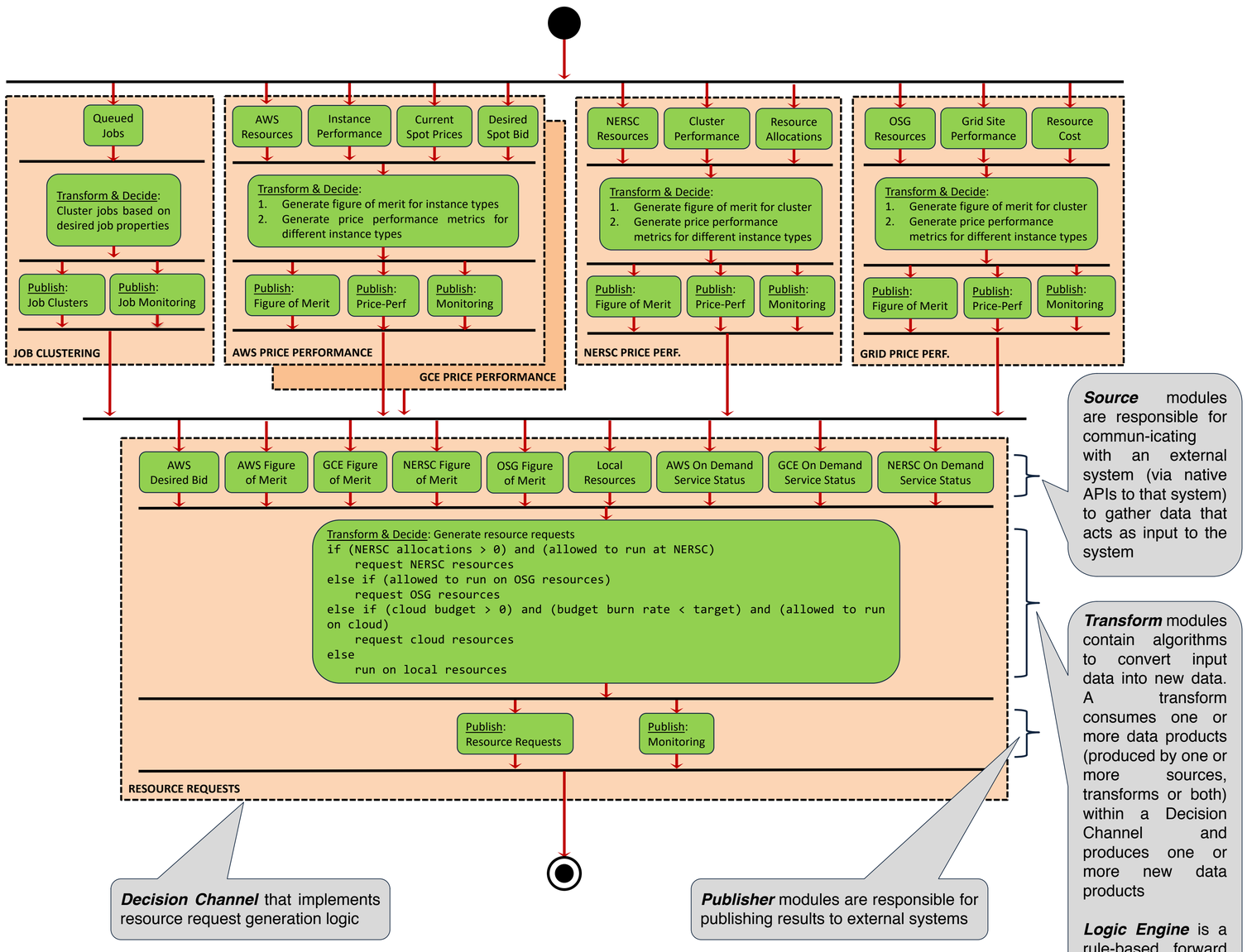
INTELLIGENTLY-AUTOMATED FACILITIES EXPANSION WITH THE HEPCLOUD DECISION ENGINE

M. Altunay, W. Dagenhart, S. Fuess, B. Holzman, J. Kowalkowski, D. Litvintsev, Q. Lu,
P. Mhashilkar, A. Moibenko, M. Paterno, P. Spentzouris, S. Timm, and A. Tiradani
Fermi National Accelerator Laboratory

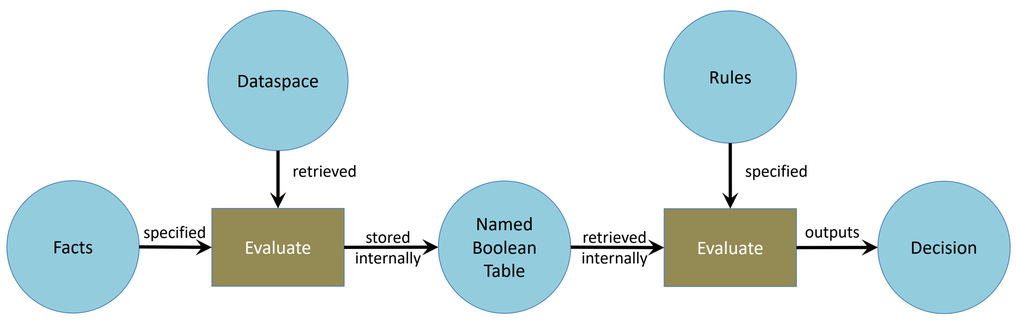
Abstract

The next generation of High Energy Physics experiments are expected to generate exabytes of data two orders of magnitude greater than the current generation. In order to reliably meet peak demands, facilities must either plan to provision enough resources to cover the forecasted need, or find ways to elastically expand their computational capabilities. Commercial cloud and allocation-based High Performance Computing (HPC) resources both have explicit and implicit costs that must be considered when deciding when to provision these resources, and to choose an appropriate scale. In order to support such provisioning in a manner consistent with organizational business rules and budget constraints, we have developed a modular intelligent decision support system (IDSS) to aid in the automatic provisioning of resources – spanning multiple cloud providers, multiple HPC centers, and grid computing federations.

Provisioning resources with the Decision Engine



Logic Engine

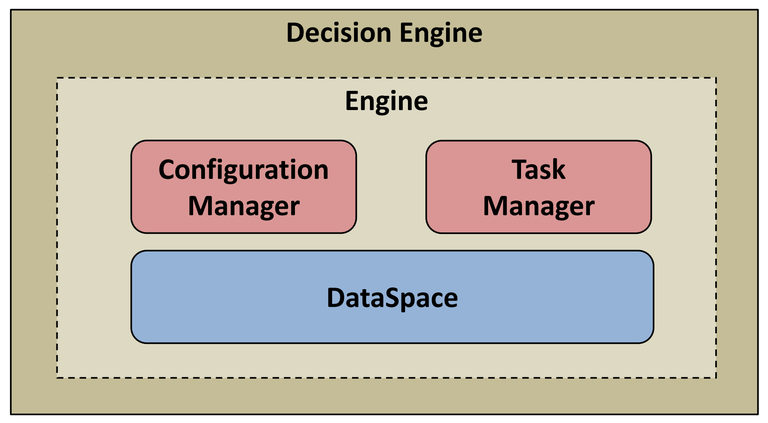


- Each ***fact*** has a name and an expression that evaluates to a Boolean
- The value of fact is the value of the expression
- Expressions can access and operate on data produced by ***Source*** and ***Transform*** modules
- A ***rule*** consists of condition composed of references to facts and Boolean operations on their values
- Actions are triggered when the rule evaluates to Boolean ***'True'***
- Logic Engine rule can produce new facts that evaluate to the result of the rule's Boolean expression
  - This fact can be used in subsequent rules

Design Drivers

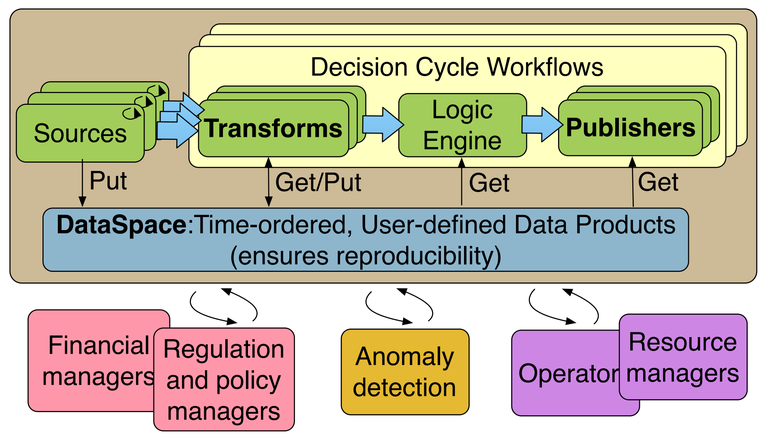
- The need for a framework that enforces the processing stages defined and implemented by the resource provisioning program, and which provides for the injection of user-supplied code and expert knowledge
- The need for a configuration and assembly system that instantiates the appropriate user-supplied code, and that provides the necessary context-dependent information to realize different parameterizations of that code
- A means to manage the data being processed and the varying timescales for the relevance and validity of those data

Decision Engine Architecture



- The ***DataSpace*** acts as a Knowledge Management system
- A ***Configuration Manager*** acts as the Configuration Factory
- ***Task Managers*** are responsible for the scheduling of Decision channels tasks
- A single ***Engine*** is responsible for coordinating the Task Managers
- ***Decision Channel*** is a grouping of tasks that generates decision
- ***Decision*** consists of recommendation of one or more actions that should be executed, actions that are directly executed, or both

Decision Channel Components




```
"logicengines" : {
    "logicengine_aws" : {
        "module" : "framework.logicengine.LogicEngine",
        "name" : "LogicEngine",
        "parameters" : {
            "rules": {
                "allow_to_publish_AWS": {
                "expression":"(allow_AWS)",
                "actions":["AWSFigureOfMerit", "AWSPricePerformance"],
                "facts":["allow_AWS"],
            },
        },
        "facts" : {
        "allow_AWS":"(True)"
    },
}
```

ACKNOWLEDGEMENT: This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.