LEAD Workflow Orchestration

Lavanya Ramakrishnan

Renaissance Computing Institute

University of North Carolina – Chapel Hill

Duke University

North Carolina State University

renci
renaissance computing institute

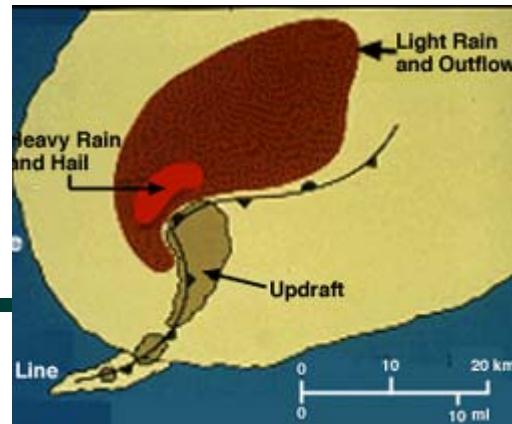
VGrADS
Virtual Grid Application Development Software Project

Outline

- Background and LEAD Service Architecture
  - —web services, BPEL
- LEAD Workflows
  - —status, site visit example run
  - —characteristics, workflow structures
- LEAD Virtual Grid Integration
  - —resource management

Linked Environments for Atmospheric Discovery

- **Rationale**
  - —each year, mesoscale weather – floods, tornadoes, hail, strong winds, lightning, hurricanes and winter storms – causes hundreds of deaths, routinely disrupts transportation and commerce, and results in annual economic losses in excess of $13B.
- **From "offline" to "online" forecasting**
  - —data assimilation and adaptive evaluation

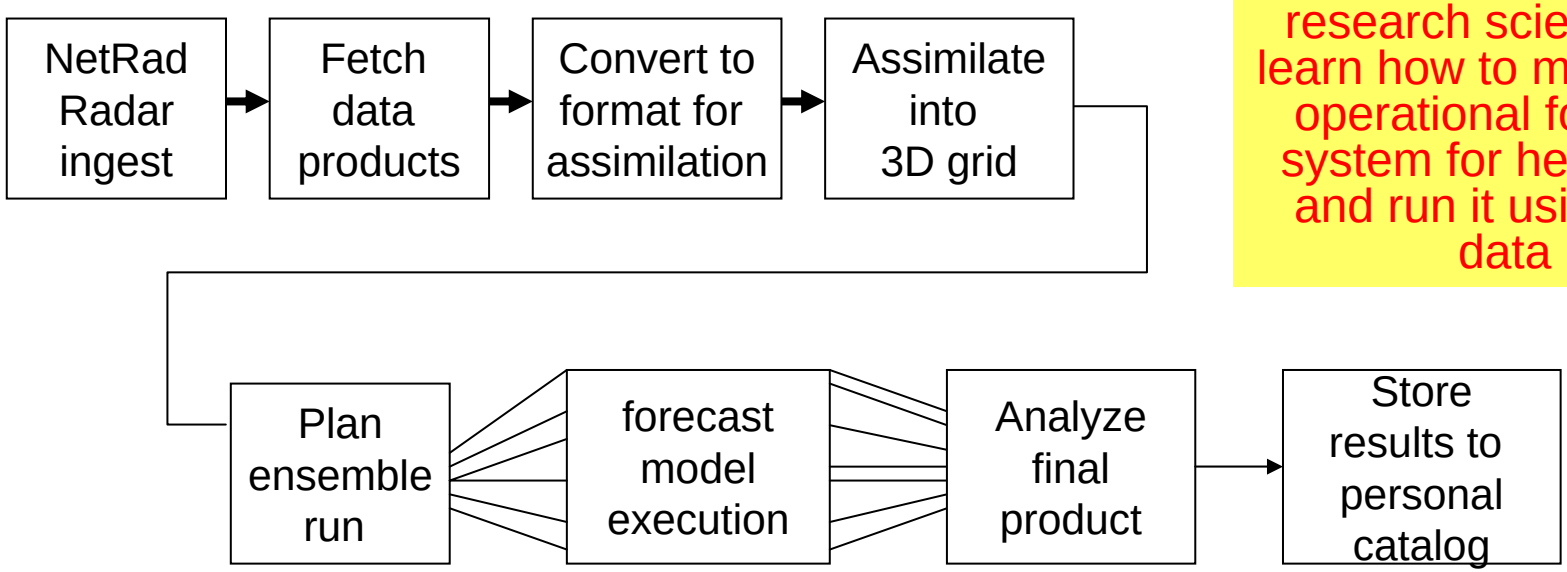


Static Forecasting

- **Large-scale tightly coupled components**
  - **—not adaptable to weather or resource behavior**
  - **—modifications are not trivial**

Center for Analysis and Prediction of Storms (CAPS, OU):

It took an MS student 6 months, working with a research scientist, to learn how to modify the operational forecast system for her needs and run it using real data

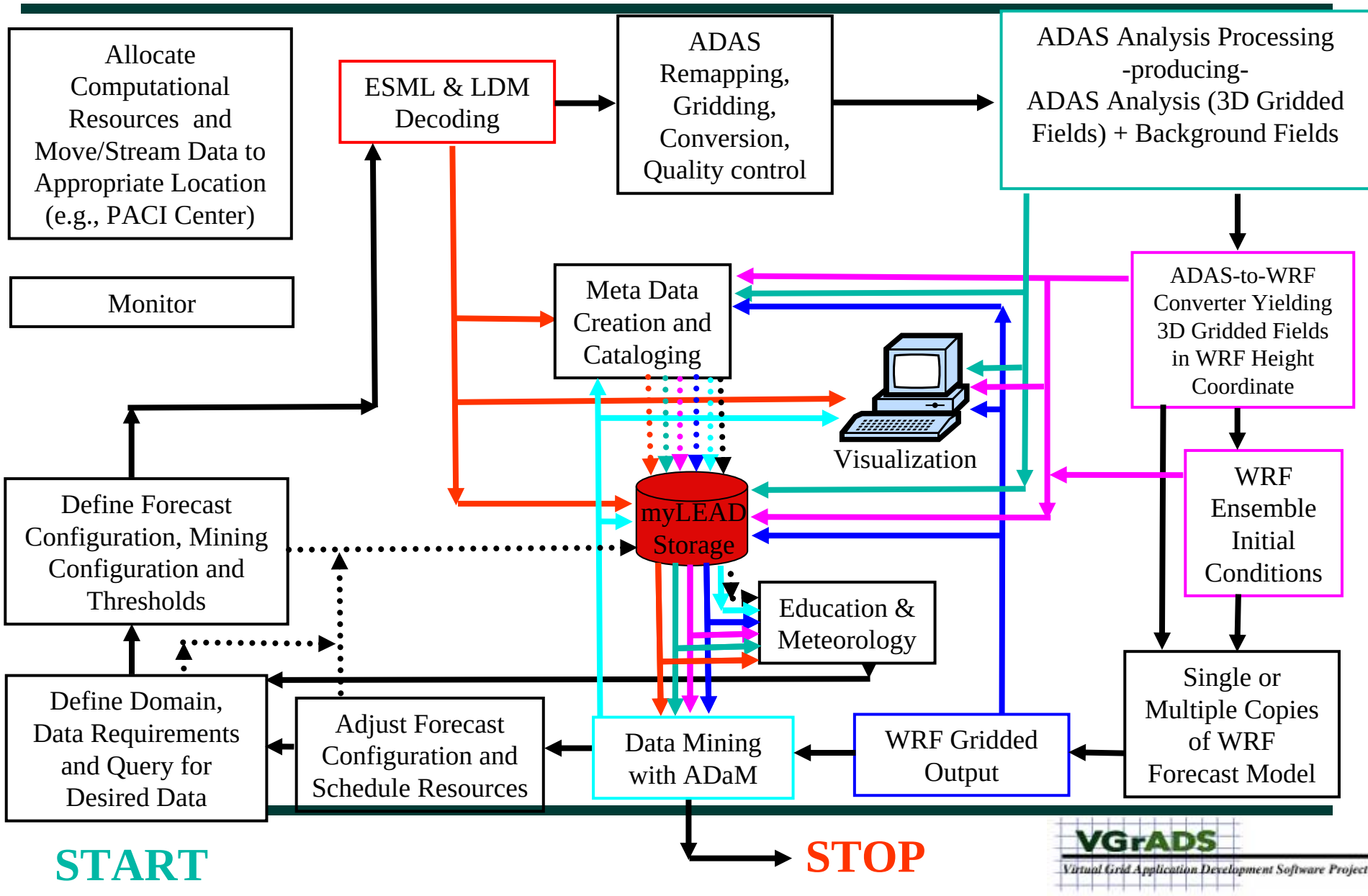


Dynamic Adaptive LEAD System

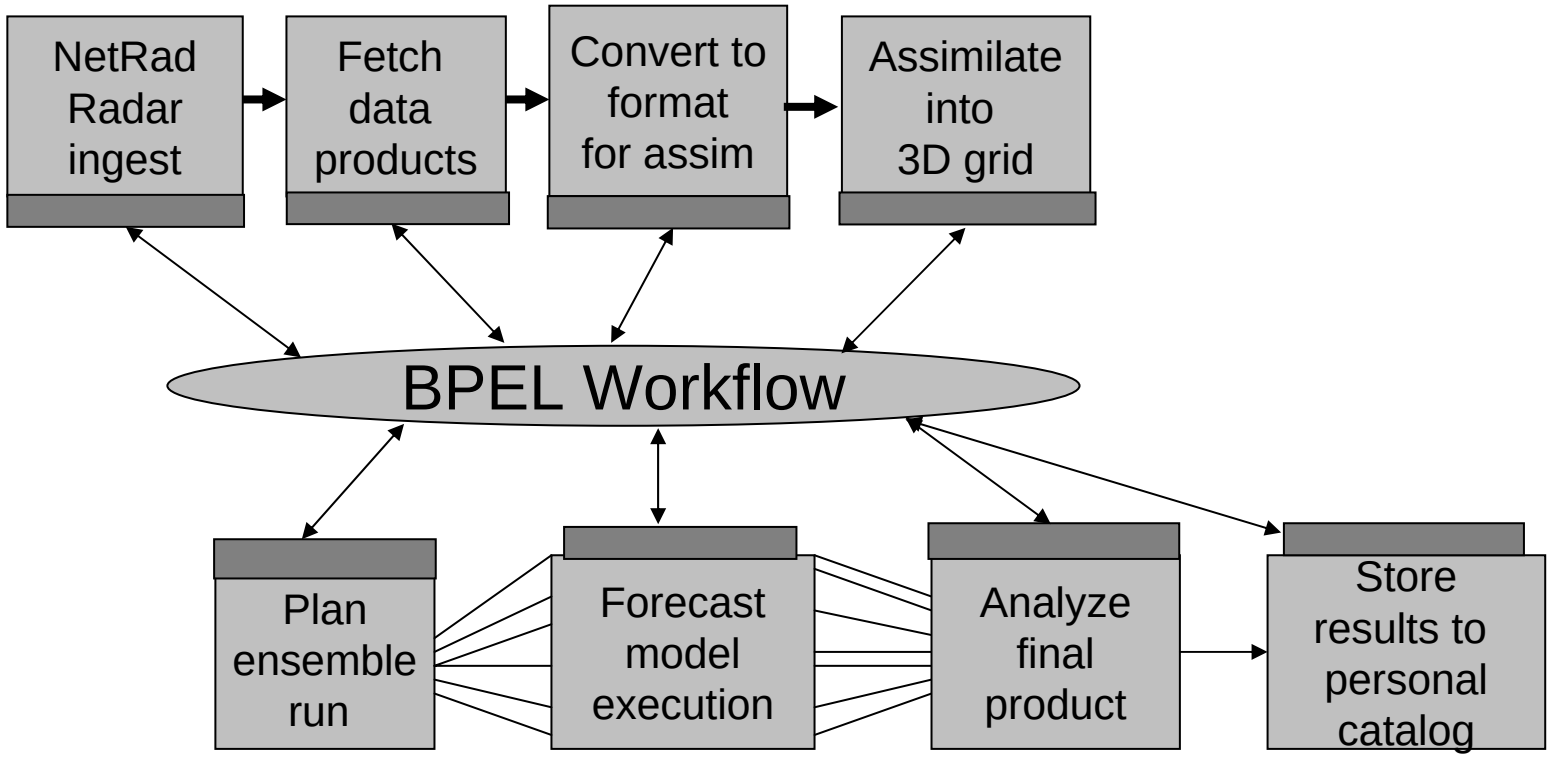
- **Meteorology goal**
  - —to provide timely and accurate forecasts using dynamic adaptation
- **Computer Science goal**
  - —map application requirements to resource capabilities
    - redundant runs, scheduling policies
  - —adapt to weather as well as resource behavior

Need real time monitoring to make adaptation decisions

LEAD Control and Data Flow



Service Oriented LEAD Workflows

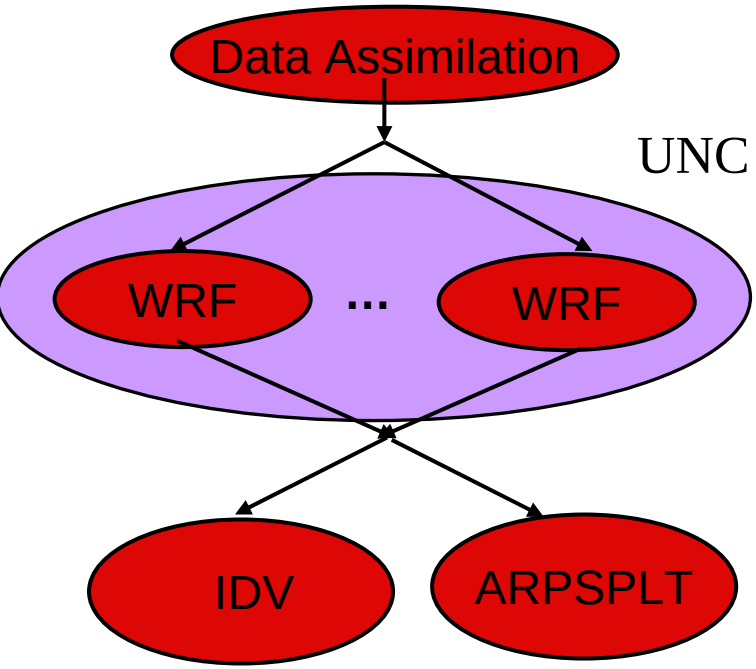


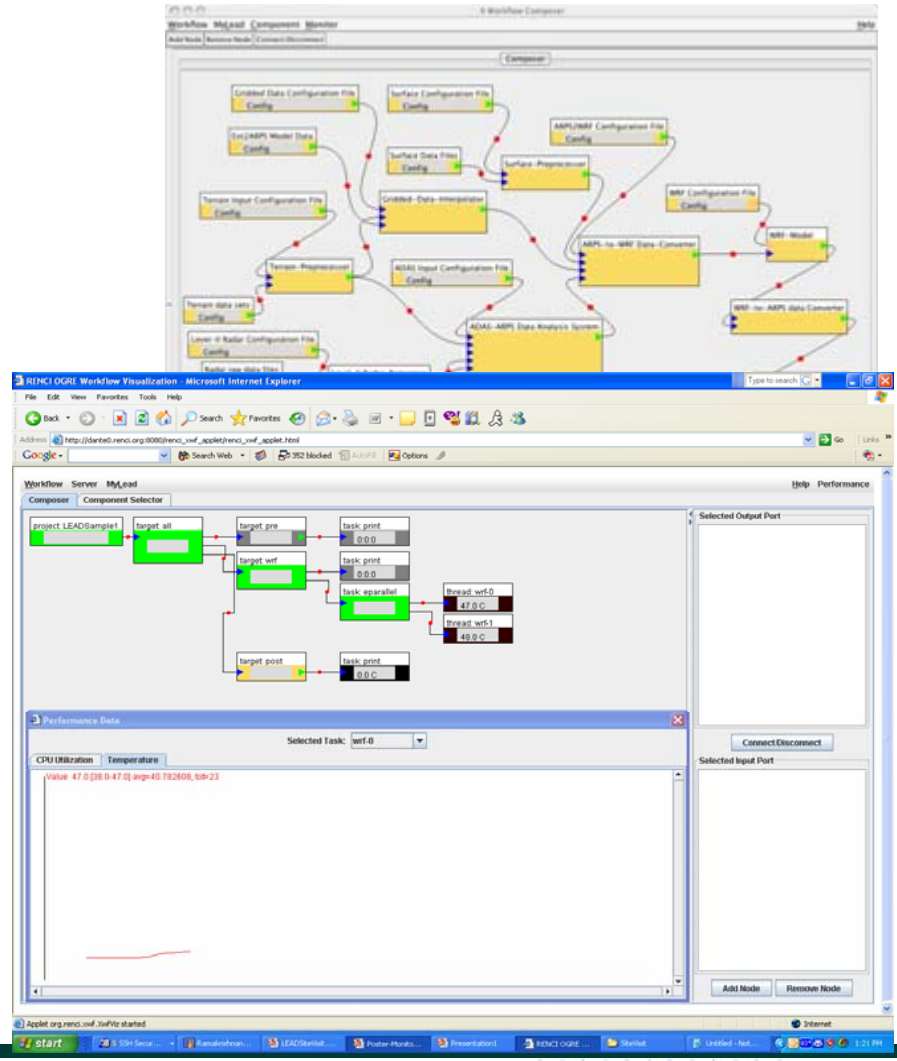
Source: Beth Plale

Business Process Execution Language (BPEL)

- **BPEL (aka WSBPEL and BPEL4WS)**
  - — enable cross-enterprise automated business processes
  - —multiple vendor participation
    - Oracle, Microsoft, IBM, SUN, SAP, BEA, EDS, Adobe, …
  - —multiple implementations
    - Oracle BPEL, IBM BPWS4J and Microsoft BizTalk Process Managers, Active BPEL (www.activebpel.org) Open Source Process Manager
- **Features**
  - —loop and control logic, synchronous/asynchronous communication
  - —composition and concurrent execution

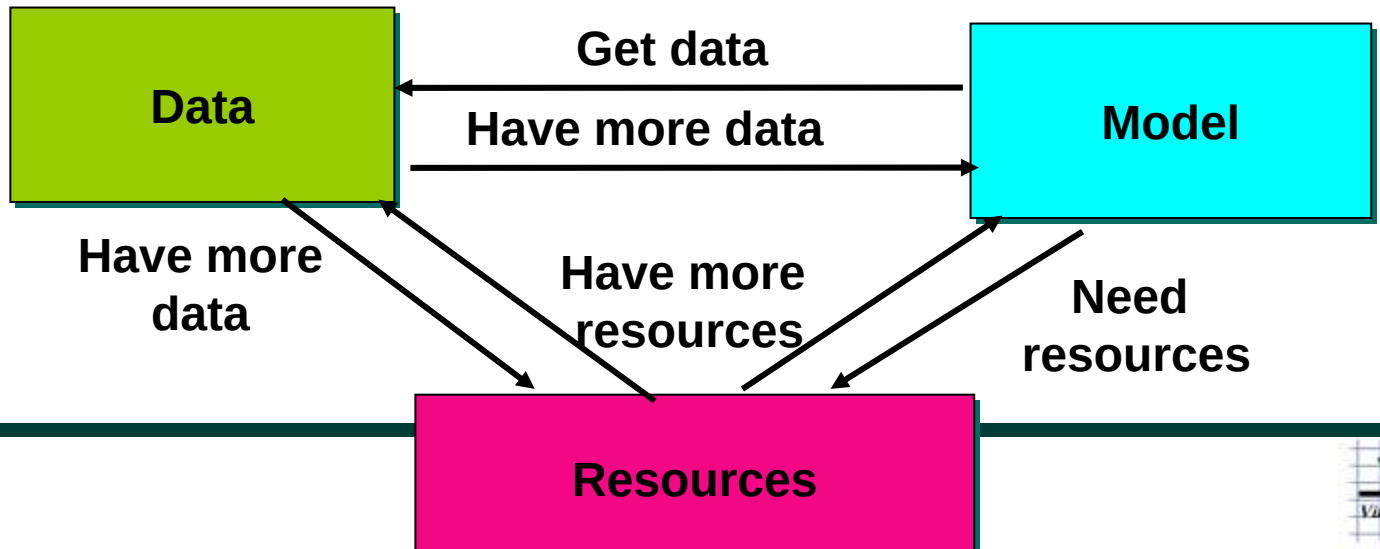
LEAD Site Visit Experimental Ensemble





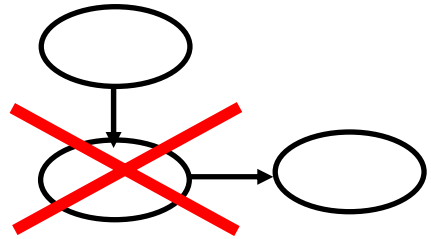
Characteristics of LEAD Workflows

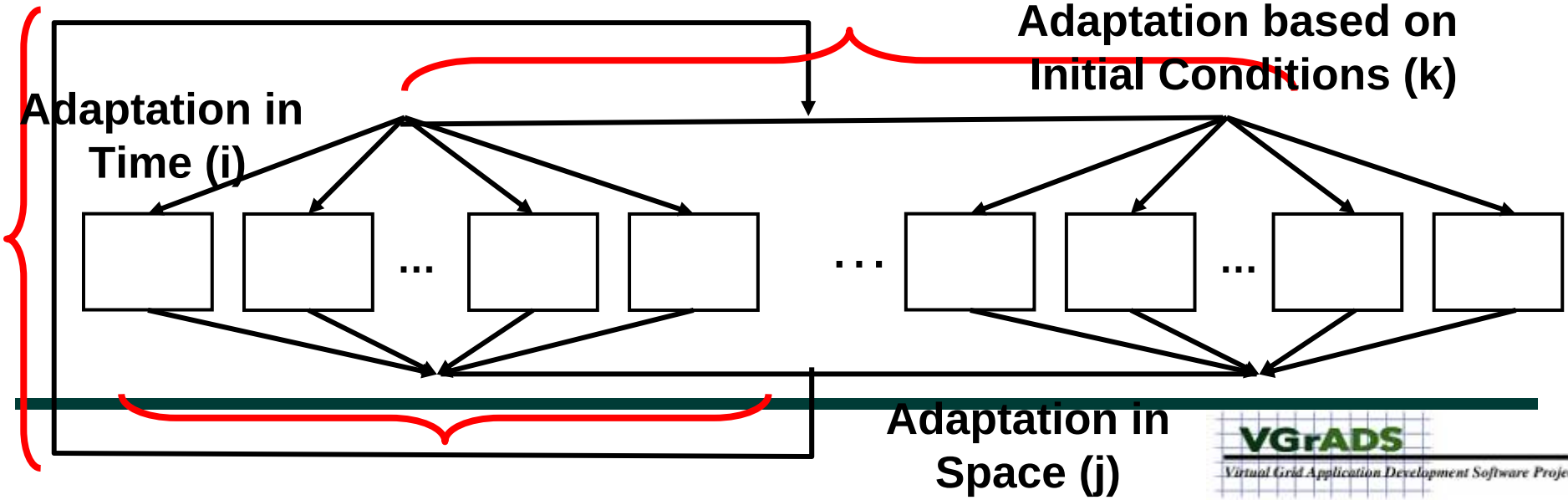
- **Coupled analysis and assimilation tools, data repositories**
  - —change configuration rapidly and automatically in response to weather
  - —Streaming data, steer remote observing technologies
- **Multilevel monitoring and intelligent control**
  - —workflow, resource, application, service
  - —performance and reliability guarantees of the resources



Adapting Workflow Structure

- **Reactive Adaptation**
  - —e.g. service failure, resource behavior
- **Proactive prediction and decisions**
  - —adjust i, j, k (weather science meets the infrastructure science) to meet individual workflow guarantees
  - —global optimization of number of workflows being serviced

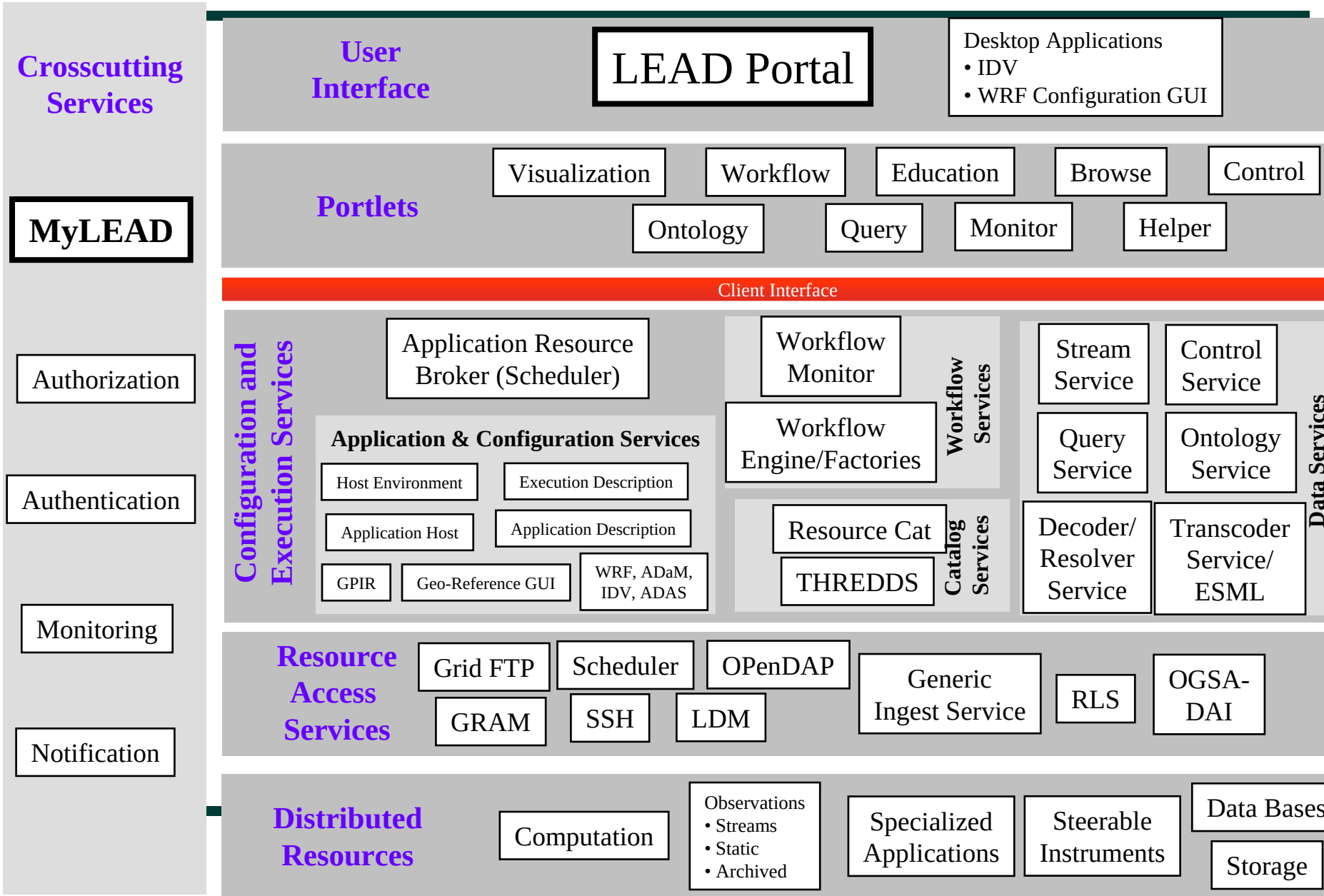




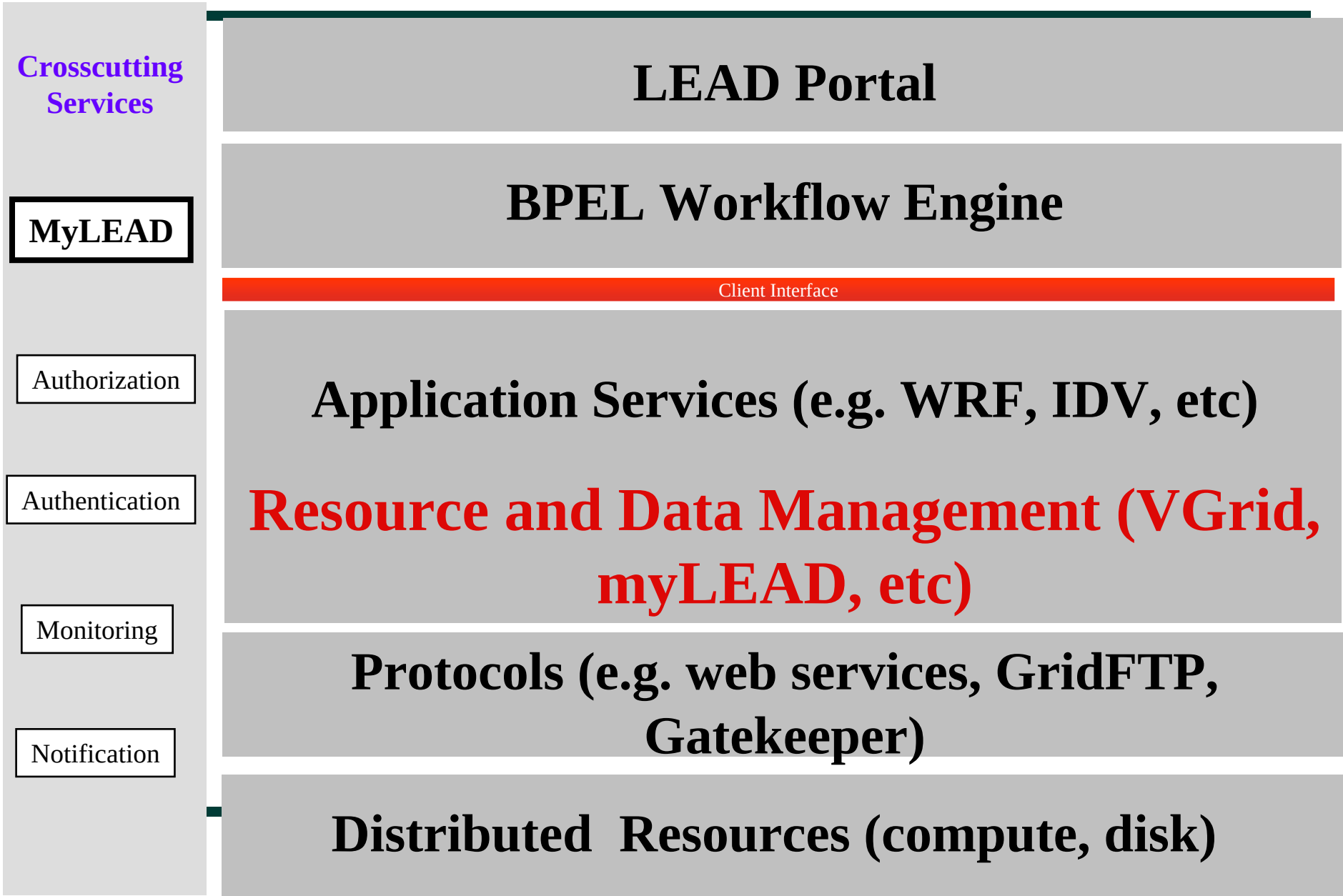
Outline

- ✓ Background and LEAD Service Architecture
  - ✓ web services, BPEL
- ✓ LEAD Workflows
  - ✓ status, site visit example run
  - ✓ characteristics, workflow structures
- LEAD Virtual Grid Integration
  - —resource management

LEAD Architecture

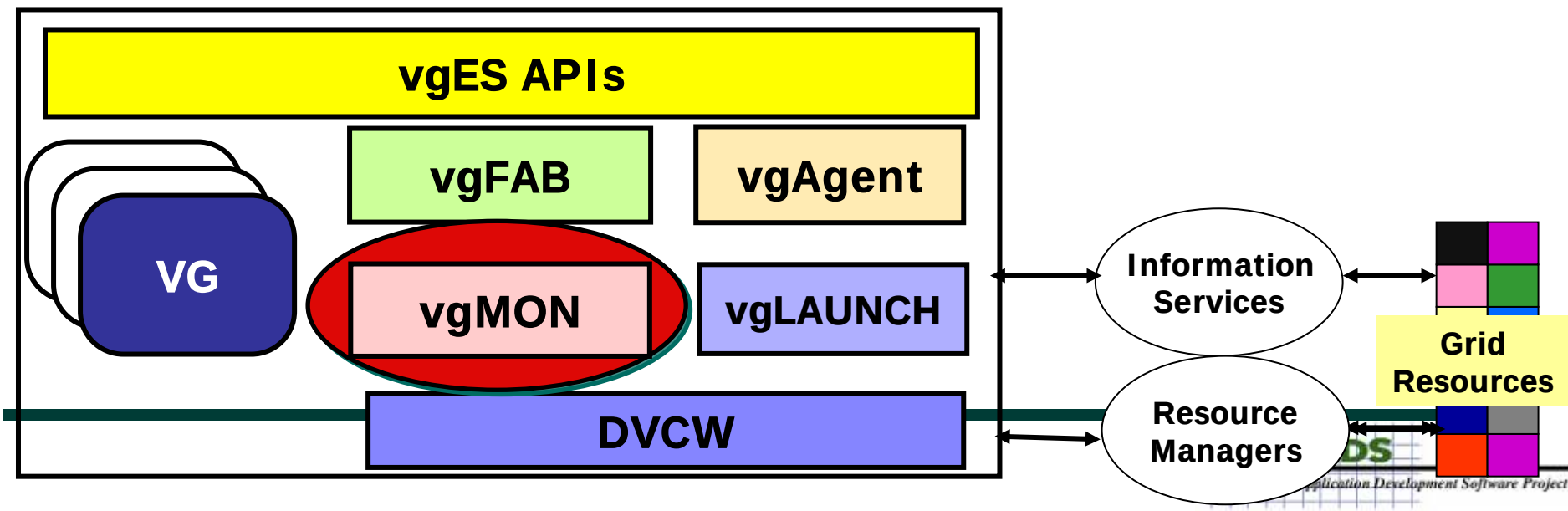


LEAD Architecture

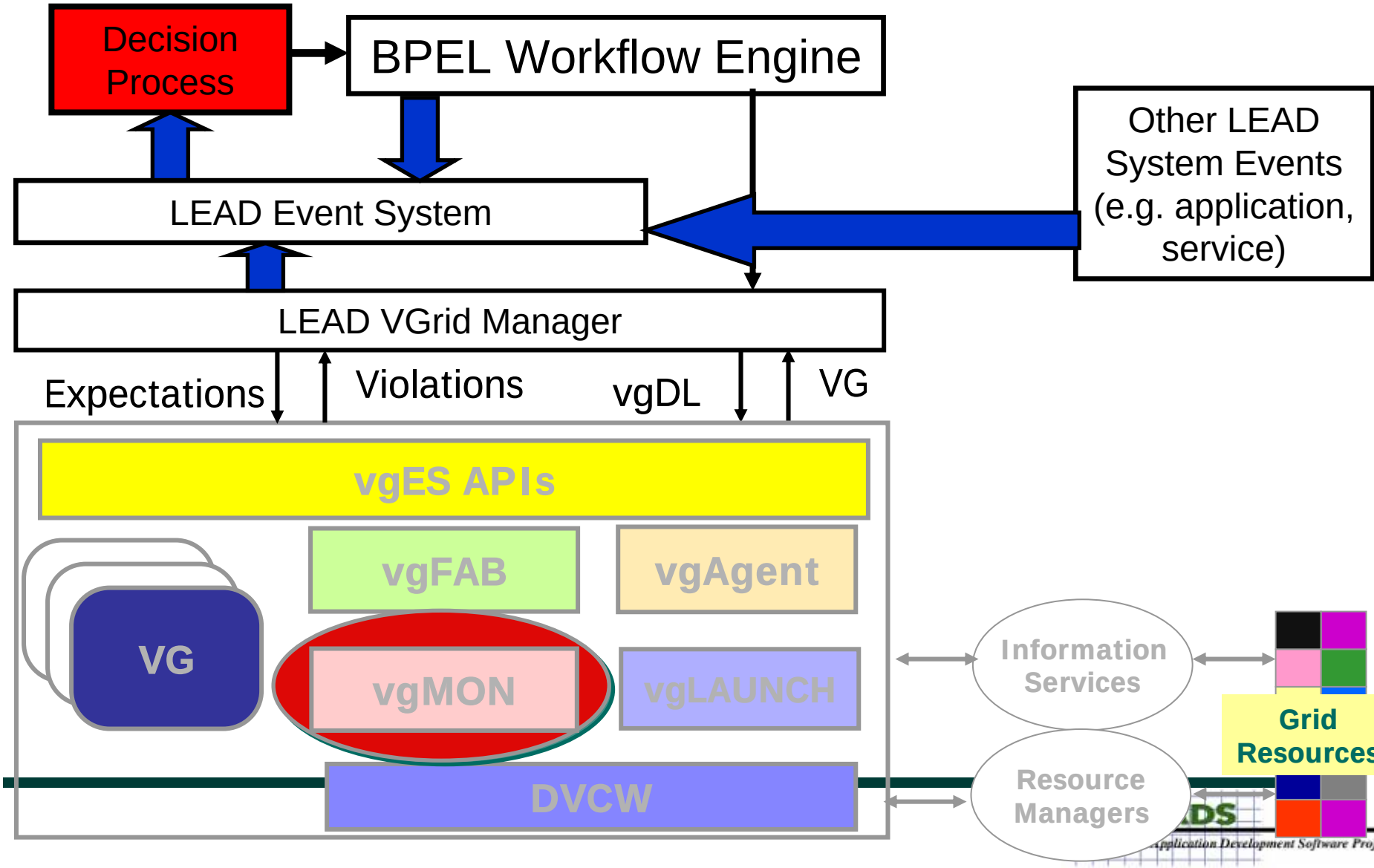


Virtual Grid provides LEAD …

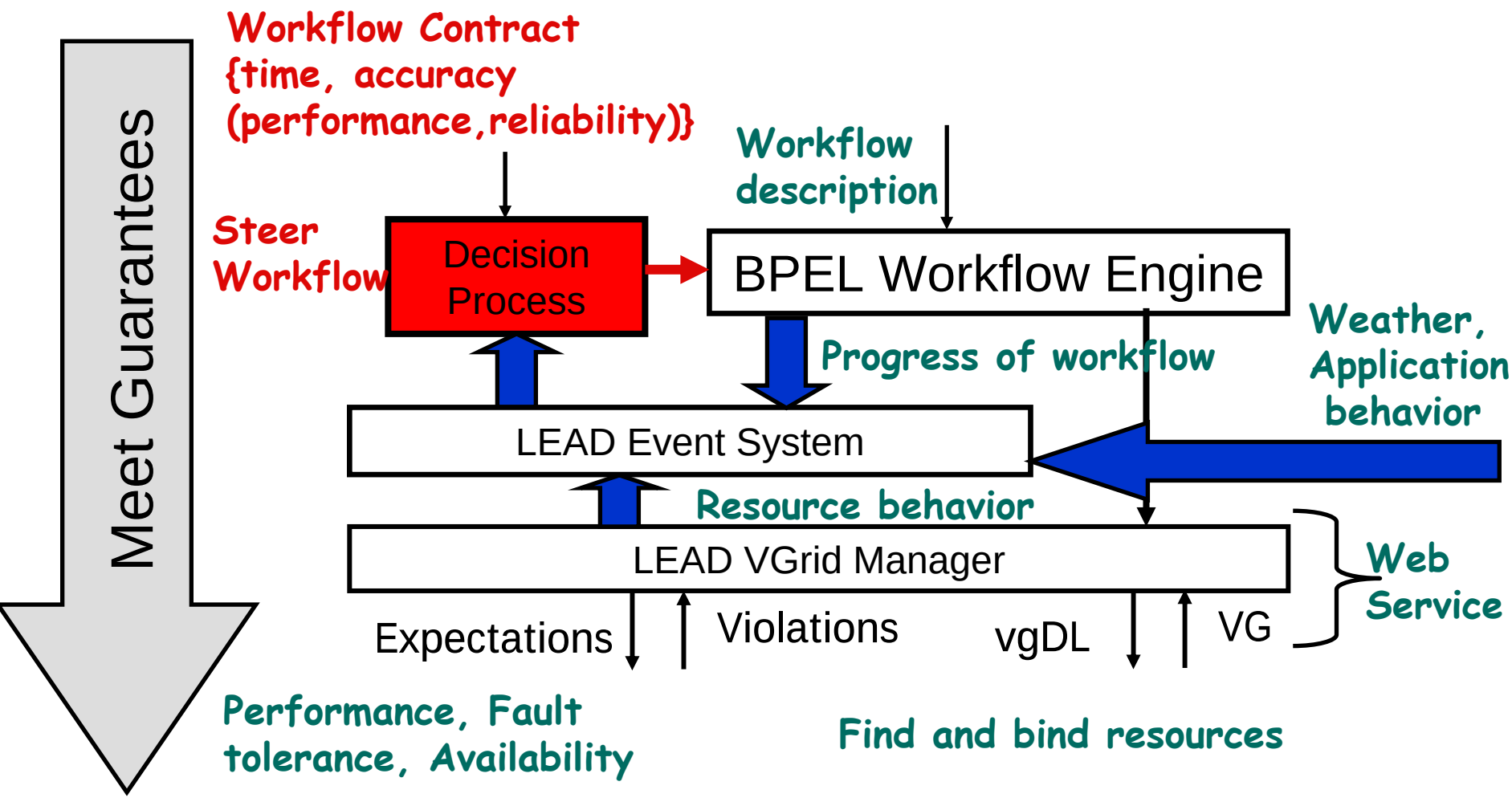
- Dynamic, scalable resource abstraction framework
  - —scheduler, resource broker
- Integrated monitoring and notification of resource behavior
  - —NWS, NWS-HAPI



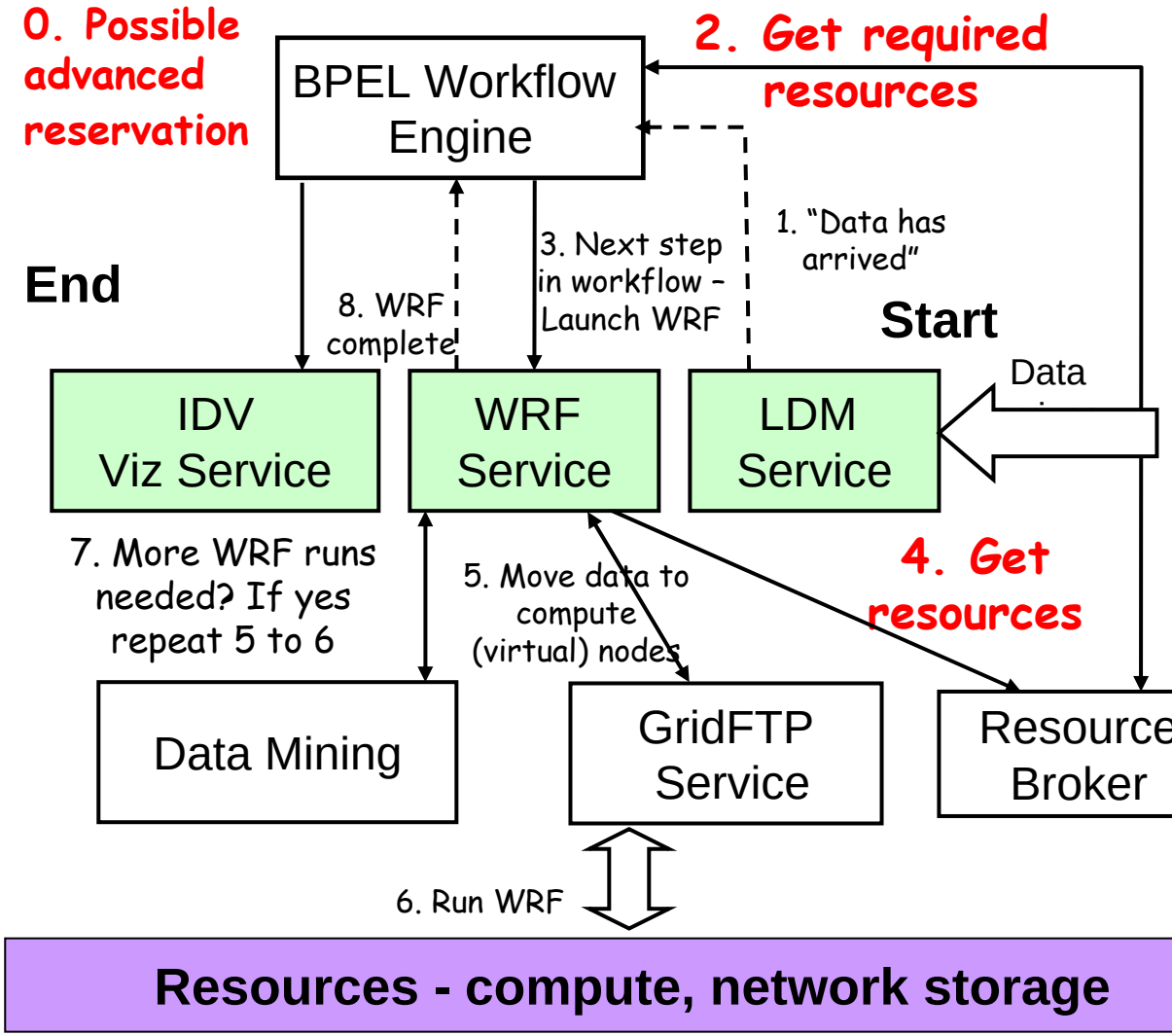
Proposed LEAD – VGrADS Architecture



Some Thoughts …



LEAD Dynamic Workflows

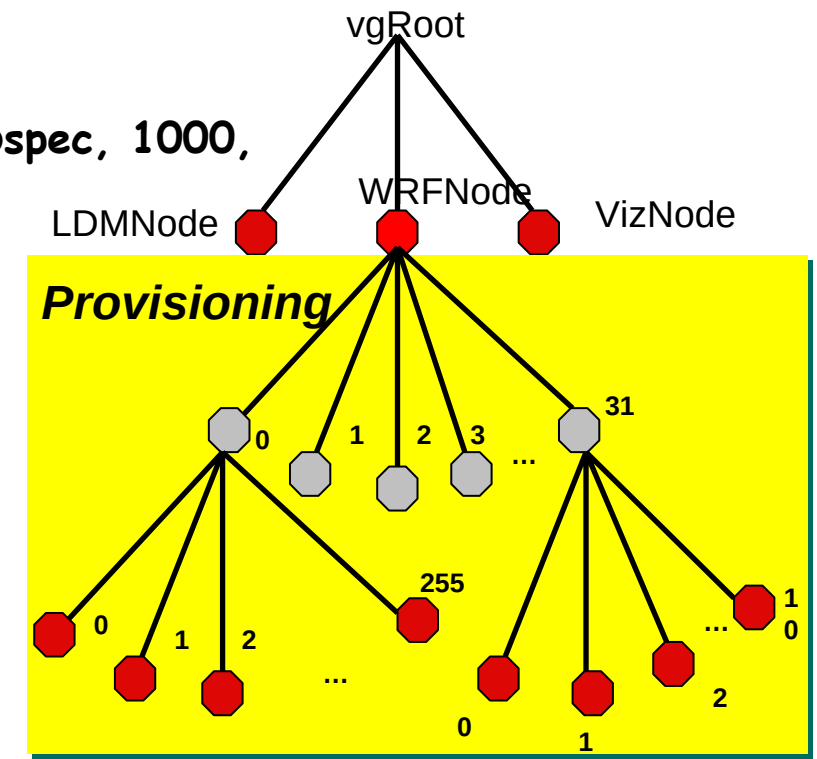


- Multilevel monitoring
- Multiple decision points
  - Streaming data
    - Unidata LDM
  - Service monitoring
    - web service load
  - Application
    - behavior on resources
  - Resource
    - performance
    - reliability

vgDL Specification for LEAD

```
LEADSpec = LDMNode = {isLDMNode}
     // Note the loose coupling
       far WRFNode = {memory >= 500MB, cpu > 2000,
    diskspace > 4GB}
     far VizNode = {memory >=4GB, cpu > 4000}
vgidLEAD = vgCreateVG(vgESsrv.renci.org, LEADspec, 1000,
    null);
vgRoot = vgGetRoot(vgidLEAD);


WRFspec = LooseBagof<C> [1:32];
       C = Clusterof<node>[4:256];
       node = {node.memory > 500MB,
                      node.cpu > 2000};
       C = {C.hasSharedFileSystem = true}
vgidLEAD = vgGetMyVG(); wrfNode = vgGetMyNode();
status = vgAddToVG(vgidLEAD, WRFspec, wrfNode,1000, init_WRF);
```

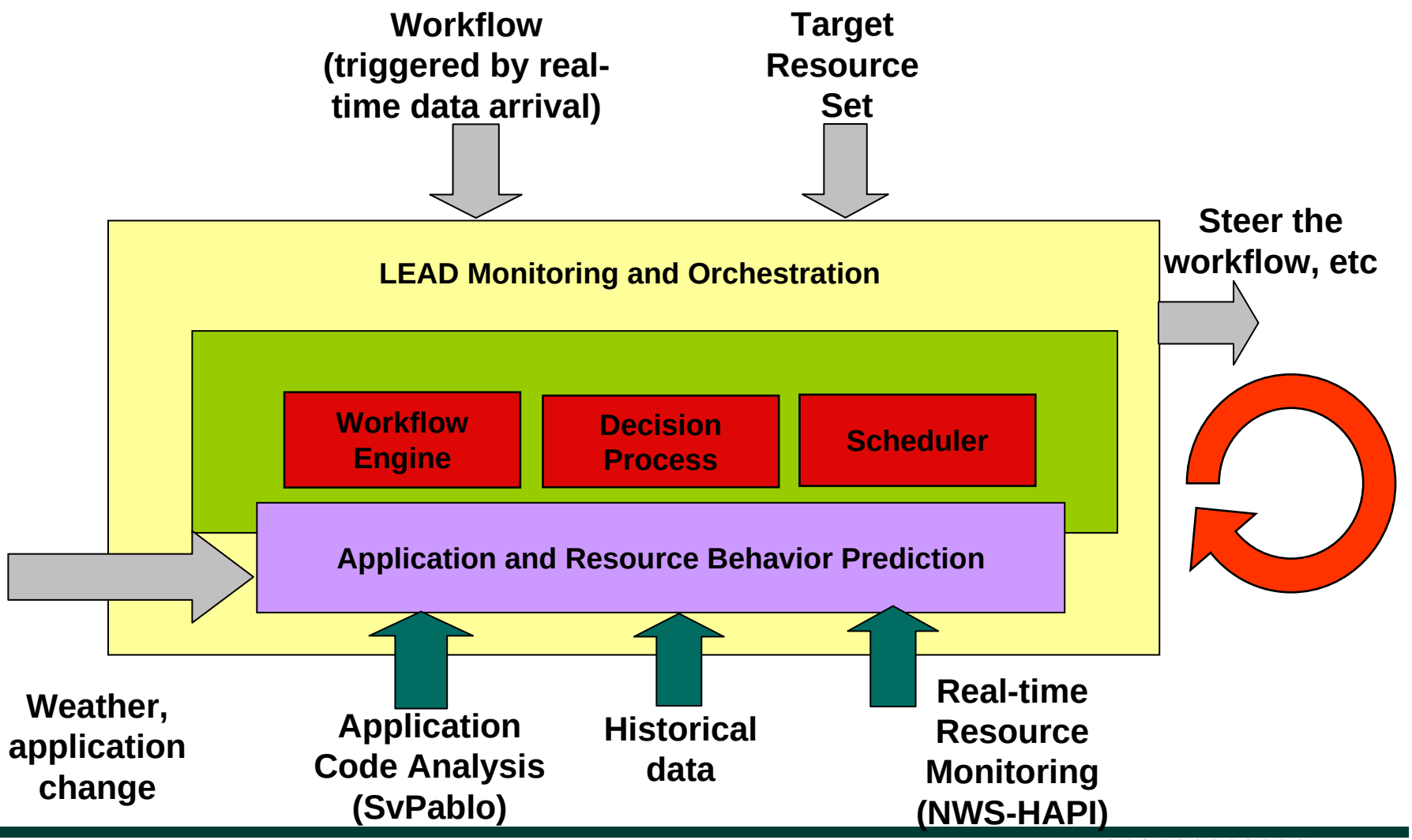


LEAD Virtual Grid Research Implications

- **Streaming data in resource scheduling**
  - —fixed point makes scheduling complicated
- **Persistent and transient services**
  - —service directory, monitoring, scheduling
- **Data management across the virtual grid**
  - —run time knowledge of data location
- **Integrated decision process**
  - —performance and reliability guarantees
  - —application, weather

Comments?

LEAD Monitoring Architecture



LEAD Control and Data Flow
LEGEND