
vgES Version 0.7 Release Overview

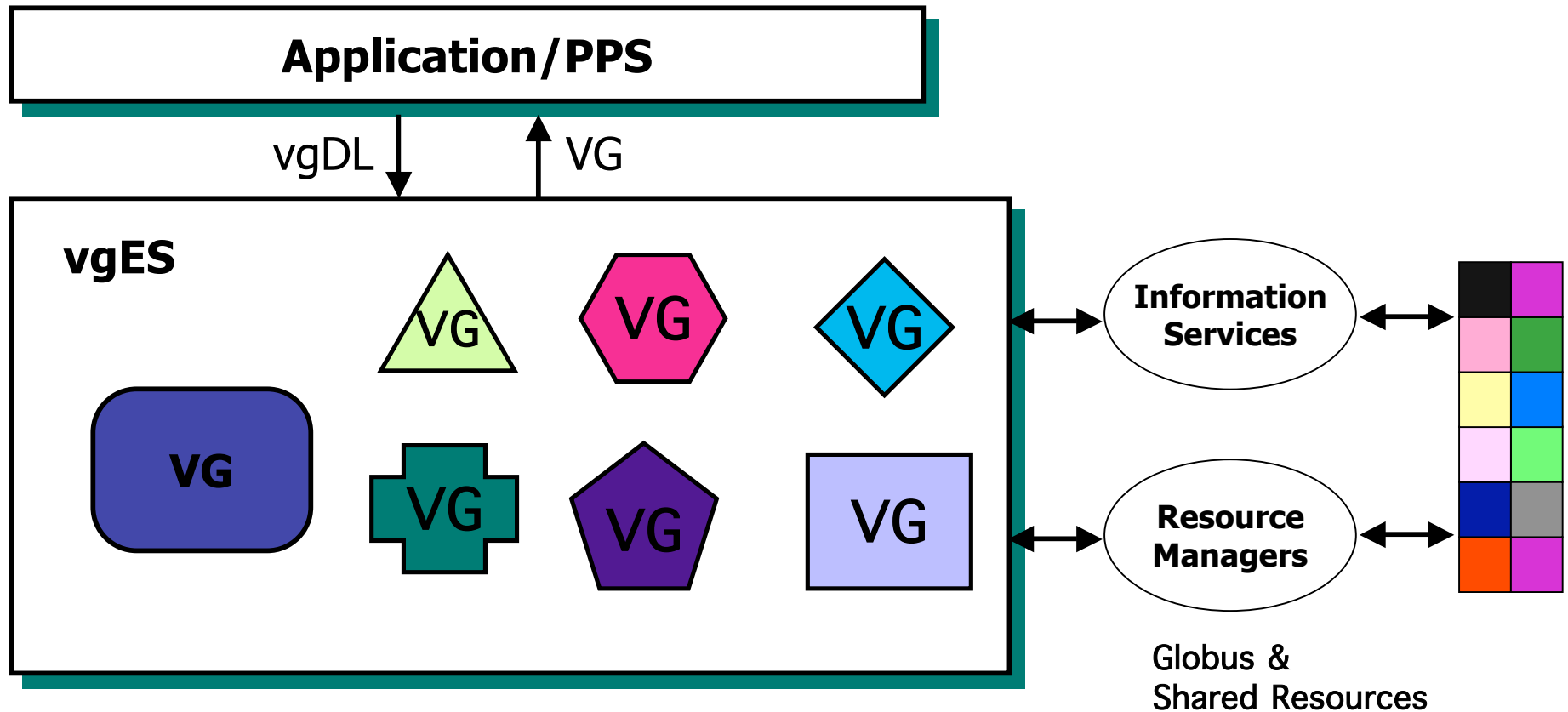
UCSD VGrADS Team

vgrads@cs.ucsd.edu

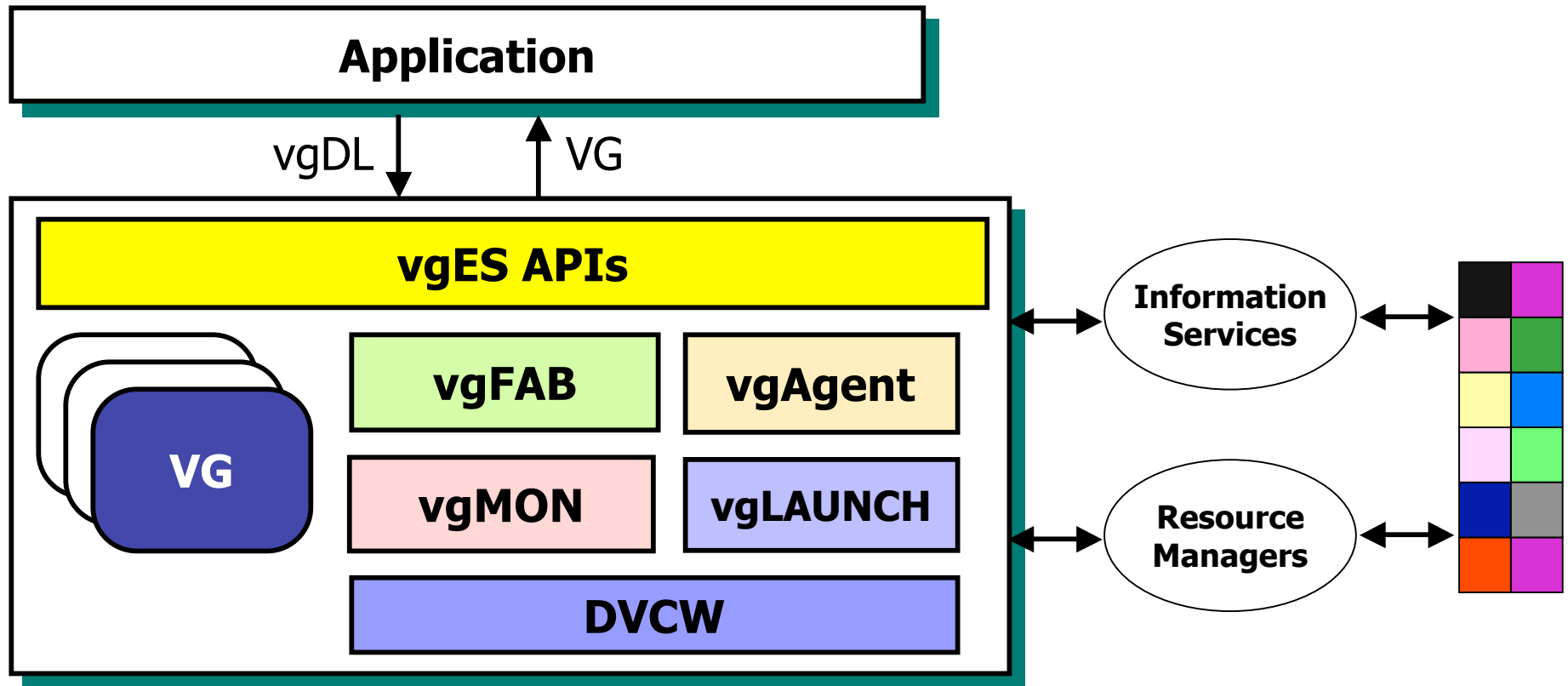
Andrew A. Chien, Henri Casanova, Yang-suk Kee,
Jerry Chou, Dionysis Logothetis, Richard Huang,
Ken Yocum

March 17, 2005

vgES Big Picture



vgES Architecture



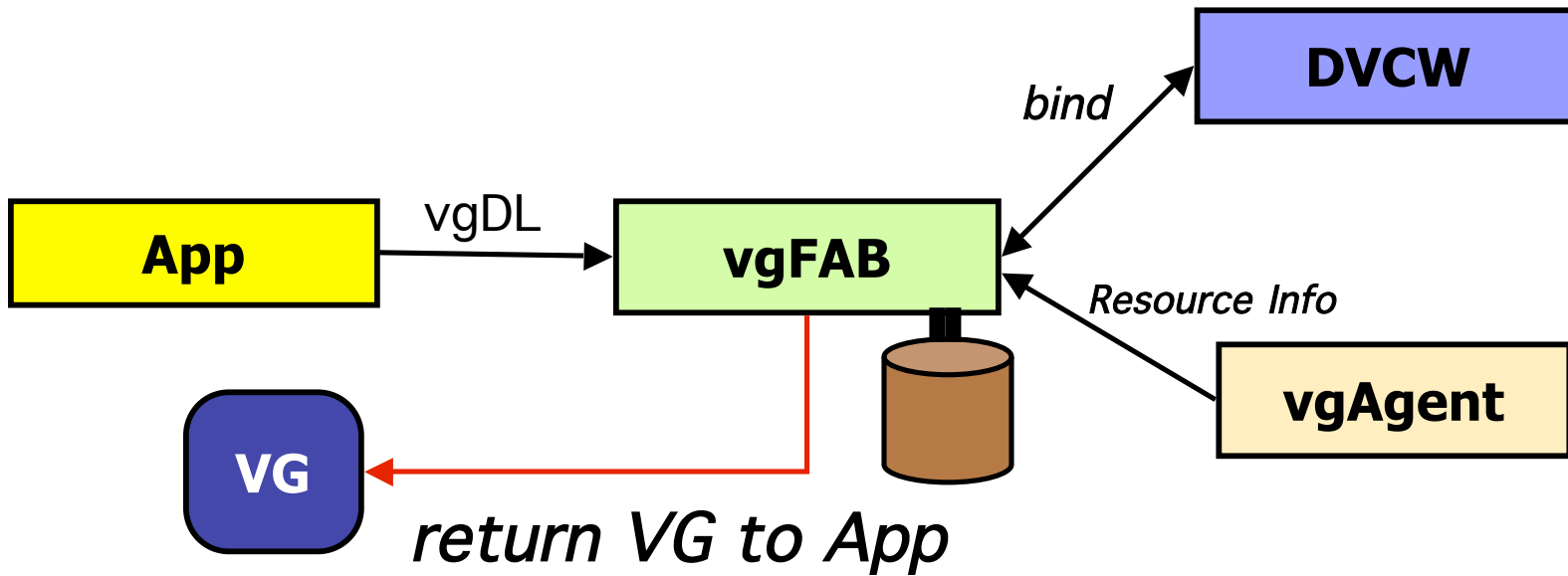
vgES Components

- **vgFAB**
 - A “finder and binder” that performs integrated resource selection and binding
- **vgLaunch**
 - An application launcher that initiates the application on the bound resources
- **DVCW**
 - Low level resource management interfaces to Globus
- **vgAgent**
 - A component that retrieves static/dynamic resource information from existing information services systems
- **vgMON**
 - A distributed monitoring component that ensures resource performance expectations

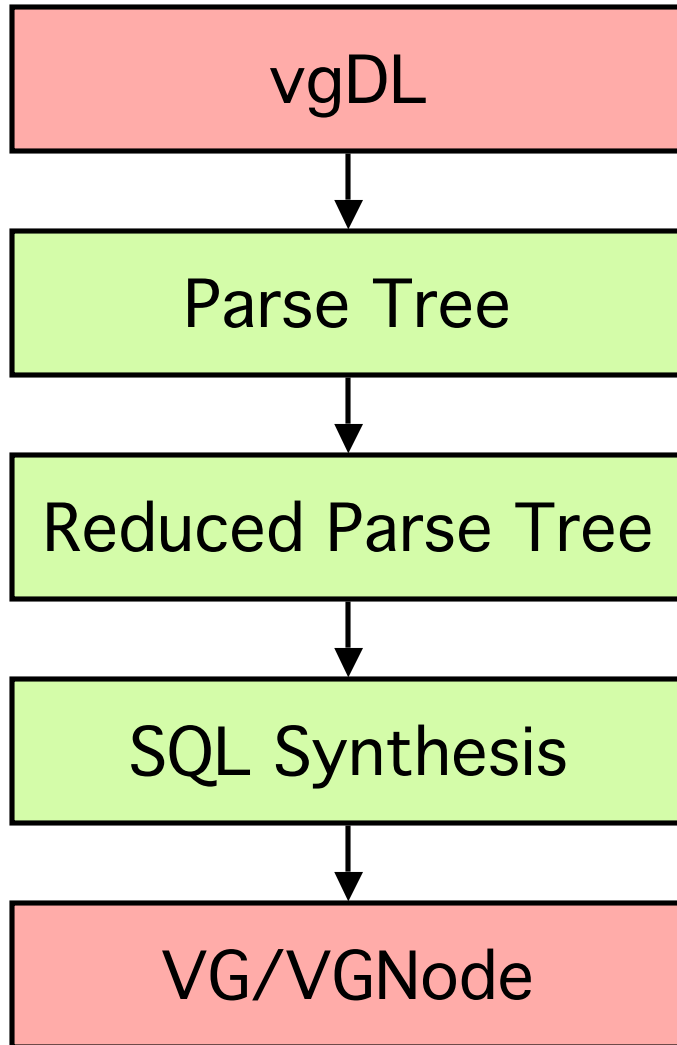
vgES Features

- **Implemented vgFAB features**
 - Virtual Grid creation and termination
 - `createVG()`, `terminateVG()`
- **Implemented VG/VGNode Features**
 - Application launch and control
 - `runCmd()`, `done()`, `terminate()`
 - Information/Attributes
 - `readAttribute()`, `getAttributes()`, `writeAttribute()`, `readInterAttribute()`
 - VG navigation
 - `getNumChildren()`, `getParent()`, `getChild()`
 - File Transfer
 - `copyFromNode()`,
- **NOT Implemented**
 - Separate resource finding and binding
 - `findVG()`, `bindVG()`
 - Dynamic VG's: Add, Remove, Adapt
 - `addNode()`, `removeNode()`, `getMyVG()`, `getMyNode()`, `getDesc()`
 - Resource Performance Monitoring
 - `setCallback()`, `vgMON` and expectations

vgFAB Architecture

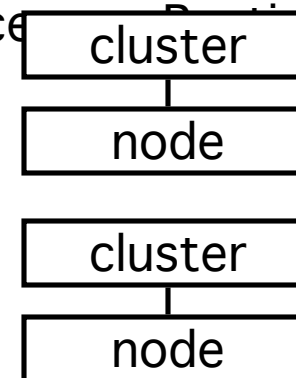


vgFAB Resource Selection



```
Rsc=ClusterOf(node)[1:10  
]
```

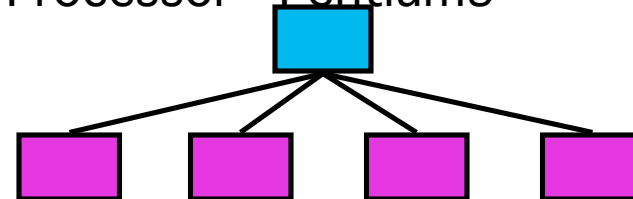
```
{node  
=[Processor=Pentium3]}
```



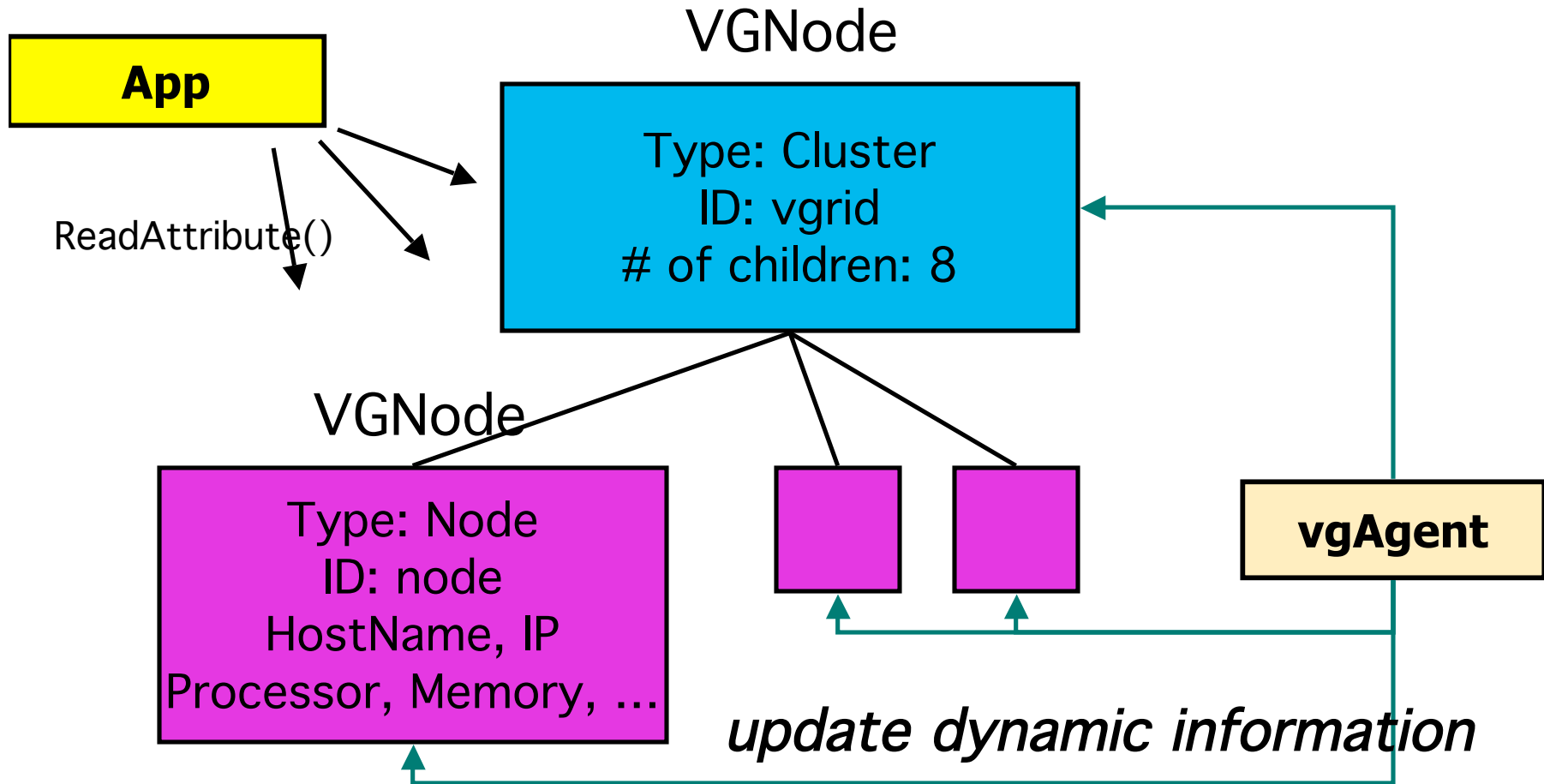
```
SELECT ID FROM Cluster
```

```
WHERE
```

```
Processor='Pentium3'
```

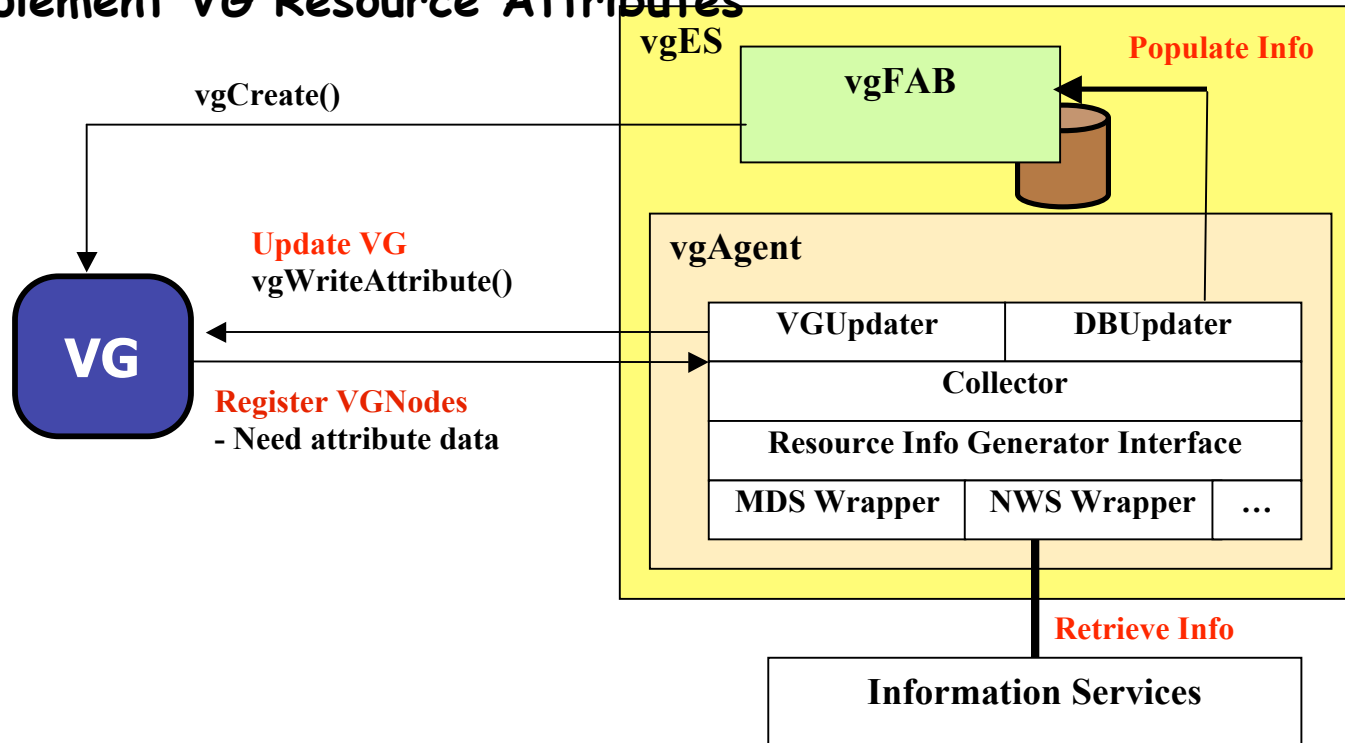


Virtual Grid (VG)



vgAgent

- Retrieve Resource Information from Information Services
- Populate the vgFAB Information store, supports resource selection and binding
- Implement VG Resource Attributes



vgAgent Support for vgFAB

- Retrieve Information and Populate vgFAB Information Store for Selection
 - Collect Attributes from Information Services
 - Done - MDS, NWS
 - Future - Ganglia, Others
 - Classifies Hosts into Clusters and TightBags
 - Cluster: Same CPU Model and Subnet
 - TightBag: Good Connectivity
 - Populates vgFAB Information Store
 - Flexible Periodic Updates
 - Adjust to tradeoff freshness vs. overhead

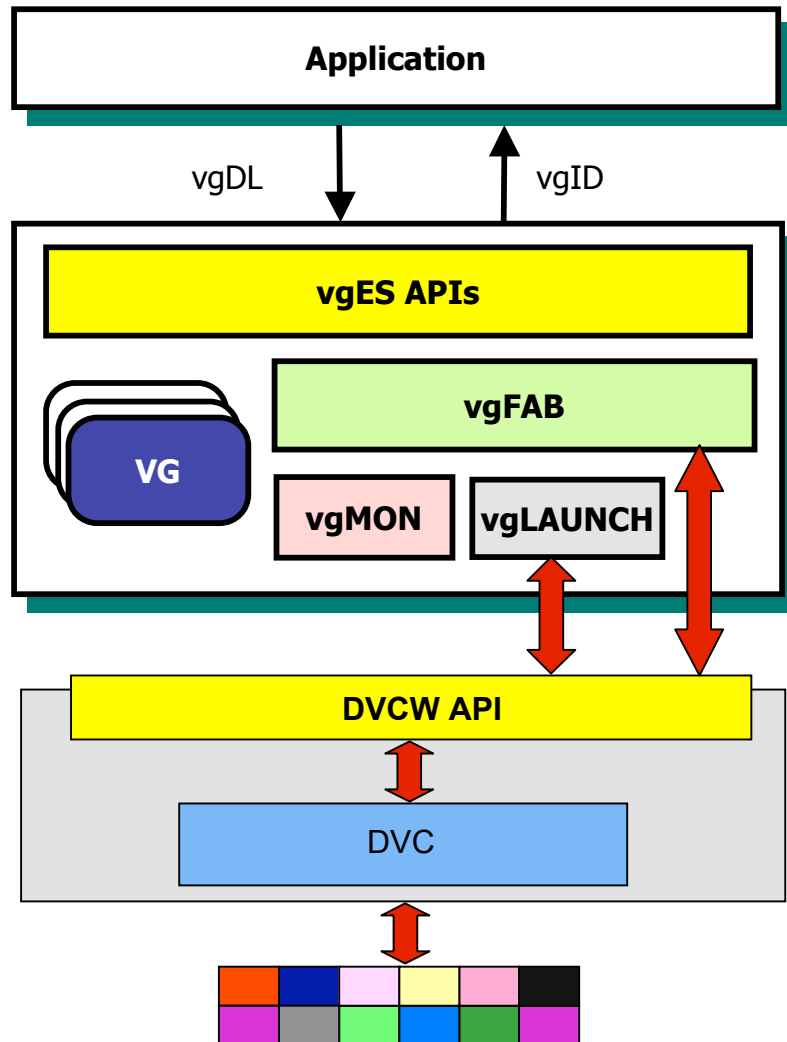
vgAgent Support for VG

- Define Attributes on VG; Interface to Information Services
- Batch Update
 - Similar to vgFAB update
 - vgAgent collects Attributes relevant to VG from underlying information service
 - Batch/Periodic update into the VG
 - Fast access by the Applications
 - Currently Implemented for MDS
- Pass Through / On-Demand
 - Application Query on Attribute Triggers vgAgent
 - vgAgent accesses underlying Information Service provider for attribute
 - Returns Value to Application, and caches if appropriate
 - Currently implemented for NWS

vgAgent Futures

- **More Sophisticated Cluster / Tightbag classification**
- **Wide range of more Efficient Attribute Implementations (range of dynamism)**
- **Advanced Inter Attributes and Management**
- **Distributed vgAgent for Scale and Robustness**

DVCW



- **Encapsulates Underlying Resource Environment and Protocols**
 - No DVC visible above
 - No Globus visible above
- **Services**
 - Resource Binding and Release
 - Launch and Manage Application Jobs
 - Move Files

DVCW Features

- **Resource Management (vgFAB)**
 - **vgBind()**
 - Low-level call to Bind an array of resources using Globus GRAM.
 - Returns an array of hosts which were successfully bound into the DVCWorkspace
 - Can access a wide range of Globus resources
 - **vgTerminate()**
 - Releases a set of bound resources
 - Any further attempted operations on the resources have no effect.
- **Application Launch, Monitoring, and Control (vgLAUNCH)**
 - **vgRunCmdOnHost()**
 - Submits a job on a remote resource and executes it
 - Returns a handle to the job, enabling monitoring
 - **vgCmdDone()**
 - Returns the job's status
 - **vgTerminateCmd()**
 - Asynchronously terminates a job
- **File Transfer (vgLAUNCH)**
 - **vgCopyToNode(), vgCopyFromNode()**
 - Copy a file to/from a node.
 - Use Globus GridFTP service.

DVCW Futures

- Use of Globus Toolkit 4
- Management of Application Execution environment
 - Globus Kickstart or other?

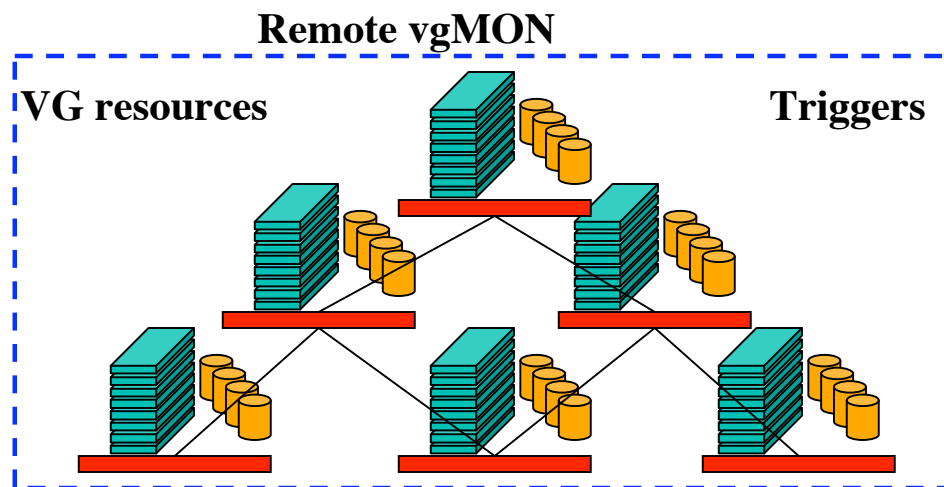
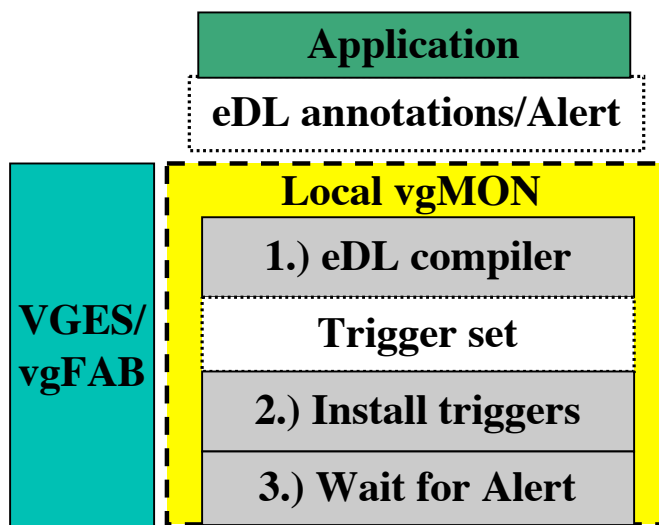
vgMON

Problem Statement:

How do we monitor a wide-area set of resources efficiently and accurately, and notify virtual grid application for debugging, adaptation, and fault tolerance?

Example:

- **Application:** $\text{Sum}(\text{cluster.CPU})[30\text{sec}] < X$
- **Trigger for each resource in cluster**
- **Trigger “fabric” aggregates CPU information**
- **Root trigger calls application “Alert” upon failed expectation condition.**



Trigger-Based Monitoring 0.1

- **Expectation Defn Language (eDL)**
 - Annotations on vgDL
 - Subset of CQL, Sophia
 - Intervals, logical operators, and functions
 - Aggregates of CPU, memory, and network
 - Extensible to application-defined attributes
- **Triggers**
 - Dynamically installed at each node
 - Form an overlay network
 - Support Variance and Sum
 - Fabrics: RandomTree and Star
- **Alerts**
 - Application-defined functions
 - Upcalled from a trigger when
- **vgMON: vgES integration**
 - VGES operations drive monitor lifecycle
 - Create/Terminate, or Add/Remove[0.2]
 - Expose expectation management directly to application[0.2]
 - Apps can write their own information providers
 - Example:
 - Failure: report after every reboot
 - `Sum(myCluster.failure)[time 1 year]>X`

vgES Documents (In Release)

- **Release Notes**
- **Installation Guide**

- **vgES CCGrid Paper (overview)**
- **vgES Application Programming Interface for Java**
- **Virtual Grid Resource Attributes**
- **vgDL Update and vgDL Design Rationale (8/2004 tech report)**

Questions?