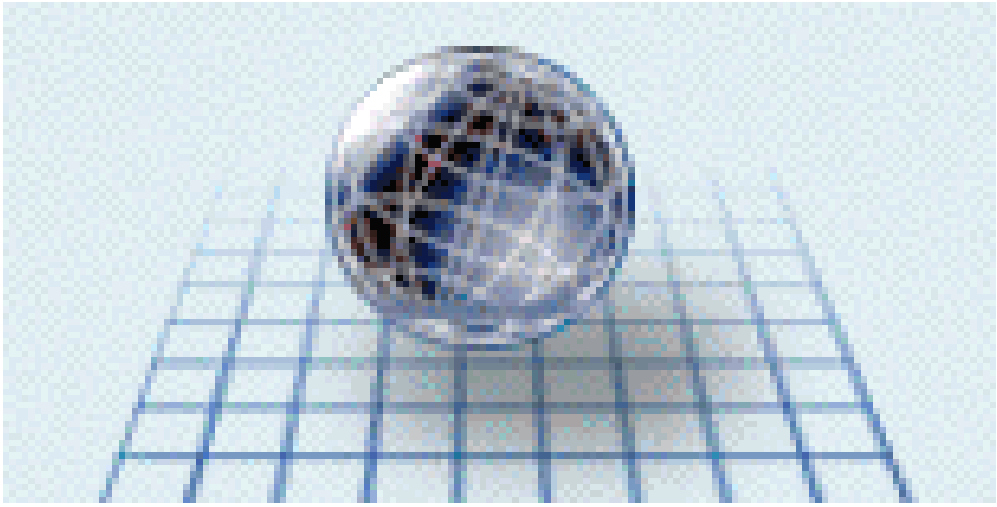
Open Grid Services Architecture: distributed computing framework

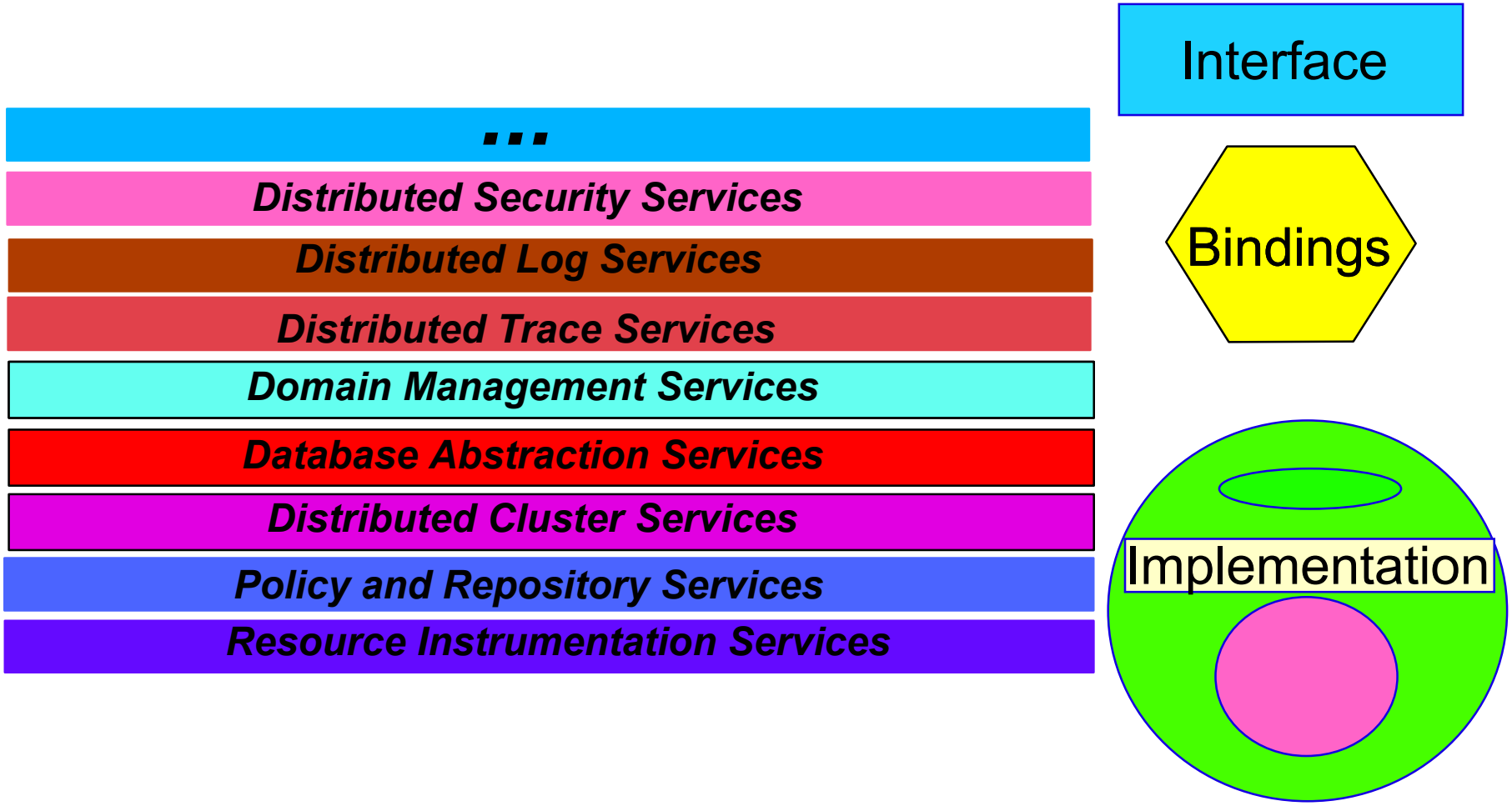


Jeff Nick, IBM Fellow
jnick@us.ibm.com
7.23.02

Open Grid Services Architecture Objectives

- Distributed Resource Management across heterogeneous platforms
- Seamless QoS delivery
- Common Base for Autonomic Management Solutions
- Common infrastructure building blocks to avoid "stovepipe solution towers"
- Open and Published Interfaces
- Industry-standard integration technologies
  - web services, soap, xml…
- Seamless integration with existing IT resources

OGSA Meta-OS Functions



OGSA build-out

End-Users and Admin Staff

S&TC eBusiness xSP etc.

New classes of applications and Grid's

Common GUI

Common

Platform Unique

Install
Config
Deployment
Maintenance

OGSA Autonomic Management Services

eWorkload Management
eCluster Management
eUtility
eAutomation
eDisaster Recovery

OGSA Grid Middleware Services

Globus
Platform
Avaki
Entropia...

OGSA Meta-OS Services

Clustering
Policy
Data Access / Replication
Resource Instrumentation
Problem Determination
Logging

OGSA Domain Services

Provisioning
Service Collections

OGSA Infrastructure

Discovery
Lifecycle
Registry
Manageability (serviceData+)
Factory
HandleMap
Notification

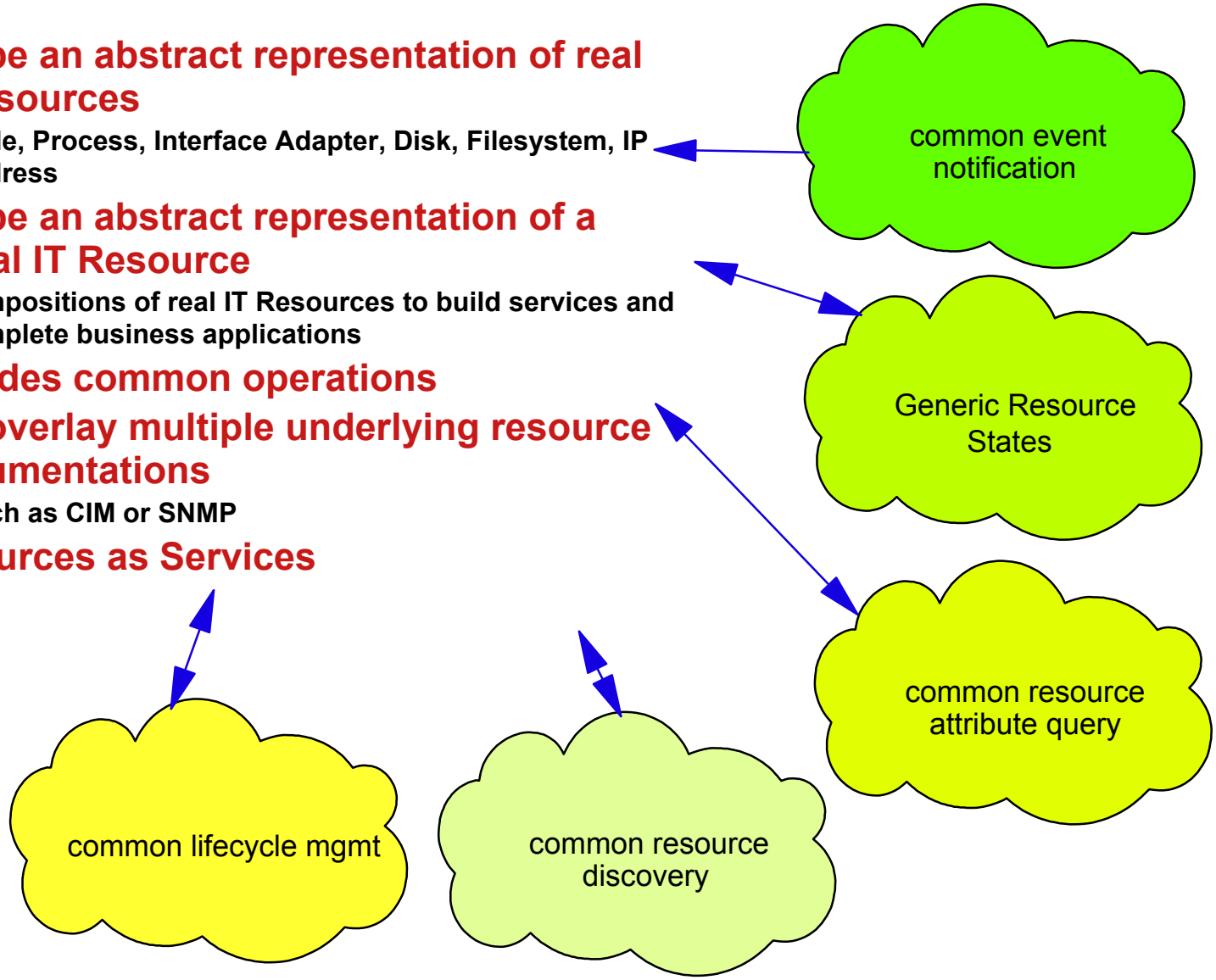
.Net JBOS WebSphere. software WS-Security

WSAD OGSA Service Plug-in's

z/OS	AIX	OS/400	Linux	Windows	Storage	Solaris & HP/UX	Linux Windows	Network	Storage
zSeries	pSeries	iSeries	z,p,i,xSeries	xSeries	Shark	Sun & HP	Dell Compaq ...	Cisco Nortel ...	Veritas ...

OGSA Common Resource Models

- Can be an abstract representation of real IT Resources
  - Node, Process, Interface Adapter, Disk, Filesystem, IP Address
- Can be an abstract representation of a logical IT Resource
  - compositions of real IT Resources to build services and complete business applications
- Provides common operations
- Can overlay multiple underlying resource instrumentations
  - such as CIM or SNMP
- Resources as Services



Common Resource Model

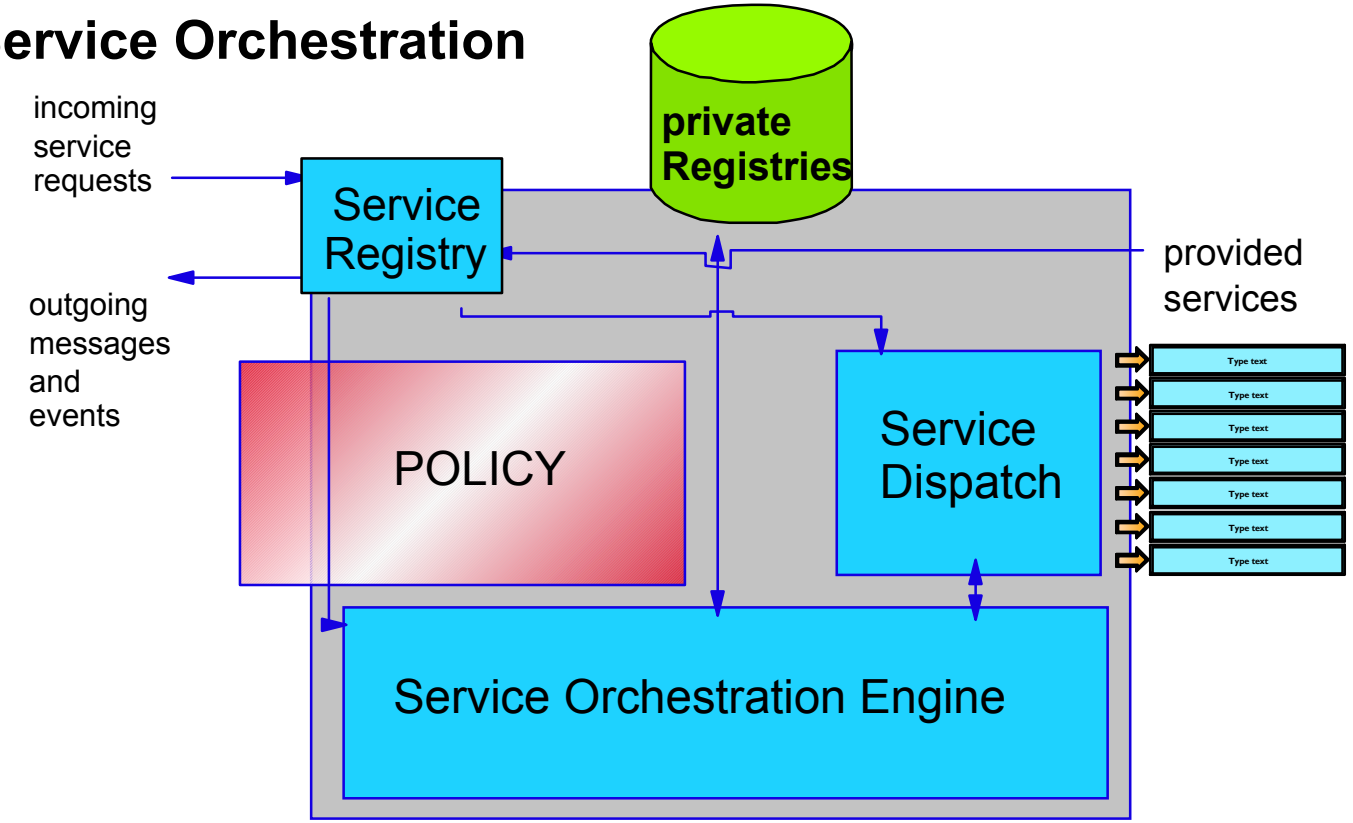
- Resources expressed as Grid Services
  - Use CIM models as base for resource models as applicable
  - Express in WSDL/GSDL as grid service
  - Mix in the base grid service port types as needed
- CIM as the basis for the meta-model
  - Add, delete, change as necessary, but not be constrained by CIM or DMTF work
  - e.g. use constructs from xml/xsd where similar ones exist in CIM
- A resource may have more than one binding to access that resource's manageability information
- A service is the concrete description of the binding

Service Domains

- Service Domain is a collection of services that satisfy the requirements of one or more usage domains. It is a functional component of a distributed system.
- Services in a service domain can be -
  - Resource oriented, such as CPU, storage space, network bandwidth, etc.
  - Systems and infrastructure oriented, such as security, routing, management, etc.
  - Application oriented, such as purchase orders, stock transactions, travel, etc.
- A service domain can be targeted towards
  - compute-intensive functions, such as financial calculations, scientific and engineering computing
  - transactional and business-process functions such as ERP, CRM etc.
- Services in a service domain may be
  - a homogeneous or heterogeneous collection of services.

Service Domains: Distributed System Components

- Service Registration and Collection
- Service Routing and Selection
- Service Interoperation and Transformation
- Flexible Service Composition
- Autonomic Service Orchestration



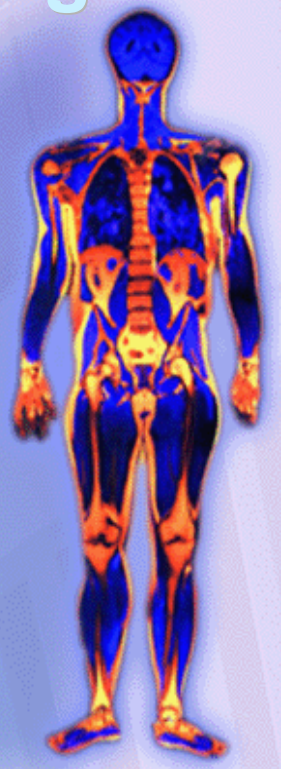
Proposal:

- Architect Service Domain as a core OGSA Domain Service (or services)
  - represents the base unit of a trusted collection of services to form a Service Domain
  - provides servicetypes for the registration, discovery, selection, routing, fail-over, creation, destroying, enumeration, iteration, and topological mapping of service instances represented by this Service Domain
- Architect the service policy format required for the operations of Service Domains
- Architect ways to relate a master service policy format to multiple Service Domains

Towards Autonomic Computing

Self-configuring Self-optimizing

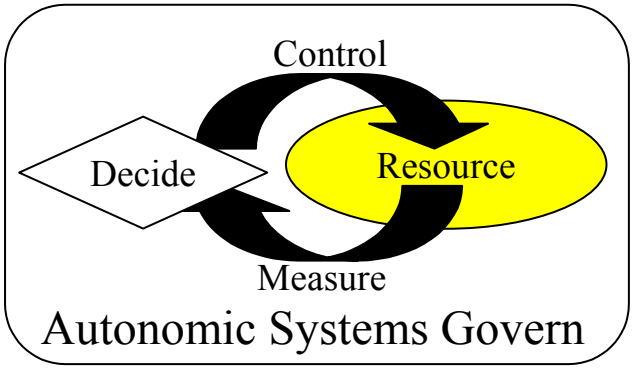
- *Build* an open, integrated infrastructure
  - Open Grid Services Architecture
- *Access and Share* the infrastructure
  - Grid Computing
- *Manage* the infrastructure
  - Autonomic Computing



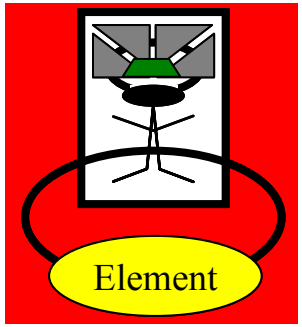
Self-healing Self-protecting

Autonomic Systems Govern

The autonomic nervous system governs many of the body's involuntary functions, including the heart rate, the respiratory rate, the blood's sugar and oxygen levels, the body temperature, the digestion, and the pupil dilation. It frees the conscious brain from the burden of having to deal with these vital but lower-level functions." Jeff Kephart

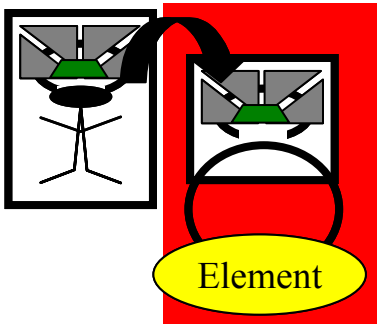


Management



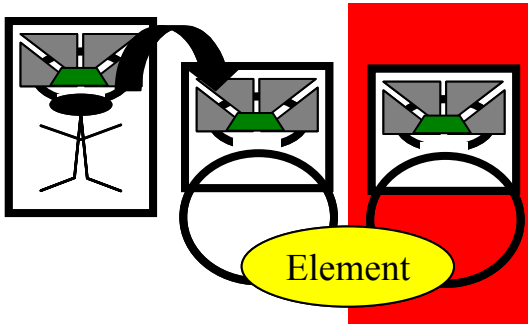
Human knowledge about the environment is used to make decisions.

Automation



Human delegates decisions by automating/encoding them into the system.

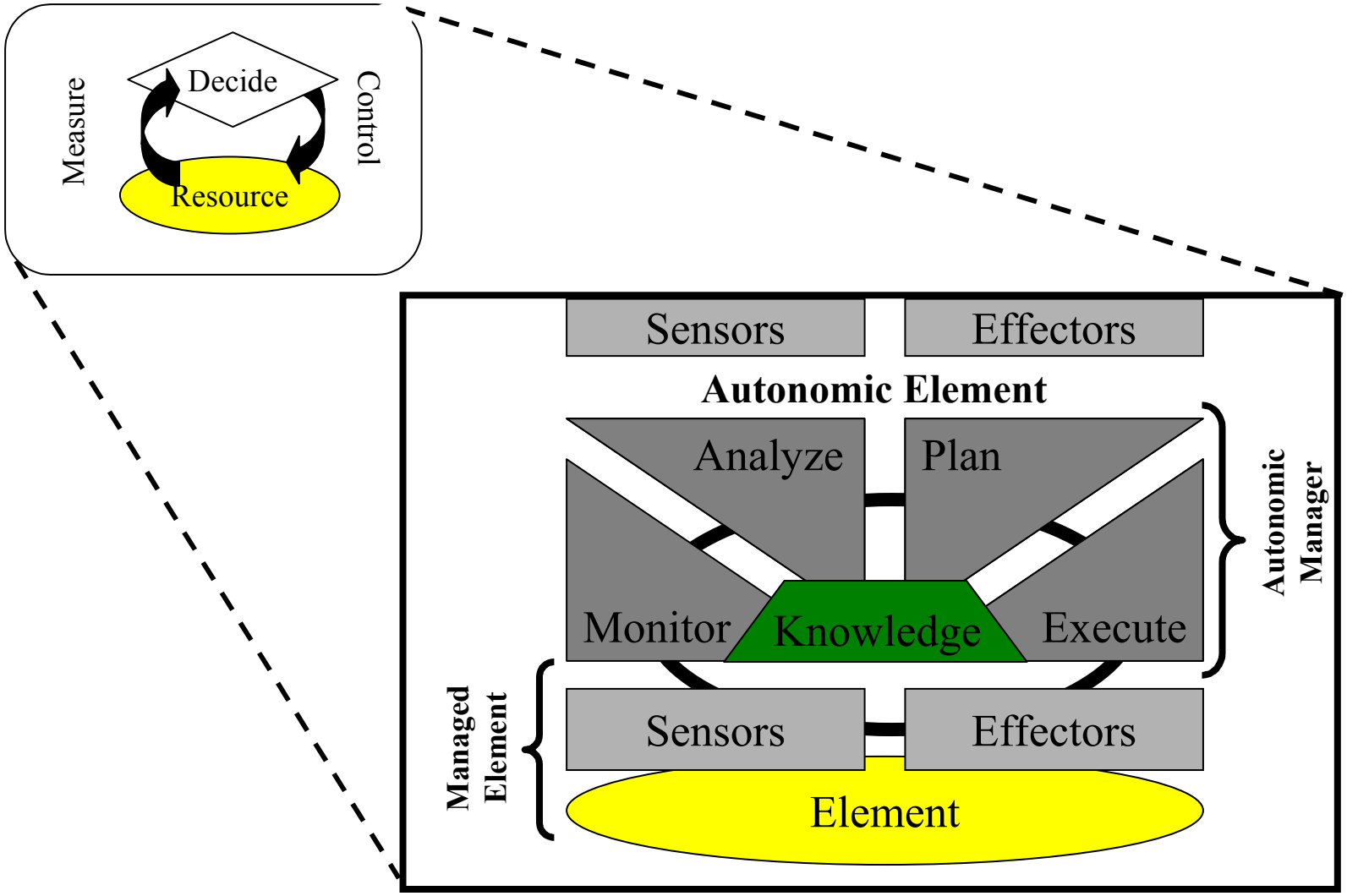
Autonomic



Human does not need to encode the knowledge.

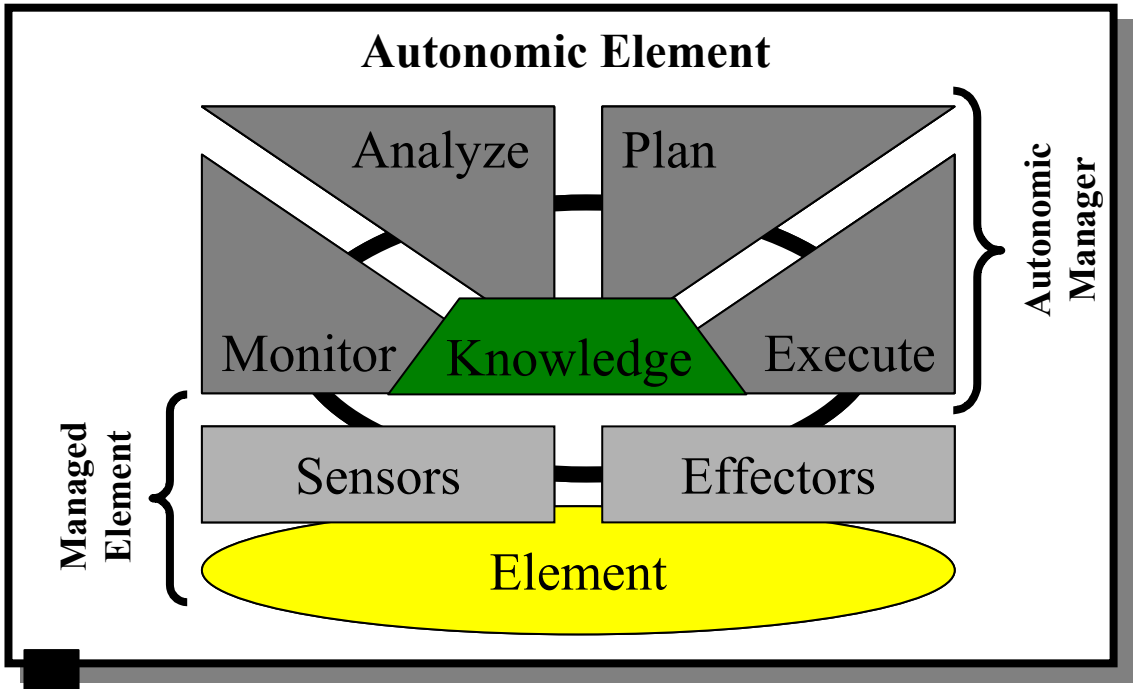
Building autonomic computing system is a continuation of the management journey.

Autonomic Element

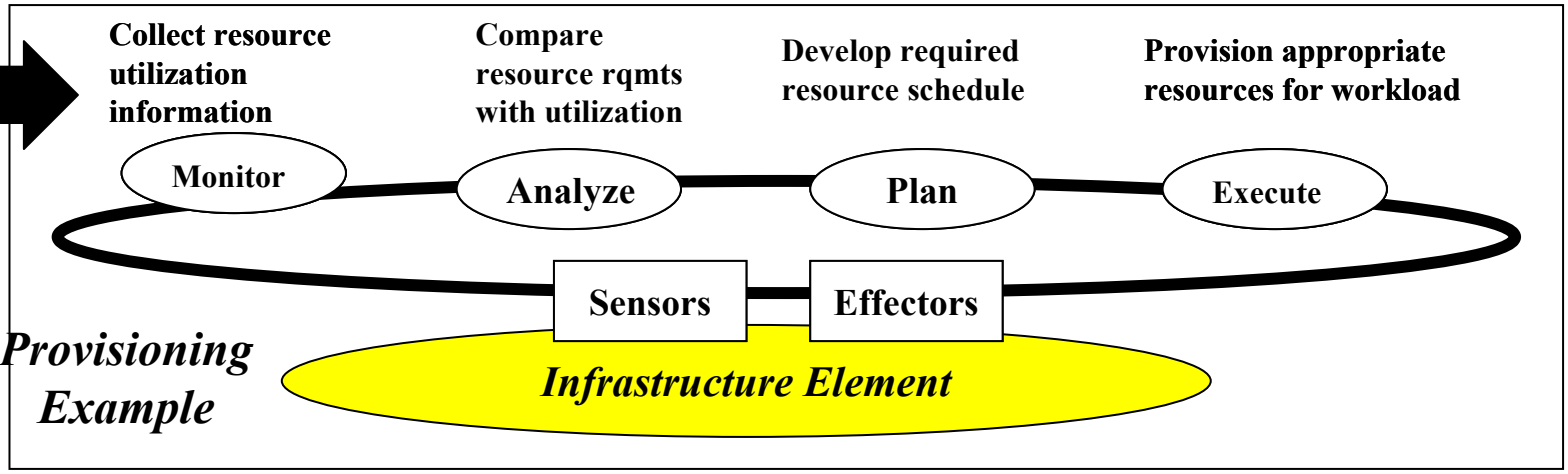


Autonomic elements are functions in a system that monitor activities and adjust the system to accomplish system wide policy.

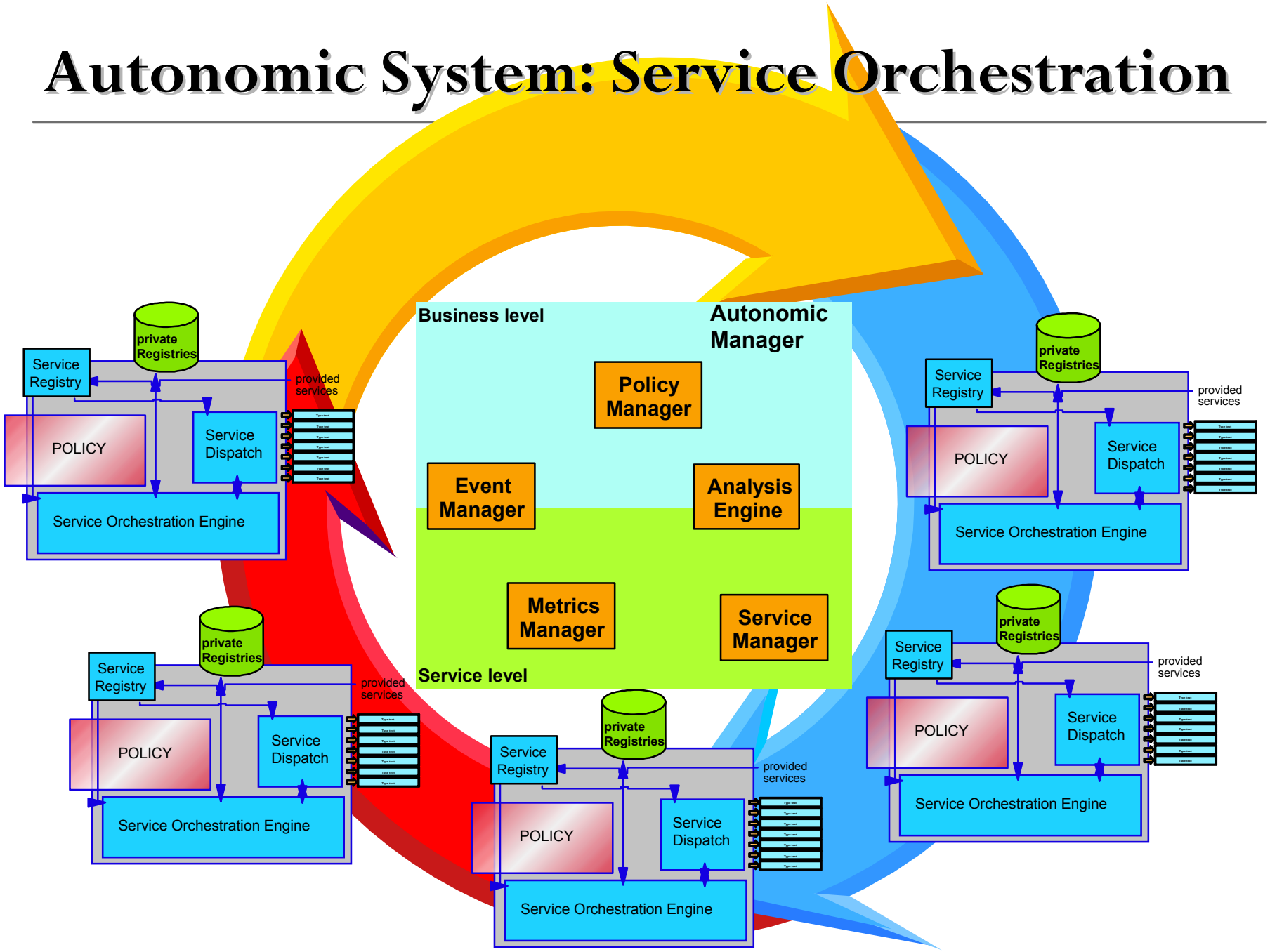
Provisioning: Autonomic Computing Architecture



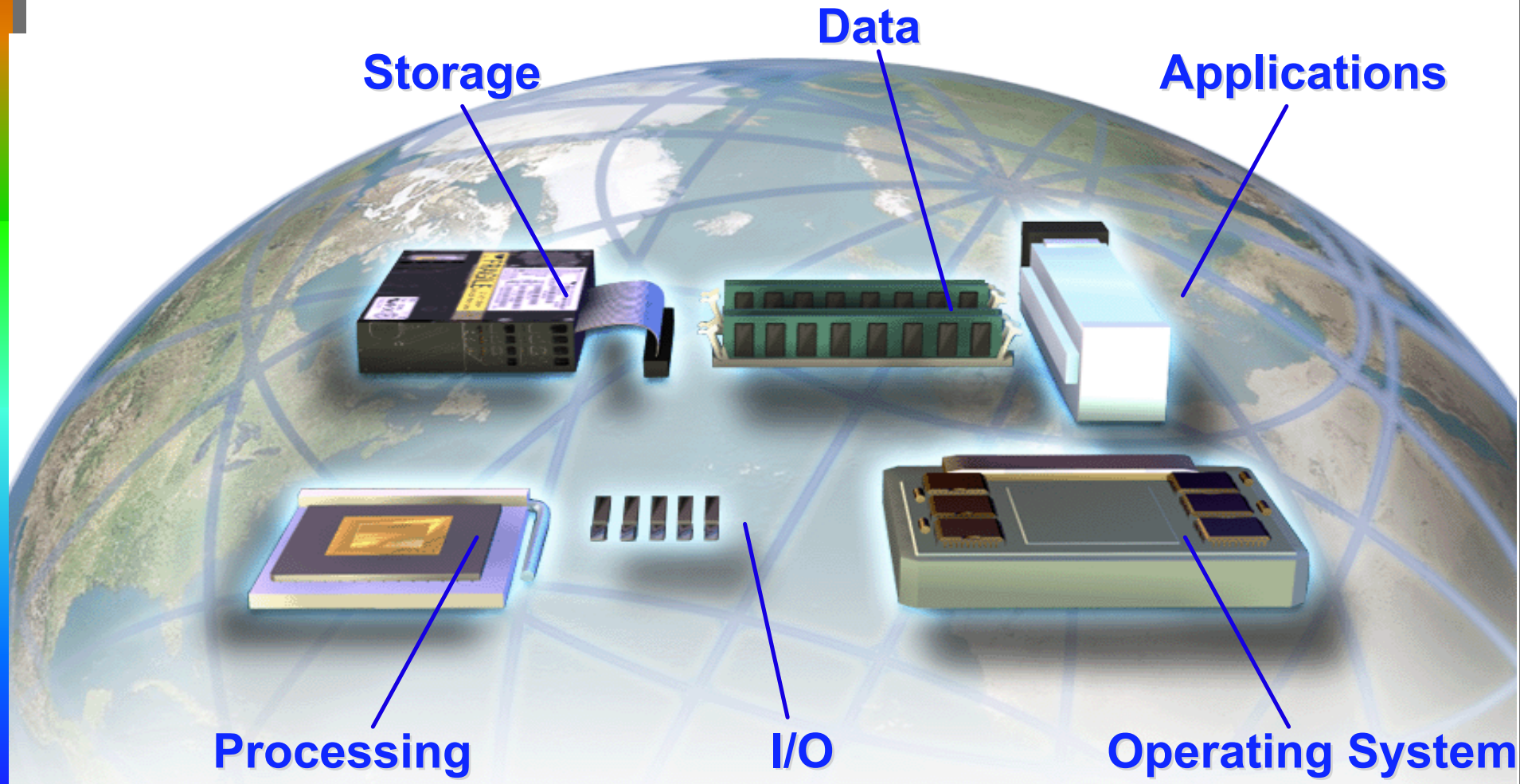
Provisioning is a task that needs to be instrumented in all applicable autonomic elements. The element provides appropriate sensors for sending resource utilization status, and effectors for re-allocation of the resource.



Autonomic System: Service Orchestration



Grid Computing and OGSA



Virtual, distributed computing platforms, 'limitless' global resources