OGSA - Basic Services

Reference: Joshy Joseph & Criag Fellenstein,

Grid Computing – On Demand Series, Pearson Education, IBM press, 2011

OGSA Services Overview

- Common Management Model (CMM)
- Service domains
- Policy
- Security
- Provisioning and resource management
- Accounting/metering
- Common distributed logging
- Monitoring
- Scheduling
- Distributed data access and replication

Open Architecture

OGSA – Open Grid Services Architecture

Applications & systems built on standards

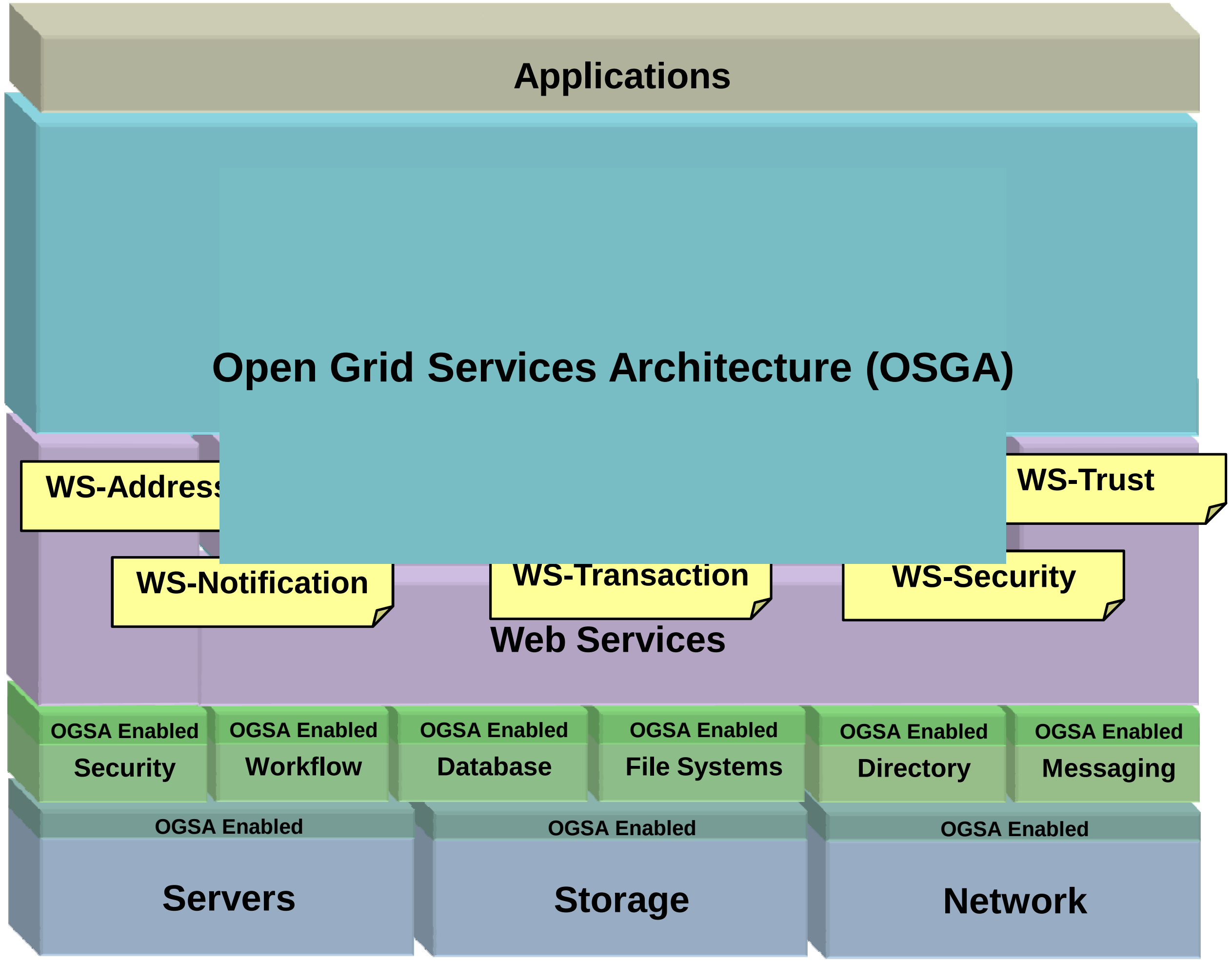
Open and value-added vendor implementations

Open architecture for interoperability

Support for web services on a variety of platforms, languages and protocols

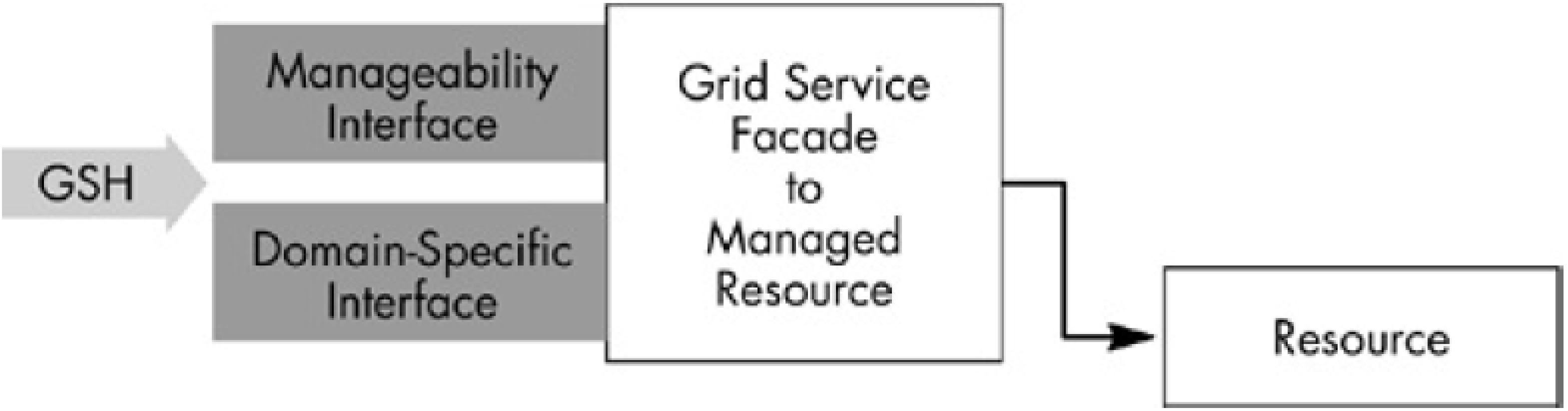
Enabled “general purpose” middleware

Enabled Hardware and Operating System Platforms



Common Management Model (CMM)

- The Open Grid System Architecture (OGSA) **Common Management Model (CMM)** is an abstract representation of real IT resources such as disks, file systems, operating systems, network ports and IP addresses.
- The CMM is a "single" model for management that can be utilized with and extended for multiple grid resource models.

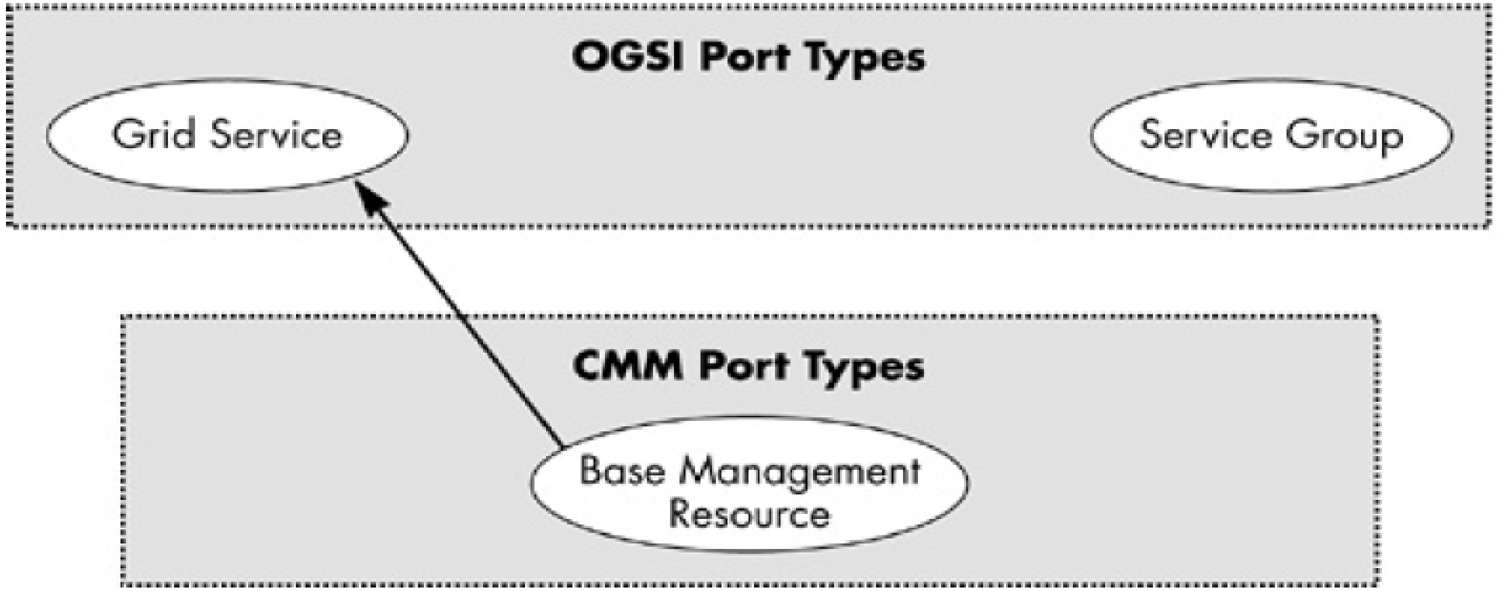

GSH
Manageability Interface
Domain-Specific Interface
Grid Service Facade to Managed Resource
Resource

Common Management Model (CMM)

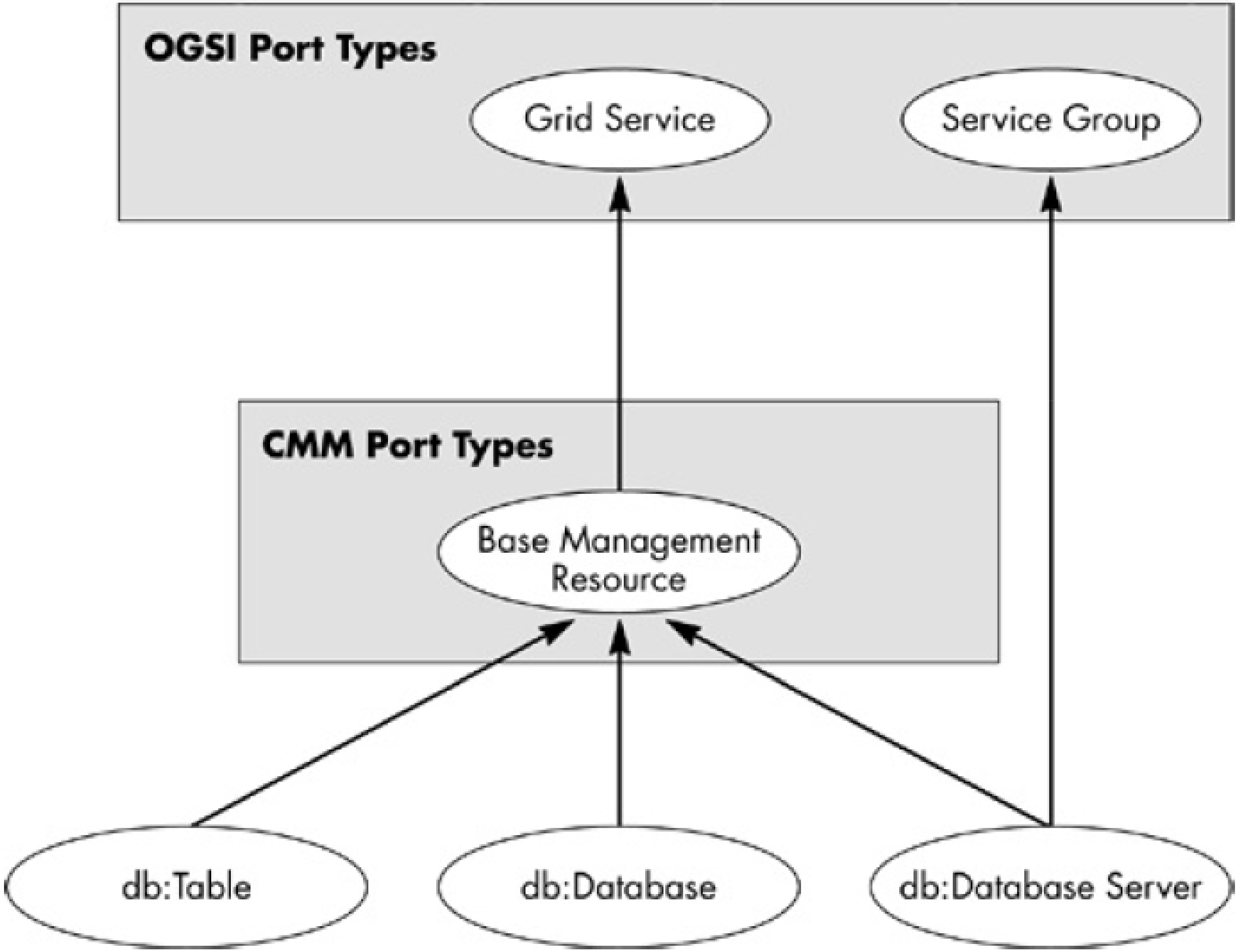
- **Manageability Interfaces:** It exposes a set of canonical interfaces and behaviors common to all the CMM services.
- **Domain-Specific Interfaces:** domain-specific interfaces are tightly coupled to the domain in which these resources are defined.
- The OGSA CMM specification defines three aspects of manageability:
  - An XML schema (XSD) for modeling the resource manageability information
  - A collection of manageability portTypes
  - Guidelines for modeling resource

CMM Port Types

- **GridService port type** is the core interface and is present in all grid services, and it provides a set of common behaviors and operations. The other interface defined by
- **BaseManageablePortType:** The behaviors represented by this port type include resource lifecycle data, relationships to other resource types and data, searchable resource properties, and resource groups to which this resource instance belongs within the respective environment.



Example ManageablePortType



Resource Modeling

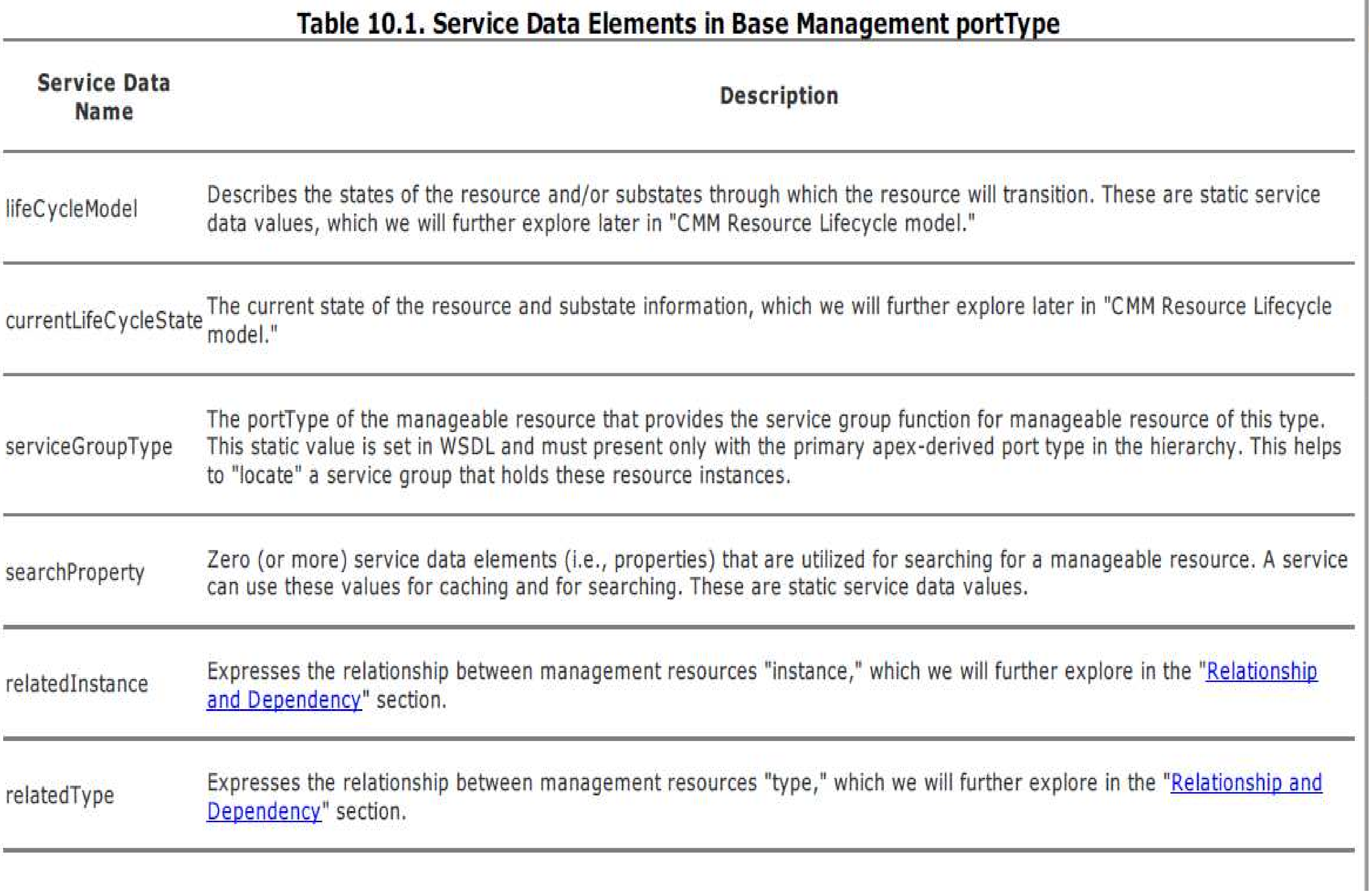
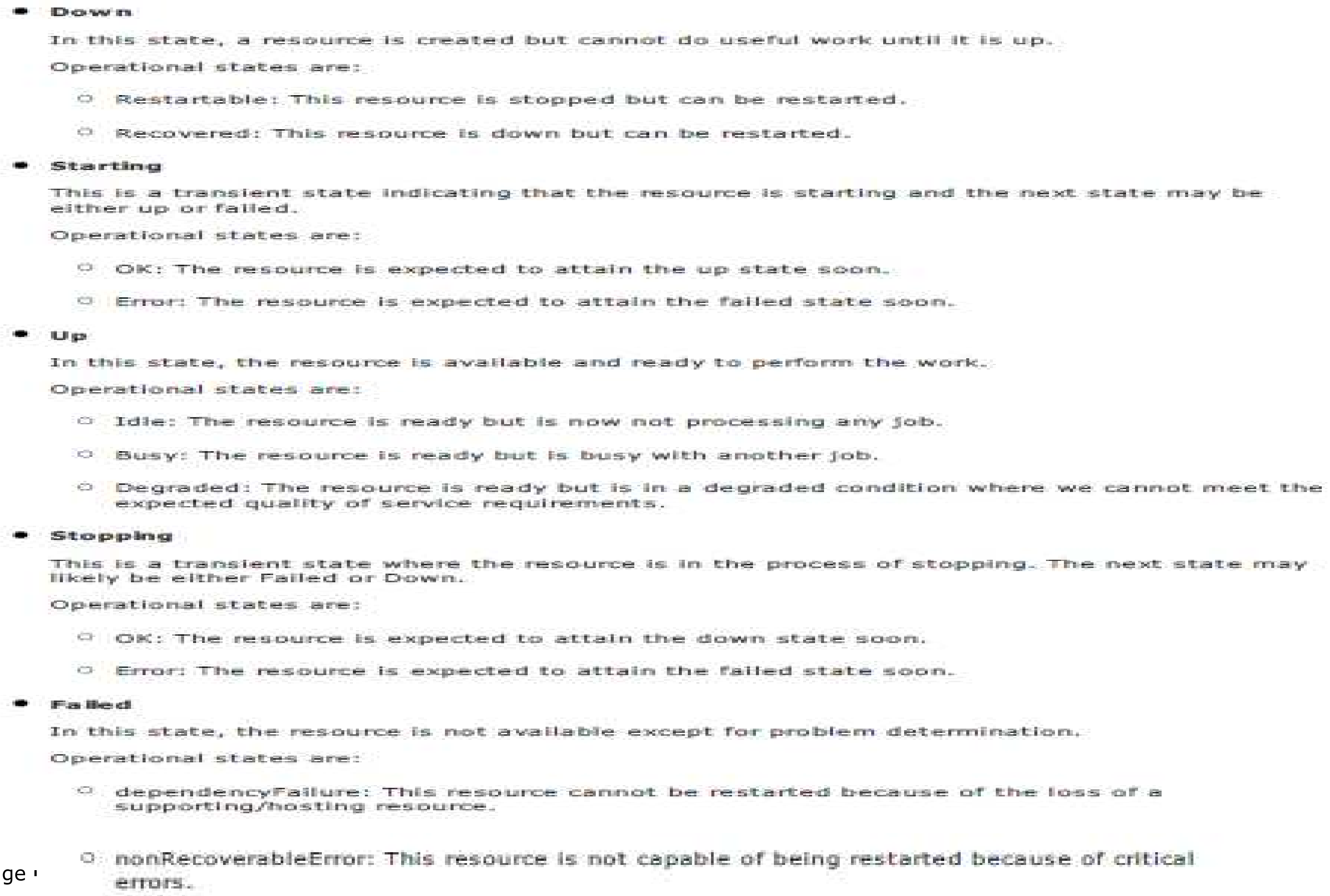


Table 10.1. Service Data Elements in Base Management portType

Service Data Name	Description
lifeCycleModel	Describes the states of the resource and/or substates through which the resource will transition. These are static service data values, which we will further explore later in "CMM Resource Lifecycle model."
currentLifeCycleState	The current state of the resource and substate information, which we will further explore later in "CMM Resource Lifecycle model."
serviceGroupType	The portType of the manageable resource that provides the service group function for manageable resource of this type. This static value is set in WSDL and must present only with the primary apex-derived port type in the hierarchy. This helps to "locate" a service group that holds these resource instances.
searchProperty	Zero (or more) service data elements (i.e., properties) that are utilized for searching for a manageable resource. A service can use these values for caching and for searching. These are static service data values.
relatedInstance	Expresses the relationship between management resources "instance," which we will further explore in the "Relationship and Dependency" section.
relatedType	Expresses the relationship between management resources "type," which we will further explore in the "Relationship and Dependency" section.

Resource Lifecycle Modeling
Lifecycle States



- **Down**

  In this state, a resource is created but cannot do useful work until it is up.

  Operational states are:

  - Restartable: This resource is stopped but can be restarted.
  - Recovered: This resource is down but can be restarted.

- **Starting**

  This is a transient state indicating that the resource is starting and the next state may be either up or failed.

  Operational states are:

  - OK: The resource is expected to attain the up state soon.
  - Error: The resource is expected to attain the failed state soon.

- **Up**

  In this state, the resource is available and ready to perform the work.

  Operational states are:

  - Idle: The resource is ready but is now not processing any job.
  - Busy: The resource is ready but is busy with another job.
  - Degraded: The resource is ready but is in a degraded condition where we cannot meet the expected quality of service requirements.

- **Stopping**

  This is a transient state where the resource is in the process of stopping. The next state may likely be either Failed or Down.

  Operational states are:

  - OK: The resource is expected to attain the down state soon.
  - Error: The resource is expected to attain the failed state soon.

- **Failed**

  In this state, the resource is not available except for problem determination.

  Operational states are:

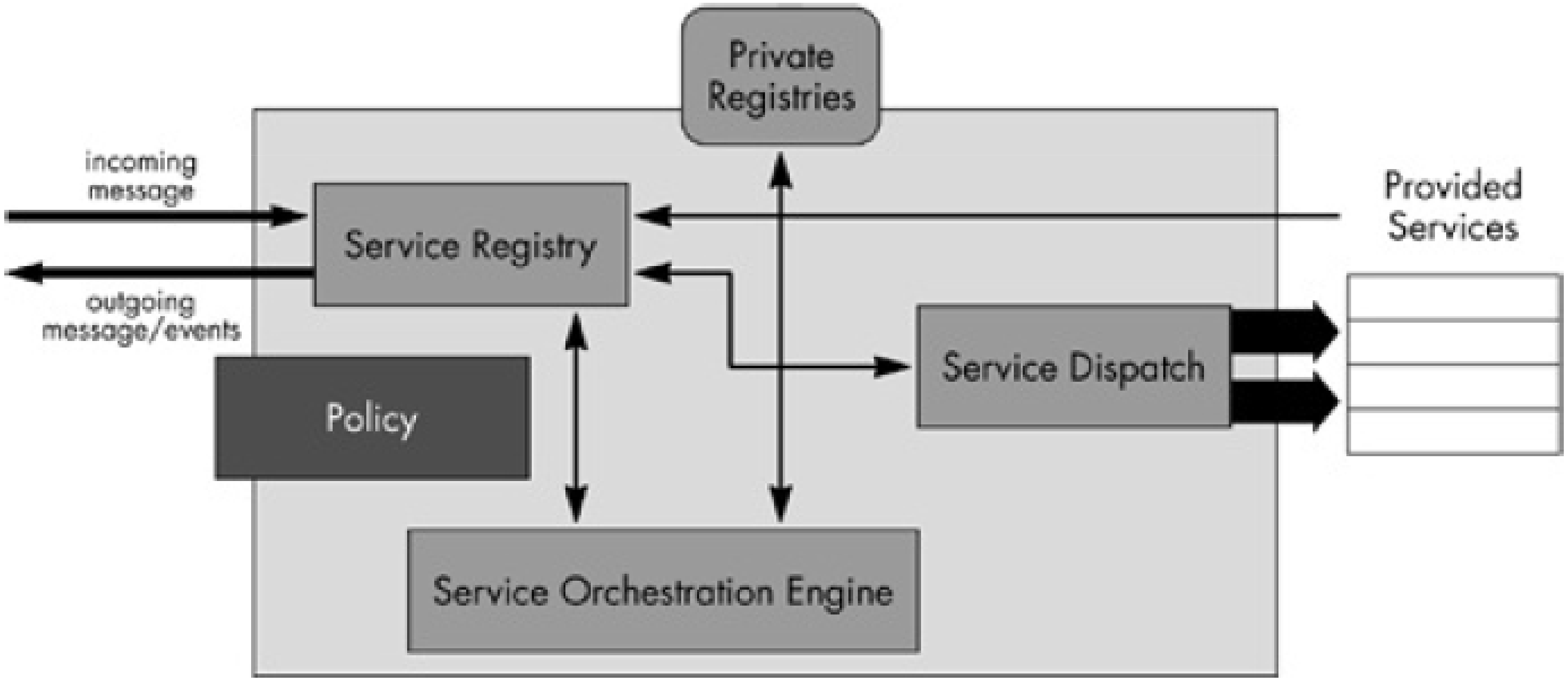
  - dependencyFailure: This resource cannot be restarted because of the loss of a supporting/hosting resource.
  - nonRecoverableError: This resource is not capable of being restarted because of critical errors.

Resource Properties

- **Resource Grouping**
  - Problem of locating fine-grained resources in the system
  - CMM solves this problem by naturally grouping resources. CMM uses ServiceGroup interface in OGSA/OGSI for managing resources
- **Relationship and Dependency among Resources**
  - Relationship describes which resources are connected to each other.
  - Dependencies add additional info to relationship on exactly how one resource depends on another.
  - Eg: Database resource uses storage device & provides details of needs such as storage space.

Service Domains

- High-level abstraction model to describe the behaviors, attributes, operations, and interfaces to allow a collection of services to function as a single unit.



Service Domains: Components

- Service Registration and Collection
- Service Routing and Selection
- Service interoperation and transformation
- Flexible service composition
- Automatic service orchestration
- Note:Uses OGSI ServiceCollection port Type.

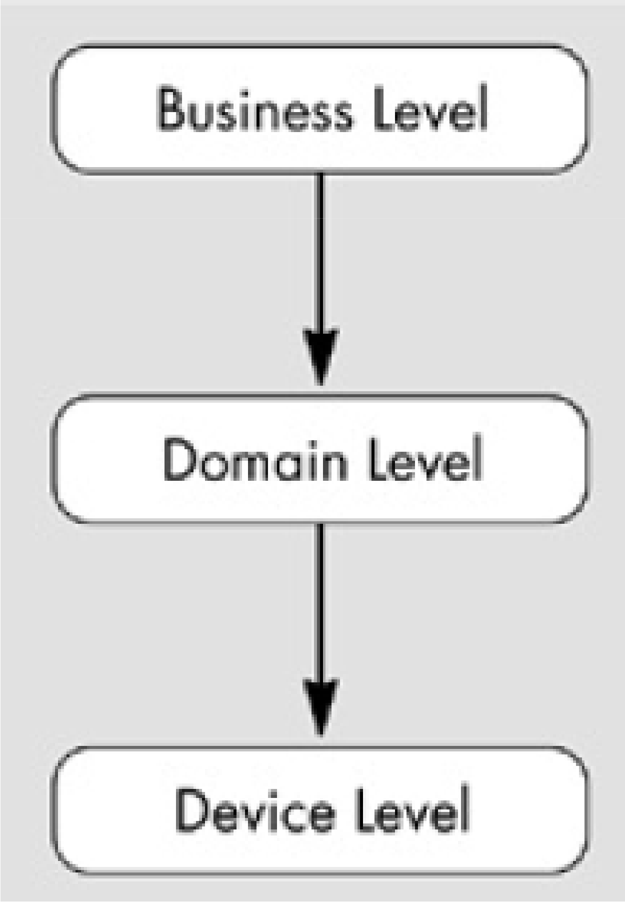
Service Domains: behaviours

- Filter
- Selection
- Topology
- Enumeration
- Discovery
- Policy

Policy Architecture

- Provides a framework for creating, managing, validating, distributing, transforming, resolving, and enforcing policies in distributed environment.

- **Levels of Policy Abstraction**

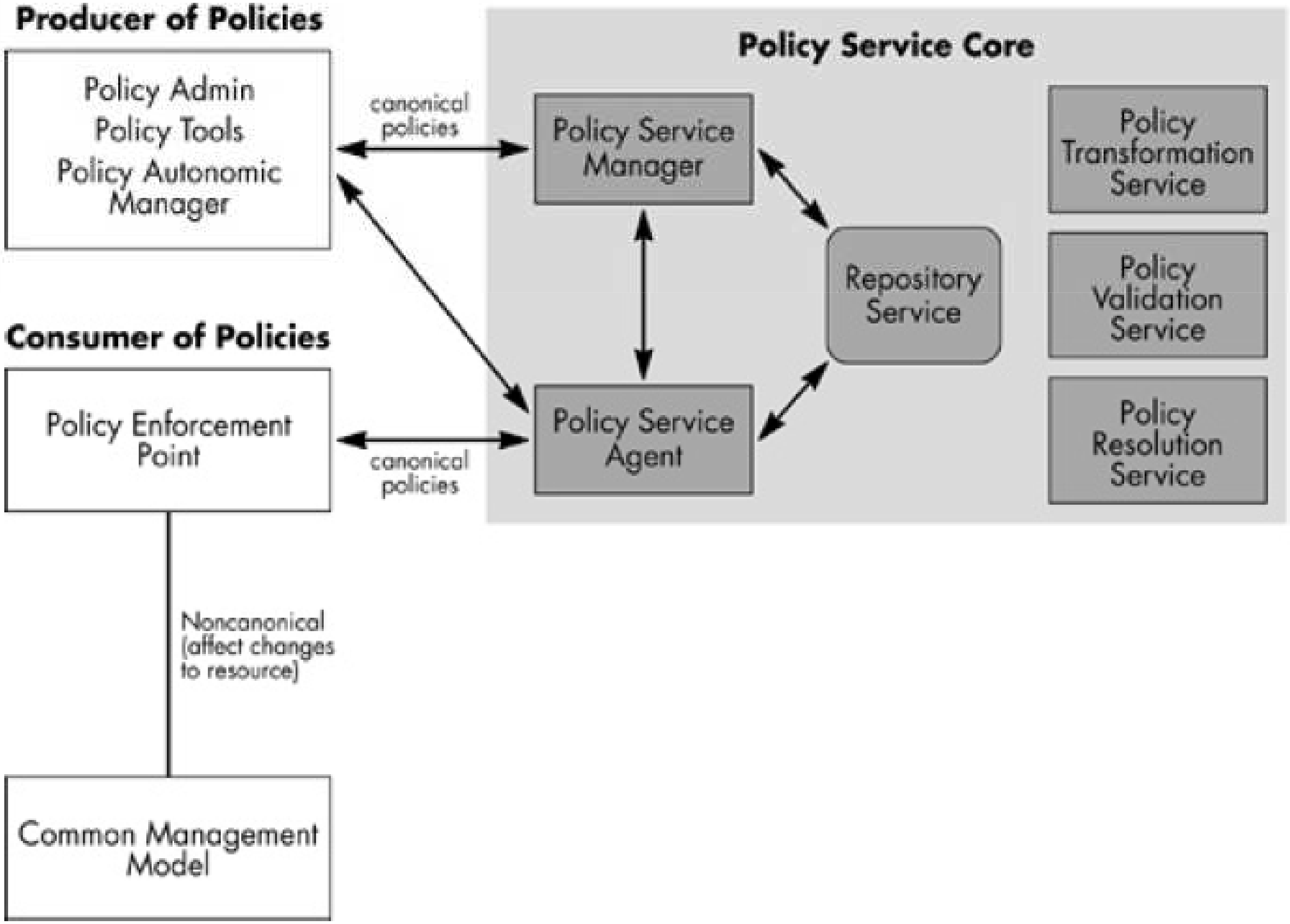


Policy Architecture

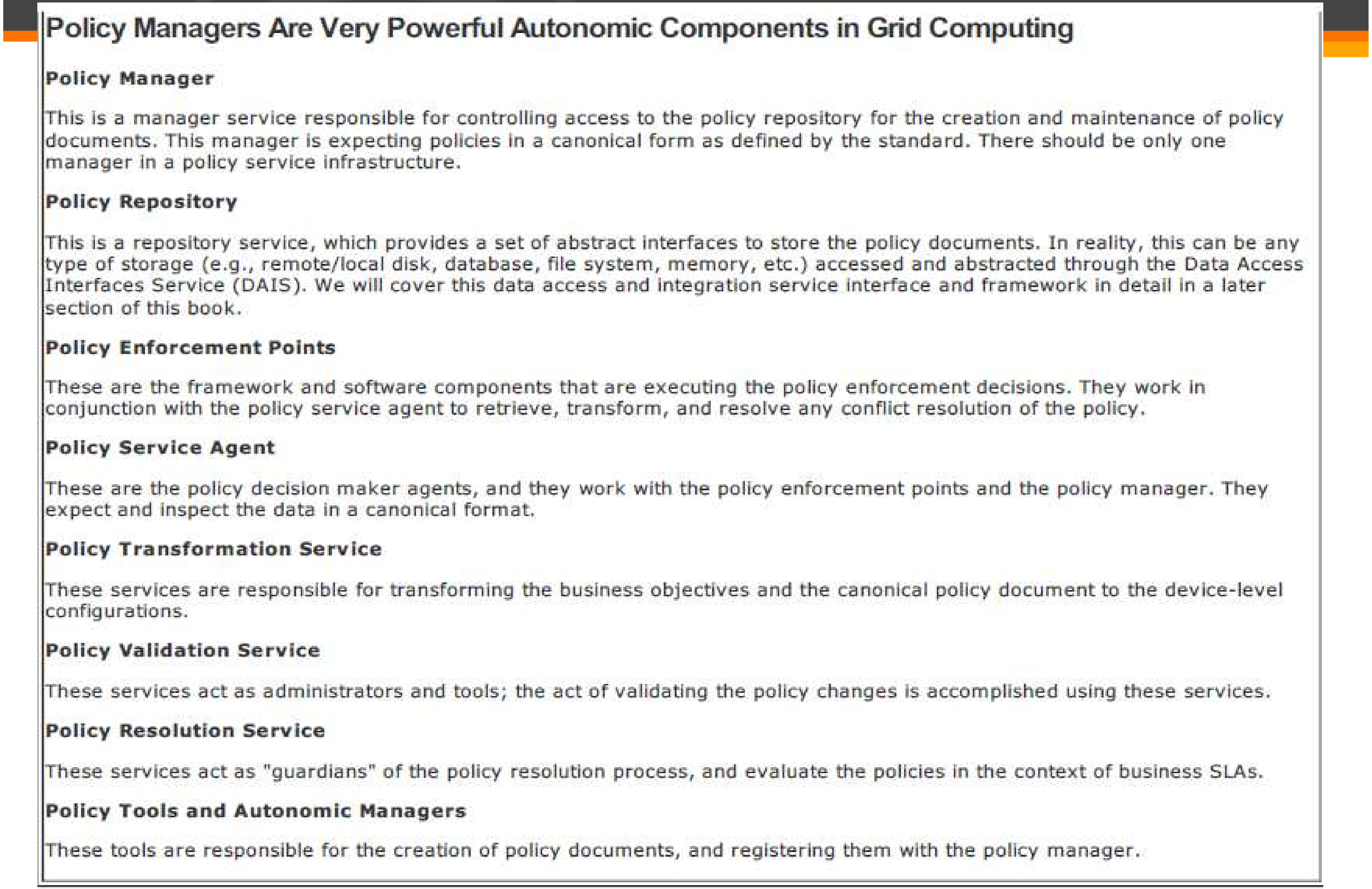
- PolicySet
- PolicyRule
- PolicyGroup
- PolicyCondition
- PolicyAction

Policy Service Framework

Figure 10.7. The defined OGSA Policy Service Core.



Policy Service Framework



Policy Managers Are Very Powerful Autonomic Components in Grid Computing

Policy Manager

This is a manager service responsible for controlling access to the policy repository for the creation and maintenance of policy documents. This manager is expecting policies in a canonical form as defined by the standard. There should be only one manager in a policy service infrastructure.

Policy Repository

This is a repository service, which provides a set of abstract interfaces to store the policy documents. In reality, this can be any type of storage (e.g., remote/local disk, database, file system, memory, etc.) accessed and abstracted through the Data Access Interfaces Service (DAIS). We will cover this data access and integration service interface and framework in detail in a later section of this book.

Policy Enforcement Points

These are the framework and software components that are executing the policy enforcement decisions. They work in conjunction with the policy service agent to retrieve, transform, and resolve any conflict resolution of the policy.

Policy Service Agent

These are the policy decision maker agents, and they work with the policy enforcement points and the policy manager. They expect and inspect the data in a canonical format.

Policy Transformation Service

These services are responsible for transforming the business objectives and the canonical policy document to the device-level configurations.

Policy Validation Service

These services act as administrators and tools; the act of validating the policy changes is accomplished using these services.

Policy Resolution Service

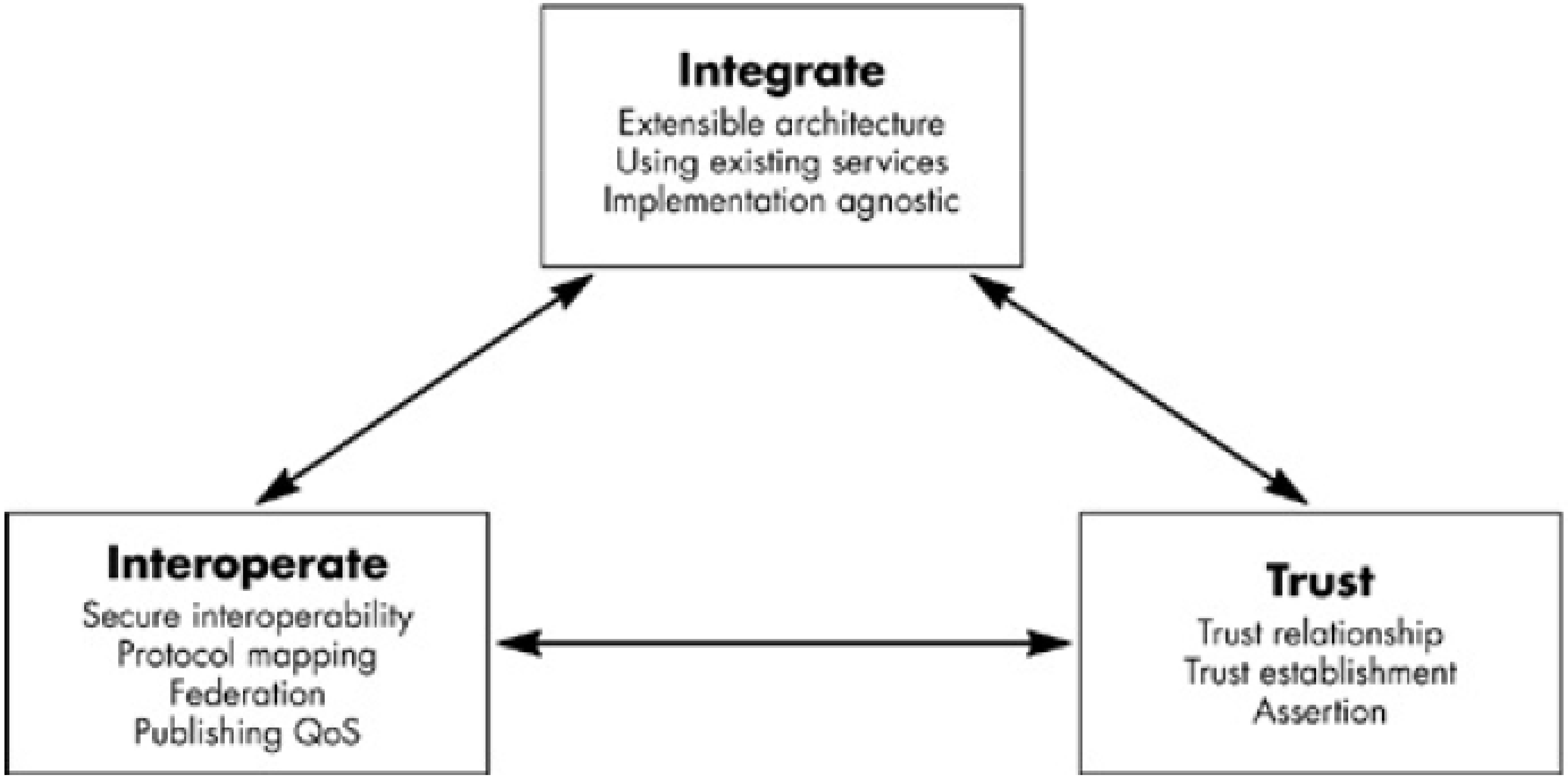
These services act as "guardians" of the policy resolution process, and evaluate the policies in the context of business SLAs.

Policy Tools and Autonomic Managers

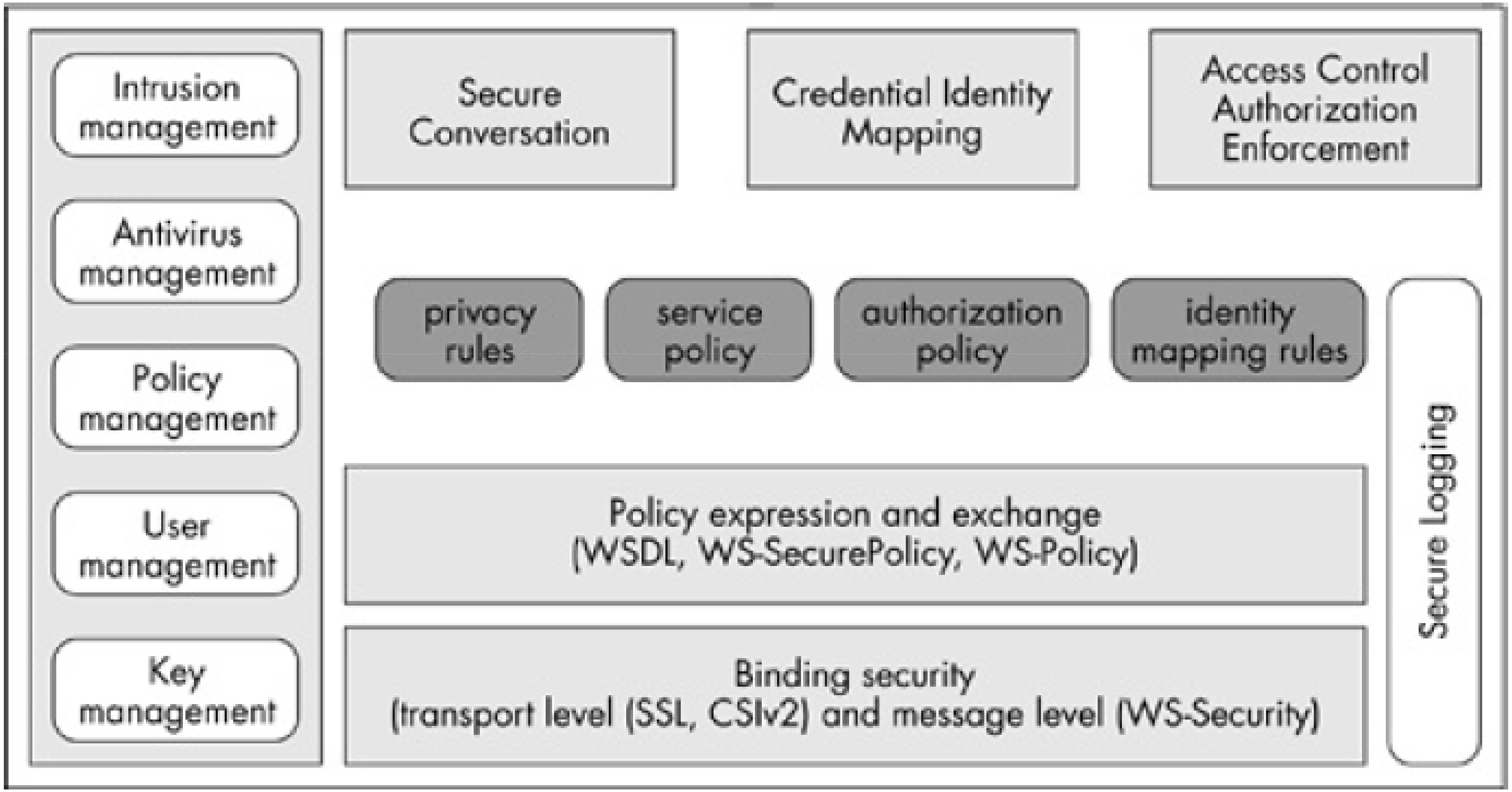
These tools are responsible for the creation of policy documents, and registering them with the policy manager.

Security Architecture

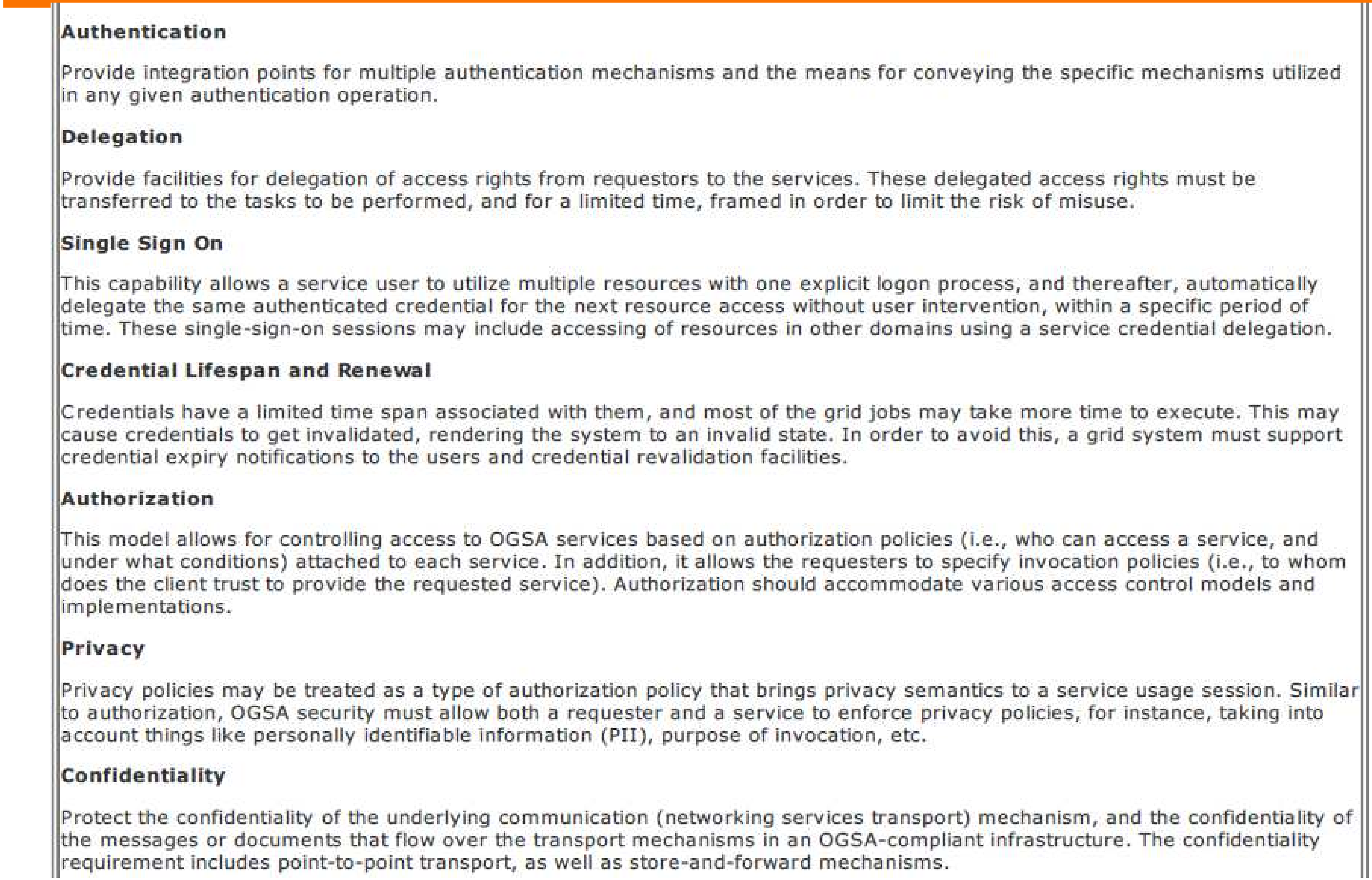
- Resource sharing among heterogeneous virtual organization participants is a complex process because of the challenges faced in **integration, interoperability and trust relationship**



OGSA Architecture- *Security Architecture*


Intrusion management
Antivirus management
Policy management
User management
Key management
Secure Conversation
Credential Identity Mapping
Access Control Authorization Enforcement
privacy rules
service policy
authorization policy
identity mapping rules
Secure Logging
Policy expression and exchange (WSDL, WS-SecurePolicy, WS-Policy)
Binding security (transport level (SSL, CSIv2) and message level (WS-Security)

Common Securities Elements Required for a Grid Environment



Authentication

Provide integration points for multiple authentication mechanisms and the means for conveying the specific mechanisms utilized in any given authentication operation.

Delegation

Provide facilities for delegation of access rights from requestors to the services. These delegated access rights must be transferred to the tasks to be performed, and for a limited time, framed in order to limit the risk of misuse.

Single Sign On

This capability allows a service user to utilize multiple resources with one explicit logon process, and thereafter, automatically delegate the same authenticated credential for the next resource access without user intervention, within a specific period of time. These single-sign-on sessions may include accessing of resources in other domains using a service credential delegation.

Credential Lifespan and Renewal

Credentials have a limited time span associated with them, and most of the grid jobs may take more time to execute. This may cause credentials to get invalidated, rendering the system to an invalid state. In order to avoid this, a grid system must support credential expiry notifications to the users and credential revalidation facilities.

Authorization

This model allows for controlling access to OGSA services based on authorization policies (i.e., who can access a service, and under what conditions) attached to each service. In addition, it allows the requesters to specify invocation policies (i.e., to whom does the client trust to provide the requested service). Authorization should accommodate various access control models and implementations.

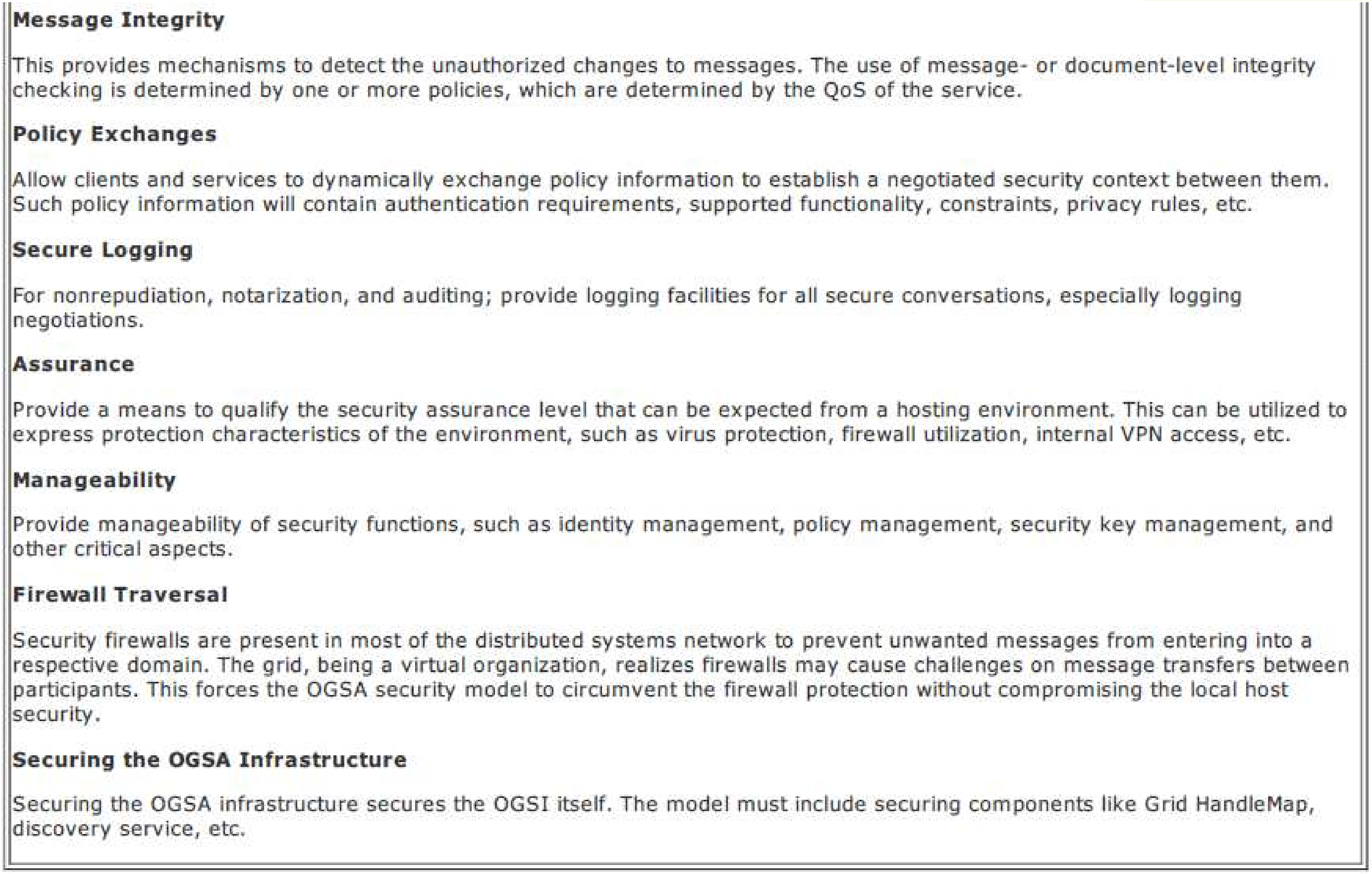
Privacy

Privacy policies may be treated as a type of authorization policy that brings privacy semantics to a service usage session. Similar to authorization, OGSA security must allow both a requester and a service to enforce privacy policies, for instance, taking into account things like personally identifiable information (PII), purpose of invocation, etc.

Confidentiality

Protect the confidentiality of the underlying communication (networking services transport) mechanism, and the confidentiality of the messages or documents that flow over the transport mechanisms in an OGSA-compliant infrastructure. The confidentiality requirement includes point-to-point transport, as well as store-and-forward mechanisms.

Common Securities Elements Required for a Grid Environment



Message Integrity

This provides mechanisms to detect the unauthorized changes to messages. The use of message- or document-level integrity checking is determined by one or more policies, which are determined by the QoS of the service.

Policy Exchanges

Allow clients and services to dynamically exchange policy information to establish a negotiated security context between them. Such policy information will contain authentication requirements, supported functionality, constraints, privacy rules, etc.

Secure Logging

For nonrepudiation, notarization, and auditing; provide logging facilities for all secure conversations, especially logging negotiations.

Assurance

Provide a means to qualify the security assurance level that can be expected from a hosting environment. This can be utilized to express protection characteristics of the environment, such as virus protection, firewall utilization, internal VPN access, etc.

Manageability

Provide manageability of security functions, such as identity management, policy management, security key management, and other critical aspects.

Firewall Traversal

Security firewalls are present in most of the distributed systems network to prevent unwanted messages from entering into a respective domain. The grid, being a virtual organization, realizes firewalls may cause challenges on message transfers between participants. This forces the OGSA security model to circumvent the firewall protection without compromising the local host security.

Securing the OGSA Infrastructure

Securing the OGSA infrastructure secures the OGSI itself. The model must include securing components like Grid HandleMap, discovery service, etc.

Metering and Accounting

- There is a general requirement that resource utilization should be monitored for cost allocation, capacity analysis, dynamic provisioning, grid-service pricing, fraud and intrusion detection, and/or billing.
- **Metering Service Interface**
  - Metering subsystems are used for measuring the resource consumption and aggregating their own respective utilization measurements
- **Accounting Service Interface**
  - Accounting services can make use of the rated financial information retrieved through rating services in order to calculate user subscription costs over a specific period of time, per use (i.e., On Demand), or on a monthly basis
- **Billing/Payment Service Interface**
  - These services work in collaboration with the accounting service to collect the payments

Common Distributed Logging

- **Common Distributed Logging** capability can be viewed as typical messaging applications where message producers generate log messages, which may or may not be consumed by the interested message consumers over a period of time
- Messages can be
  - Informational
  - Trace
  - Error
  - debug
- Separates implementation from service
- Decoupling helps to provide a clear separation of the roles of the log producers and log consumers.

Common Distributed Logging

- **Transformation and Common Representation:** This facility provides plug-in transformation scripts to convert from one log format to another. Most notable among these transformation scripts is XSLT, which acts as an XML data transformation script.
- **Filtering and aggregation:** Most of the logging may result in a huge amount of data and filtering of these data into certain buckets of desirable segments; this is a value-added feature. The OGSA logging service provides registration of such filtering criteria for each consumer, and aggregates the messages based on these criteria
- **Configurable persistency:** The durability of the logs is a major feature provided by the OGSA framework. W e can enable this feature based on a per service and/or a per message basis (i.e., On Demand).
- **Consumption patterns:** Logging services should provide both synchronous (pull) and asynchronous (push) models of interaction of messages by the consumers.
- **Secure logging:** As we already know, many of these logs are critical, sensitive, and private; therefore, the need to store them and transport them in a secure fashion is an absolute

Distributed Data Access an Replication

- The **complexity** of **data** access and management on a grid arises from the **scale**, **dynamism**, **autonomy** and the **geographical distribution** of the data sources.
- These **complexities** should be made **transparent** to grid **applications** through a layer of **grid data virtualization services**.
- Data Grid should provide the following services.
  - Data access service.
  - Data replication.
  - Data caching service
  - Metadata catalog and services
  - Schema transformation services
  - Storage services

Summary of OGSA Services

- Common Management Model (CMM)
- Service domains
- Policy
- Security
- Provisioning and resource management
- Accounting/metering
- Common distributed logging
- Monitoring
- Scheduling
- Distributed data access and replication