IBM SOA Technology Summit

Moving Ahead With SOA

Managing Web Services : Life-cycle management and SLAs

Eric DATEI
Senior IT Architect – IBM Certified

SOA on your terms and our expertise

ON DEMAND BUSINESS™

Agenda

- What is SOA Management

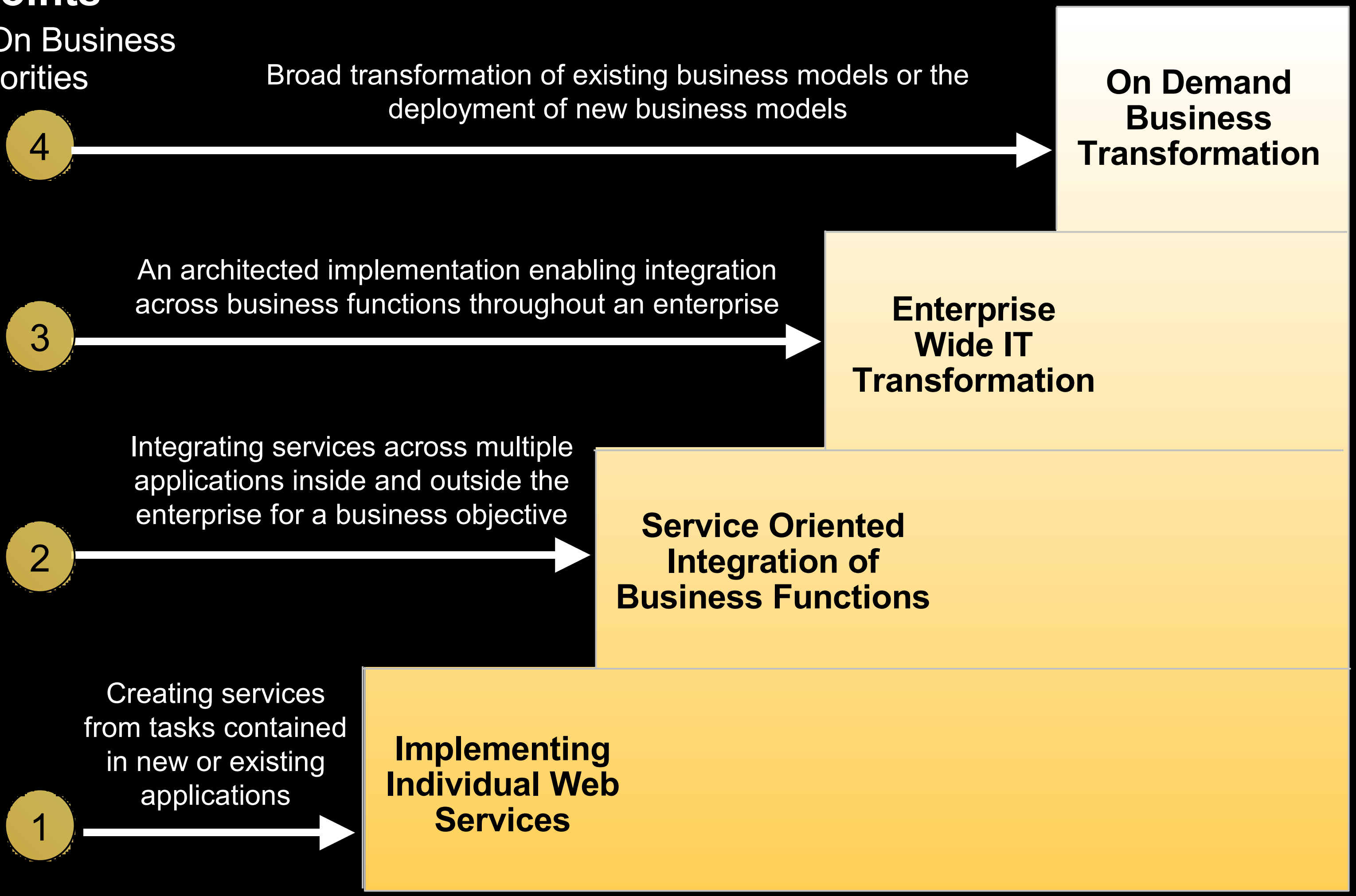
- Services provisioning & discovery
  - Services lifecycle management
  - Versioning
  - Service Discovery : UDDI

- SLA Management and enforcement
  - Quality of service management
  - Service monitoring
  - SLA Enforcement

Organizations can take different paths to eventual adoption of SOA depending on your business goals and IT constraints

Entry Points

Based On Business Priorities



4 Broad transformation of existing business models or the deployment of new business models → **On Demand Business Transformation**

3 An architected implementation enabling integration across business functions throughout an enterprise → **Enterprise Wide IT Transformation**

2 Integrating services across multiple applications inside and outside the enterprise for a business objective → **Service Oriented Integration of Business Functions**

1 Creating services from tasks contained in new or existing applications → **Implementing Individual Web Services**

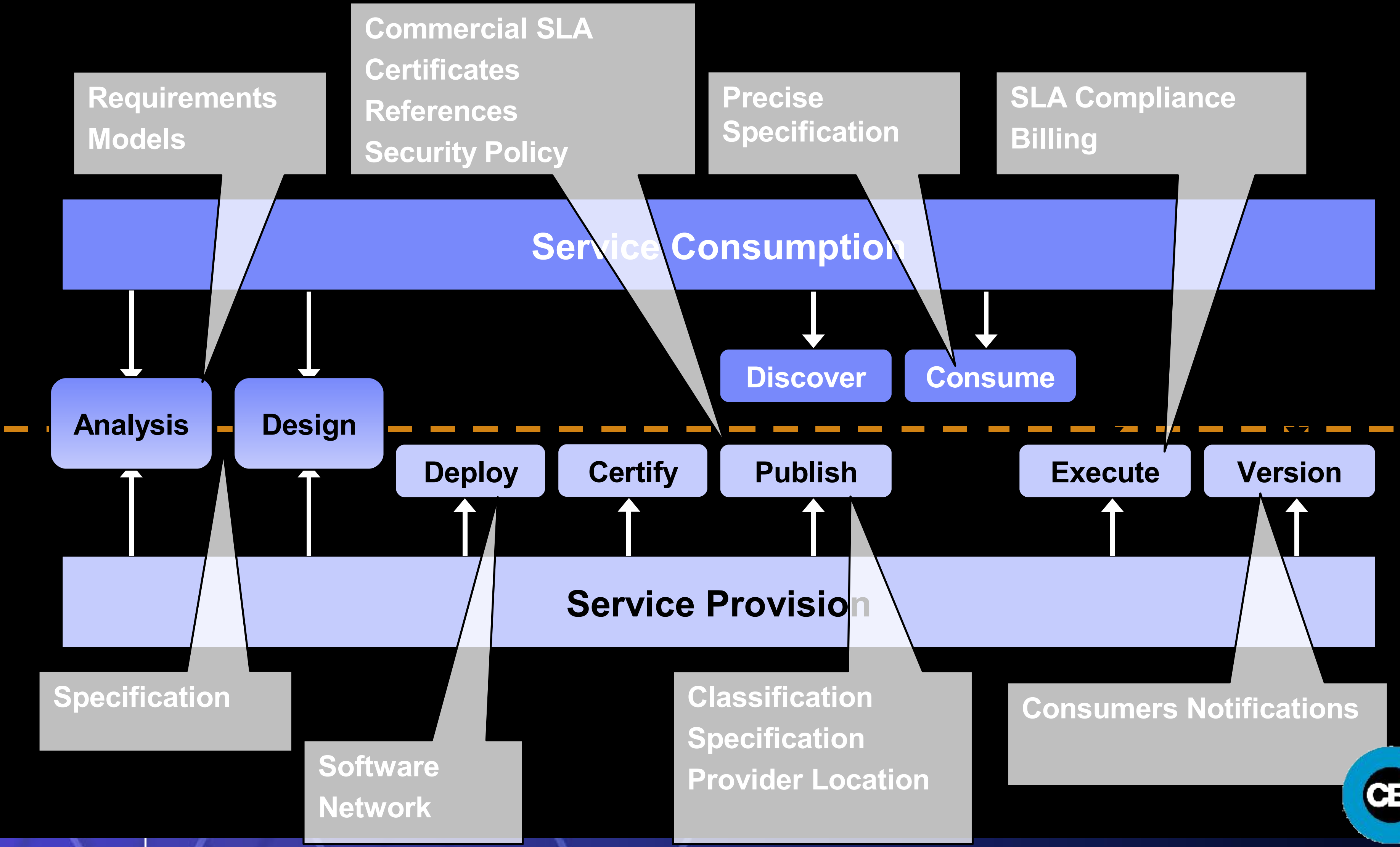
Do Web Services Solve Everything ?

- Web Services are just a deployment technology

- How are application service-enabled ?
- How are services exposed ?
- How do Service Providers and Consumers agree with what they are supposed to do and exchange specifications ?
- How are Web Services managed ?
- How is security enforced ?
- How are new versions delivered ?
- How are performance and high-availability objectives met ?
- …

Need to think about Service across the whole lifecycle – It is not just Web Services at deployment.

Service Consumption & Provision: Information Exchange



Agenda

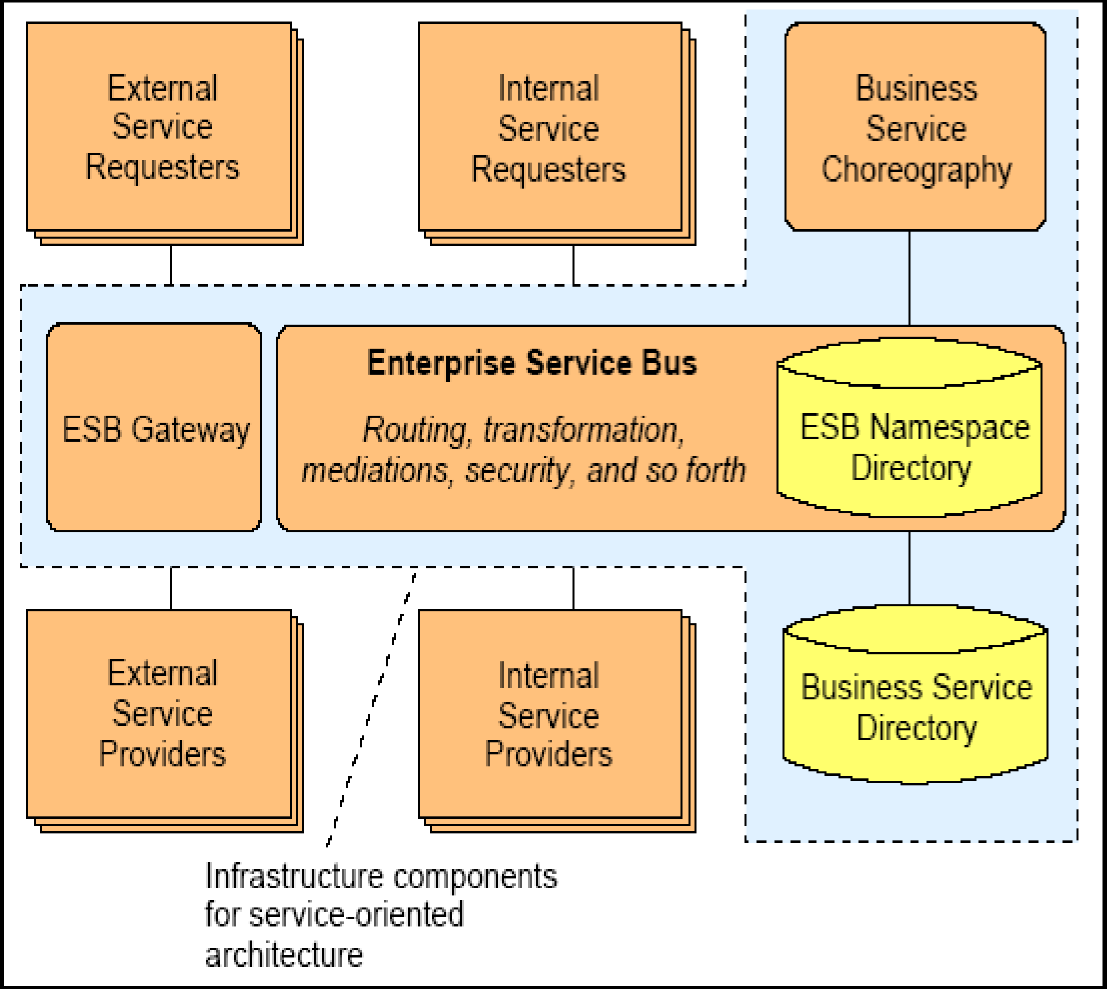
- What is SOA Management
- Services provisioning & discovery
  - Services exposition and provisioning
  - Versioning
  - Service Discovery : UDDI
- SLA Management and enforcement
  - Quality of service management
  - Service monitoring
  - SLA Enforcement

Services Exposition
SOA Infrastructure & Architectural Patterns

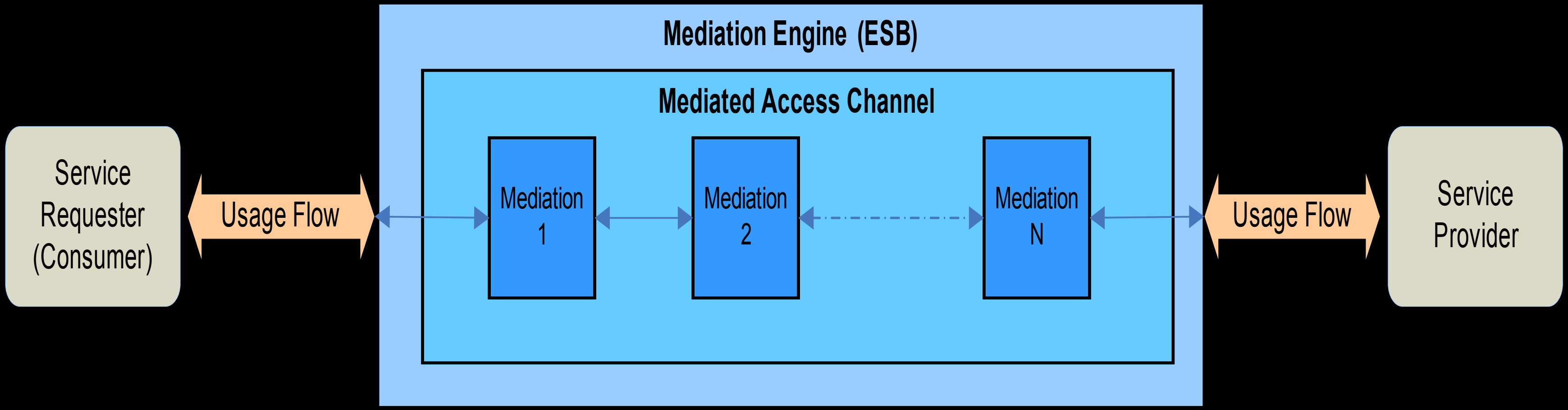
- **Issues**
  - Manage multiple services and endpoints
  - Expose services to both internal and external requesters (Internal applications, External applications and 3rd Parties) with support for different invocation protocols
- **Objectives**
  - Provide a homogeneous and coherent architecture to expose and manage services
  - Provide a centralized administration infrastructure to control and manage the exposition of all services
  - Ability to seamlessly plug additional processing as intermediaries without disrupting exposed services
  - Comply with non functional requirements: extensibility, manageability, scalability, performance, security…
- **Solution**
  - Design and implement an **Enterprise Service Bus** according to the ESB Pattern and its Exposed ESB variant

SOA Infrastructure: ESB Pattern
(Routing, Transformation, Mediation, Security and so forth…)

- **Overall Drivers**
  - Support large numbers of service interactions in a manageable way. Remove spaghetti / point-to-point interactions.
  - Provide support for advanced service interaction capability, e.g. transactions, store and forward, infrastructure services, security, quality of service etc.
  - Support a variety of interaction styles such as synchronous request / response, messaging, publish/subscribe and events.
  - Provide a robust, manageable, distributed integration infrastructure consistent with the principles of SOA.
  - Support service routing and substitution, protocol transformations and other message processing.
  - Support both Web Services and traditional EAI communication standards and technologies.

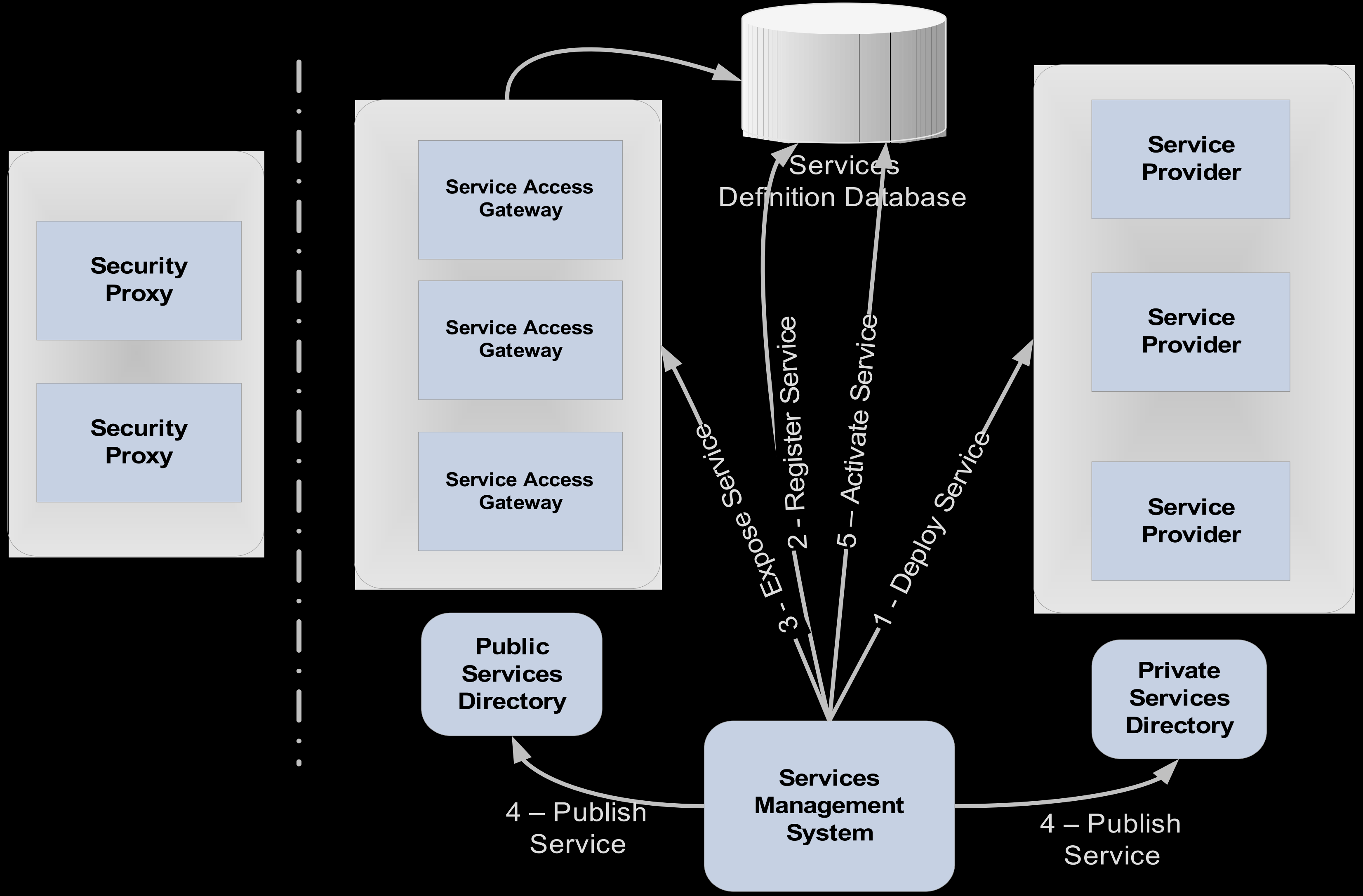

External Service Requesters
Internal Service Requesters
Business Service Choreography
ESB Gateway
Enterprise Service Bus
Routing, transformation, mediations, security, and so forth
ESB Namespace Directory
External Service Providers
Internal Service Providers
Business Service Directory
Infrastructure components for service-oriented architecture

Leveraging the ESB Mediation Capacities



- **Support for Mediations (a.k.a. Intermediaries)**
  - Ability to plug additional processing within the service interaction model without disrupting Service Requesters and Service Providers
- **Mediations can be leveraged to support various functional and non functional requirements in a non-intrusive (or less-intrusive) manner**
  - Security
  - Logging
  - Routing
  - …

Service provisioning process



Service provisioning process
Project feedbacks and best practices

- Provide only SOAP or local Java access support for service invocation
  - SOA infrastructure and Services provisioning process will be much more simple
  - It is no use today to support RMI / IIOP access for performance consideration

- Provide asynchronous mechanisms support at the ESB level. Support of standard patterns :
  - Send and forget
  - Store and forward
  - Publish and subscribe

- Service Access Gateway implementation
  - Provide a generic end point access instead of expose each service interface if the gateway doesn't support real provisioning API and clustering mode

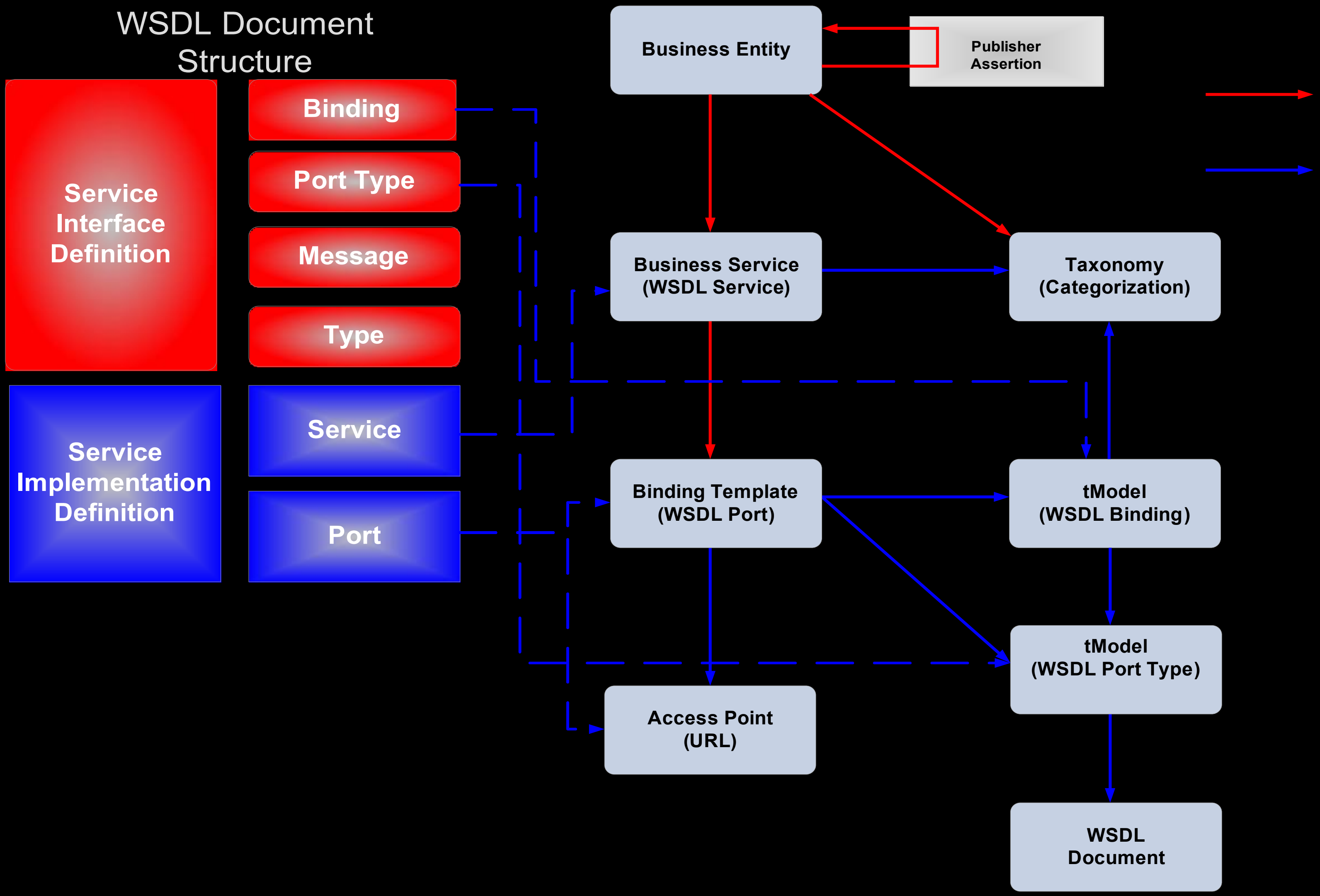
Service Discovery

- Issues
  - How is Service-related information governed (stored, managed and maintained, accessed) ?
  - How do Service Requesters determine which Services to use ?
  - How do Service Requesters locate Service endpoints ?
- Objectives
  - Manage service-related information (interface, service location, additional information such as specification…) in a centralized manner
  - Provide categorization and versioning capabilities to leverage service-related information
  - Provide service requesters with extensive discovery and notification capabilities
  - Provide administration capabilities
- Solution
  - Design and implement a Service Directory (Enabler Directory)

Service Directory (UDDI)

- Contains services description and services-related information model
  - Provider Business Entity (including Contacts and Assertions)
  - Services
  - References to WSDL documents (stored in the Service Definition Database)
- Contains additional categorization information
  - Service Name
  - Version
  - Service Status (active / inactive)
  - Service Lifecycle Phase (prototype / production / retirement)

UDDI Data Model and WSDL document mapping



Service Directory (UDDI): Feedbacks from projects

- Very flexible model: almost too flexible !
  - Not much feedback
  - Hard to make guesses about best and bad practices
- Policy and ACL model still too formal and not very practical
  - Difficult to manage information-level ACL in an efficient way
- Still too technical
  - UDDI model is complex: consumers must somehow be provided with a simpler perspective
  - Subscription mechanism is too verbose and full of technical details, providing no immediate value
- Systinet
  - Promotion model (Staging – Production) not very flexible: could not be leveraged to support Orange requirements
  - Systinet GUI is too close to the underlying model: not very intuitive and business-oriented
- In a few words, the business value provided by a Service Directory is not crystal clear and does not balance the technical intricacies experienced in using UDDI

Service Versioning

- Structuring Assumptions
  - Versioning is performed at the Service level (including its Services & Operations)
  - There is no backward compatibility between successive versions of a Service
  - Switching between successive versions on the requestor side requires migration work.
  - However, Clients must not be required to migrate to new versions of Services but must be able to keep on using previous ones provided they still exist.
- Principles
  - In order not to impact (or to minimize such an impact) service requestors, all existing versions of a Service (including its Services & Operations) must still be supported when deploying a new version.
  - The support of multiple version of a Service is performed on a best-effort basis. This imply that a given version of a Service might only be supported for a given period of time.
  - Some deprecated Operations can be removed from a previous version of Service without implying the removal of the entire version
  - All versions of a given Service rely on the same Target System (backend components)

Agenda

- What is SOA Management

- Services provisioning & discovery
  - Services lifecycle management
  - Versioning
  - Service Discovery : UDDI

- SLA Management and enforcement
  - Quality of service management
  - Service monitoring
  - SLA Enforcement

SLA Management

- There is two very different issues in SLA management which can (has to) be managed with different mechanisms and solutions
  - Technical issues (Service Oriented)
  - Business issues (Customer Oriented)

- The market offering is moving very quickly. There is two kinds of solutions
  - Software solutions : Amberpoint, Actional, Digital Evolution …
  - Appliance (hardware and Software solutions) : DataPower, Network Appliance, Layer 7 technologies …

- The integration of the target products into the global architecture must be early and deeply studied
  - Provisioning aspects
  - Integration with subscription and billing systems
  - Failover and scalability aspects
  - Security …

SLA management and Enforcement
Business objectives

- Business Issues are Customer oriented
  - Define and manage SLA related to a client business contract
- SLA monitoring and reporting
  - Monitor the quality of service for Services accesses for each client
  - Usage statistics for each client
  - Access reports for billing
- Enforce SLA related to a client business contract
  - Enforce authorizations (Block requests on non authorized operations)
  - Enforce agreements defines on throughput to limit the number of access during a given period (10 requests / hour, 200 requests / day, 2000 requests / month …)

SLA management and Enforcement
Technical objectives

- Technical objectives for SLA management
  - Manage SLA and service monitoring with a centralized system
  - Automatic Provisioning of the SLA management system with clients and services defined in the Service Directory and in the User Directory
  - Provide a system compliant with the performance and traffic objectives
- SLA monitoring and reporting
  - Real time collect of information related to services accesses
    - Availability, Errors, Response time, Number of calls …
  - Centralization of alerts
  - Platform general performances monitoring
  - Usage statistics for each service
- SLA Enforcement
  - Global traffic regulation to avoid saturation of the target services and Deny of Services

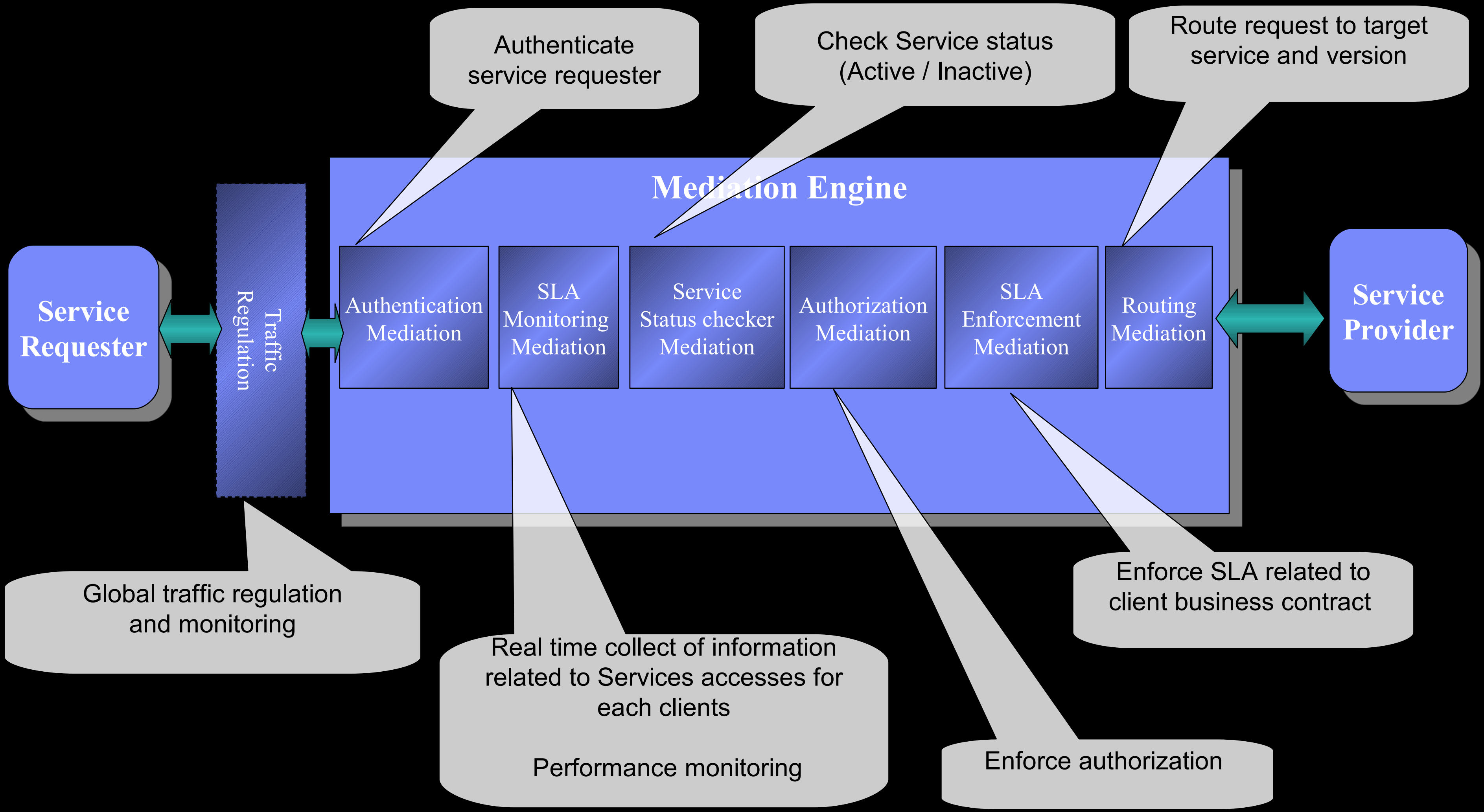
Service Level Agreement & Objectives
What is measured ?

- Service Level Agreements (SLA) define services access quality of service for users. They are based on four instruments:
  - ***Response time***: the Response Time instrument measures the average amount of time a service requires to respond to a request message
  - ***Throughput***: the Throughput instrument counts the number of request that have been successfully received, processed, and responded to. A request that generates a fault is not counted by the Throughput instrument. Throughput is measured individually for each service operation
  - ***Fault***: the Fault instrument counts the number of messages received that generated faults during processing an operation
  - ***Availability***: the Availability instrument measures the percentage of time that a service operation is available.
- Service level objectives (SLO) can be defined with these instruments for each target service or service operation
  - Response time < 2s
  - Availability > 99,9 %
  - Throughput < 100 requests / hour
- A SLA, composed of a list of SLO, can be associate to a user to define the required quality of service

SLA Definition

- There is two kinds of Agreements
  - Quality of service Agreements based on response time, fault or availability instruments …
    - Response time < 2s
    - Availability > 99,9 %
  - Limiting Agreements based on throughput instrument
    - Throughput for a given service < 1000 requests / hour
    - Throughput for a given client < 500 requests / day
- Quality of service Agreements are monitored for statistics and billing
- Limiting Agreements can be monitored or enforced
  - Limiting agreement defines quotas on throughput for a given service for all accesses or for each client
  - When the limit is reached :
    - The request can be blocked to enforce SLA related to a client business pre paid contract or for a global traffic regulation
    - Or and alert can be sent

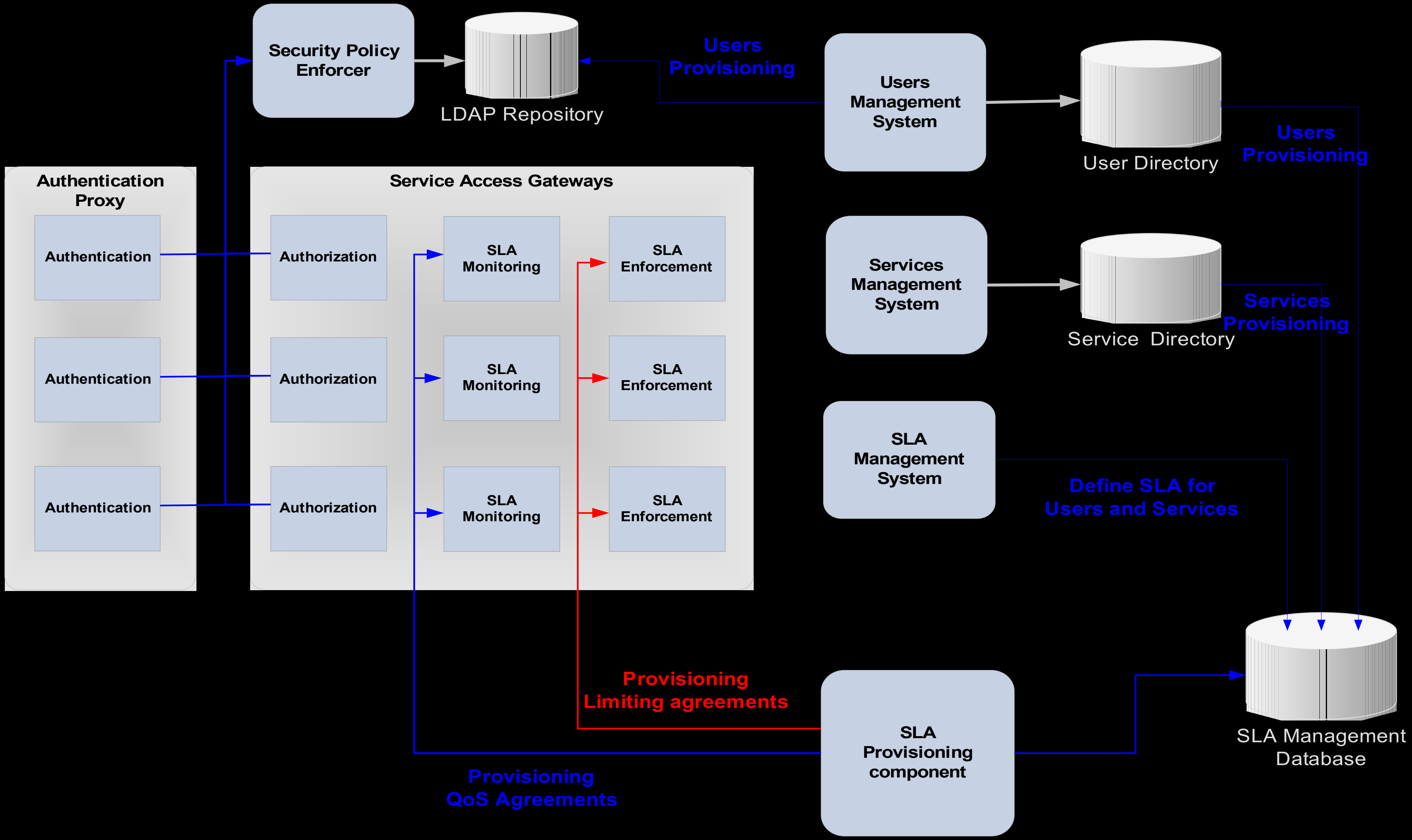
SLA Management
General architecture overview


Authenticate service requester
Check Service status (Active / Inactive)
Route request to target service and version
Mediation Engine
Service Requester
Traffic Regulation
Authentication Mediation
SLA Monitoring Mediation
Service Status checker Mediation
Authorization Mediation
SLA Enforcement Mediation
Routing Mediation
Service Provider
Global traffic regulation and monitoring
Real time collect of information related to Services accesses for each clients
Performance monitoring
Enforce authorization
Enforce SLA related to client business contract

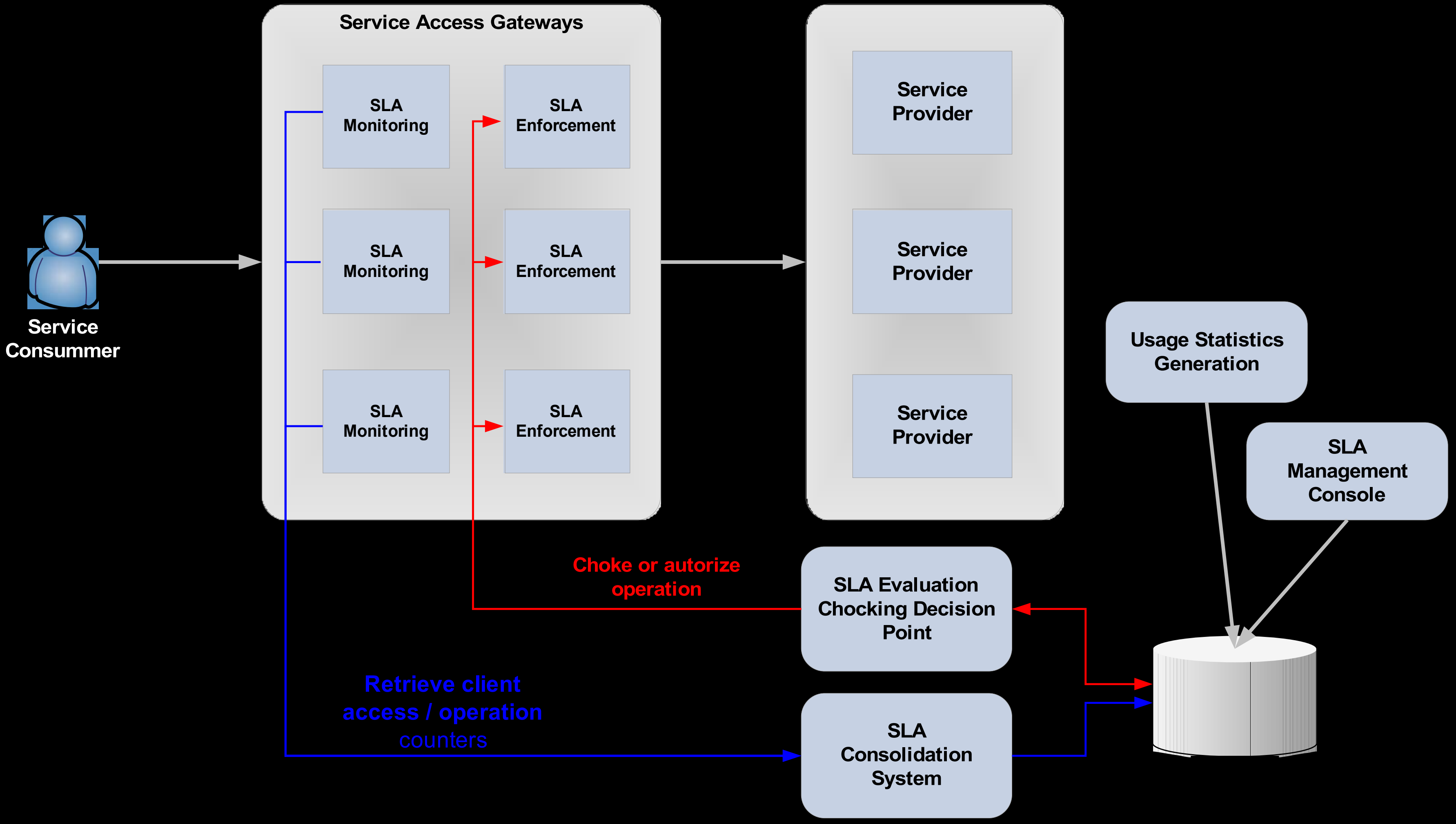
SLA Management
Services subscription issues and objectives

- Services subscription deeply impact SLA management process
  - Standard profiles can be defined for service accesses. These profiles contain
    - API which can be requested
    - Service Level Agreements which will be used including Quality of Service and limiting agreements
  - Standard profiles are then associated to clients
- Objectives
  - Automatic provisioning of the SLA management system when a profile is associated with a client
  - Provide self care service subscription capabilities
    - Allow a Third Party administrator to register Service requester
    - Allow a third party to subscribe to predefined API package

SLA definition and provisioning
Architecture overview



SLA monitoring and enforcement
Architecture overview



Feedbacks from a large scale SOA project

- Non functional requirements
  - 10 000 Third parties
  - 100 requests / second
  - About 20 services and 100 operations for Stage 1 : (10 000 * 100 SLA definition)
  - Multiple version for each services
  - Multiple service consumption policies

- SLA provisioning issues
  - SLA management system has to provide API to manage and provision SLA
  - Lots of data (clients definition, SLA definition, SLA / Clients association …) can be updated during SLA provisioning process. Transaction mechanisms have to be supported by the provisioning API
  - Incremental update must be supported

- Performance and scalability issues
  - To manage efficiently lots of clients, SLA Management System must be clients oriented and not services oriented.
  - Mediations based on XSLT filters can strongly impact performances
  - Real time SLA monitoring and enforcement for each client have a strong impact on scalability

- Availability issues
  - SLA management system has to provide automatic failover mechanisms