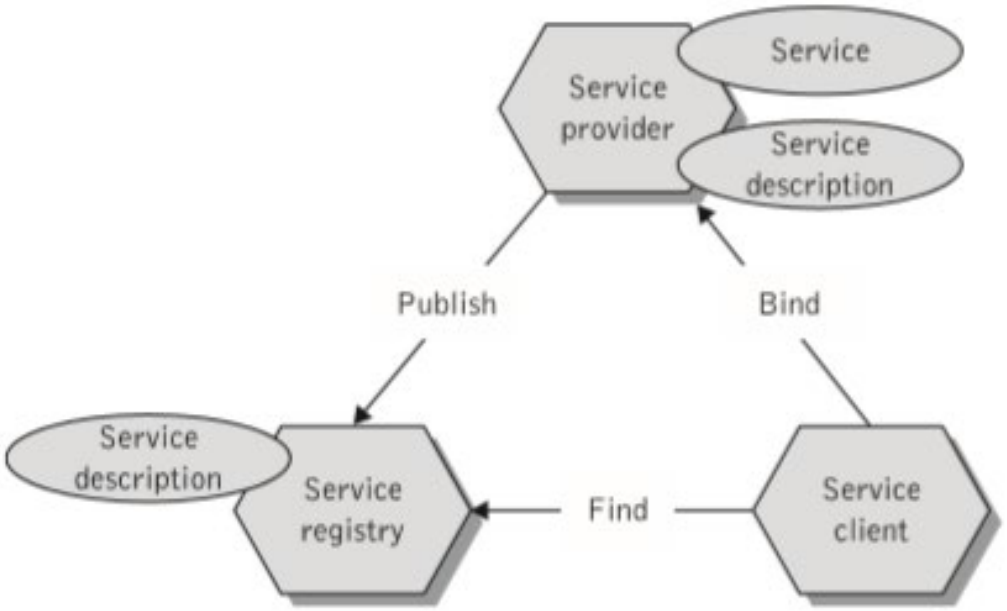
Service-oriented architecture (SOA)

Registering and discovering web service

SOA

- Service-oriented architecture (SOA) is a logical view of designing a software system to provide services to either end user applications or to other services distributed in a network, via published and discoverable interfaces.

Operations in the SOA



Web service roles and operations

Web services provider

- From a business perspective the Web services provider is the organization that owns the Web service and implements the business logic that underlies the service.
- From an architectural perspective this is the platform that hosts and controls access to the service.
- Web services provider is responsible for publishing the Web services it provides in a service registry hosted by a service discovery agency. This involves describing the business, service, and technical information of the Web service, and registering that information with the Web services registry in the format prescribed by the discovery agency.

Web services requestor

- From a business perspective this is the enterprise that requires certain functions to be satisfied.
- From an architectural perspective, this is the application that is looking for, and subsequently invoking, the service.
- Web services requestor searches the service registry for the desired Web services.
- This means discovering the Web services description in a registry provided

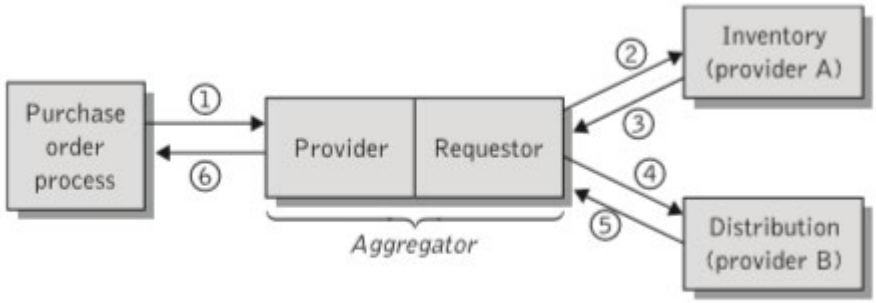
Web services registry

- A searchable directory where service descriptions can be published and searched.
- Service requestors find service descriptions in the registry and obtain binding information for services.
- Web services discovery agency is responsible for providing the infrastructure required to enable the three operations in the Web services architecture
- publishing the Web services by the Web services provider, searching for Web services by Web services requestors, and invoking the Web services.

Operations

- Publish
- Find
- Bind

Ex:-



SOA: composite service example

Service-oriented architectures key characteristics

- SOA services have **self-describing interfaces** in platform-independent XML documents. WSDL is the standard used to describe the services.
- SOA services **communicate with messages formally defined via XML**. Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider. Messages between services can be viewed as key business documents processed in an enterprise.
- SOA services are maintained in the enterprise by a **registry** that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration (UDDI) is the standard used for service registry.
- Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services.

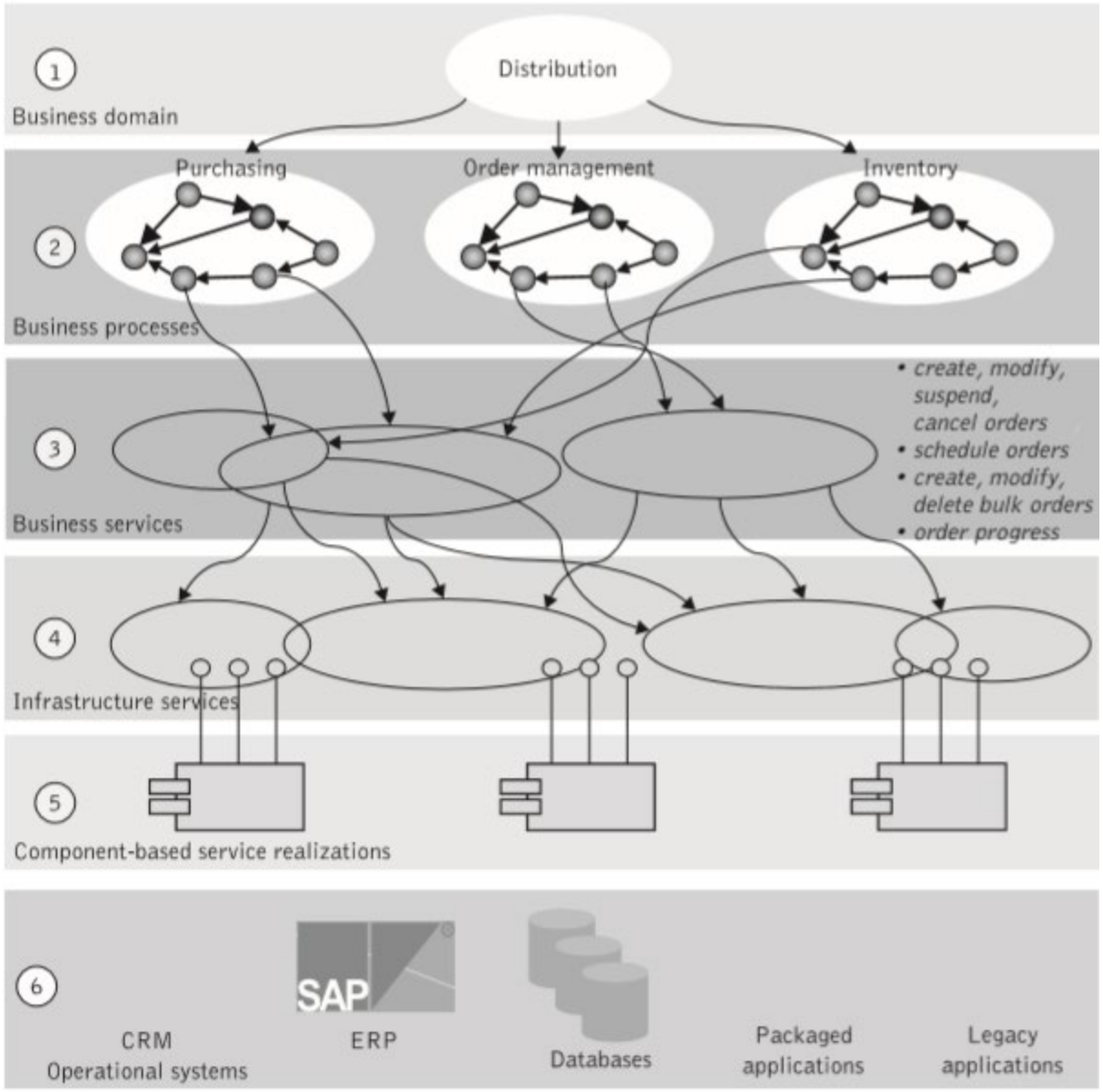
Benefits of SOA

- SOA differs from existing distributed technologies in that most vendors accept it and have an application or platform suite that enables SOA.
- SOA, with a ubiquitous set of standards, brings better reusability of existing assets or investments in the enterprise and lets you create applications that can be built on top of new and existing applications.
- SOA enables changes to applications while keeping clients or service consumers isolated from evolutionary changes that happen in the service implementation.
- SOA enables upgrading individual services or services consumers; it is not necessary to completely rewrite an application or keep an existing system that no longer addresses the new business requirements.
- Finally, SOA provides enterprises better flexibility in building applications and business processes in an agile manner by leveraging existing application infrastructure to compose new services

SOA Layers

Layer 1: **Business domain**

- based on the observation that all business process constellations in an enterprise target a particular business domain.
- set of current and future business processes that share common capabilities and functionality and can collaborate with each other to accomplish a higher-level business objective
- Ex:- loans, insurance, banking, finance, manufacturing, marketing, human resources, etc.



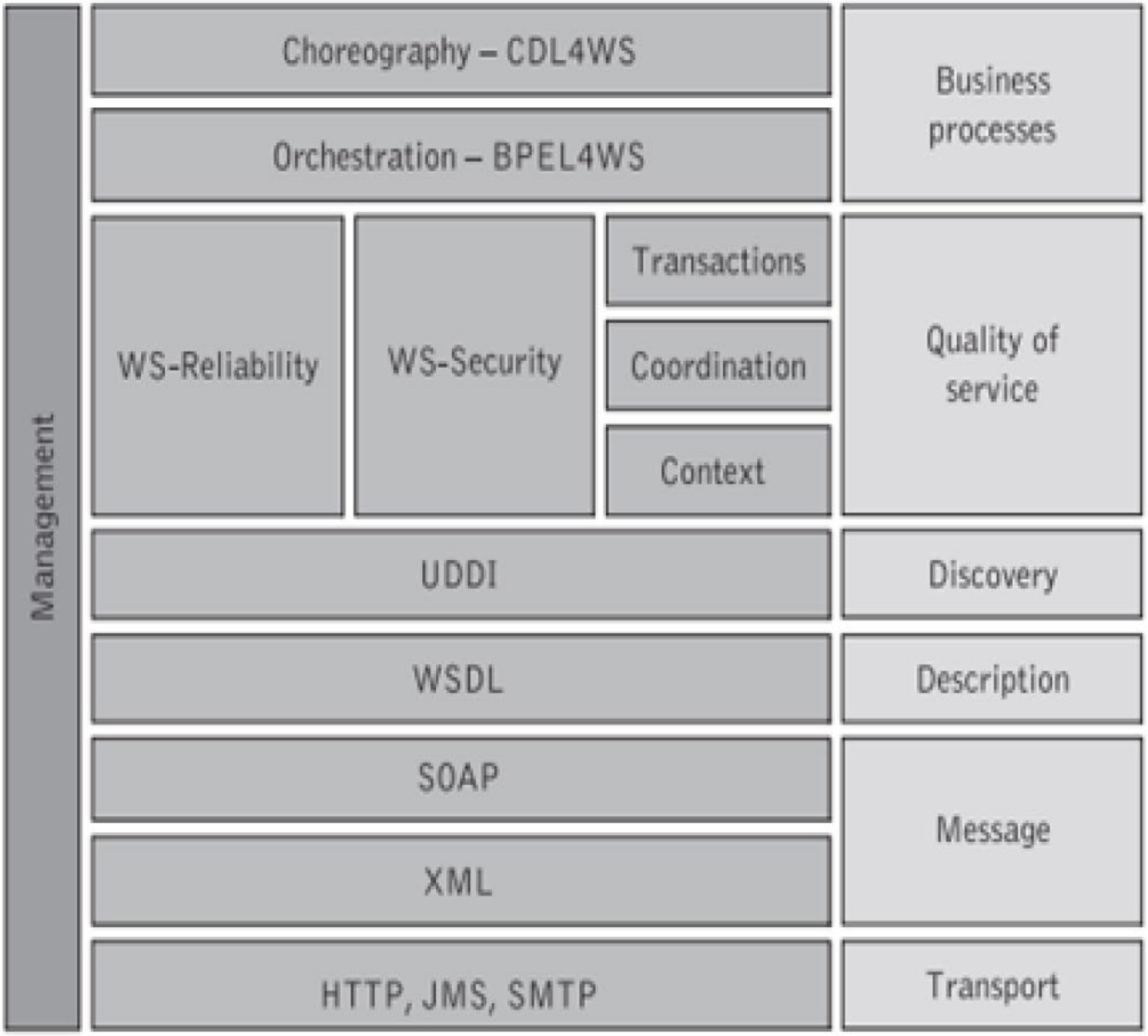
Layers in an SOA

Web services technology stack?

- The goal of Web services technology is to allow applications to work together over standard Internet protocols, without direct human intervention.
- By doing so, we can automate many business operations, creating new functional efficiencies and new, more effective ways of doing business.
- In order to simply things we provide a classification scheme for the most important standards in the Web services technology stack

Web services technology stack?

- Enabling technology standards : Although not specifically tied to any specific transport protocol, Web services build on ubiquitous Internet connectivity and infrastructure to ensure nearly universal reach and support.
- Extensible Markup Language (XML). XML is a widely accepted format for all exchanging data and its corresponding semantics.


Management
Choreography – CDL4WS
Orchestration – BPEL4WS
Business processes
WS-Reliability
WS-Security
Transactions
Coordination
Context
Quality of service
UDDI
Discovery
WSDL
Description
SOAP
XML
Message
HTTP, JMS, SMTP
Transport

Service composition and collaboration standards

Service composition

- Describes the execution logic of Web-services-based applications by defining their control flows
- prescribing the rules for consistently managing their unobservable business data.
- enterprises can describe complex processes that span multiple organizations – such as order processing, lead management, and claims handling – and execute the same business processes in systems from other vendors.
- The Business Process Execution Language (BPELcan achieve service composition for Web services

Web services technology stack?

Service collaboration:

- Describes cross-enterprise collaborations of Web service participants by defining their common observable behavior
- synchronized information exchanges occur through their shared contact points
- when commonly defined ordering rules are satisfied. Service collaboration is materialized by the Web Services Choreography Description Language (WS-CDL) which specifies the common observable behavior of all participants engaged in business collaboration.
- Each participant could be implemented not only by BPEL but also by other executable business process languages

Web services technology stack?

Coordination/transaction standards:

- Solving the problems associated with service discovery and service description retrieval is the key to success of Web services.
- The WS-Coordination and WS-Transaction initiatives complement BPEL to provide mechanisms for defining specific standard protocols for use by transaction processing systems, workflow systems, or other applications that wish to coordinate multiple Web services.
- These three specifications work in tandem to address the business workflow issues implicated in connecting and executing a number of Web services that may run on disparate platforms across organizations involved in e-business scenarios

Web services technology stack?

- Value-added standards:
- Additional elements that support complex business interactions must still be implemented before Web services can automate truly critical business processes.
- Value-added services standards include mechanisms for security and authentication, authorization, trust, privacy, secure conversations, contract management.

QoS

- A significant requirement for an SOA-based application is to operate in such a way that it functions reliably and delivers a consistent service at a variety of levels.
- This requires not only focusing on the functional properties of services but also concentrating on describing the environment hosting the Web service, i.e., describing the non-functional capabilities of services.
- Each service hosting environment may offer various choices of QoS based on technical requirements regarding demands for around-the-clock levels of service availability, performance and scalability, security and privacy policies, and so on, all of which must be described.
- It is thus obvious that the QoS offered by a Web service is becoming the highest priority for service providers and their customers

QoS

- ability of the Web service to respond to expected invocations and to perform them at the level commensurate with the mutual expectations of both its provider and its customers
- Several quality factors that reflect customer expectations, such as constant service availability, connectivity, and high responsiveness, become key to keeping a business competitive and viable as they can have a serious impact upon service provision.
- Delivering QoS on the Internet is a critical and significant challenge because of its dynamic and unpredictable nature

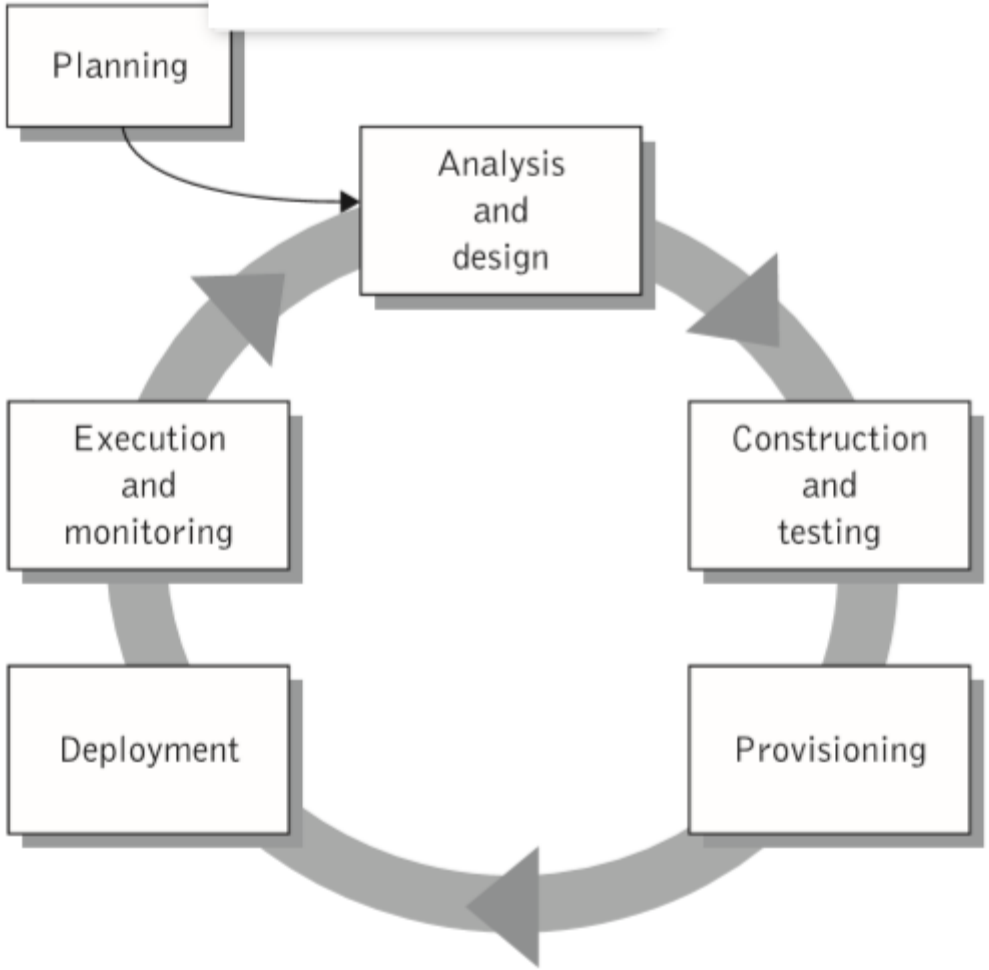
Qos

- Availability
- Accessibility
- Conformance to standard
- Integrity
- Performance-*Higher throughput and lower latency values represent good performance of a Web service*
- Reliability
- Scalability
- Security
- Transactionality
- An SLA is a formal agreement (contract) between a provider and client, formalizing the details of a Web service (contents, price, delivery process, acceptance and quality criteria, penalties, and so on, usually in measurable terms) in a way that meets the mutual understandings and expectations of both the service provider and the service requestor.

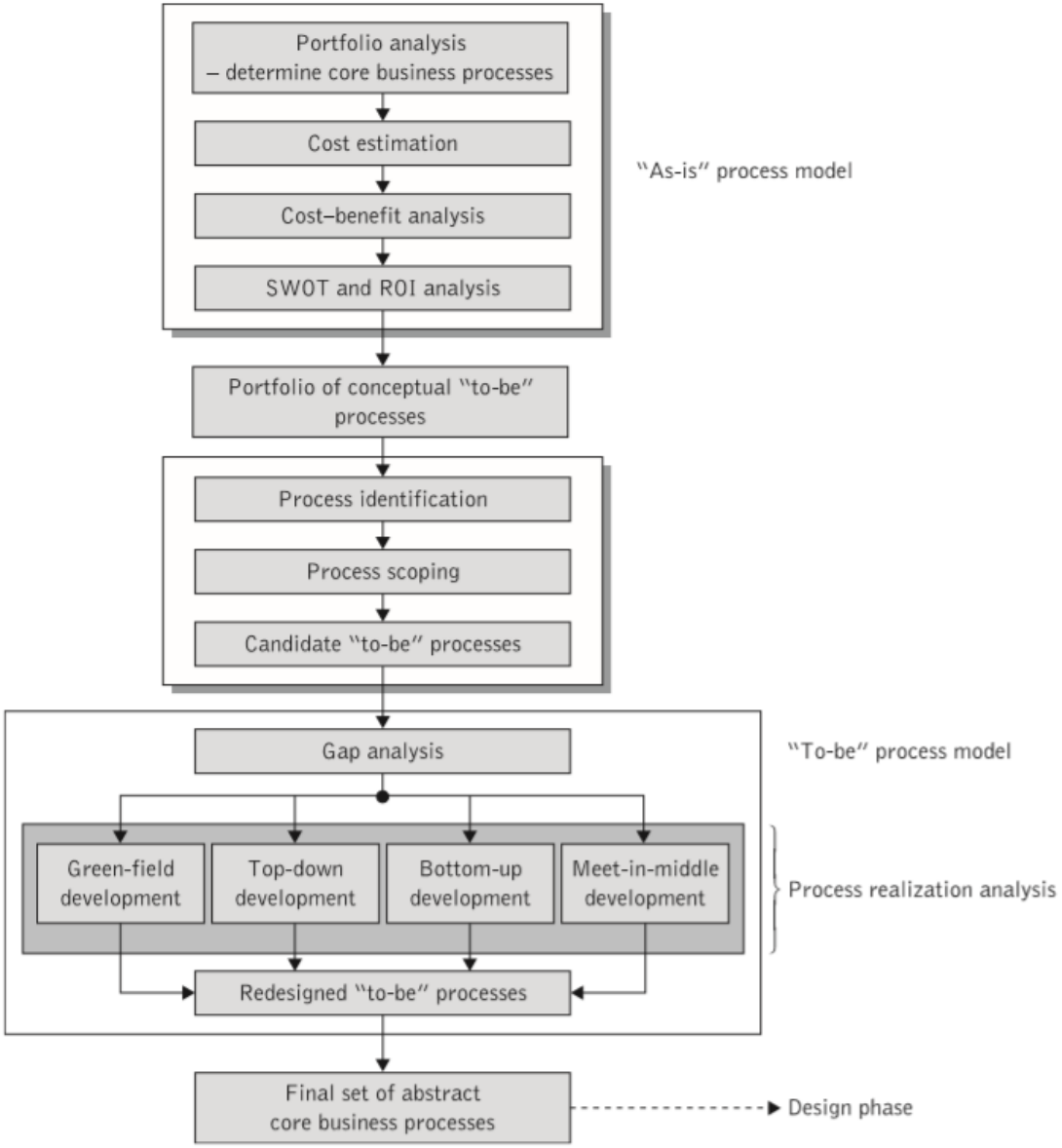
An SLA parts

- **Purpose**: This field describes the reasons behind the creation of the SLA.
- **Parties**: This field describes the parties involved in the SLA and their respective roles, e.g., service provider and service consumer (client).
- **Validity period**: This field defines the period of time that the SLA will cover. This is delimited by start time and end time of the agreement term.
- **Scope**: This field defines the services covered in the agreement.
- **Restrictions**: This field defines the necessary steps to be taken in order for the requested service levels to be provided.
- **Service-level objectives**: This field defines the levels of service that both the service customers and the service providers agree on, and usually includes a set of service level indicators, like availability, performance, and reliability. Each of these aspects of the service level will have a target level to achieve.
- **Penalties**: This field defines what sanctions should apply in case the service provider underperforms and is unable to meet the objectives specified in the SLA.
- **Optional services**: This field specifies any services that are not normally required by the user, but might be required in case of an exception.
- Exclusion terms: These specify what is not covered in the SLA.
- **Administration:** This field describes the processes and the measurable objectives in an SLA and defines the organizational authority for overseeing them.

Web services development lifecycle



analysis


Portfolio analysis
– determine core business processes
Cost estimation
Cost–benefit analysis
SWOT and ROI analysis
"As-is" process model
Portfolio of conceptual "to-be" processes
Process identification
Process scoping
Candidate "to-be" processes
Gap analysis
"To-be" process model
Green-field development
Top-down development
Bottom-up development
Meet-in-middle development
Process realization analysis
Redesigned "to-be" processes
Final set of abstract core business processes
Design phase