Lecture 4.d

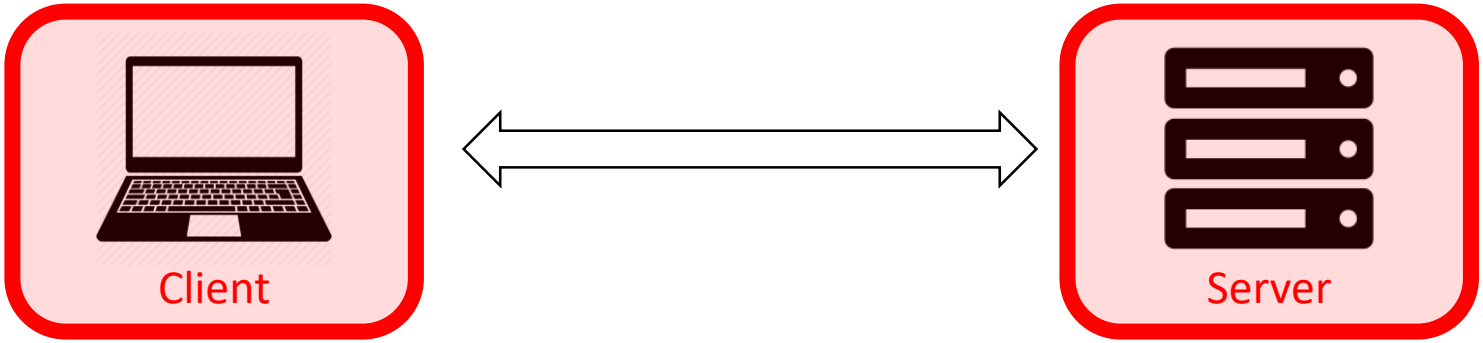
Service Oriented Architectures

Dr. Gabriele Pierantoni

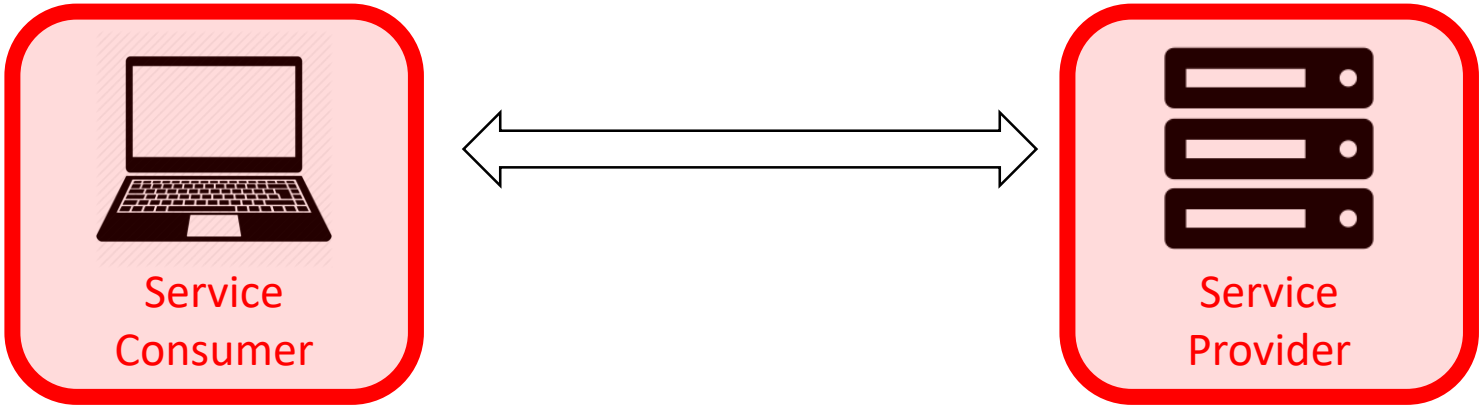
10.02.2021

Service Oriented Architectures

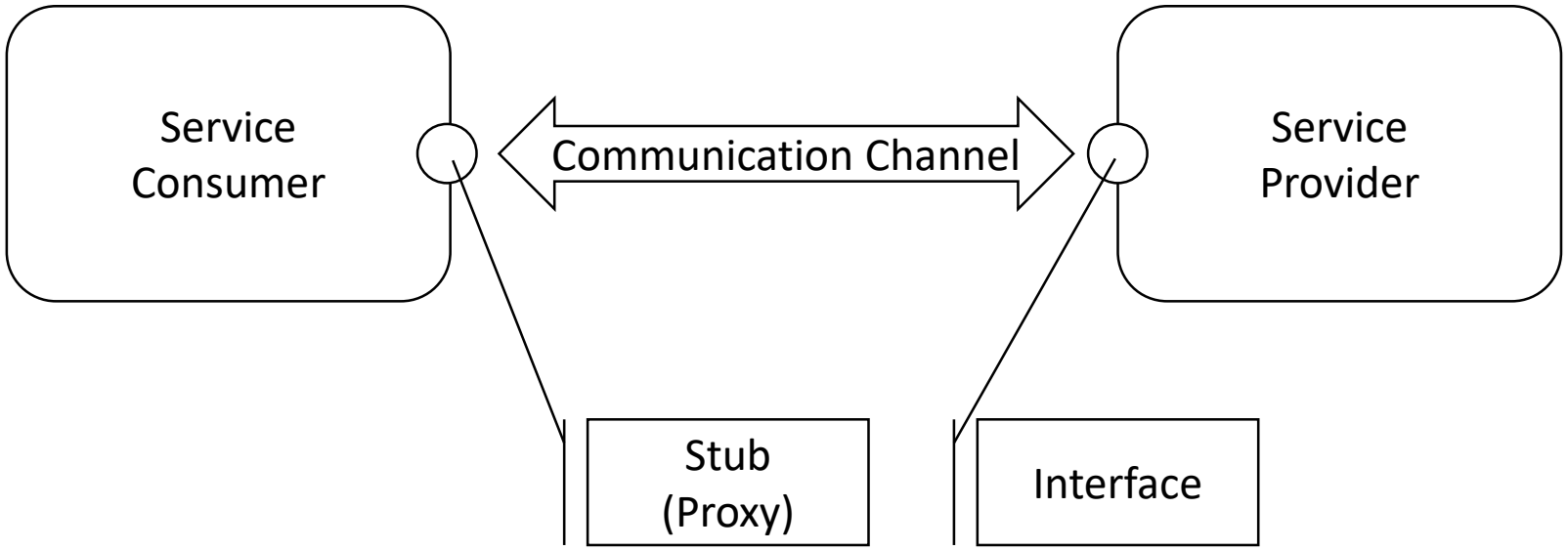
Client Server Architecture



Service Oriented Architecture



Service Oriented Architecture

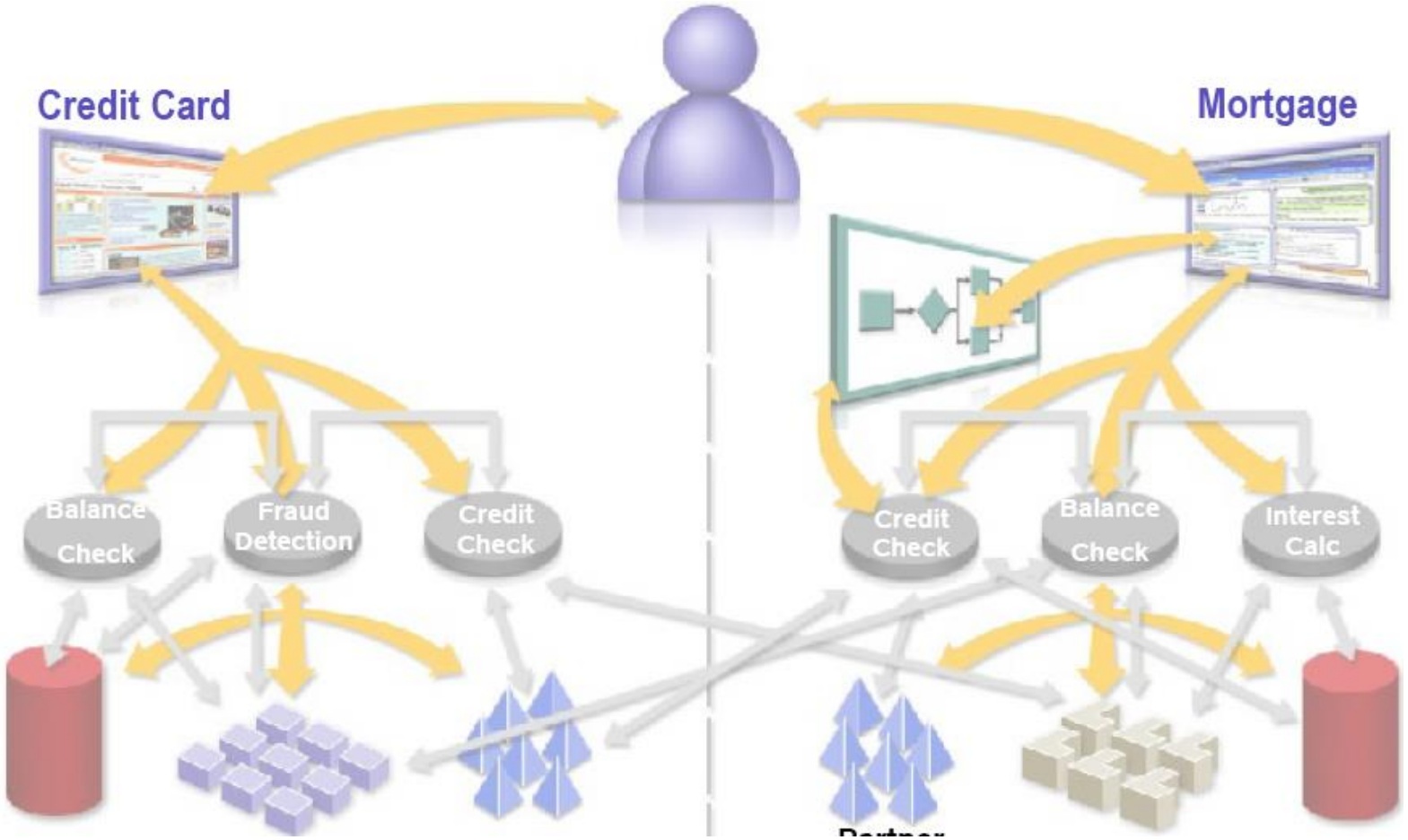


Service Oriented Architecture (SOA)

SOA definition

- An architecture for structuring systems into coarse-grained services
- Technology-neutral concept and best practice
- Emphasizes the loose coupling of services
- New services are created from existing ones in a synergistic fashion
- Services, are defined using a description language and have invokable interface that are called to perform business processes.
- Services can be re-composed when business requirements change

Pre-SOA Scenario



Customer Data

Back-End System

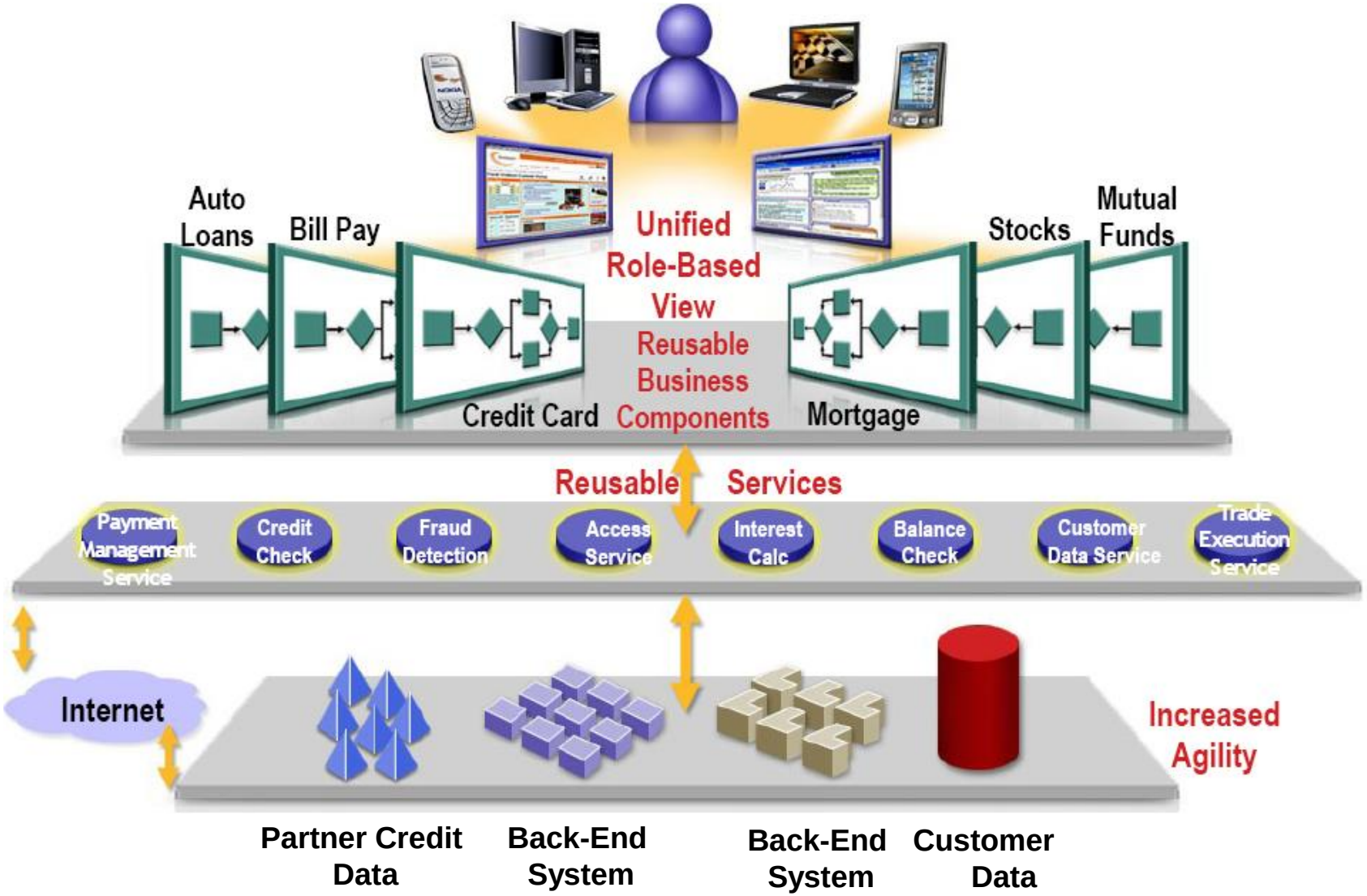
Partner Credit Data

Partner Credit Data

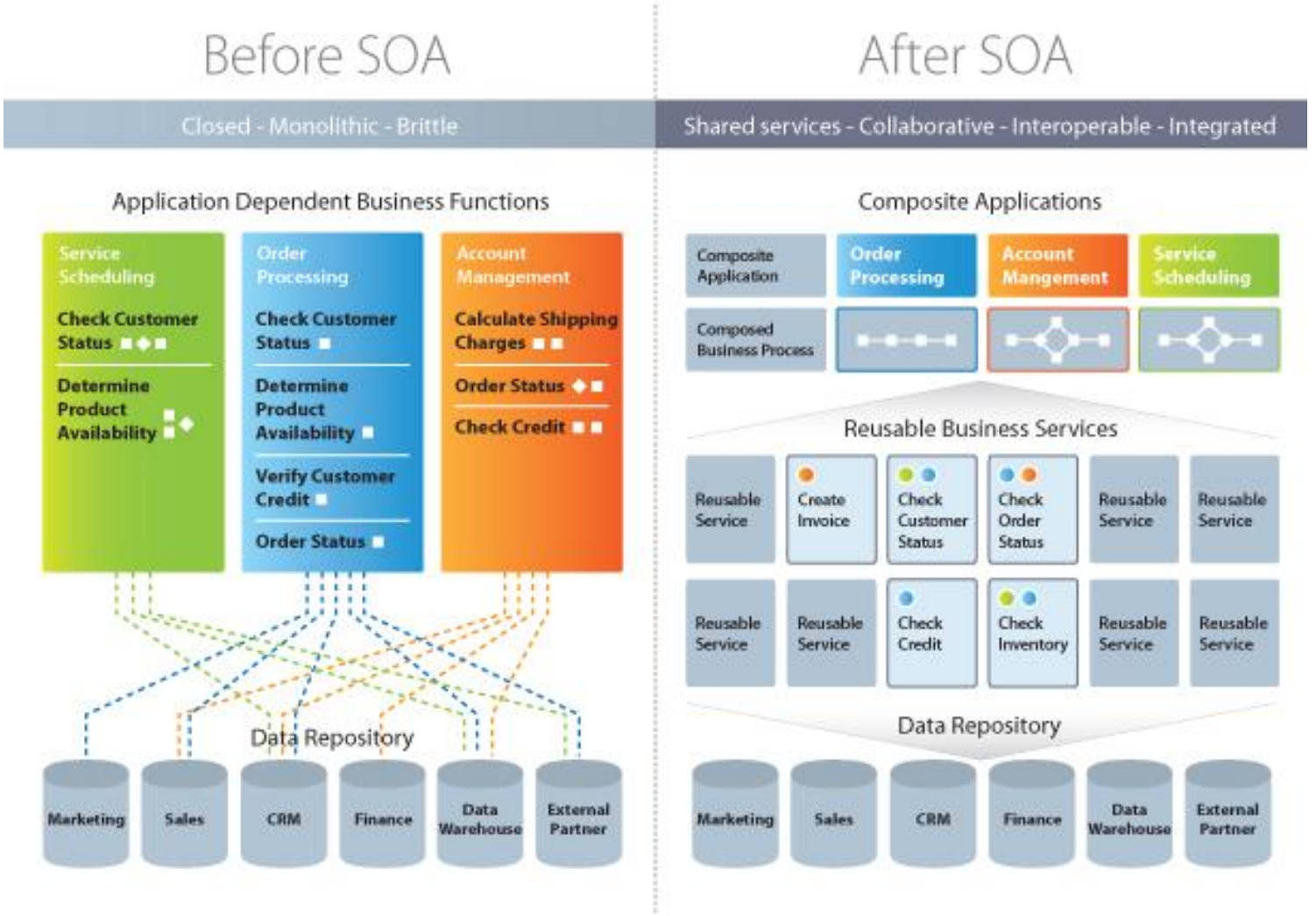
Back-End System

Customer Data

SOA Scenario



Service Oriented Architecture



SOA Entities

SOA Entities (1)

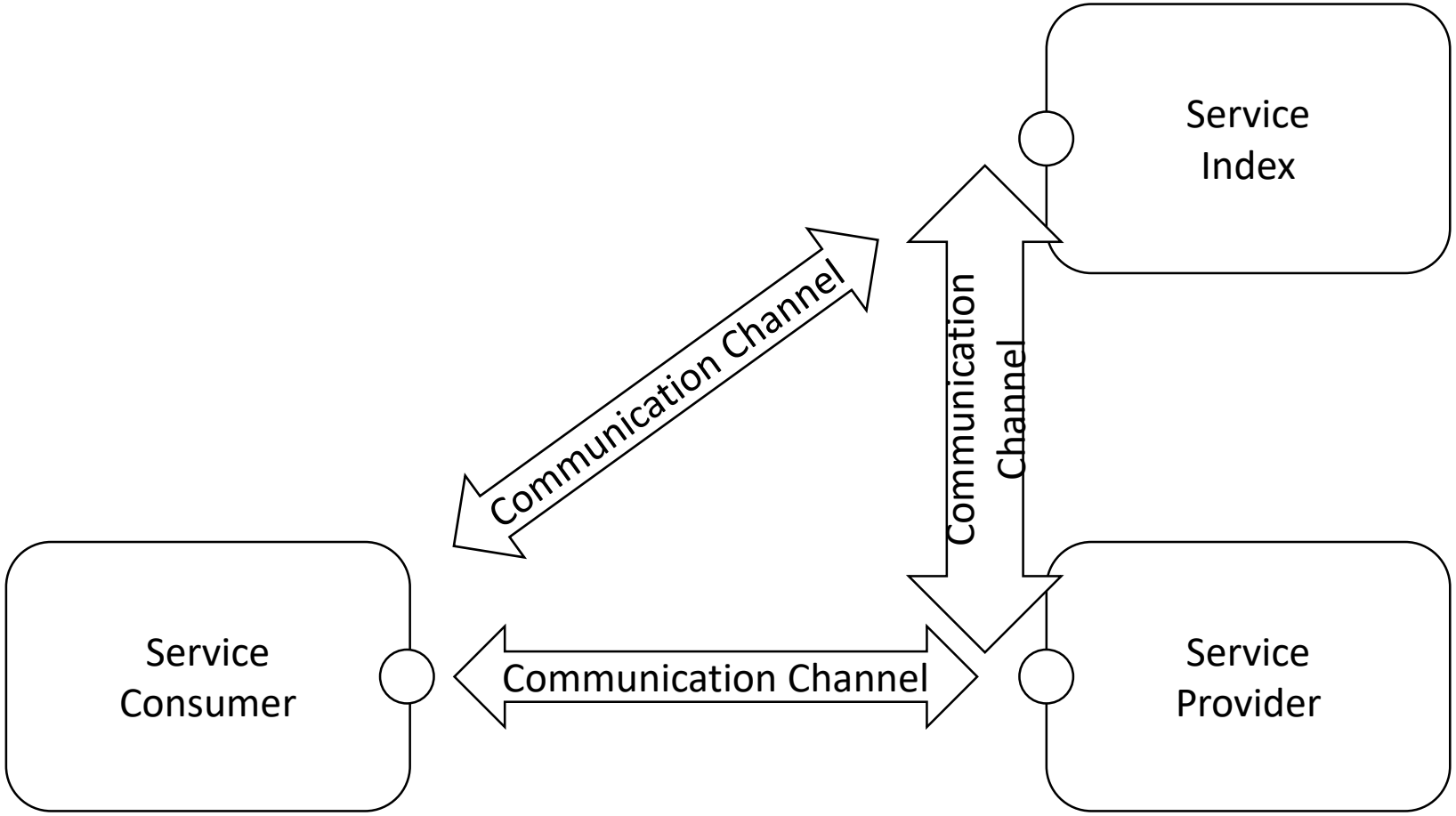
- **SOA entities** are either **functional** elements, such as service or **non-functional** elements, such as service-level data model,

- **Components** are abstract modules of software deployed as a unit onto a server with other components.

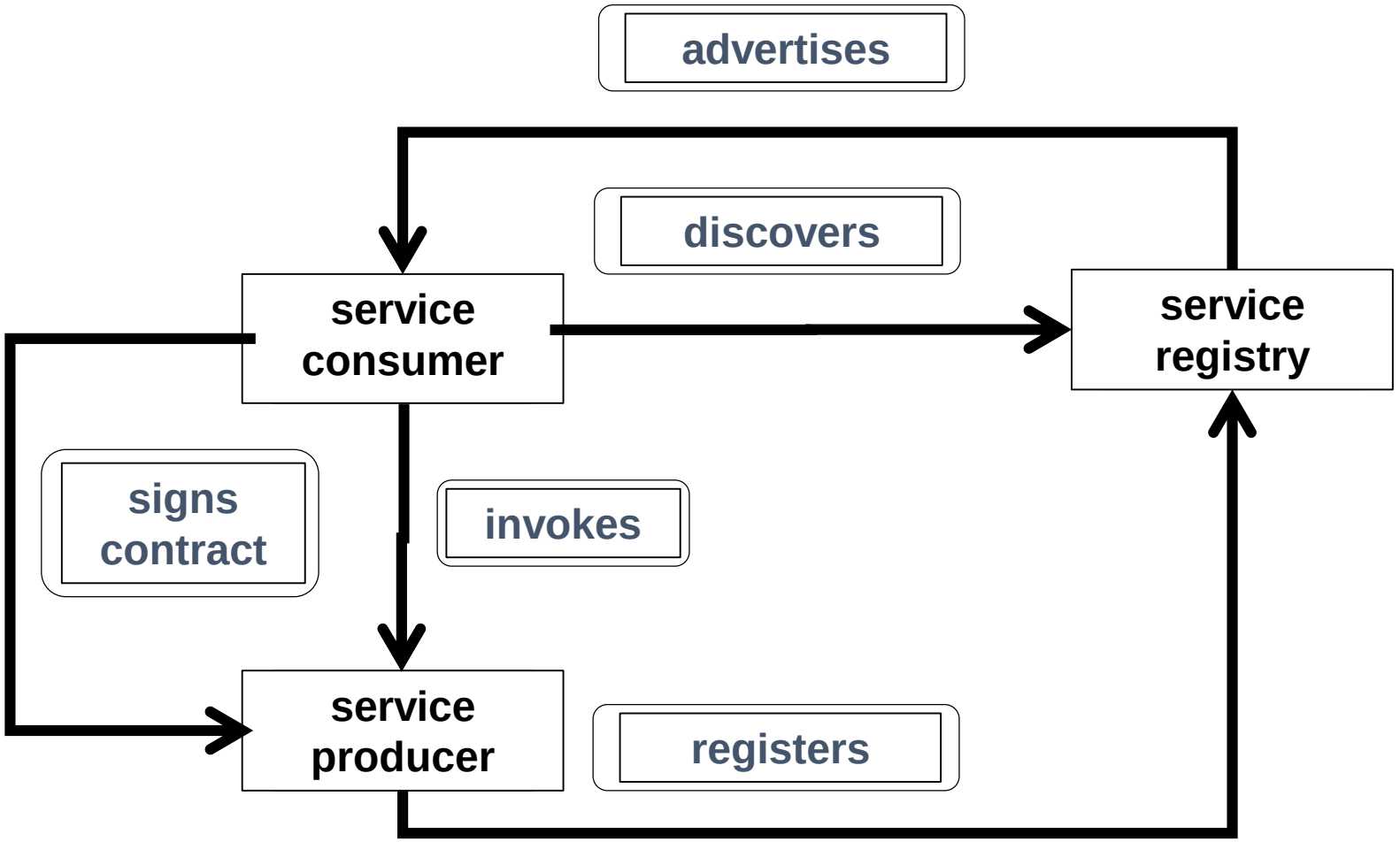
- **Interactions** between components are implemented by **connectors**

- The **configuration of components and connectors** describes the way a system is structured and behaves.

Service Oriented Architecture



SOA Entities (4)



SOA Entities (5)

Service

- **Black-Box** components with well-defined interfaces
- **Provides a specific function**, which is either a single discrete function (fine-grained service) or set of related functions (coarse-grained service)
- **Service's behaviour** is defined by a service contract that can be implemented (service providers) and offered for use by another component (service consumers)

SOA Entities (5)

example: Web service that could

- Perform business logic
- Transform data
- Route messages
- Query databases
- Apply business policy
- Handle business exceptions
- Prepare information for use by a user interface
- Orchestrate conversations between multiple services

SOA Entities (6)

Service Description

- **Service** should consist of parameters, constraints and policies that define how to invoke the service
- **Each service** should include a service definition in a standardized format, which enables applications and human actors to examine the service description and determine issues such as: what the service does, how they may bind to it, and what security protocols (if any) must be used with it.
- **Service definitions** may also include details about any implied process or other legal or business terms that occur when the service is invoked

example: Web Services Description Language (WSDL)

SOA Entities (7)

Service Contract

- It is a specification of the way a service consumer will interact with a service provider
- It specifies the format of the request to the service and response from it
- A service contract may require a set of pre- and post-conditions, where the pre- and post-conditions specify the state that the service must be in to execute a particular function
- The contract may also specify quality of service (QoS) levels, which are specifications for the non-functional aspects of the service, for instance, a quality of service attribute is the amount of time it takes to execute a service request, security, etc.

SOA Entities (8)

Service Proxy (client stub or service client) (1/3)

- The service provider supplies a service proxy to the service consumer, who sends the service request by calling an API function on the proxy
- The service proxy finds a contract and a reference to the service provider in the registry, then formats the request and executes the request on behalf of the service consumer
- It supports only methods the service itself provides, if the proxy in any way changes the interface of the remote service, then technically, it is no longer a proxy

SOA Entities (9)

Service Proxy (client stub or service client) (2/3)

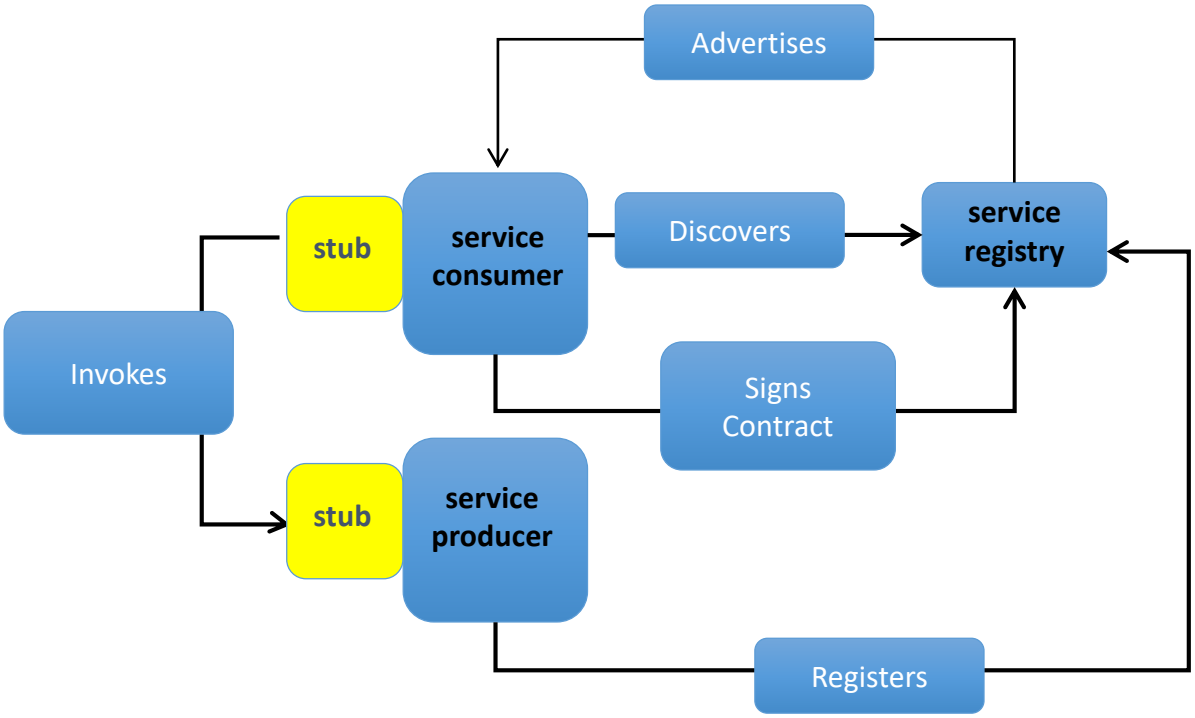
- The service proxy is simply a local reference to a remote service, which is written in the native language of the service consumer
- It is a convenience entity for the service consumer, i.e. it is not required and the service consumer developer could write the necessary software for accessing the service directly
- The service proxy can enhance performance by caching remote references and data
- When a proxy caches a remote reference, subsequent service calls do not require additional service registry calls

SOA Entities (10)

Service Proxy (client stub or service client) (3/3)

- By storing service contracts locally, the service consumer reduces the number of network communication required to execute the service
- The service proxy can improve performance by eliminating network calls altogether by performing some functions locally
- For service methods that do not require service data, the entire method can be implemented locally in the proxy
- If a method requires some small amount of service data, the proxy could download the data once and use it for subsequent method calls

SOA Entities (10)



SOA Entities (11)

Service Registry (or Repository)

- It is an interface that provides information how to bind to entities
- The owner of the entity can make an entry to reference the entity and explain how to bind to it and others may retrieve this information and use it to bind to the entities
- Its implementations allow for federation (also called replication) to allow content from one implementation to be replicated or referenced from within other implementations

SOA Entities (12)

Service Registry (or Repository)

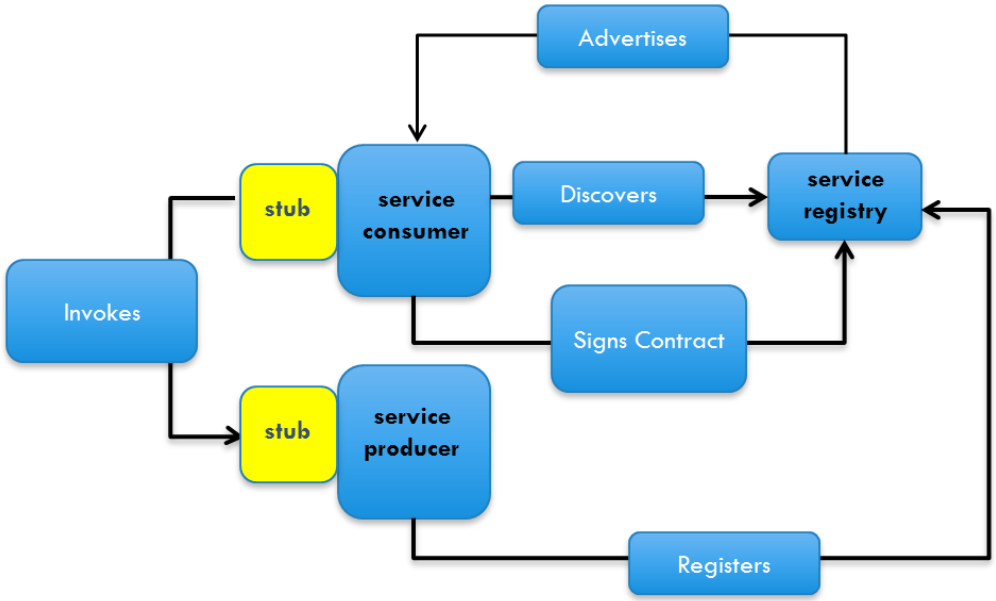
- It is an entity where service providers can store and service consumers can find entities required for their enterprise to function
- The registry provides a storage mechanism that is bound to the registry such that the registry knows about any auditable events to the entity in the repository
- It provides functionality to store and retrieve service-metadata. WSDL, XSD and Policy documents can be loaded and parsed into separate entities
- For example with a WSDL document, separate entities are created for service, binding, portType etc. XML and binary documents can be loaded as single entities,

example: **OASIS Universal Description and Discovery Interface (UDDI)**

SOA Entities (13)

Service Provider

- It is the service entity that accepts and executes requests from consumers
- It can be a mainframe system or some other type of software system that executes the service request
- It publishes its contract in the service registry for access by service consumers



SOA Entities (14)

Service Consumer

- It is an application, service, or some other type of software module that requires a service
- It is the entity that initiates the locating of the service in the service registry, binding to the service over a computer network, and requesting a service function
- It invokes the service by sending a request in a message formatted according to the contract