SERVICES (AS WEB SERVICES)

- Web services provide the potential of fulfilling primitive requirements, but they need to be intentionally designed to do so. This is because the Web services framework is flexible and adaptable.

- Web services can be designed to duplicate the behavior and functionality found in proprietary distributed systems, or they can be designed to be fully SOA-compliant. This flexibility has allowed Web services to become part of many existing application environments and has been one of the reasons behind their popularity.

- It also reveals the fact that Web services are not necessarily inherently service-oriented.

SERVICES (AS WEB SERVICES)

Most basic Web services design concepts:

Fundamentally, every Web service can be associated with:

- a temporary classification based on the roles it assumes during the runtime processing of a message
  - service roles (temporary classifications)

- a permanent classification based on the application logic it provides and the roles it assumes within a solution environment
  - service models (permanent classifications)

SERVICES (AS WEB SERVICES)

Service roles:

- Web service is capable of assuming different roles, depending on the context within which it is used.
- example, a service can act as the initiator, relayer, or the recipient of a message.
- A service is therefore not labeled exclusively as a client or server, but instead as a unit of software capable of altering its role, depending on its processing responsibility in a given scenario.

SERVICES (AS WEB SERVICES)

Service provider:

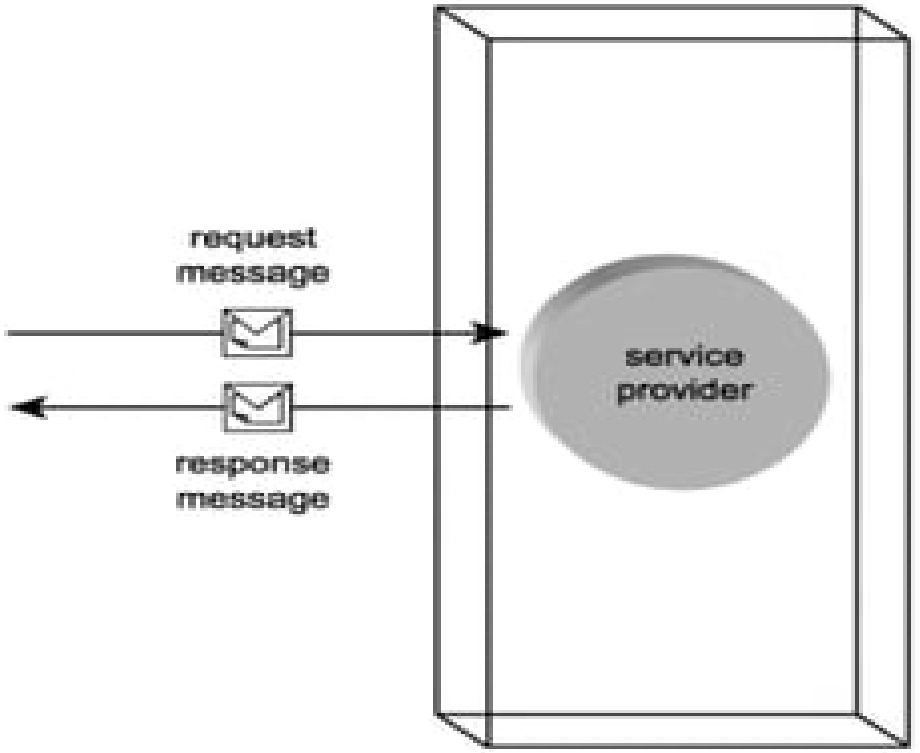
The service provider role is assumed by a Web service under the following conditions:

- The Web service is invoked via an external source, such as a service requestor
- The Web service provides a published service description offering information about its features and behavior.
- Depending on the type of message exchange used when invoking a service provider, the service provider may reply to a request message with a response message.

SERVICES (AS WEB SERVICES)

Service provider:

As the recipient of a request message, the Web service is classified as a service provider.



SERVICES (AS WEB SERVICES)

Service requestor:

- Any unit of processing logic capable of issuing a request message that can be understood by the service provider is classified as a service requestor.

- A Web service is always a service provider but also can act as a service requestor.

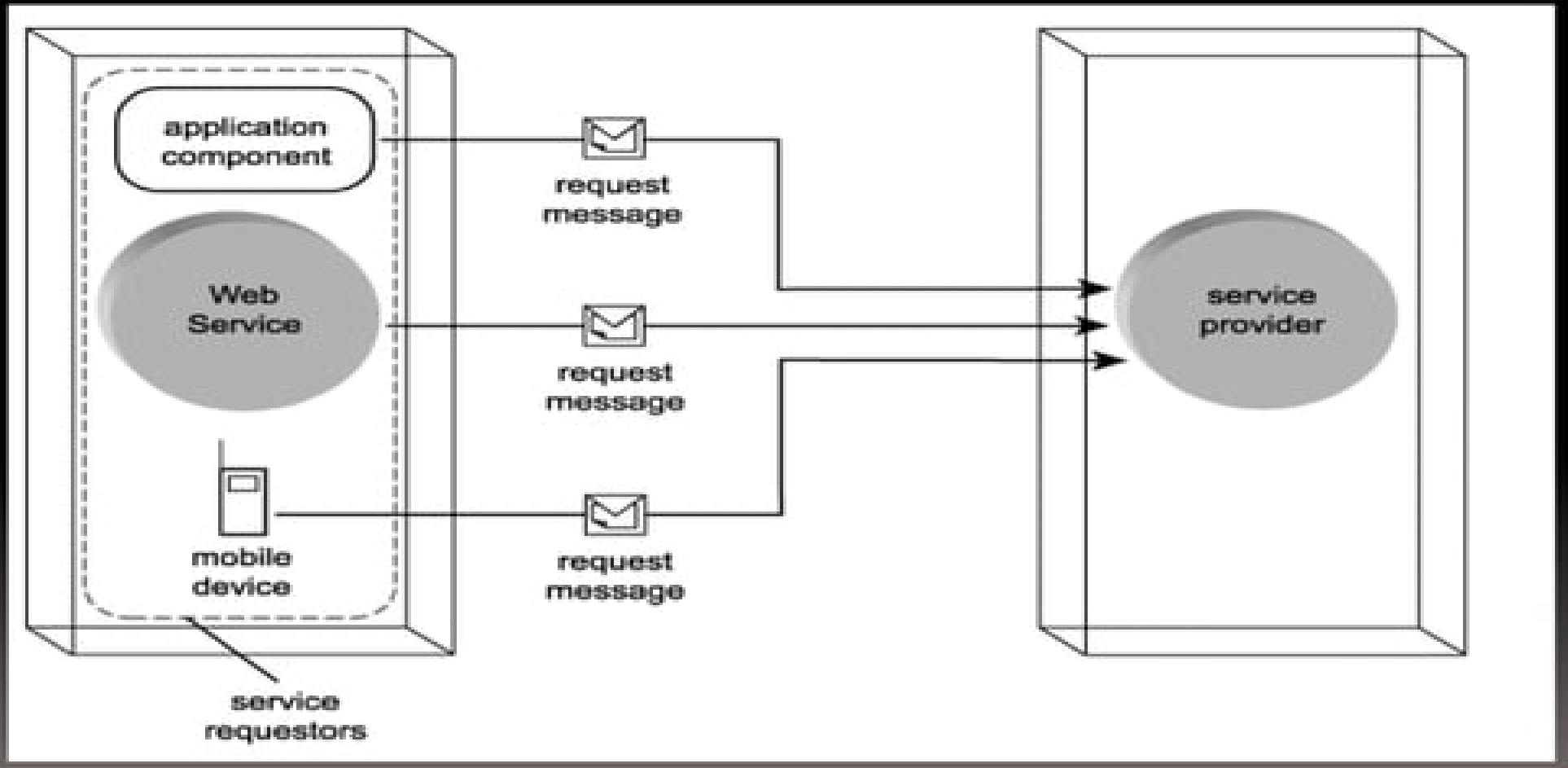
A Web service takes on the service requestor role under the following circumstances:

- The Web service invokes a service provider by sending it a message
- The Web service searches for and assesses the most suitable service provider by studying available service descriptions.

SERVICES (AS WEB SERVICES)

Service requestor:

The sender of the request message is classified as a service requestor



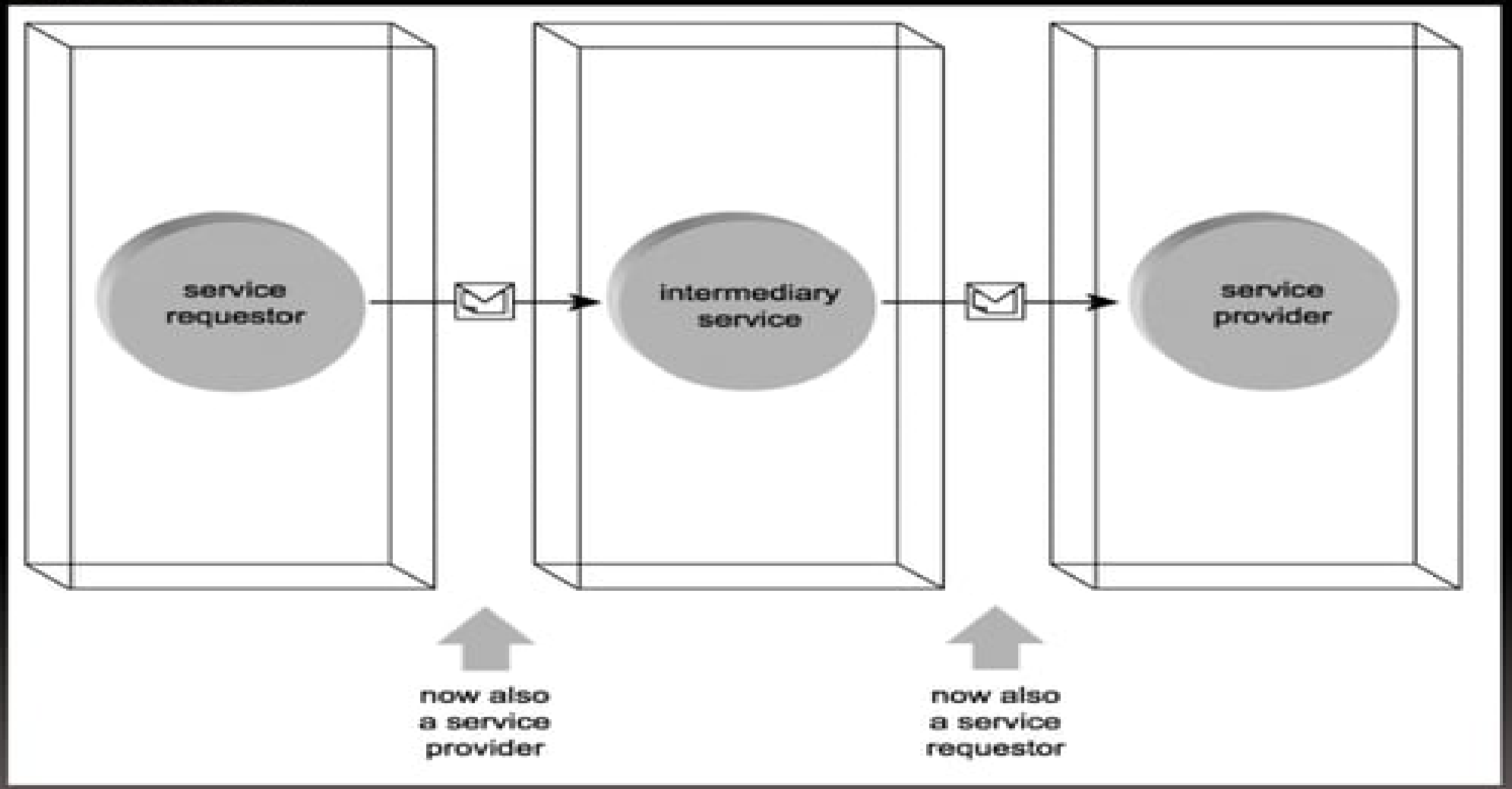
SERVICES (AS WEB SERVICES)

Intermediaries:

- The communications framework established by Web services contrasts the predictable nature of traditional point-to-point communications channels.
- Web services communication is based on the use of messaging paths, which can best be described as point-to-* paths.
- once a service provider submits a message, it can be processed by multiple intermediate routing and processing agents before it arrives at its ultimate destination.
- Web services and service agents that route and process a message after it is initially sent and before it arrives at its ultimate destination are referred to as intermediaries or intermediary services.
- Because an intermediary receives and submits messages, it always transitions through service provider and service requestor roles

SERVICES (AS WEB SERVICES)

Intermediaries:



The intermediary service transitions through service provider and service requestor roles while processing a message

SERVICES (AS WEB SERVICES)

Intermediaries:

Types of intermediaries.

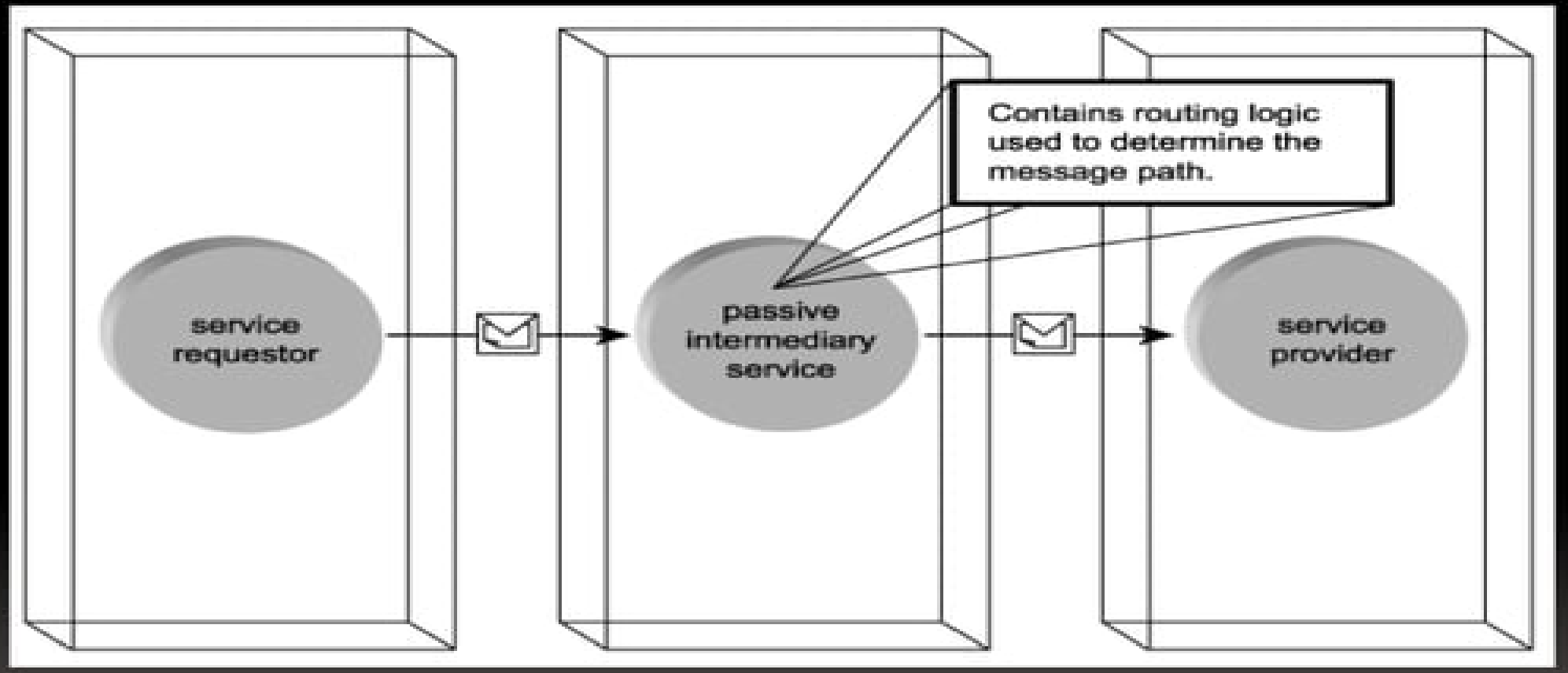
1. passive intermediary
2. active intermediaries

1. passive intermediary

- typically responsible for routing messages to a subsequent location .
- It may use information in the SOAP message header to determine the routing path, or it may employ native routing logic to achieve some level of load balancing.
- intermediary is because it does not modify the message.

SERVICES (AS WEB SERVICES)

passive Intermediaries:



A passive intermediary service processing a message without altering its contents

SERVICES (AS WEB SERVICES)

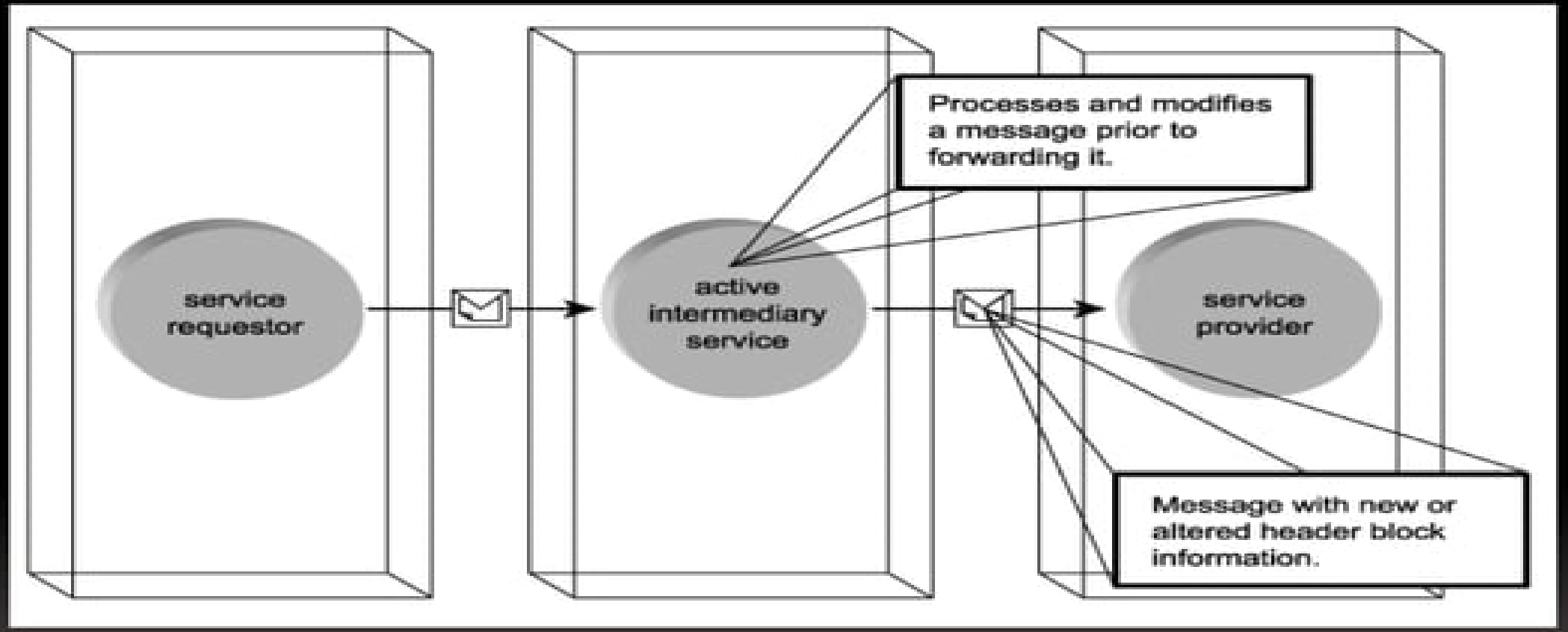
Intermediaries:

2. Active intermediary

- route messages to a forwarding destination.
- Prior to transmitting a message, however, these services actively process and alter the message contents .
- Typically, active intermediaries will look for particular SOAP header blocks and perform some action in response to the information they find there.
- They almost always alter data in header blocks and may insert or even delete header blocks entirely.

SERVICES (AS WEB SERVICES)

Active Intermediaries:



An active intermediary service

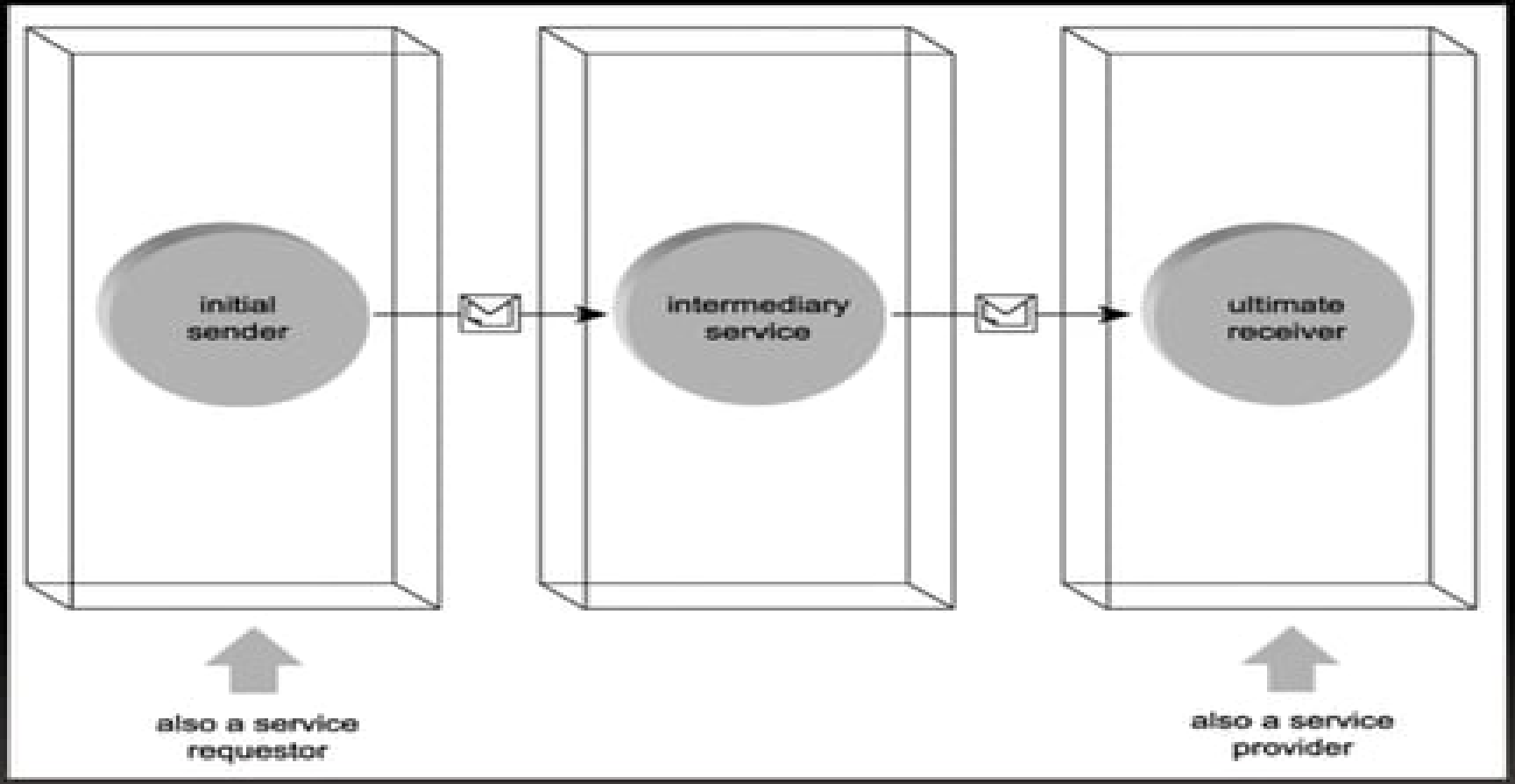
SERVICES (AS WEB SERVICES)

Initial sender and ultimate receiver:

- Initial senders are simply service requestors that initiate the transmission of a message.
- the initial sender is always the first Web service in a message path.
- The counterpart to this role is the ultimate receiver.
- This label identifies service providers that exist as the last Web service along a message's path
- intermediary services can never be initial senders or ultimate receivers within the scope of a service activity.

SERVICES (AS WEB SERVICES)

Initial sender and ultimate receiver :



Web services acting as initial sender and ultimate receiver

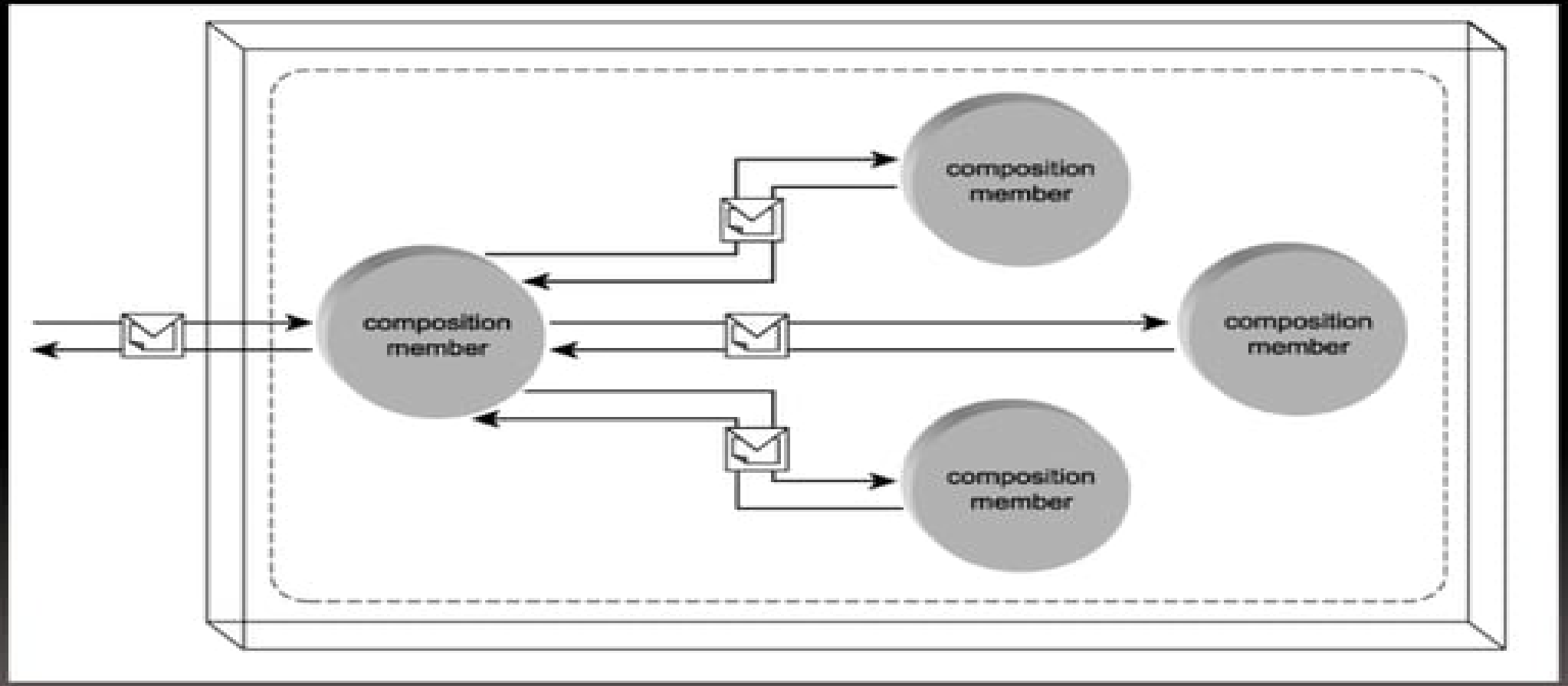
SERVICES (AS WEB SERVICES)

Service compositions

- The term does not apply to a single Web service, but to a composite relationship between a collection of services.
- Any service can enlist one or more additional services to complete a given task.
- any of the enlisted services can call other services to complete a given sub-task.
- each service that participates in a composition assumes an individual role of service composition member
- Service compositions also are referred to as service assemblies

SERVICES (AS WEB SERVICES)

Service compositions



A service composition consisting of four members

SERVICES (AS WEB SERVICES)

Service models

- The roles explored so far are agnostic to the nature of the functionality being provided by the Web service.
- They are generic states that a service can enter within a generic context.
- The manner in which services are being utilized in the real world, though, has led to a classification based on the nature of the application logic they provide, as well as their business-related roles within the overall solution. These classifications are known as service models.

SERVICES (AS WEB SERVICES)

Business service model

- Within an SOA, the business service represents the most fundamental building block.
- It encapsulates a distinct set of business logic within a well-defined functional boundary.
- It is fully autonomous but still not limited to executing in isolation, as business services are frequently expected to participate in service compositions.

Business services are used within SOAs as follows:

- as fundamental building blocks for the representation of business logic
- to represent a corporate entity or information set
- to represent business process logic
- as service composition members

SERVICES (AS WEB SERVICES)

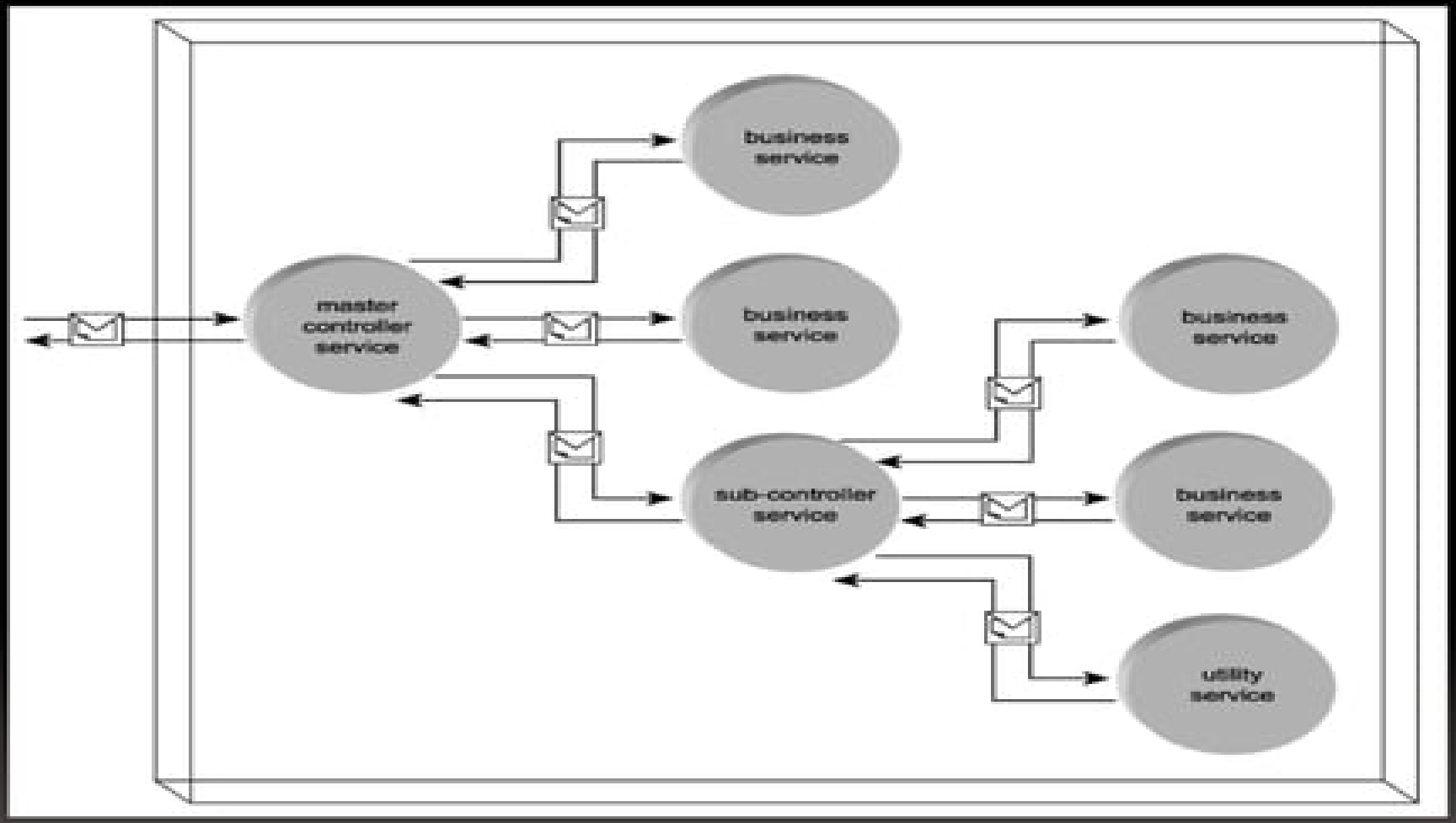
Utility service model

- Any generic Web service or service agent designed for potential reuse can be classified as a utility service.
- the reusable functionality be completely generic and non-application specific in nature.

Utility services are used within SOAs as follows:

- as services that enable the characteristic of reuse within SOA
- as solution-agnostic intermediary services
- as services that promote the intrinsic interoperability characteristic of SOA
- as the services with the highest degree of autonomy

SERVICES (AS WEB SERVICES)

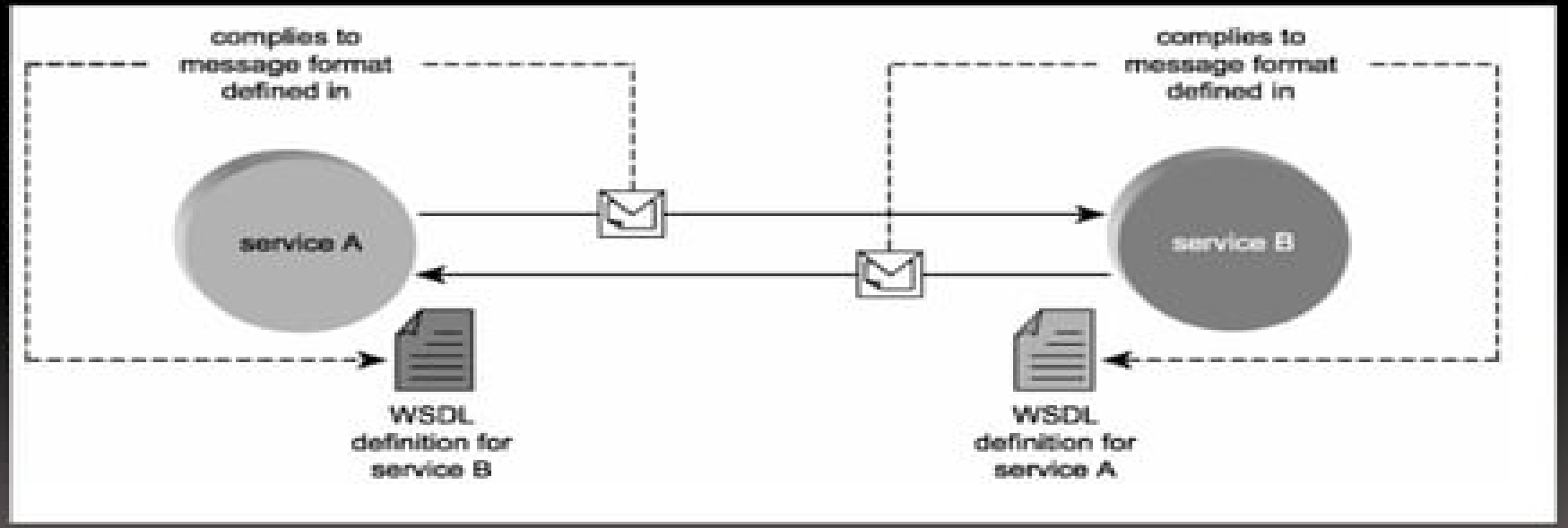


A service composition consisting of a master controller, a subcontroller, four business services, and one utility service

SERVICE DESCRIPTIONS (WITH WSDL)

- description documents are required to accompany any service wanting to act as an ultimate receiver.
- The primary service description document is the WSDL definition

WSDL definitions enable loose coupling between services



SERVICE DESCRIPTIONS (WITH WSDL)

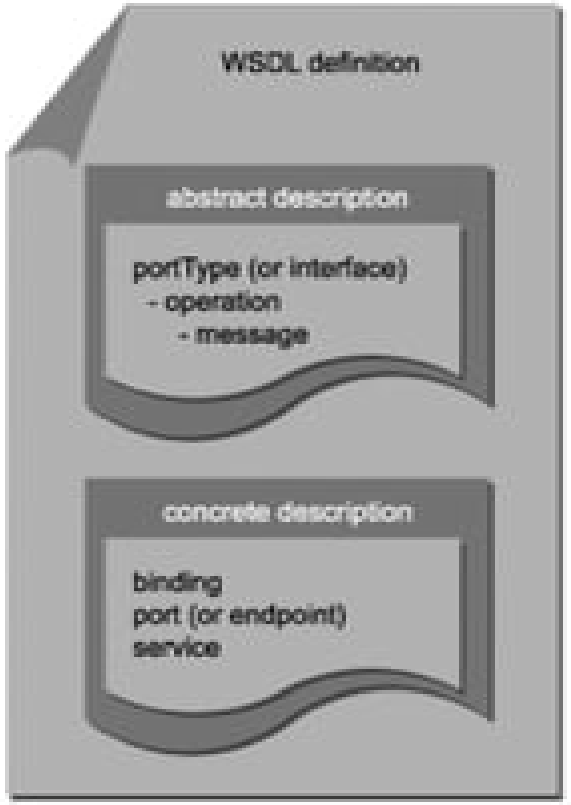
Service endpoints and service descriptions:

- A WSDL describes the point of contact for a service provider, also known as the service endpoint or just endpoint.
- It provides a formal definition of the endpoint interface (so that requestors wishing to communicate with the service provider know exactly how to structure request messages) and also establishes the physical location (address) of the service.
- A WSDL service description (also known as WSDL service definition or just WSDL definition) can be separated into two categories:
  - abstract description
  - concrete description

SERVICE DESCRIPTIONS (WITH WSDL)

Service endpoints and service descriptions:

WSDL document consisting of abstract and concrete parts that collectively describe a service endpoint



SERVICE DESCRIPTIONS (WITH WSDL)

Semantic descriptions

- Most of the metadata currently provided by services focuses on expressing technical information related to data representation and processing requirements.
- The most challenging part of providing a complete description of a Web service is in communicating its semantic qualities.

Examples of service semantics include:

1. how a service behaves under certain conditions
2. how a service will respond to a specific condition
3. what specific tasks the service is most suited for

SERVICE DESCRIPTIONS (WITH WSDL)

Semantic descriptions

- Most of the time service semantics are assessed by humans, either verbally by discussing the qualities of a service with its owner, or by reading supplementary documentation published alongside service descriptions.
- The ultimate goal is to provide sufficient semantic information in a structured manner so that, in some cases, service requestors can go as far as to evaluate and choose suitable service providers independently.
- Semantic information is important when dealing with external service providers, where your knowledge of another party's service is limited to the information the service owner decides to publish.
- But even within organizational boundaries, semantic characteristics tend to take on greater relevance as the amount of internal Web services grows.

SERVICE DESCRIPTIONS (WITH WSDL)

Service description advertisement and discovery

- The requirement for one service to contact another is access to the other service's description.
- As the amount of services increases within and outside of organizations, mechanisms for advertising and discovering service descriptions may become necessary.
- For example, central directories and registries become an option to keep track of the many service descriptions that become available.
- These repositories allow humans (and even service requestors) to:
  - locate the latest versions of known service descriptions
  - discover new Web services that meet certain criteria

SERVICE DESCRIPTIONS (WITH WSDL)

Service description advertisement and discovery