Web Services and Primitive SOA

The Web Services Framework

- **What is Framework?**
  - A collection of things

- Web Services framework includes:
  - Web Services
  - Communication Agreement
  - Messaging Framework
  - Extensions of Second Generation

Service as Web Service in SOA

- A service can be a web service in SOA if
  - It ensures basic **service orientation**
  - It implements a **business functionality**

Classification of Web Services

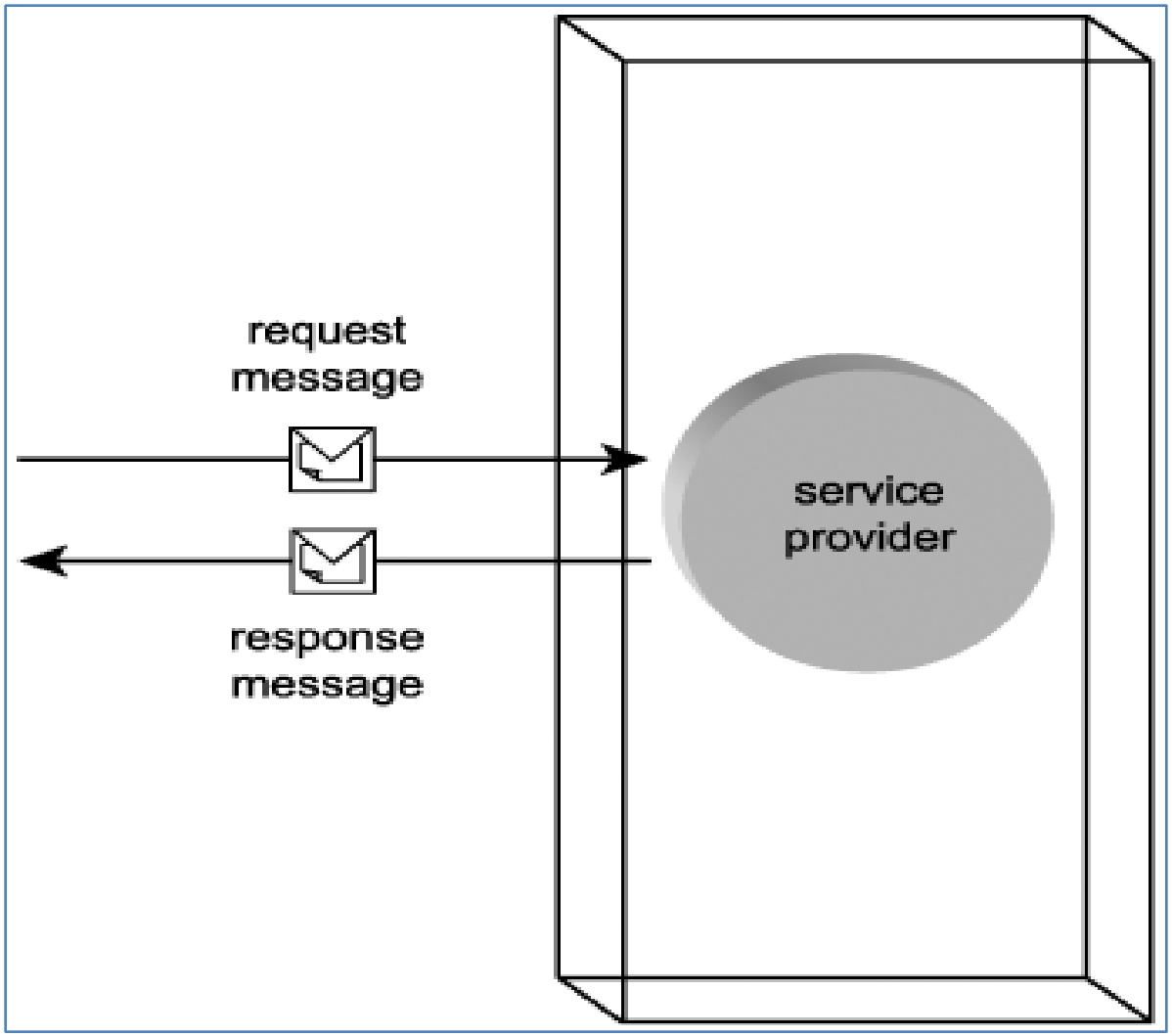
- **Temporary classification:**
  - Based on the roles it assumes during the runtime processing of a message
  - Also called Role based classification

- **Permanent classification:**
  - Based on the application logic it provides and the roles it assumes within a solution environment
  - Also called Model based classification

Service roles (Temporary Classification)

- A web service can assume different roles depending upon the context
- It can be
  - Initiator
  - Relayer
  - Recipient
- A service can change role based on context

Service roles – Service Provider



Service roles – Service Provider

- **Provider Role** requires following conditions to be fulfilled:
  1. The Web service is **invoked** via an external source, such as a service requestor.
  2. The Web service provides a published **service description** offering information about its features and behavior.

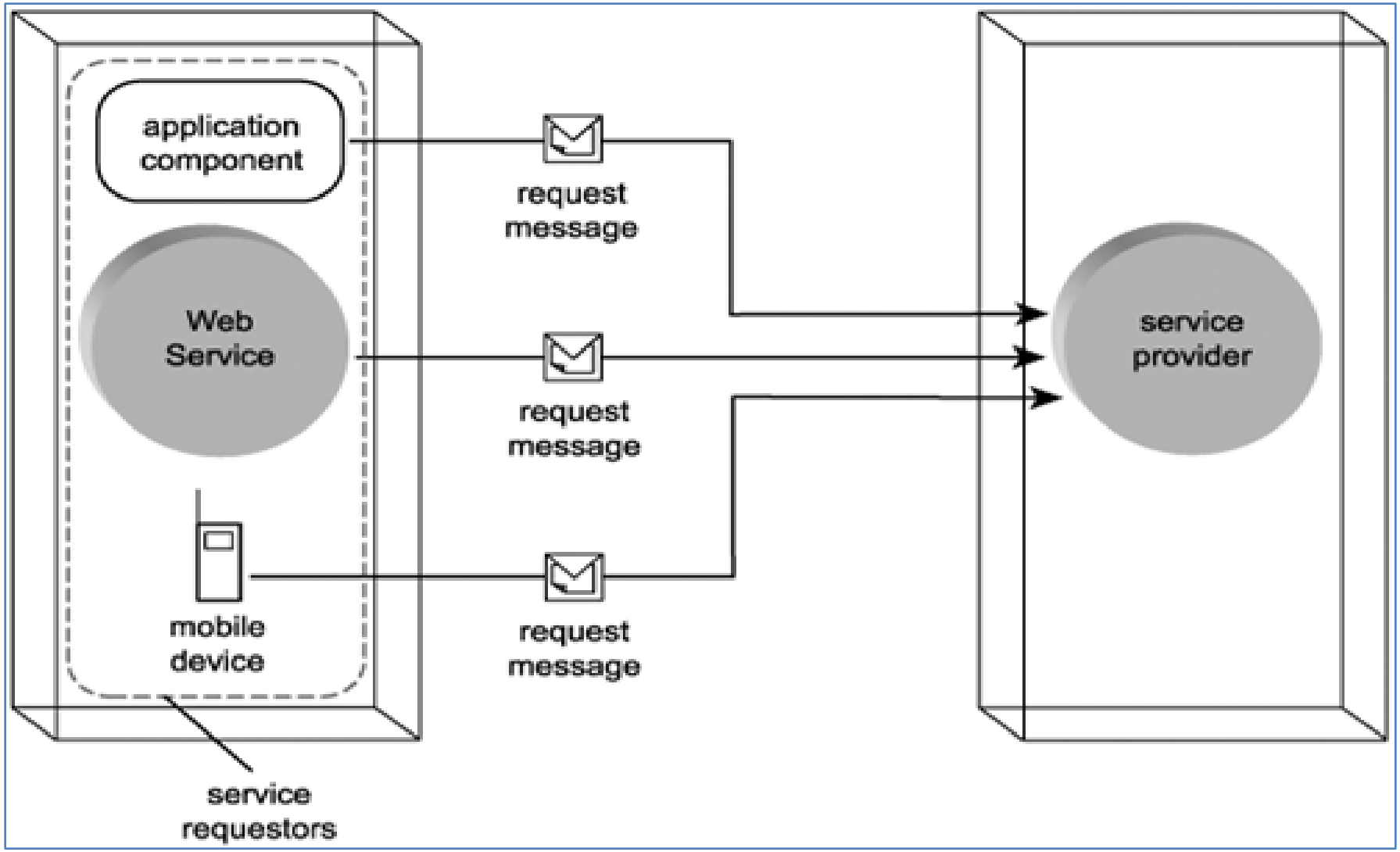
- The service provider role is synonymous with the **server role** in the classic client-server architecture.

Service Provider: Additional Terms

- **Service Provider Entity:**
  - The organization or individual providing the Web service

- **Service Provider Agent:**
  - The Web service itself, acting as an agent on behalf of its owner

Service roles – Service Requestor

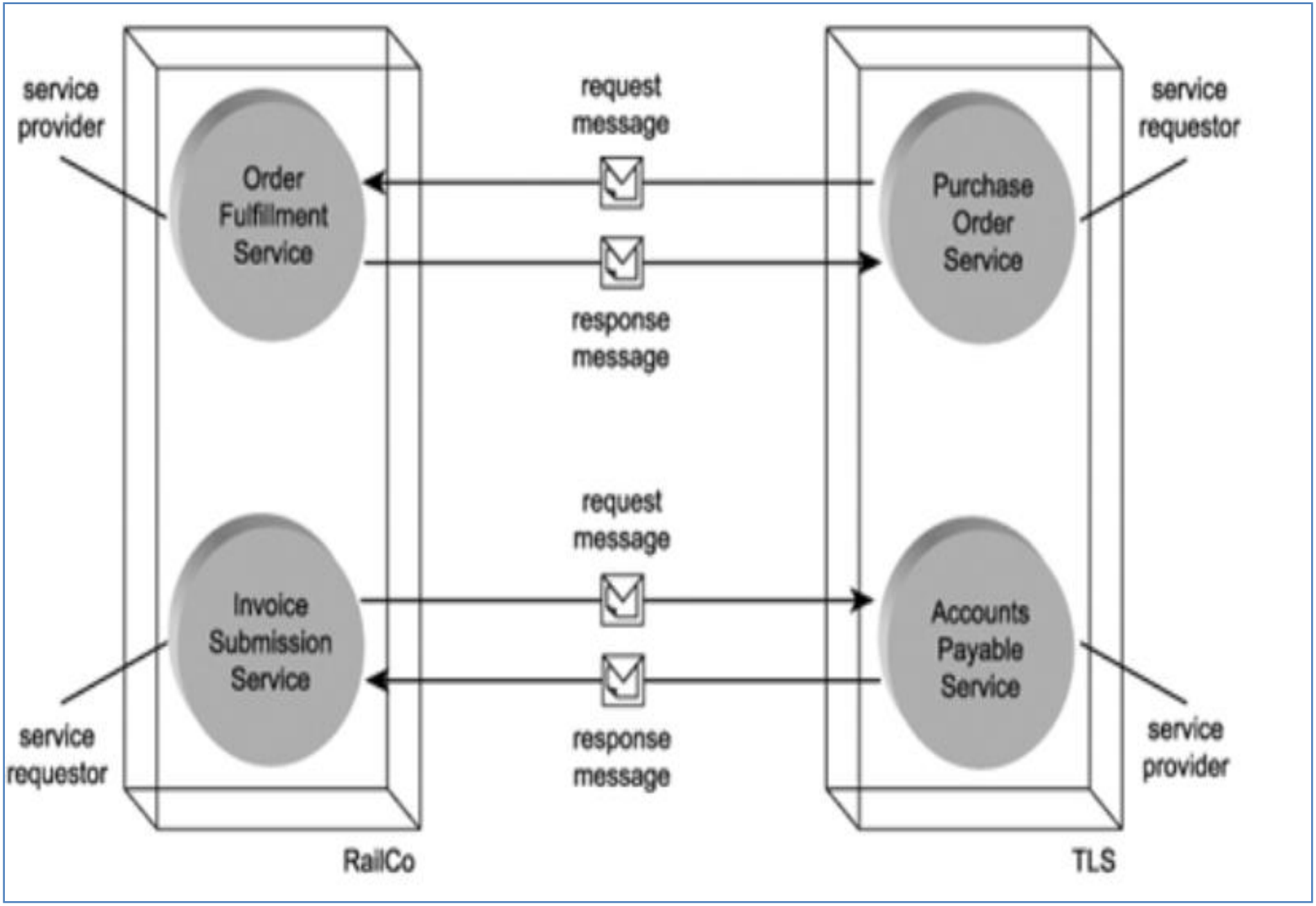


Service roles – Service Requestor

- **Requestor Role** requires following conditions to be fulfilled:
  - 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.

Service Requestor: Additional Terms

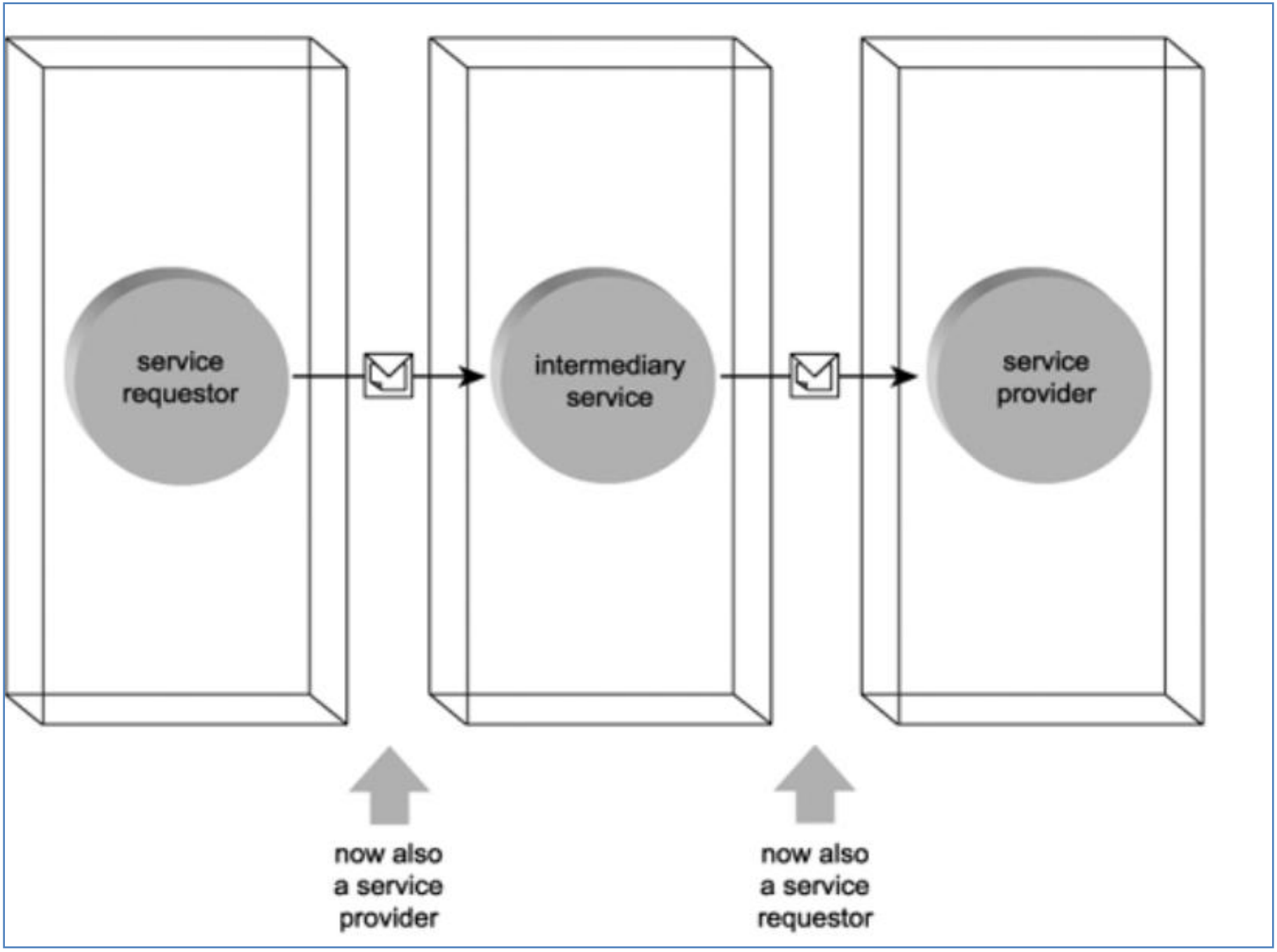
- **Service requestor entity:**
  - The organization or individual requesting the Web service
- **Service requestor agent:**
  - The Web service itself, acting as an agent on behalf of its owner


service provider
Order Fulfillment Service
request message
response message
service requestor
Purchase Order Service
request message
Invoice Submission Service
response message
service requestor
Accounts Payable Service
service provider
RailCo
TLS

Intermediaries

- Web services contrasts the predictable nature of traditional point-to-point communications channels.
- Point to point: less flexible, less scalable but easier to design
- SOA relies on Messaging paths - point-to-*

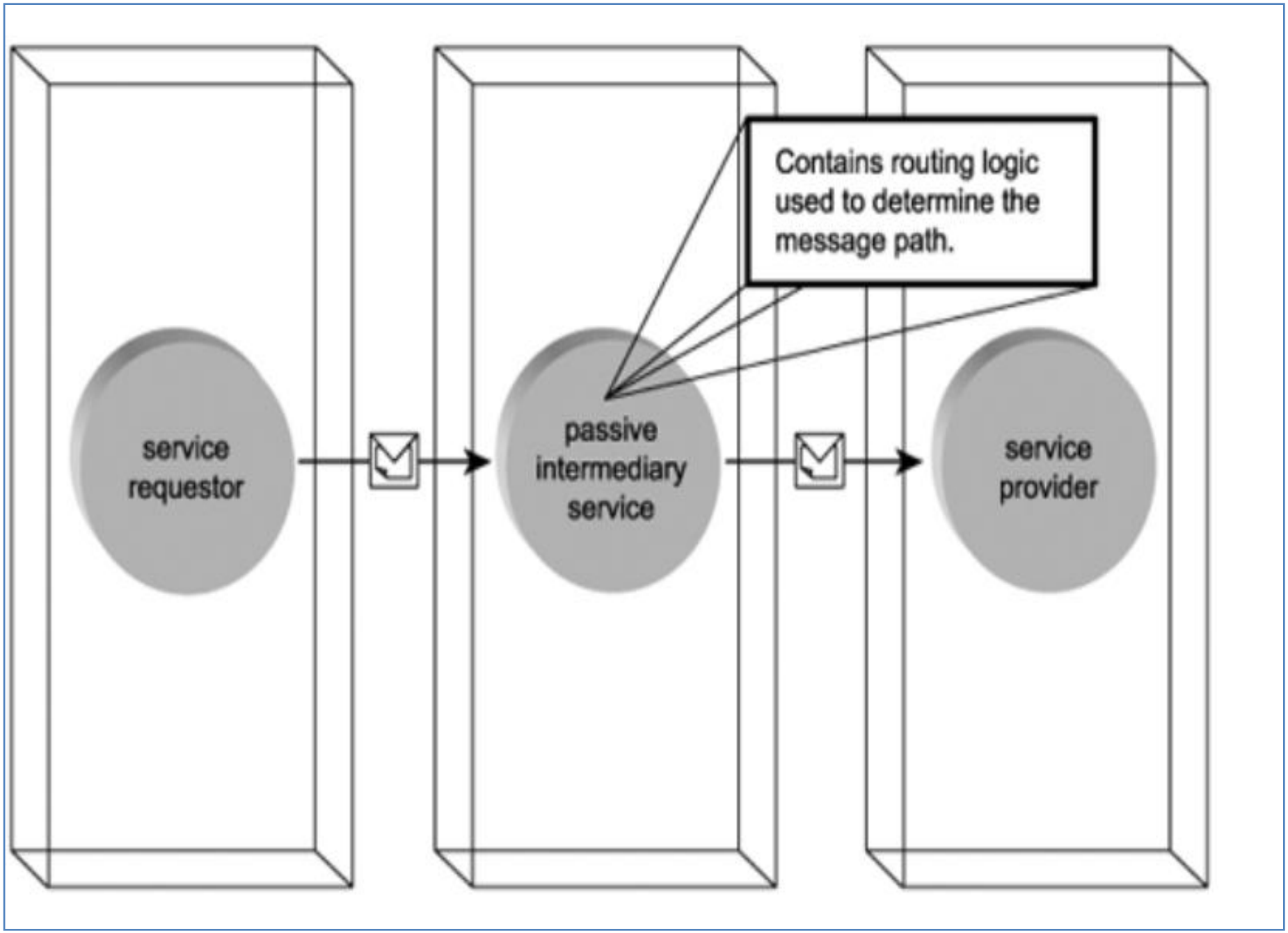
- "Web services and Agents that route and process a message after it is sent and before it arrives"

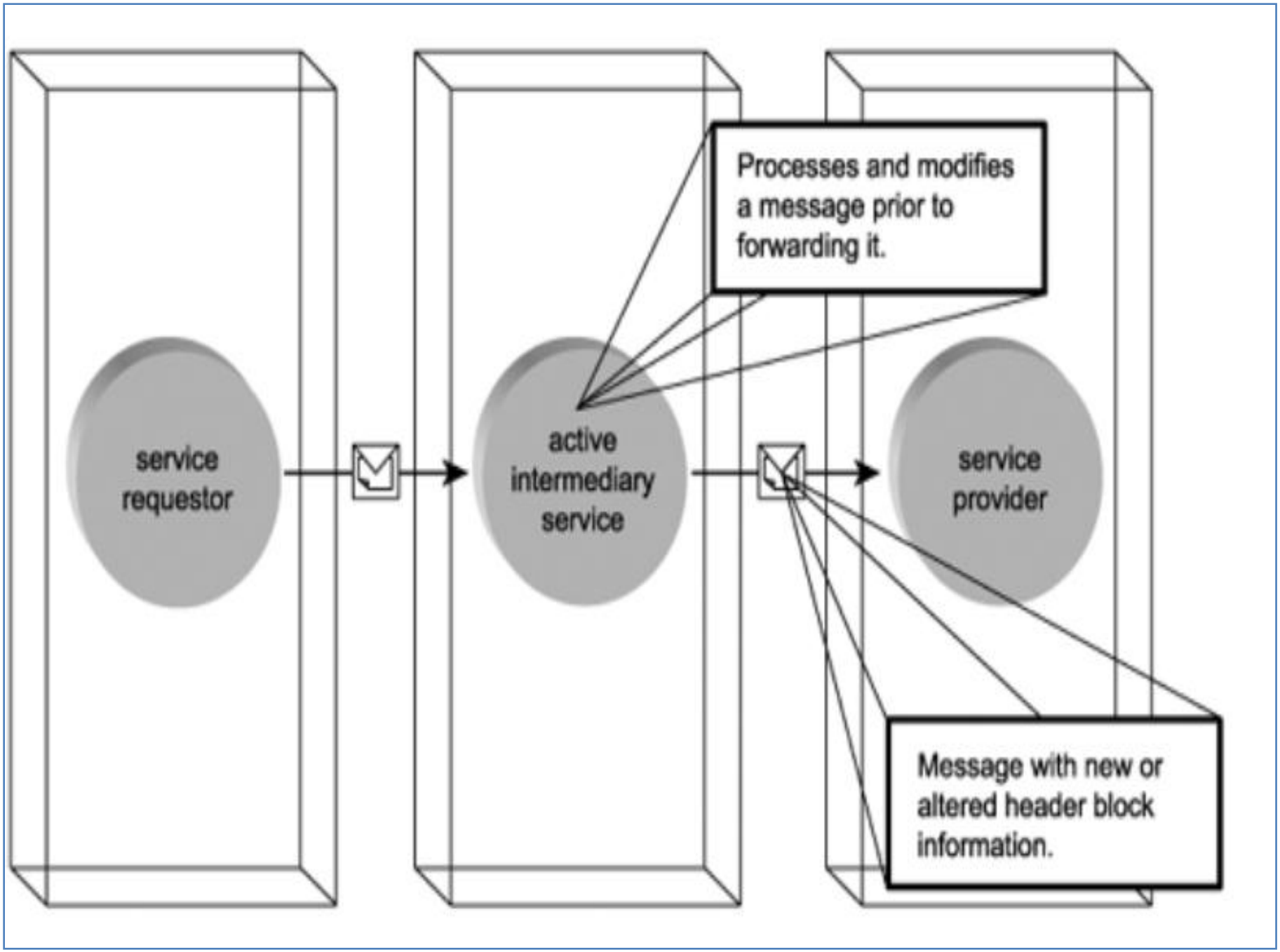

service requestor
intermediary service
service provider
now also a service provider
now also a service requestor

Types of Intermediaries

- **Passive Intermediary:**
  - Routes a message to subsequent location
  - Does not modify the message (SOAP Header)

- **Active Intermediary:**
  - Routes, processes and alters the message contents
  - Can add/remove SOAP headers

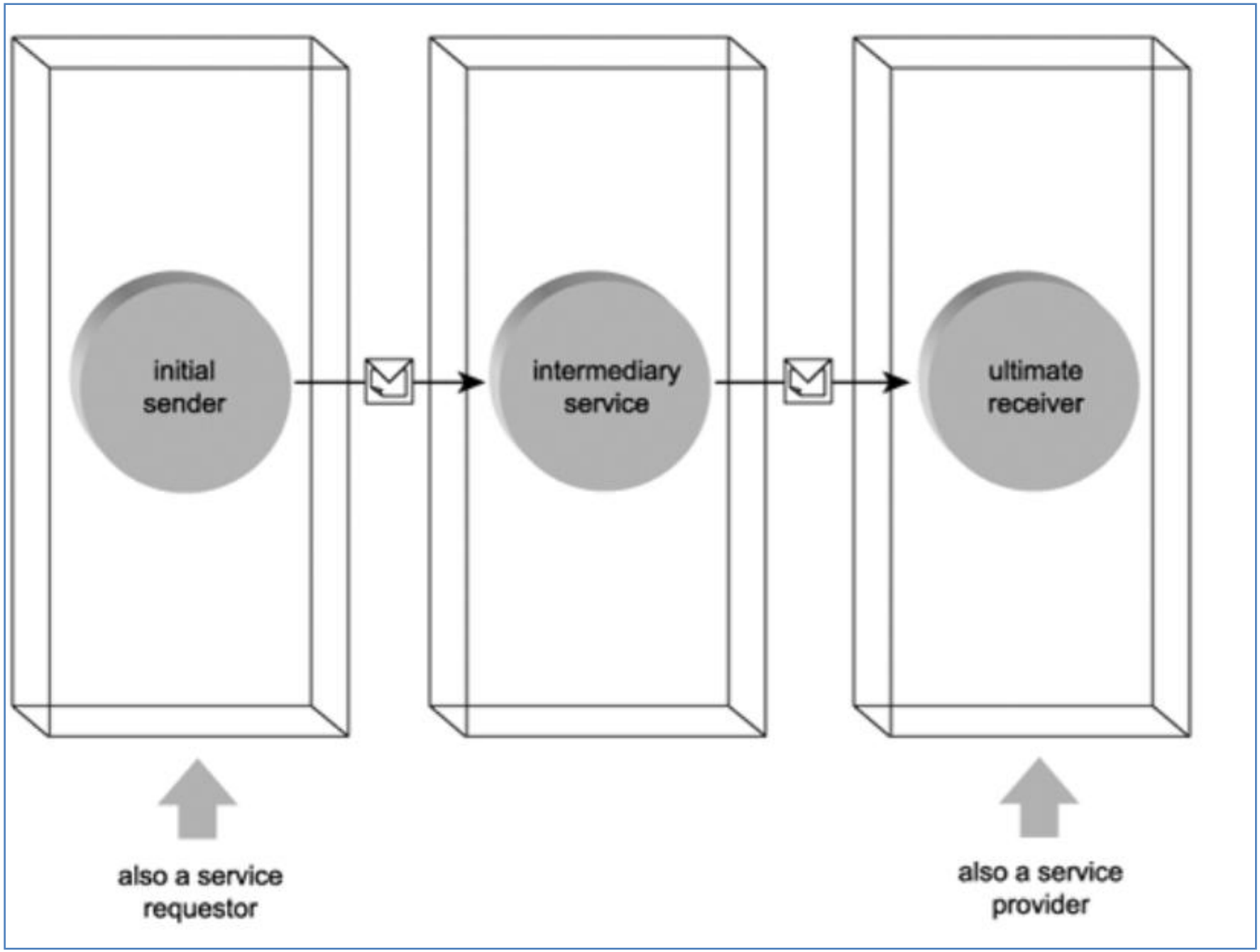

Contains routing logic used to determine the message path.
service requestor
passive intermediary service
service provider


Processes and modifies a message prior to forwarding it.
service requestor
active intermediary service
service provider
Message with new or altered header block information.

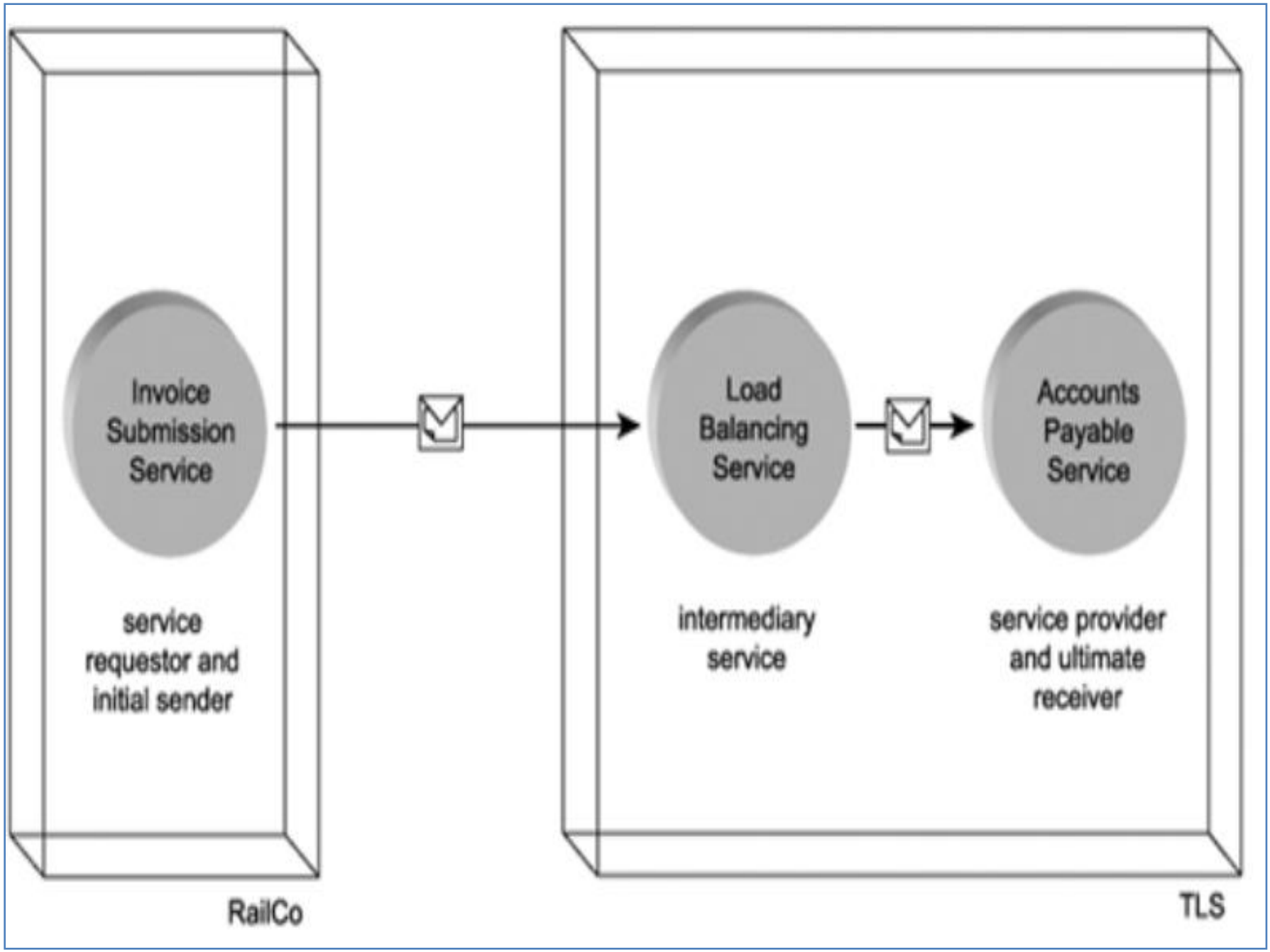
Initial Sender And Ultimate Receiver

- **Initial Sender:**
  - Requestor that initiates the transmission of a message

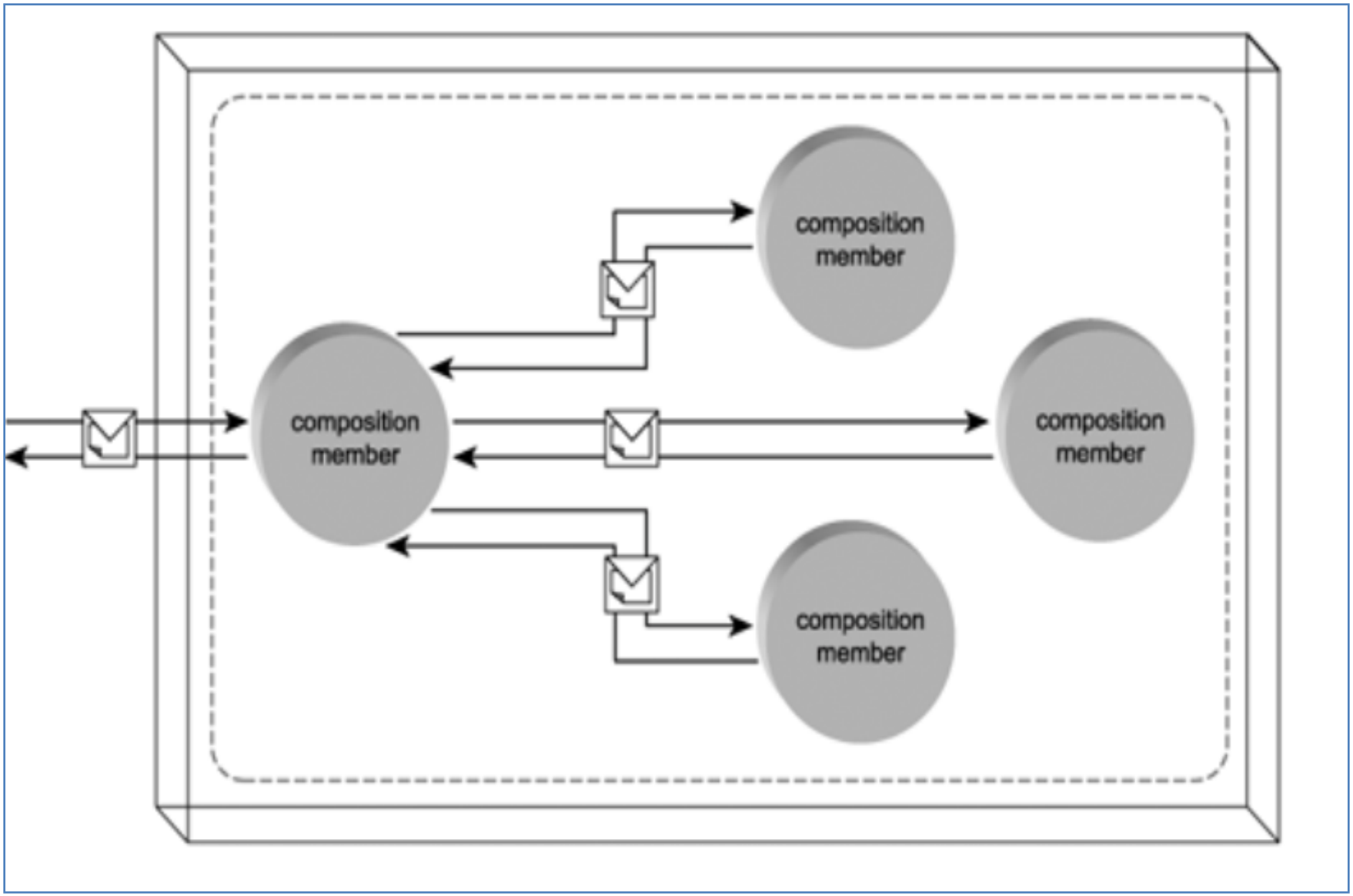
- **Ultimate Receiver:**
  - Provider that exists as the last web service on the message path

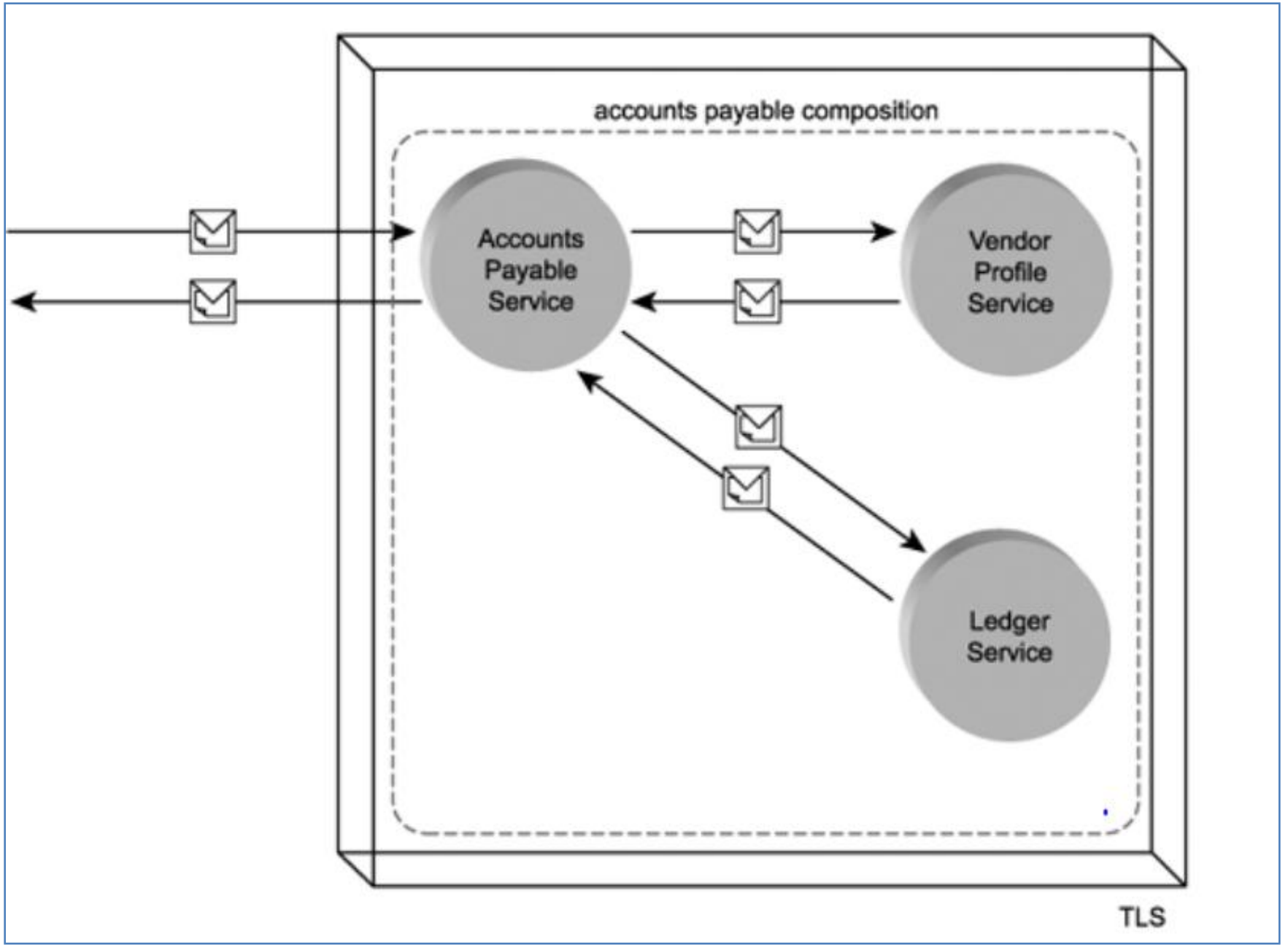

initial sender
intermediary service
ultimate receiver
also a service requestor
also a service provider

Can intermediary services ever be initial senders or ultimate receivers within the scope of a service activity?


Invoice Submission Service
Load Balancing Service
Accounts Payable Service
service requestor and initial sender
intermediary service
service provider and ultimate receiver
RailCo
TLS

Service Compositions




accounts payable composition
Accounts Payable Service
Vendor Profile Service
Ledger Service
TLS

Service Models

- Services can be classified based on the nature of the application logic

  – These classifications are known as service models.
    - Application service
    - Business service
    - Controller service
    - Utility service, etc...

Business Service Model

- Within an SOA, the business service represents the most fundamental building block.

- It is fully autonomous but still not limited to executing in isolation,
  - as business services are frequently expected to participate in service compositions.

Business Service Model

- Business services are used within SOAs as follows:
  - to represent **business process logic**
  - as service **composition members**

Utility Service Model

- A service that offers **reusable logic**.

- This category is primarily intended for the classification of **solution-agnostic** application service.

Utility Service Model

- 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**

Classify the Following

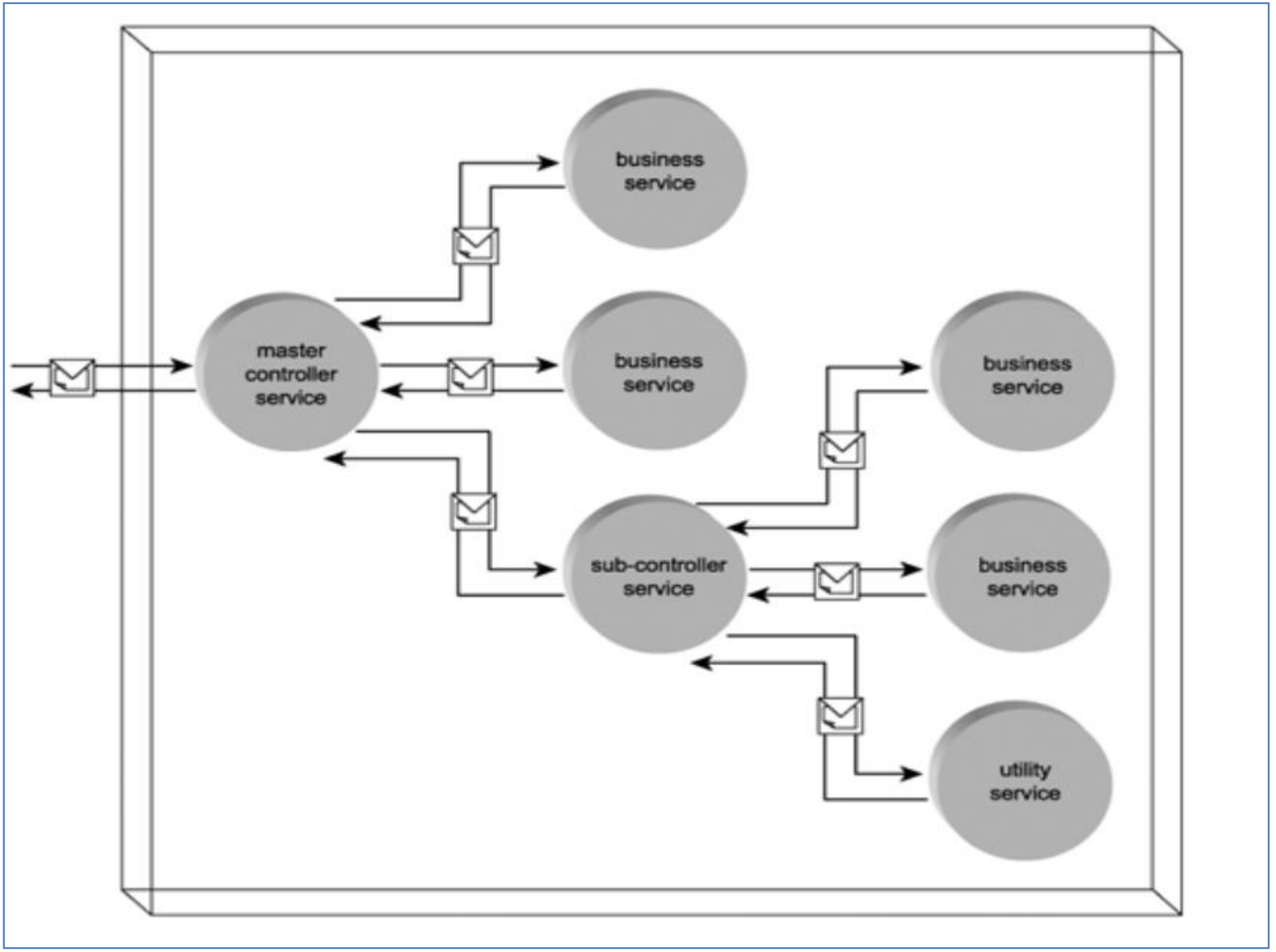
- Accounts Payable Service
- Internal Policy Service
- Invoice Submission Service
- Ledger Service
- Load Balancing Service
- Order Fulfillment Service
- Purchase Order Service
- Vendor Profile Service

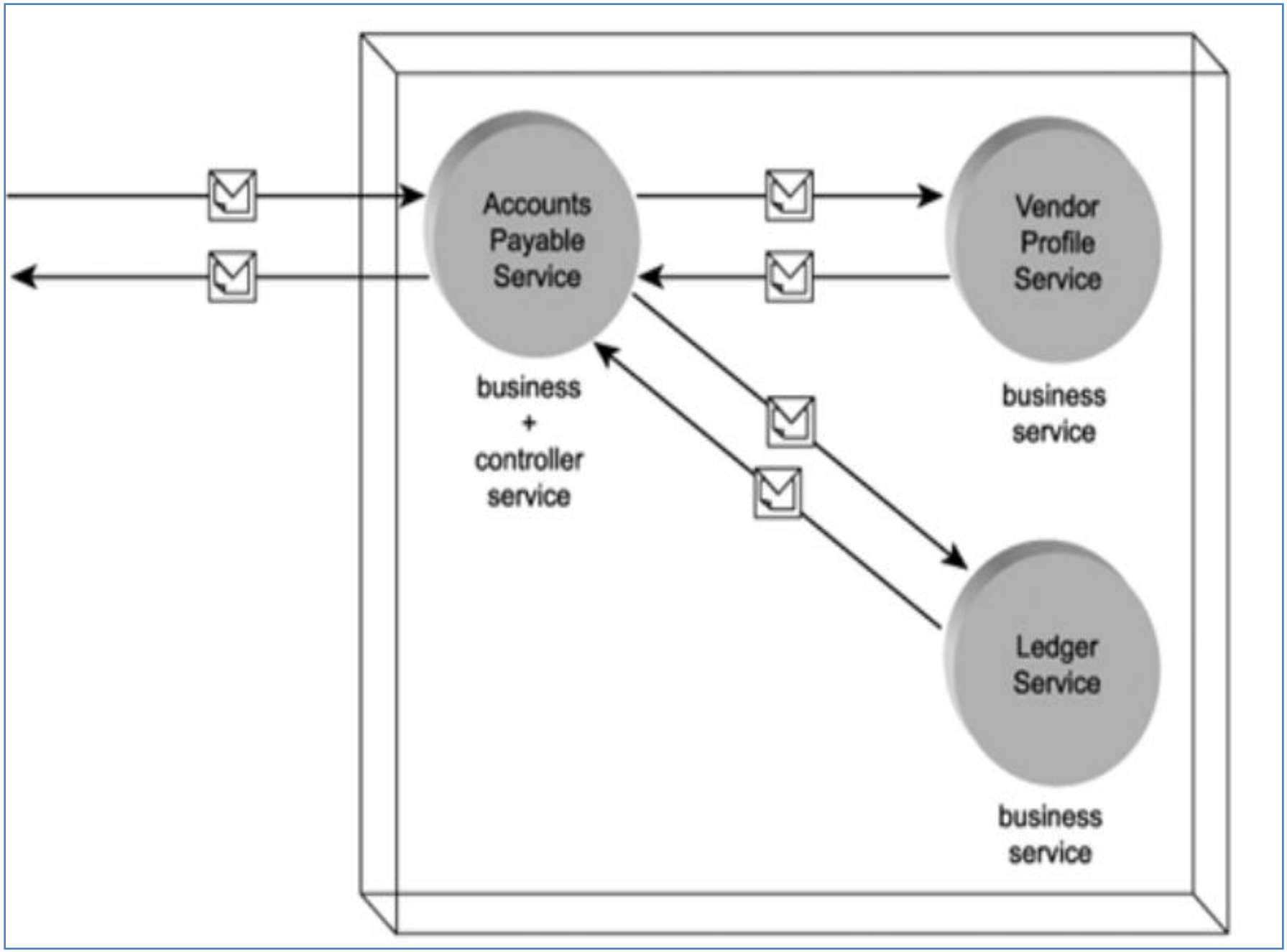
Controller Service Model

- A service that **composes** others.

- Controller services are used within SOAs as follows:
  - to support and implement the **principle of composability**
  - to leverage **reuse** opportunities
  - to support **autonomy** in other services

- Master controller and sub-controller


business service
master controller service
business service
business service
sub-controller service
business service
utility service

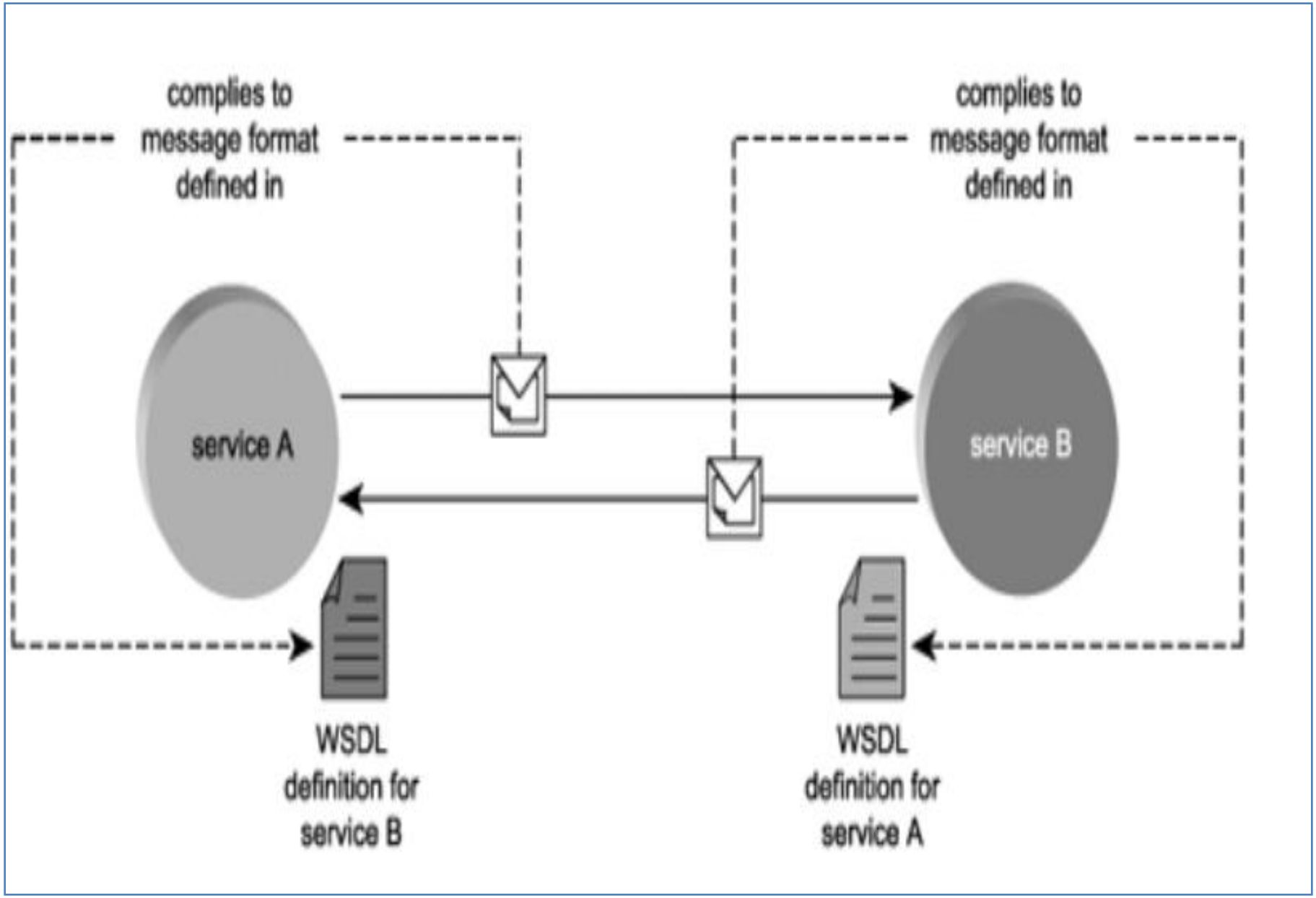

Accounts
Payable
Service
business
+
controller
service
Vendor
Profile
Service
business
service
Ledger
Service
business
service

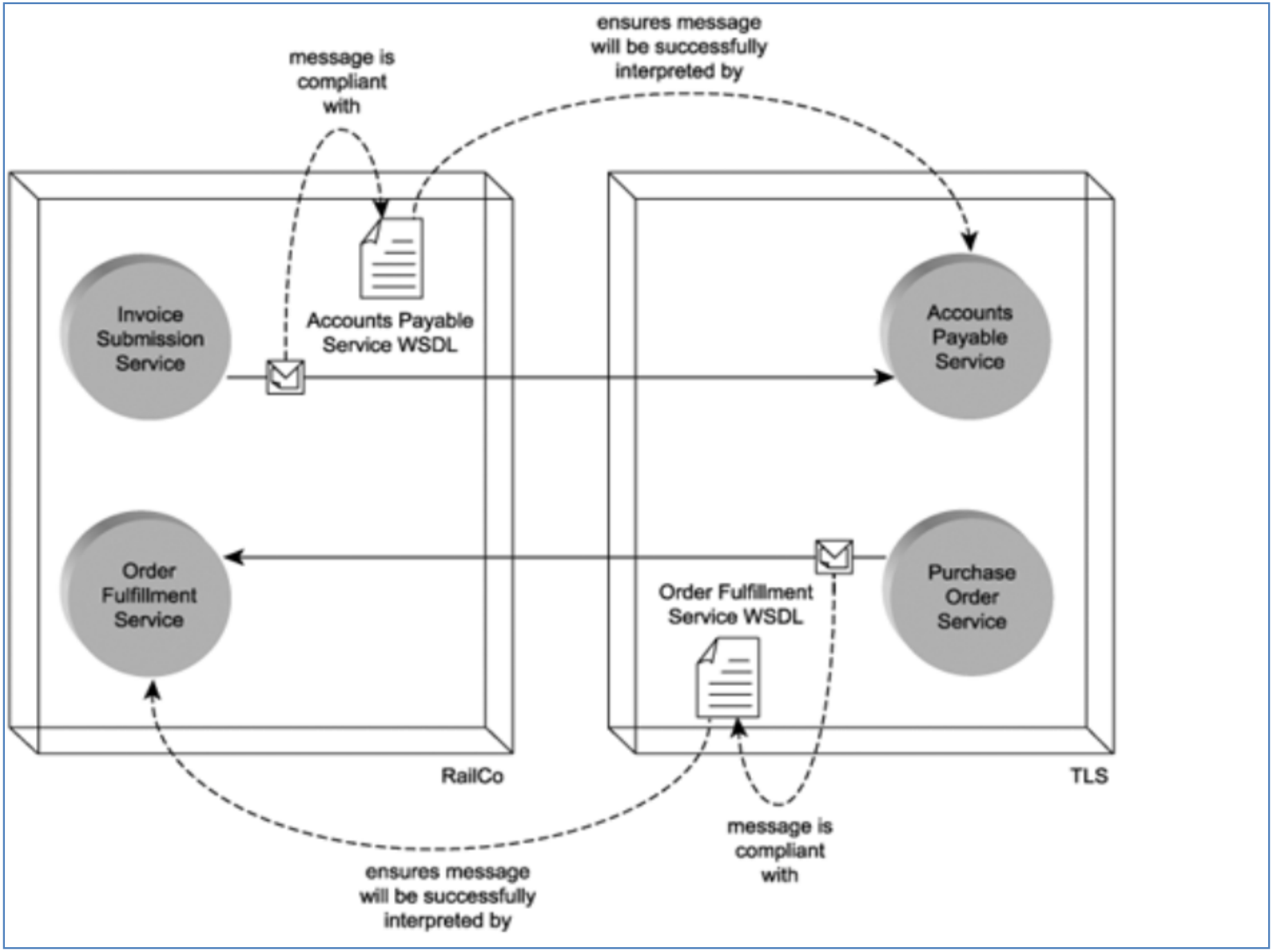
Service roles and Service models

- Web services can be labeled using **temporary and permanent** classifications.
- **Temporary classifications** relate to roles assumed by a service at runtime.
  - For example, an intermediary service can transition through different roles while processing a message.
- Service models refer to **permanent classifications** that represent the logic housed by the service, as well as its role within the overall solution.
  - "A service can belong to more than one service model."

Service Descriptions (with WSDL)

- **What is the importance of service descriptions?**
  - Descriptions are needed to ensure loose coupling among services.
  - Description documents are required to accompany any service wanting to act as an ultimate receiver.


complies to
message format
defined in
complies to
message format
defined in
service A
service B
WSDL
definition for
service B
WSDL
definition for
service A


ensures message
will be successfully
interpreted by
message is
compliant
with
Invoice
Submission
Service
Accounts Payable
Service WSDL
Accounts
Payable
Service
Order
Fulfillment
Service
Order Fulfillment
Service WSDL
Purchase
Order
Service
RailCo
TLS
message is
compliant
with
ensures message
will be successfully
interpreted by

What is Legacy system adapter?

What is WSDL?

What is UDDI?

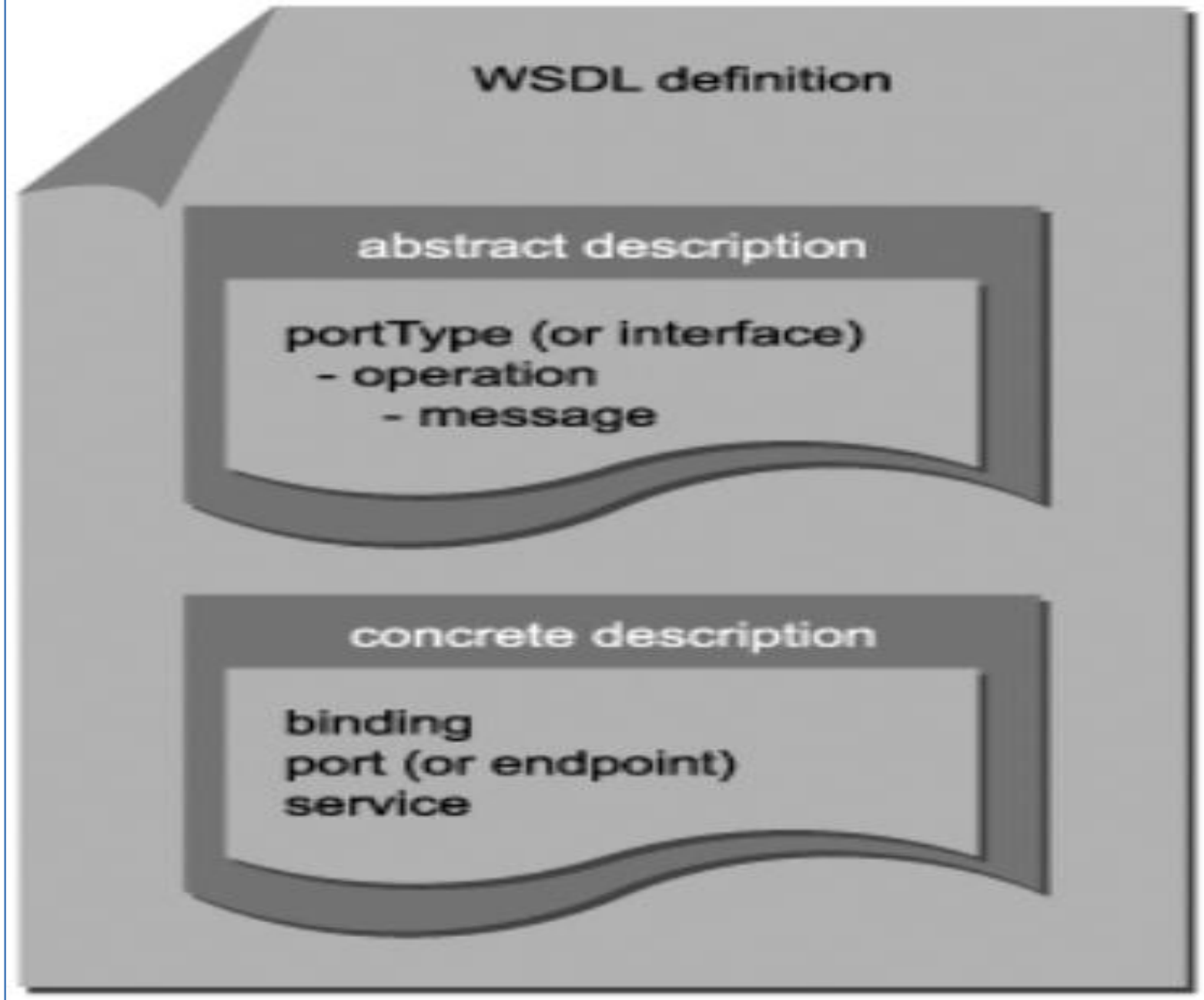
What is QoS of service?

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.

WSDL Organization

- A WSDL service description can be separated into two parts:
  - ***Abstract*** description
  - ***Concrete*** description


WSDL definition
abstract description
portType (or interface)
- operation
- message
concrete description
binding
port (or endpoint)
service

Abstract description

- “An abstract description establishes the interface characteristics of the Web service”
  - without any reference to the technology used to host the web service

- Consists of three elements:
  - portType
    - Operation
      - Message

Abstract description

- **portType:**
  - The portType section of an abstract description provides a high-level view of the service interface
  - “Sorts the messages a service can process into groups of functions, known as operations.”

Abstract description

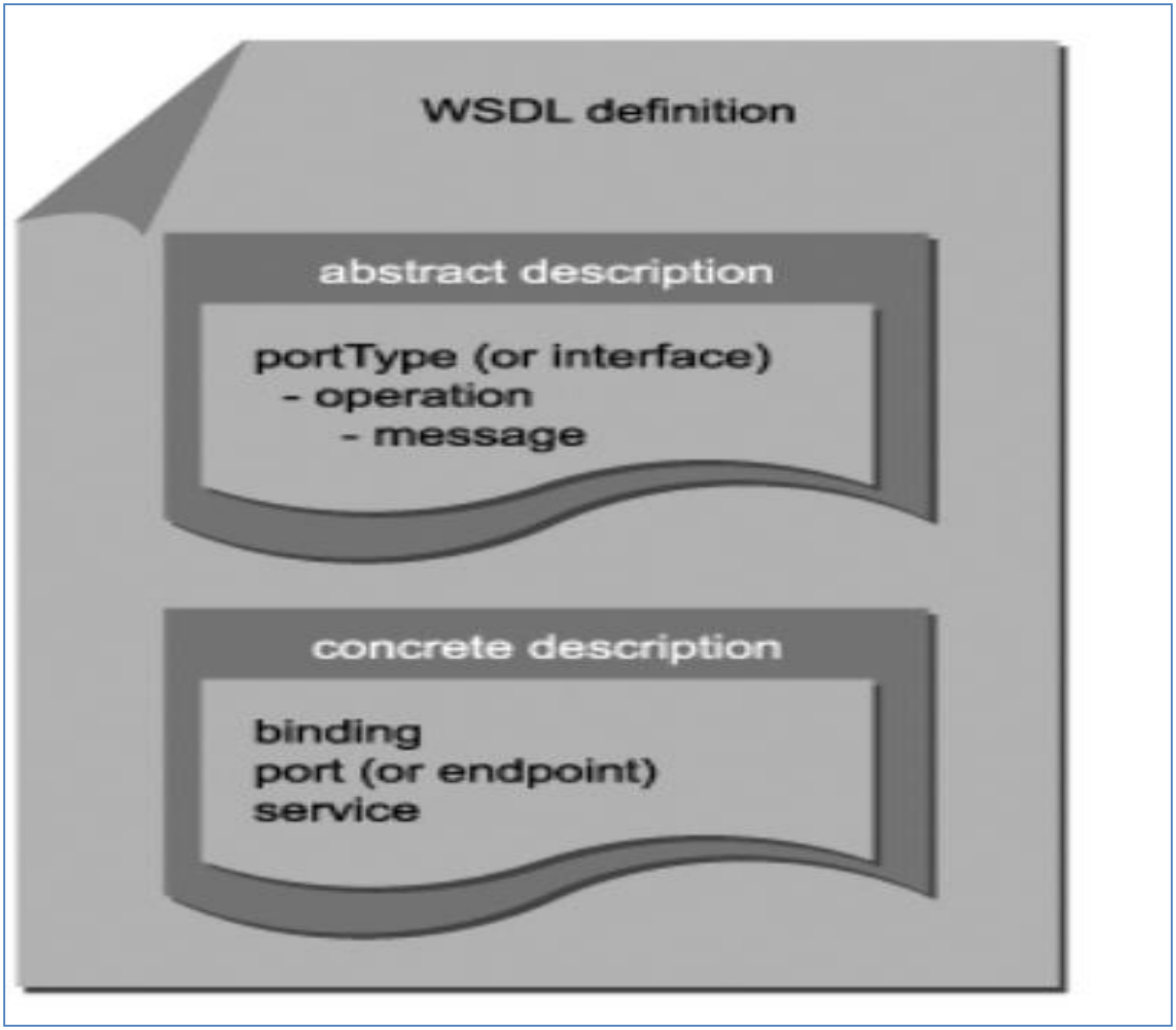
- **Operation:**
  - "Each operation represents a specific action performed by the service."
  - A service operation is comparable to a public method used by components in traditional distributed applications.
  - Much like component methods, operations also have input and output parameters.

Abstract description

- **Message:**
  - Web services rely exclusively on messaging-based communication, parameters are represented as messages.
  - Therefore, an operation consists of a set of input and output messages.

Concrete description

- For a Web service to be able to execute any of its logic,
  - it needs for its abstract interface definition to be connected to some real, implemented technology.
  - This connection is defined in the concrete description portion of the WSDL


WSDL definition
abstract description
portType (or interface)
- operation
- message
concrete description
binding
port (or endpoint)
service

Concrete description

- **Binding:**
  - A WSDL description's binding describes the requirements for
    - a service to establish physical connections with this service

Concrete description

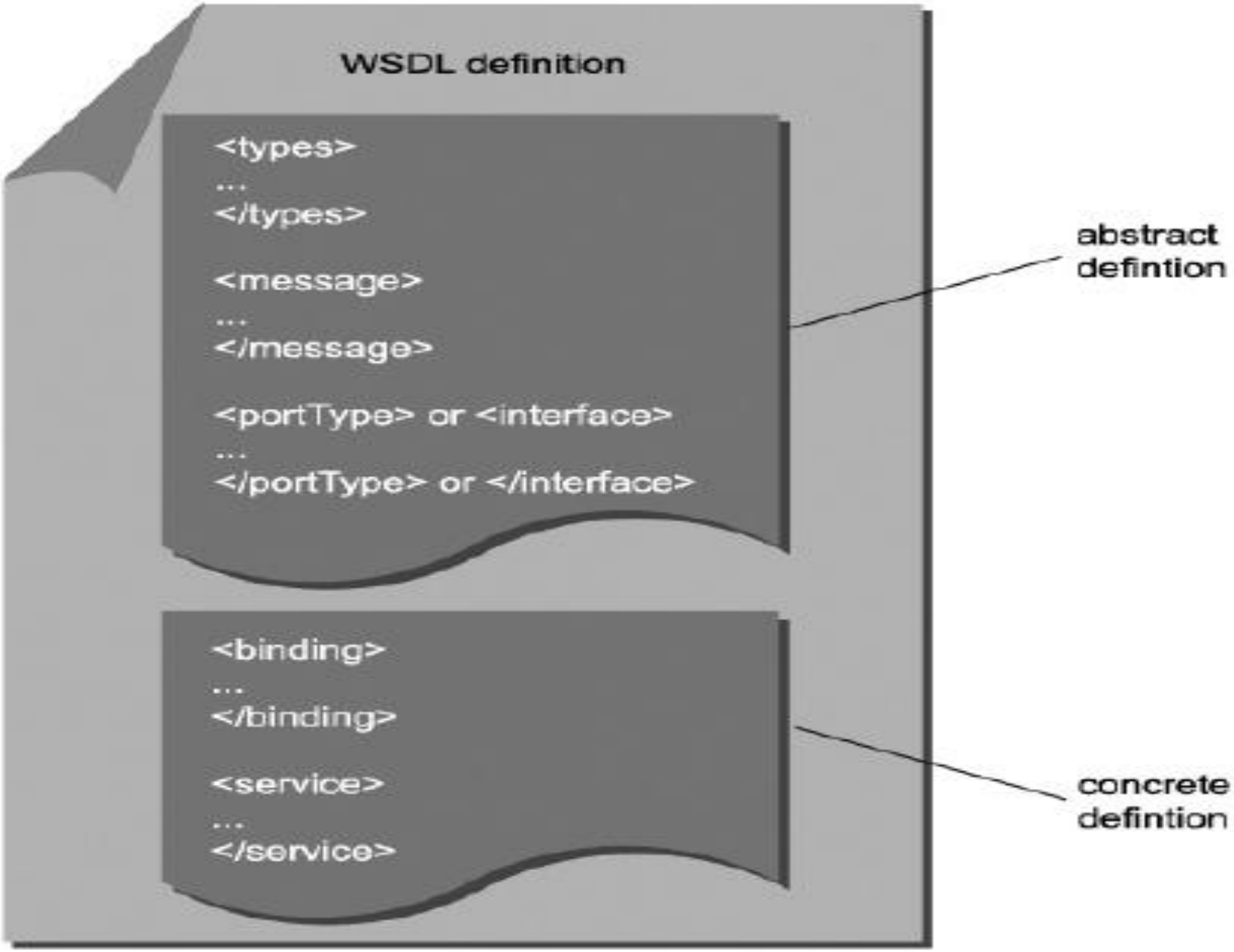
- **Port:**
  - It represents the physical address at which a service can be accessed with a specific protocol.

- **Service:**
  - Within the WSDL language, the term service is used to refer to a group of related endpoints.

Example

- For the Abstract Description, Accounts Service supports following operation:
  - SubmitInvoice
- This operation contains following messages:
  - AcceptInvoice (Input Message)
  - SendAck (Output Message)

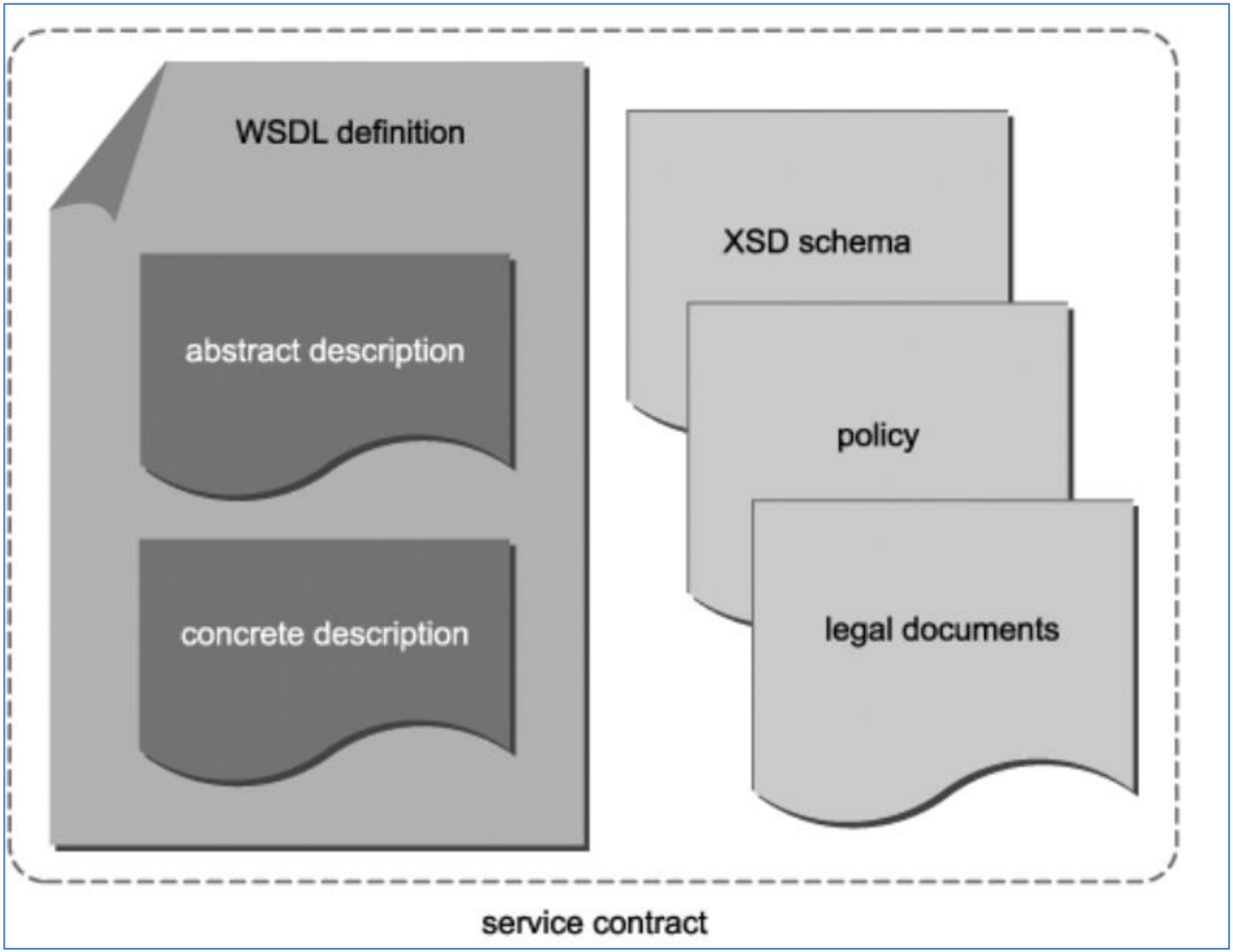
- Concrete description for the service binds above mentioned operation to SOAP and also provides the address of accounts service.


WSDL definition
<types>
...
</types>
<message>
...
</message>
<portType> or <interface>
...
</portType> or </interface>
<binding>
...
</binding>
<service>
...
</service>
abstract
defintion
concrete
defintion

Metadata and Service Contract

- **Service Metadata**
  - WSDL definition
  - XSD schema
  - Policy

- **Service Contract**
  - Above Three + Legal Documents


WSDL definition
abstract description
concrete description
XSD schema
policy
legal documents
service contract

Semantic Description

- Service's behavioral characteristics

- Examples of service semantics include:
  - how a service behaves under certain conditions
  - how a service will respond to a specific condition
  - what specific tasks the service is most suited for