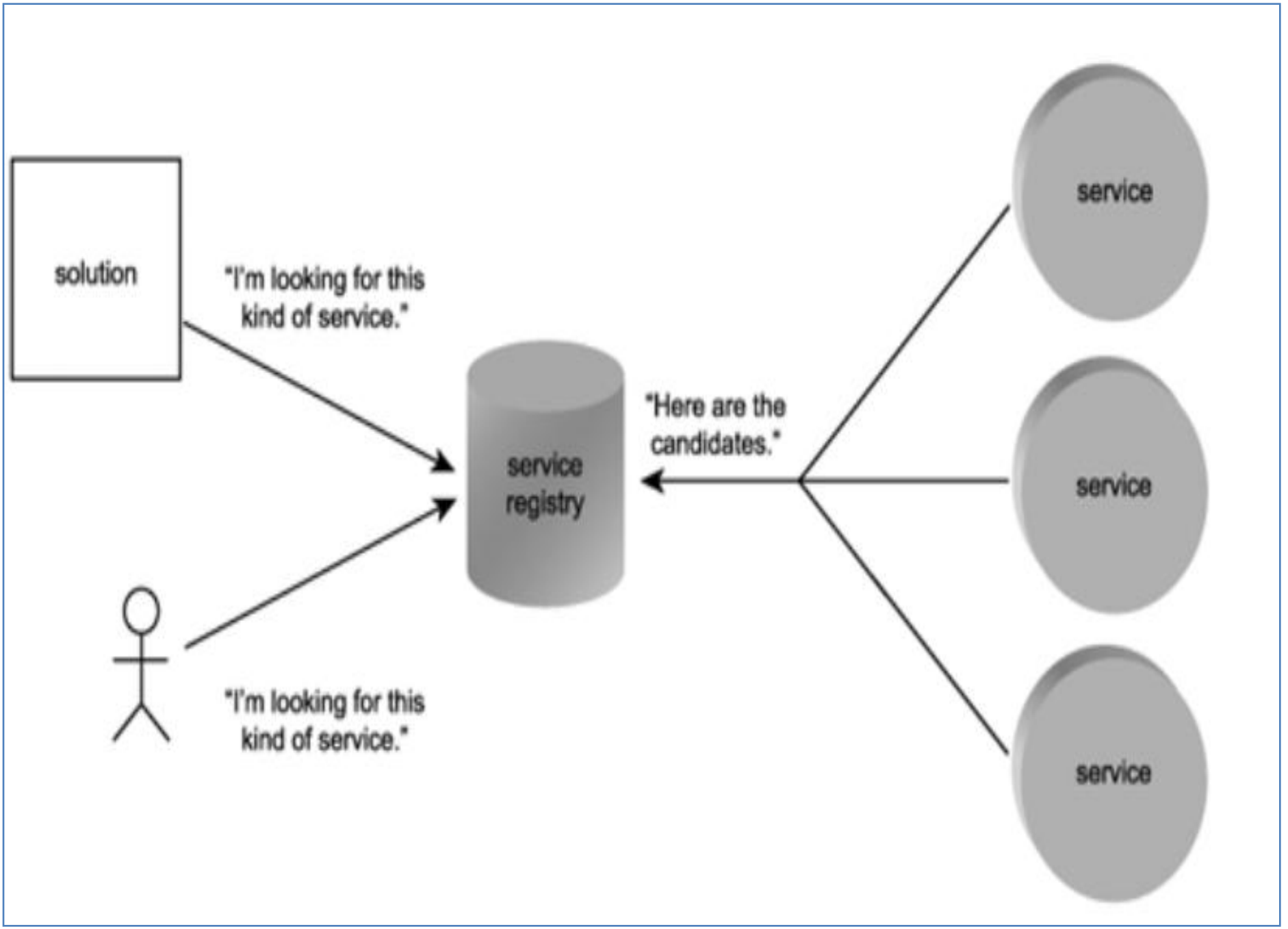
Service description advertisement and discovery

- How to locate the latest versions of known service descriptions?
- How to discover new Web services that meet certain criteria?

  – These requirements can be achieved using Central Directories and Registries


solution
"I'm looking for this kind of service."
service registry
"Here are the candidates."
service
service
service
"I'm looking for this kind of service."

UDDI as Registry Model

- UDDI has two types of registries:

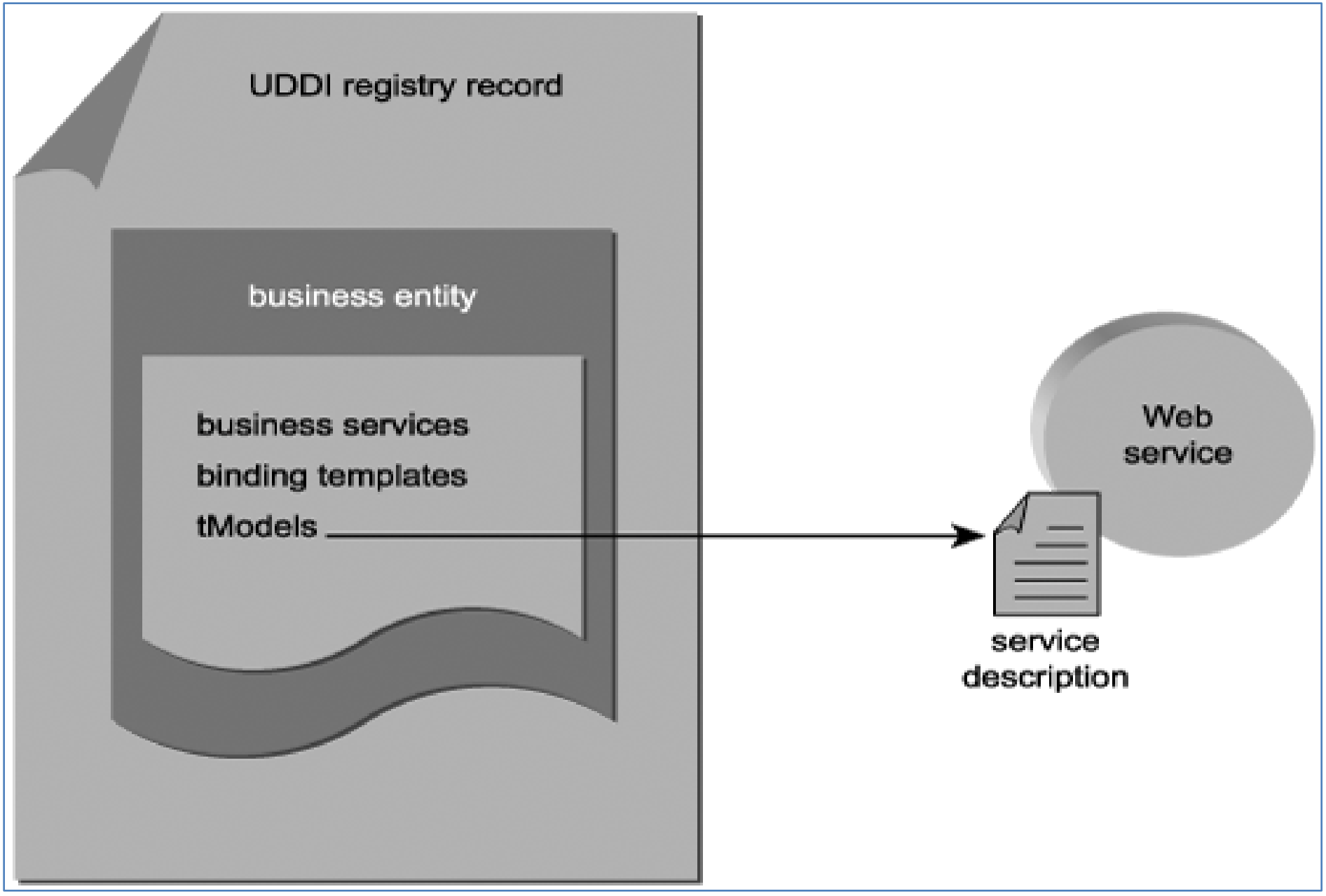
- **Public Registry:**
  - Accept registrations from any organization, regardless of whether they have any web services to offer
  - Organizations acting as Service Providers can register their services after signing up

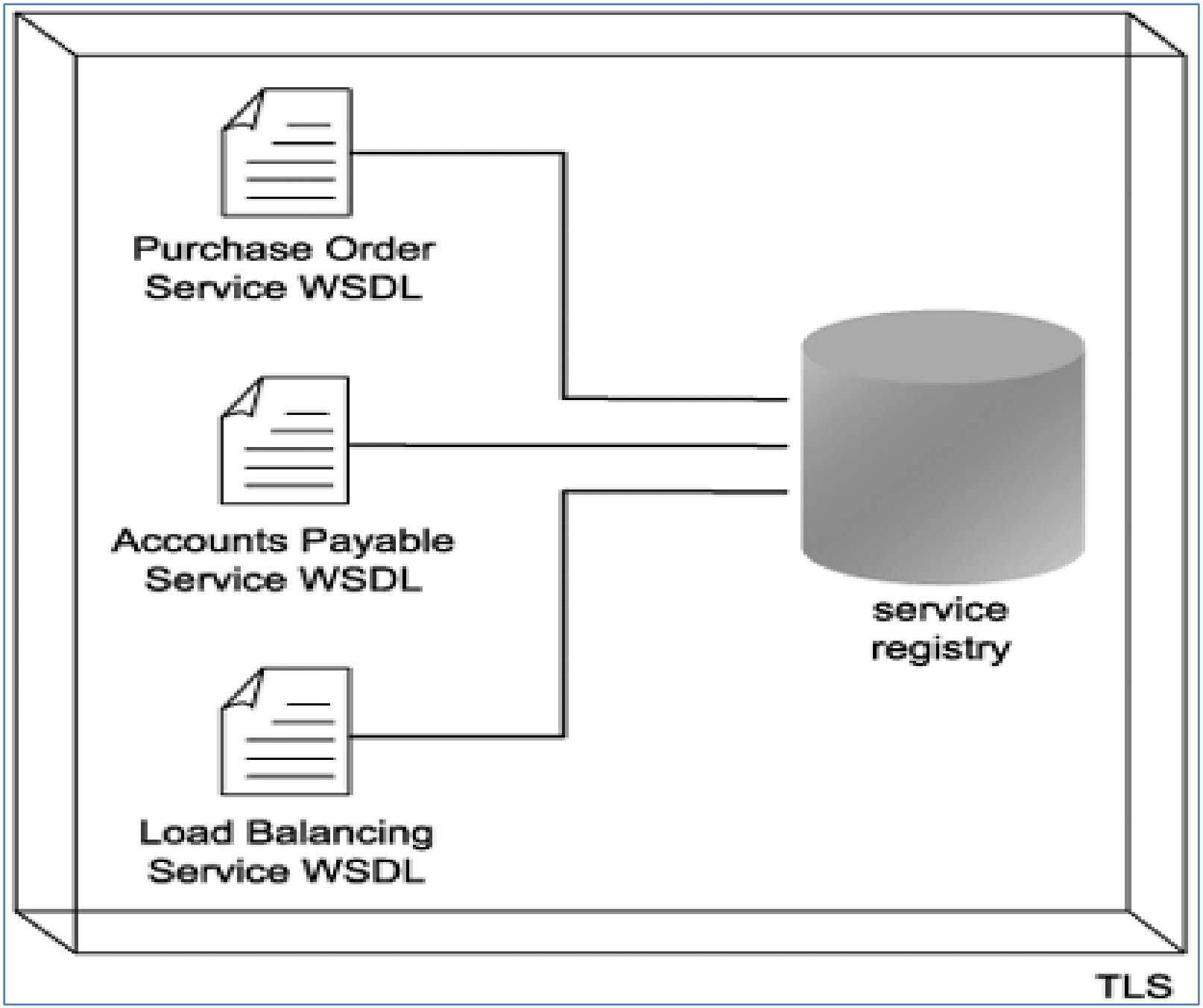
UDDI as Registry Model

- **Private Registry:**
  - Implemented within the organization boundary for storage of all service the organization develops, leases or purchases

Parts of UDDI Record

- **Business entities and business services**
  - Each public registry record consists of a business entity with basic information regarding the organization
  - It includes business service areas with services offered by the business entity
- **Binding templates and tModels**
  - Binding information is stored separately
  - Includes two aspects
    - May point to website address
    - May point to actual WSDL


UDDI registry record
business entity
business services
binding templates
tModels
Web
service
service
description


Purchase Order
Service WSDL
Accounts Payable
Service WSDL
Load Balancing
Service WSDL
service
registry
TLS

So far...

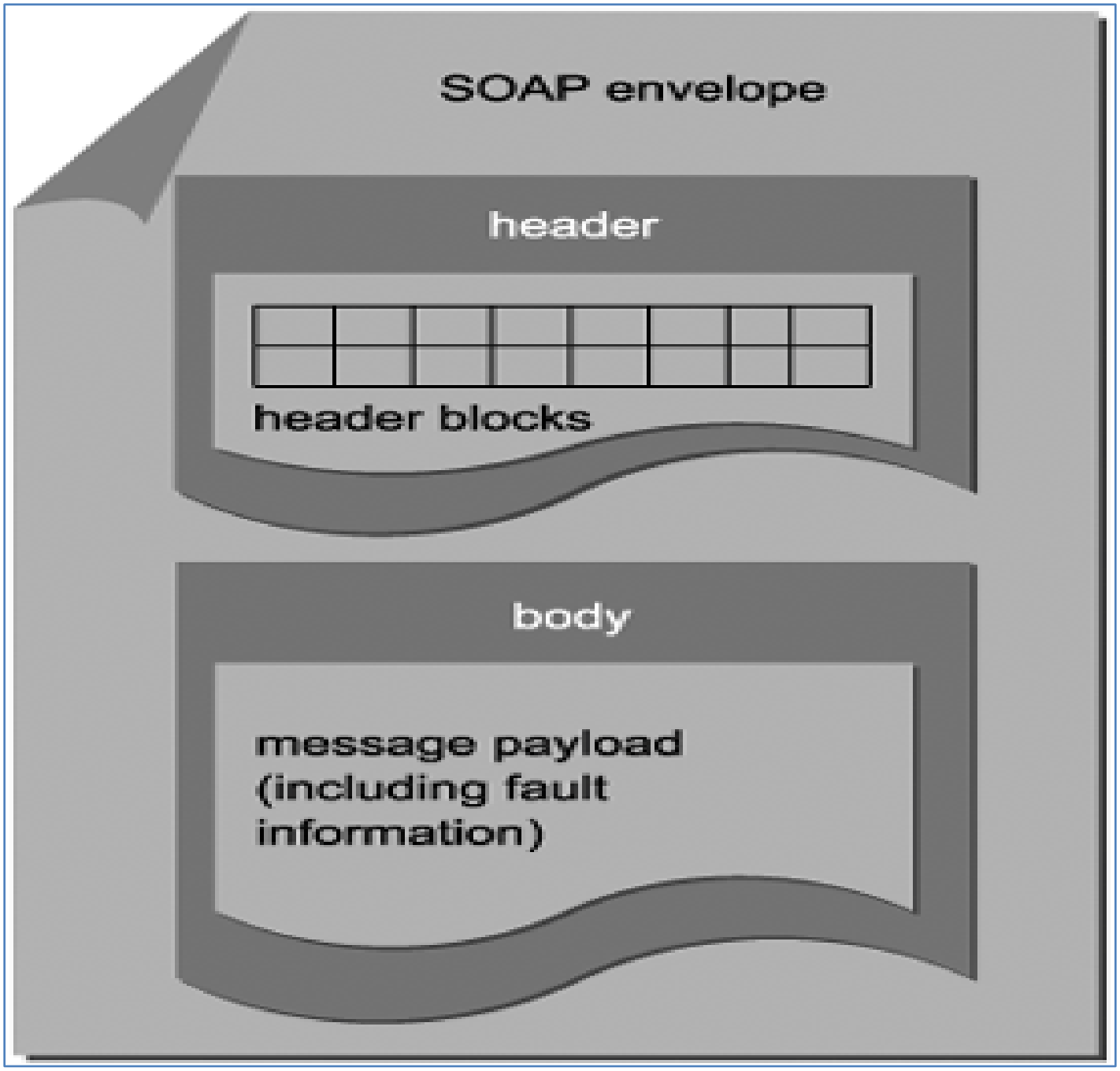
- The WSDL definition is divided into two parts: the abstract description that defines the service interface, and the concrete description that establishes the transport and location information.

- Service descriptions can be dynamically discovered by humans or applications, using private or public service registries.

Messaging with SOAP

- All services must follow **same format and protocol** for communication
- Business and Application logic can be embedded in messages
  - Hence they must be flexible and extensible

- SOAP is the standard transport protocol for messages processed by web services


SOAP envelope
header
header blocks
body
message payload
(including fault
information)

SOAP Message

- **Envelope:**
  - Every SOAP message is packaged into a container called envelope
  - It includes all parts of the message
- **Header:**
  - An area that stores meta information
  - Optional, but rarely omitted
- **Body:**
  - Contains XML formatted data
  - Named as message payload

SOAP - Header blocks

- Message independence is implemented through the use of header blocks
- Header blocks outfit a message with all of the information required for any services.
- This alleviates services from having to store and maintain message-specific logic.

Inside SOAP header blocks…

- Routing Information
- Processing Information (For Intermediary or receiver)
- Rules for delivery to ensure reliability
- Instructions and Security Measures
- Context Management Information
- Correlation Information (For Request – Reply)

SOAP Message - Advantage

- Services do not need to store message specific logic
- Services can be designed as generic

SOAP Message - Example

- Invoice sent via SOAP message should contain the following information:
  - Correlation Information with a value of date and time of message transmission (Relates original order with the response)
  - Security Information necessary to access the TLS B2B System

SOAP Attachments

- These techniques allow to handle data in its native format with the SOAP message
  - Can be used for binary files (images)

- If a document with signature is needed, it can be attached with the SOAP message

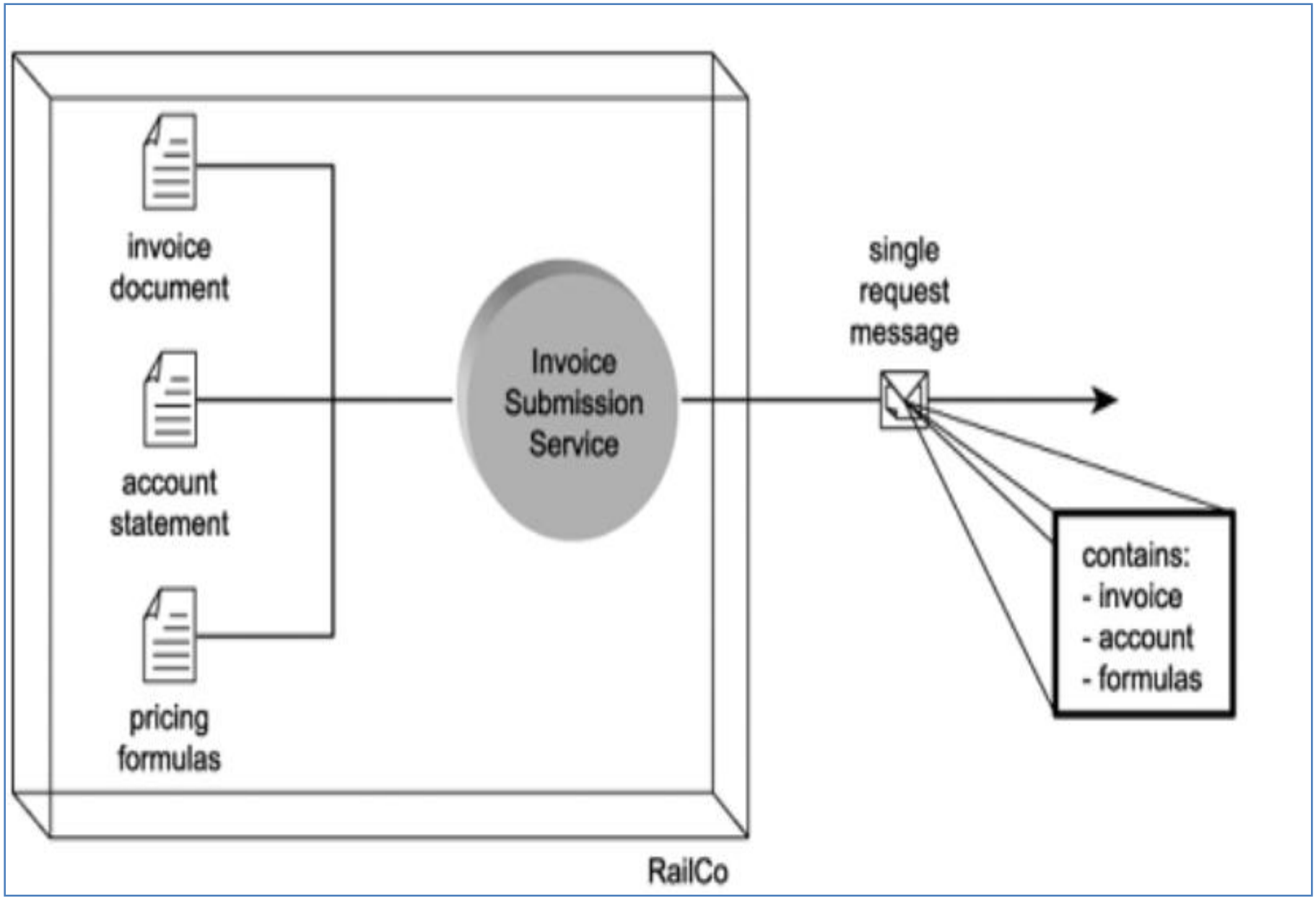
SOAP Faults

- SOAP messages can contain exception handling logic in fault section
  - As a part of SOAP Body

- Can be used to send response in case of failure

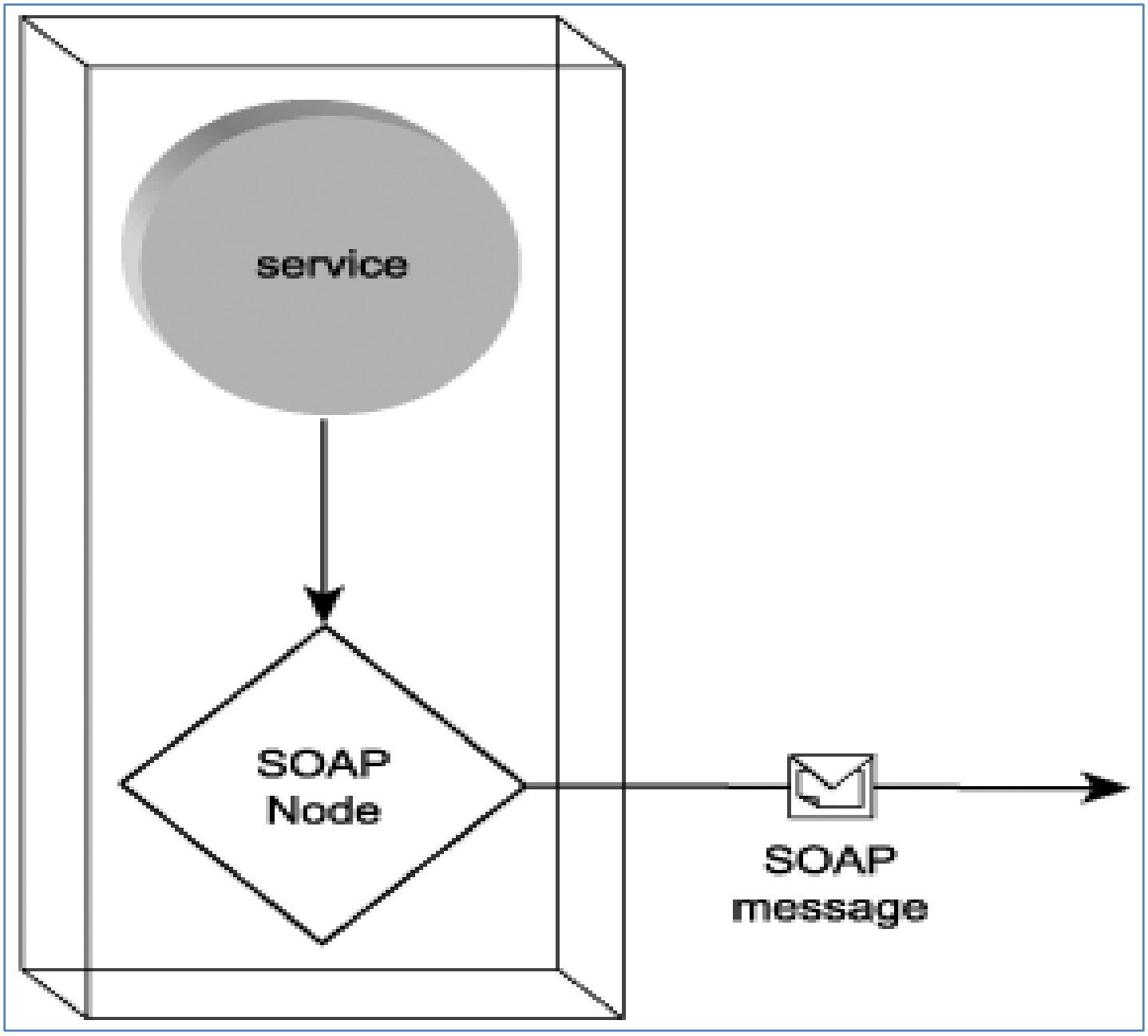
Message styles

- SOA relies on document-style messages to enable larger payloads, coarser interface operations, and reduced message transmission volumes between services.


invoice
document
account
statement
pricing
formulas
Invoice
Submission
Service
single
request
message
contains:
- invoice
- account
- formulas
RailCo

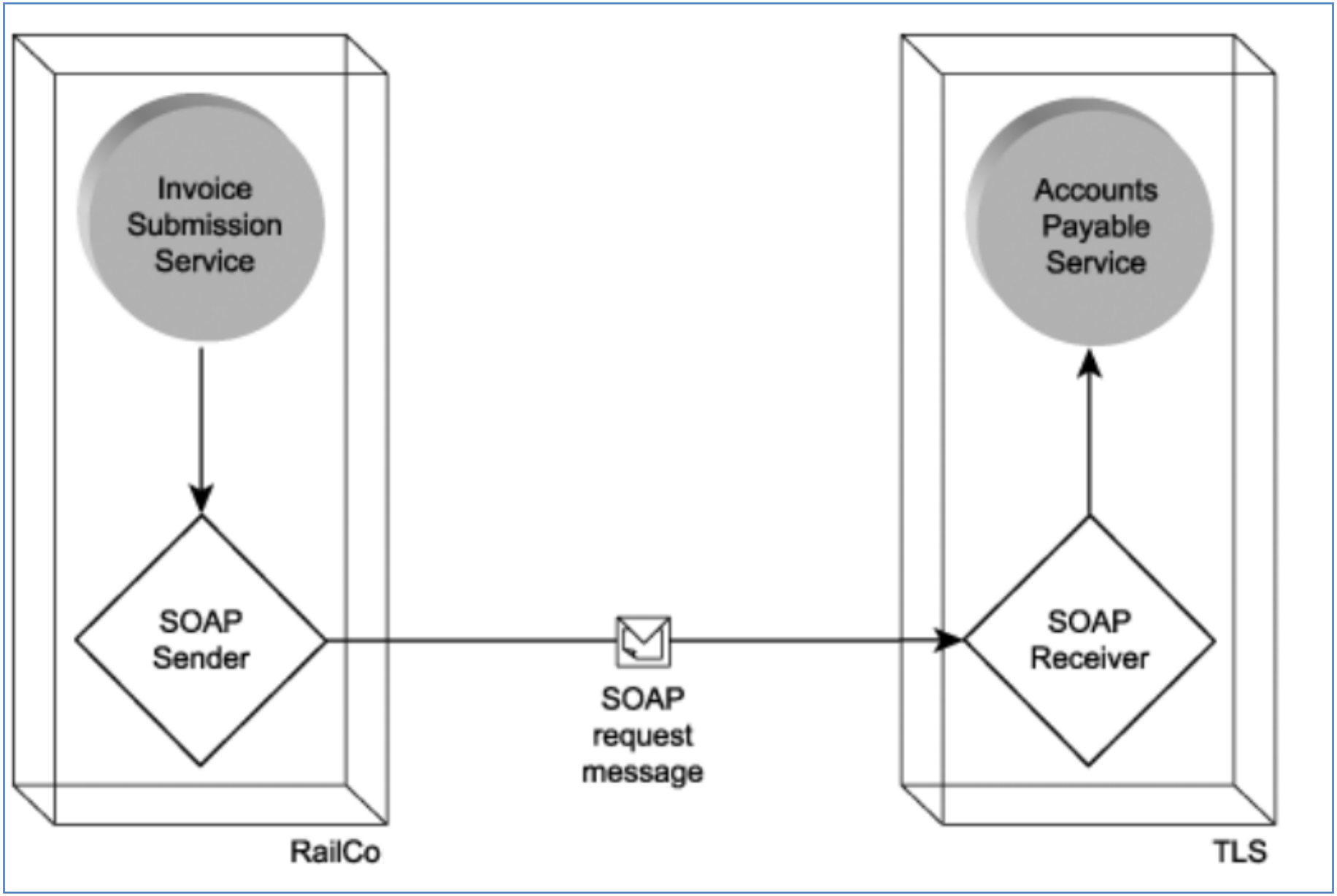
Nodes

- Web services rely on physical communication medium to process and manage the exchange
- *“The programs that services use to transmit and receive SOAP messages are referred to as SOAP nodes”*
- A message sent by SOAP node must be received by SOAP node of the other service


service
SOAP
Node
SOAP
message

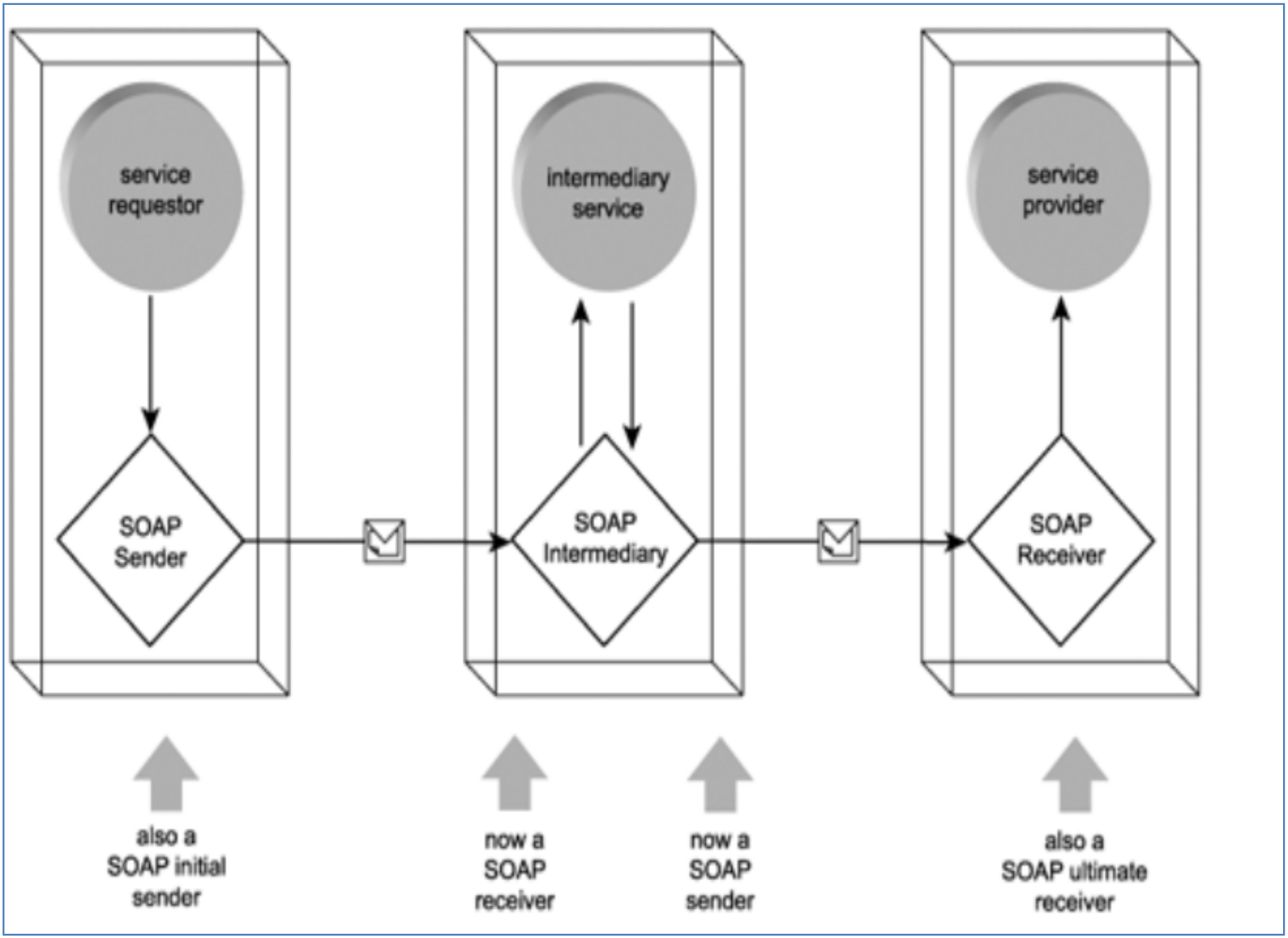
Node types

- SOAP sender – Transmits a message
- SOAP receiver – Receives a message
- SOAP intermediary – Receives and Transmits a message
- Initial SOAP sender – First SOAP node
- Ultimate SOAP receiver – Last SOAP node


Invoice Submission Service
SOAP Sender
RailCo
SOAP request message
Accounts Payable Service
SOAP Receiver
TLS

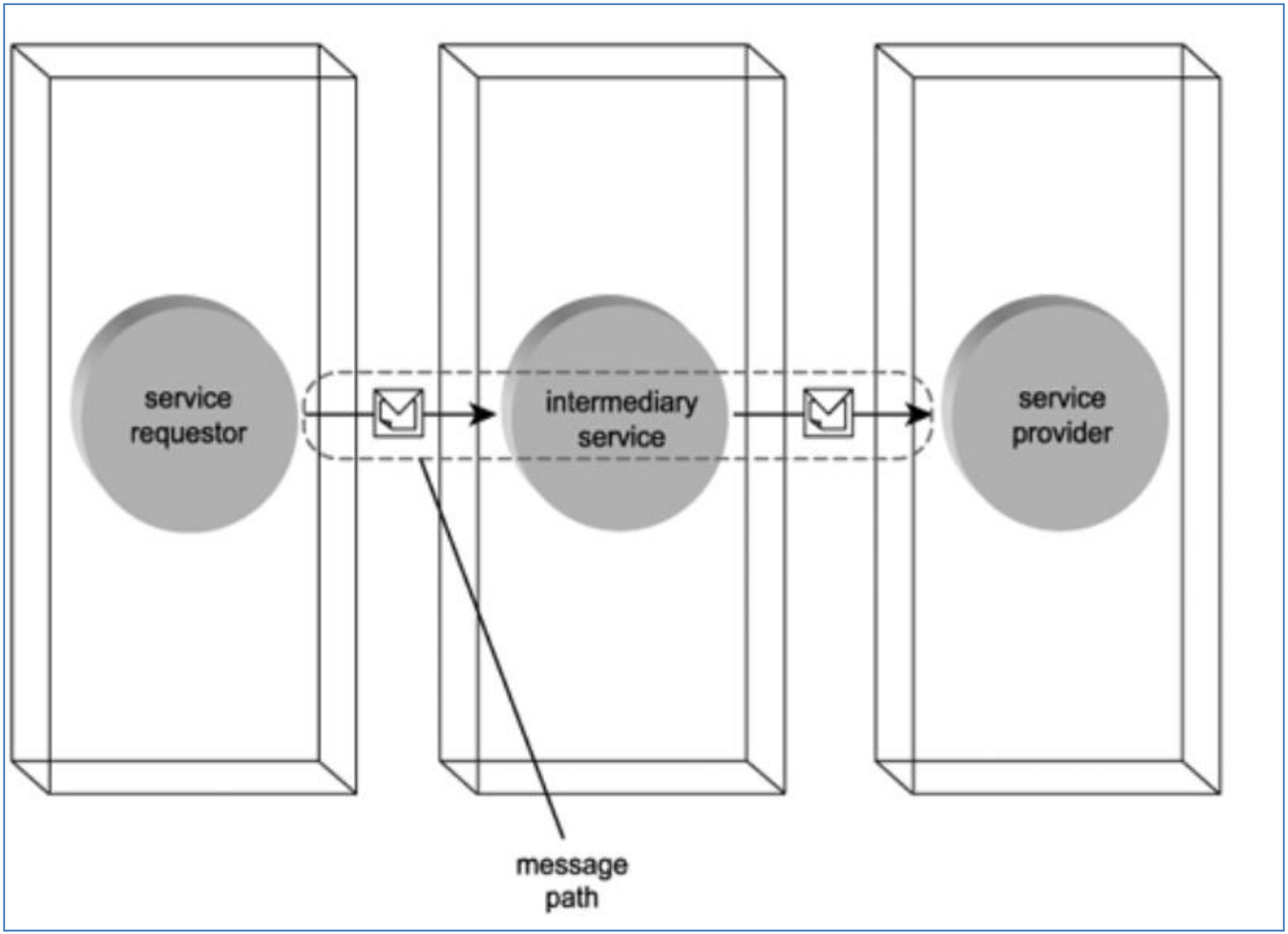
SOAP intermediaries

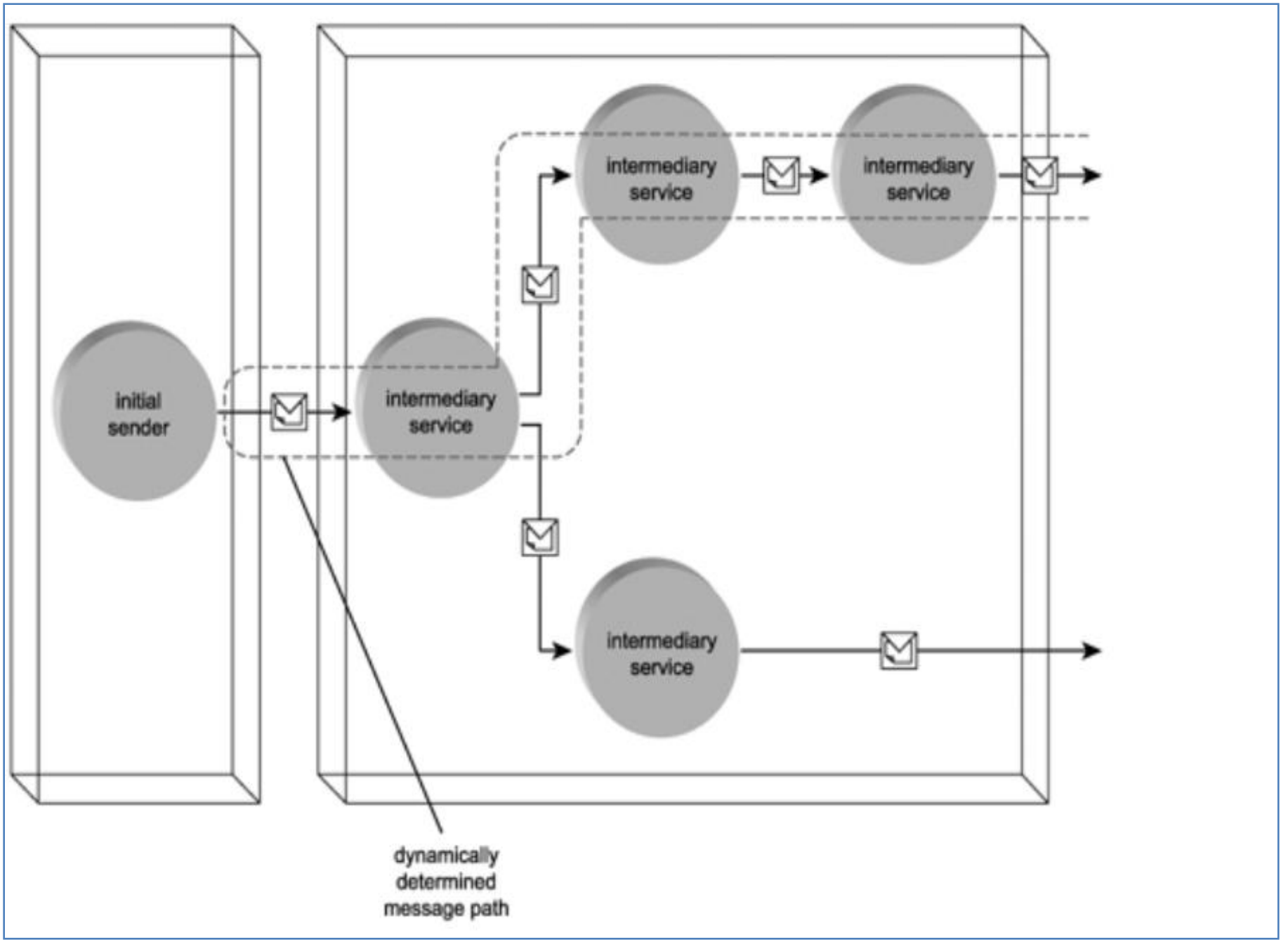
- **Forwarding intermediary nodes**
  - Relays the message to next SOAP node
  - May alter the header block by removing the blocks which can't be relayed further
- **Active intermediary nodes**
  - Can alter the header blocks by adding new ones, modifying existing ones and executing certain actions


service requestor
intermediary service
service provider
SOAP Sender
SOAP Intermediary
SOAP Receiver
also a SOAP initial sender
now a SOAP receiver
now a SOAP sender
also a SOAP ultimate receiver

Message Paths

- A message path refers to the route taken by a message from when it is first sent until it arrives at its ultimate destination.
- Therefore, a message path consists of at least one initial sender, one ultimate receiver, and zero or more intermediaries
- Message path can also be dynamic, based on the header blocks


service
requestor
intermediary
service
service
provider
message
path


initial sender
intermediary service
intermediary service
intermediary service
intermediary service
dynamically determined message path

Standards Organizations

- **W3C**
  - World Wide Web Consortium
  - Covers fundamental standards (XML, XML Schema, WSDL, SOAP etc)
- **OASIS**
  - Organization for the Advancement of Structured Information Standards
  - Covers E-commerce Standards (UDDI, WS-Security, WS-BPEL etc)
- **WS-I**
  - Web services Interoperability Organization
  - Standards like Basic Profile

Past and SOA

- **Application architecture**
  - Application architecture is to an application development team what a blueprint is to a team of construction workers
  - It reflects immediate solution requirements, as well as long-term, strategic IT goals
- Enterprise architecture
- Service-oriented architecture

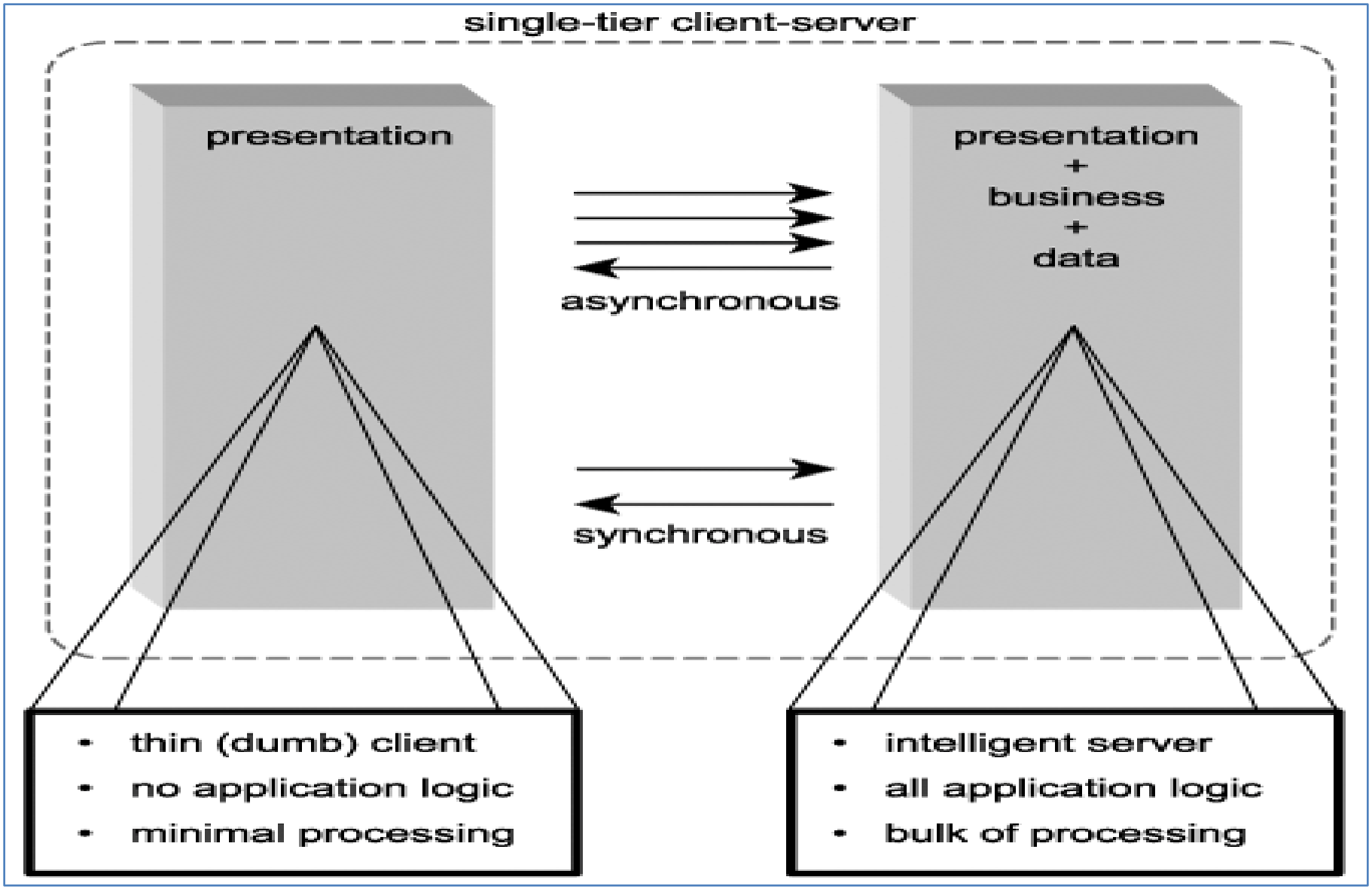
Past and SOA

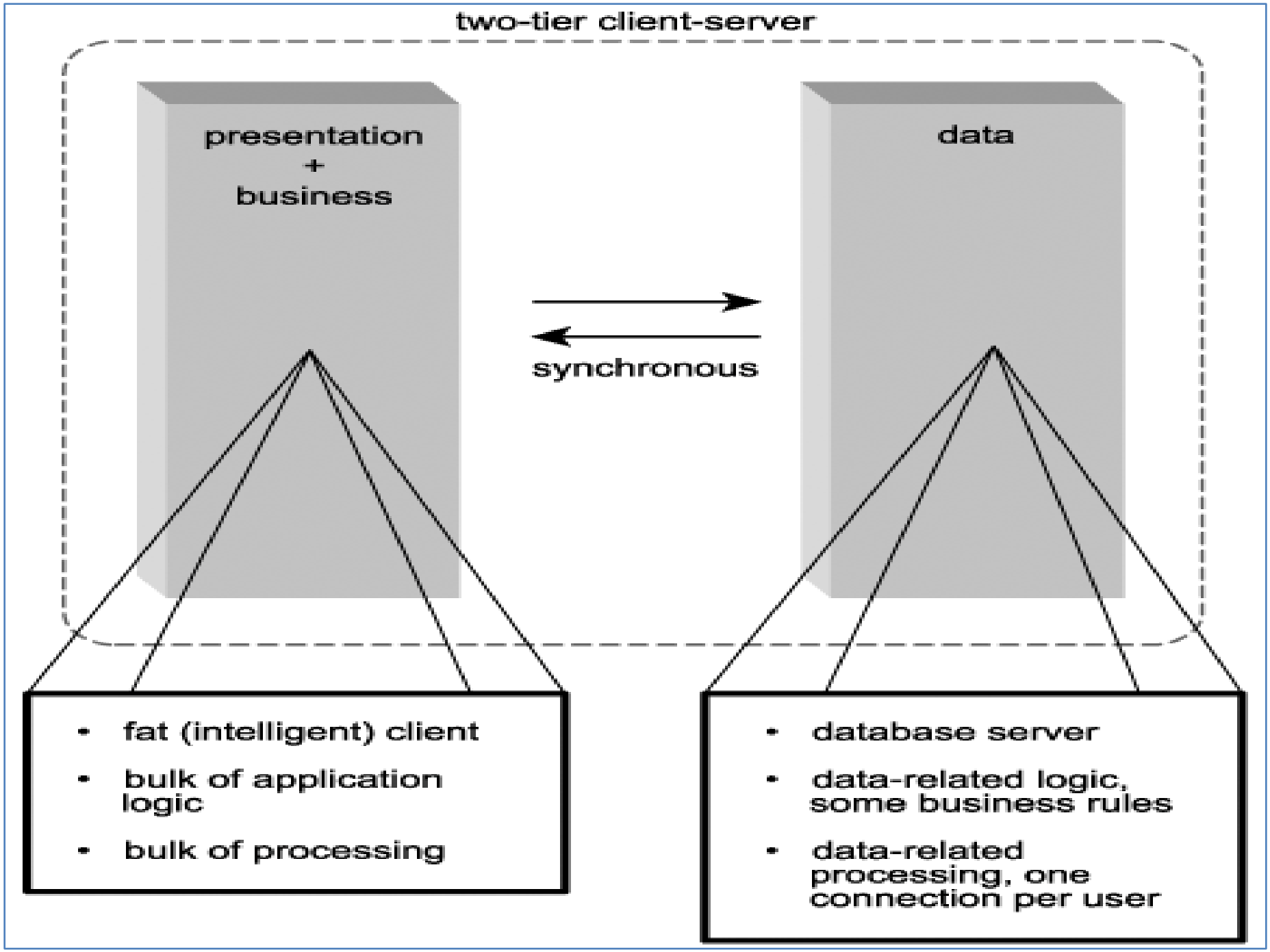
- Application architecture
- **Enterprise architecture**
  - Controls different application architectures
  - Like Urban plan for city
  - A long-term vision of how the organization plans to evolve its technology and environments
- Service-oriented architecture

Past and SOA

- Application architecture
- Enterprise architecture
- **Service-oriented architecture**
  - It spans both enterprise and application architecture domains.
  - SOA belongs in those areas that have the most to gain from the features and characteristics it introduces.

SOA vs. Client-Server Architecture




two-tier client-server
presentation
+
business
data
synchronous
• fat (intelligent) client
• bulk of application logic
• bulk of processing
• database server
• data-related logic, some business rules
• data-related processing, one connection per user

Comparison Points

- Application Logic
- Application Processing
- Technology
- Security
- Administration

Application Logic

- CS environments place the majority of application logic into the client software.
- In SOA, any software capable for SOAP exchange can be a requestor
- CS stores business rules in form of stored procedures
- "SOA is flexible" – Logic can be decomposed and placed in different services
- CS has tight coupling
- SOA has loose coupling

Application processing

- Client-server
  - The 80/20 ratio (Most processing done by client)
  - Synchronous communication
  - Stateful
  - Client demand significant resources
- SOA
  - Distributed processing, no fixed processing ratio
  - Synchronous or asynchronous communication
  - Stateless – Message Level Context Management
  - Service – Functional boundary and limited resource requirement

Technology

- Client server
  - 3GL (C++), 4GL (VB) programming languages
  - Robust RDBMSs (Oracle, Microsoft)
- SOA
  - Web technologies and XML
  - SOAP messaging framework

Security

- Client-server
  - Centralized at the server level
  - Databases are sufficiently sophisticated to manage user accounts
  - Security within client executable
- SOA
  - Complex level of security needed due to distributed nature

Administration

- Client server
  - Increasingly large maintenance costs – Each update has to go to every client
  - Managing server demands
- SOA
  - Handling variety of requestors
  - New requirements – Composition, Reusability, Registry management etc.