MESSAGING WITH SOAP

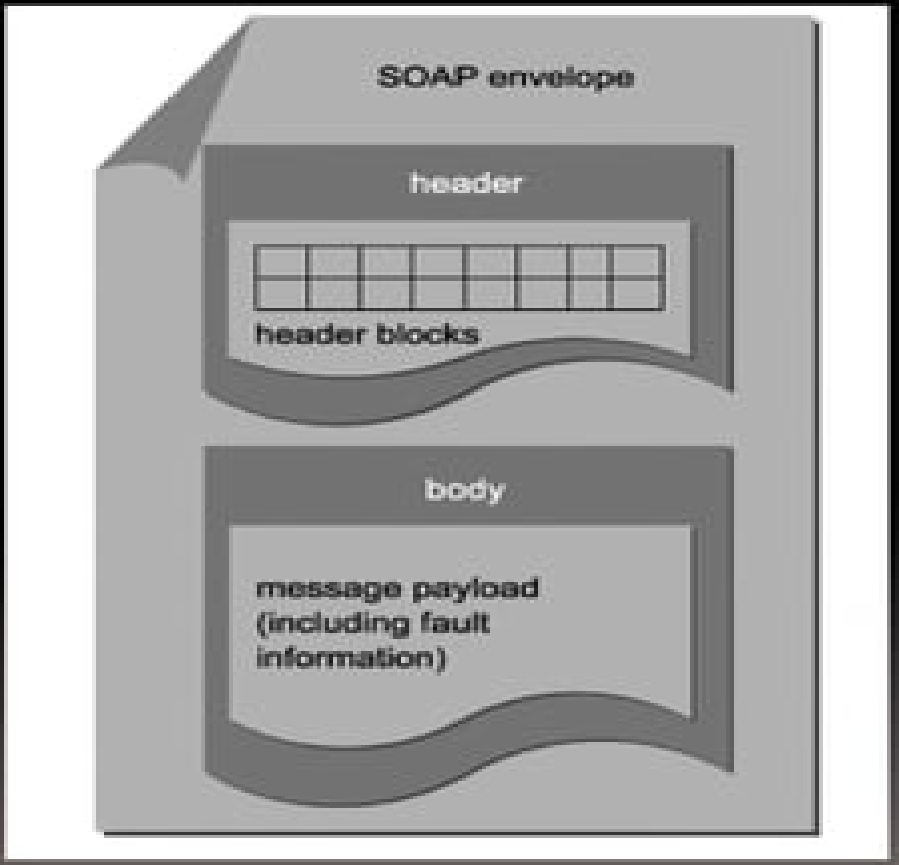
MESSAGING (WITH SOAP)

- within SOAs ,emphasis is placed on a message-centric application design that increases amount of business and application logic is embedded into messages.
- The SOAP specification has been universally accepted as the standard transport protocol for messages processed by Web services.
- A phone book is commonly compared to a service registry, and a mailing address is the equivalent of a WSDL endpoint (or port).
- An envelope represents a standardized medium for transporting mail much like SOAP represents a standardized format for transporting messages.

MESSAGING (WITH SOAP)

Messages

- Simple Object Access Protocol, the SOAP specification's main purpose is to define a standard message format.



The basic structure of a SOAP message

MESSAGING (WITH SOAP)

Envelope, header, and body

- Every SOAP message is packaged into a container known as an envelope.
- The envelope is responsible for housing all parts of the message
- Each message can contain a header, an area dedicated to hosting meta information.
- header section is a vital part of the overall architecture, and though optional, it is rarely omitted.
- Its importance relates to the use of header blocks through which numerous extensions can be implemented.
- The actual message contents are hosted by the message body, which typically consists of XML formatted data.
- The contents of a message body are often referred to as the message payload.

MESSAGING (WITH SOAP)

Header blocks

- A primary characteristic of the SOAP communications framework used by SOAs is an emphasis on creating messages that are as intelligence-heavy and self-sufficient as possible.
- This results in SOAP messages achieving a level of independence that increases the robustness and extensibility of this messaging framework qualities that are extremely important when relying on communication within the loosely coupled environment that Web services require.
- Message independence is implemented through the use of header blocks, packets of supplementary meta information stored in the envelope's header area.
- Header blocks outfit a message with all of the information required for any services with which the message comes in contact to process and route the message in accordance with its accompanying rules, instructions, and properties

MESSAGING (WITH SOAP)

Header blocks

- the use of header blocks, SOAP messages are capable of containing a large variety of supplemental information related to the delivery and processing of message contents.
- This alleviates services from having to store and maintain message-specific logic.
- It further reinforces the characteristics of contemporary SOA related to fostering reuse, interoperability, and composability.
- Web services can be designed with generic processing functionality driven by various types of meta information the service locates in the header blocks of the messages it receives.
- The use of header blocks has elevated the Web services framework to an extensible and composable enterprise-level computing platform. Practically all WS-* extensions are implemented using header blocks.

MESSAGING (WITH SOAP)

Header blocks

Examples of the types of features a message can be outfitted with using header blocks include:

1. processing instructions that may be executed by service intermediaries or the ultimate receiver
2. routing or workflow information associated with the message
3. security measures implemented in the message
4. reliability rules related to the delivery of the message
5. context and transaction management information
6. correlation information (typically an identifier used to associate a request message with a response message)

- SOAP allows the recognition and processing of header blocks to be marked as optional.
- This way messages can be safely outfitted with header blocks that implement non-critical features from newer extensions.

MESSAGING (WITH SOAP)

Message styles

- The SOAP specification was originally designed to replace proprietary RPC protocols by allowing calls between distributed components to be serialized into XML documents, transported, and then deserialized into the native component format upon arrival.
- As a result, much in the original version of this specification centered around the structuring of messages to accommodate RPC data.
- This RPC-style message runs contrary to the emphasis SOA places on independent, intelligence-heavy messages.
- SOA relies on document-style messages to enable larger payloads, coarser interface operations, and reduced message transmission volumes between services.

MESSAGING (WITH SOAP)

Attachments

- To facilitate requirements for the delivery of data not so easily formatted into an XML document, the use of SOAP attachment technologies exist.
- Each provides a different encoding mechanism used to bundle data in its native format with a SOAP message.
- SOAP attachments are commonly employed to transport binary files, such as images.

- Example:

purchase orders are not allowed to be issued in the standard electronic format, as the signature is required to be an ever-present part of the document.

MESSAGING (WITH SOAP)

Faults

- SOAP messages offer the ability to add exception handling logic by providing an optional fault section that can reside within the body area.
- The typical use for this section is to store a simple message used to deliver error condition information when an exception occurs.

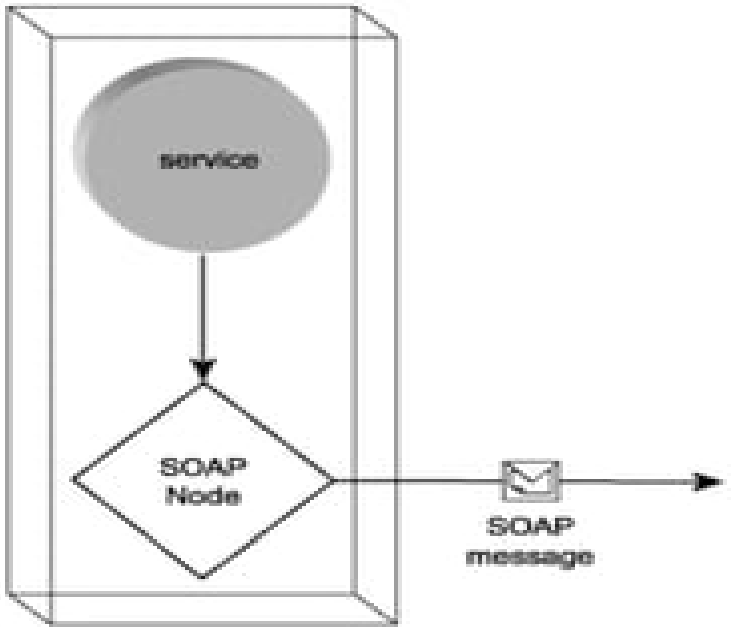
Nodes

- Although Web services exist as self-contained units of processing logic, they are reliant upon a physical communications infrastructure to process and manage the exchange of SOAP messages.
- Every major platform has its own implementation of a SOAP communications server, and as a result each vendor has labeled its own variation of this piece of software differently.
- In abstract, the programs that services use to transmit and receive SOAP messages are referred to as SOAP nodes

MESSAGING (WITH SOAP)

Nodes

A SOAP node transmitting a SOAP message received by the service logic



- SOAP message sent by the SOAP node from service A can be received and processed by a SOAP node (supporting the same version of the SOAP standard) from any other service.

MESSAGING (WITH SOAP)

Node types

- As with the services that use, the underlying SOAP nodes are given labels that identify their type, depending on what form of processing they are involved with in a given message processing scenario.
- Below is a list of type labels associated with SOAP nodes (in accordance with the standard SOAP Processing Model).
- Notice that these names are very similar to the Web service roles. The SOAP specification has a different use for the term "role" and instead refers to these SOAP types or labels as concepts.
  - SOAP sender- a SOAP node that transmits a message
  - SOAP receiver- a SOAP node that receives a message
  - SOAP intermediary- a SOAP node that receives and transmits a message, and optionally processes the message prior to transmission
  - initial SOAP sender -the first SOAP node to transmit a message
  - ultimate SOAP receiver-the last SOAP node to receive a message

MESSAGING (WITH SOAP)

SOAP intermediaries

- The same way service intermediaries transition through service provider and service requestor roles, SOAP intermediary nodes move through SOAP receiver and SOAP sender types when processing a message
- SOAP nodes acting as intermediaries can be classified as forwarding or active.
- When a SOAP node acts as a forwarding intermediary, it is responsible for relaying the contents of a message to a subsequent SOAP node. In doing so, the intermediary will often process and alter header block information relating to the forwarding logic it is executing.
- For example, it will remove a header block it has processed, as well as any header blocks that cannot be relayed any further.

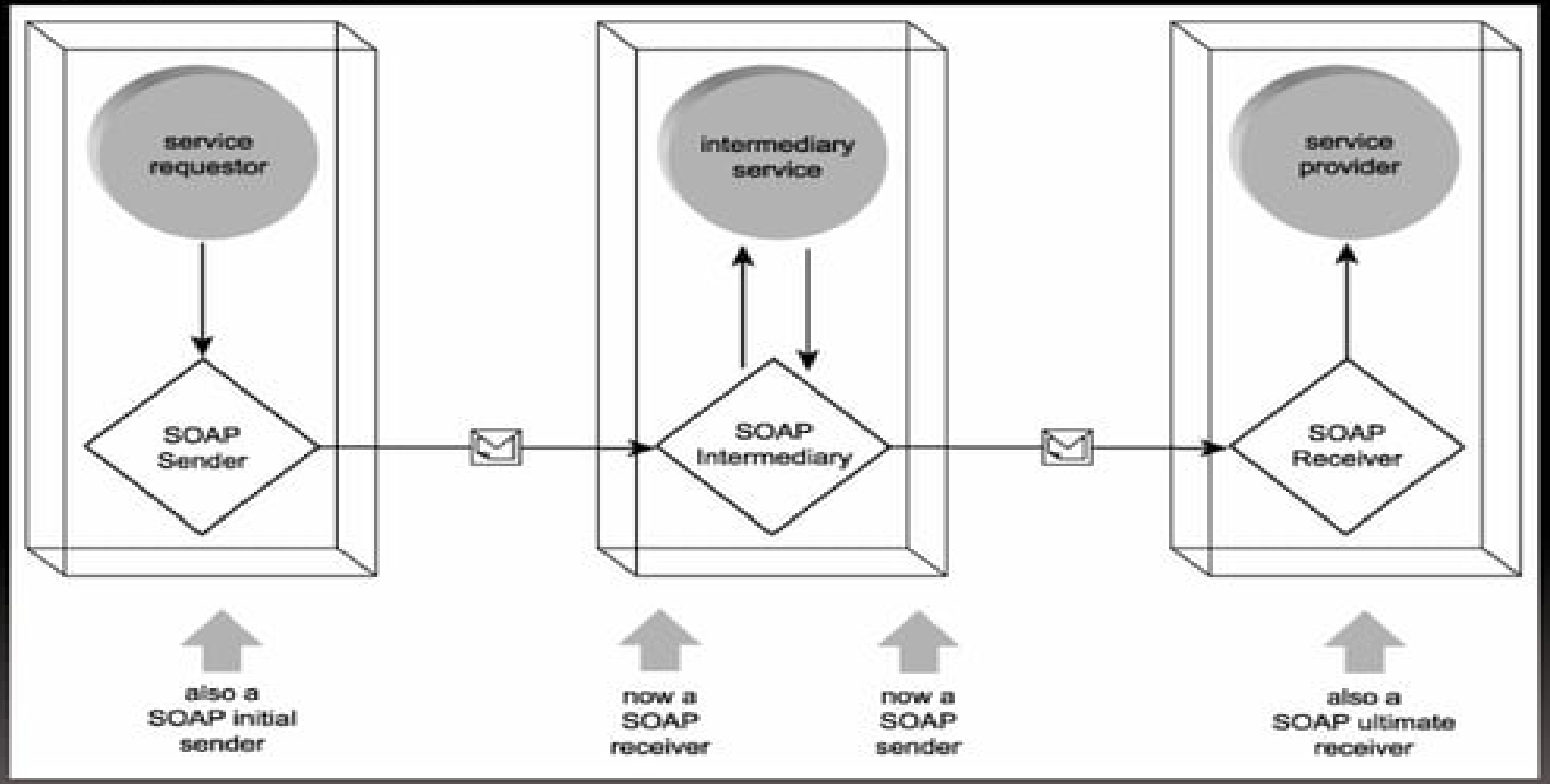
MESSAGING (WITH SOAP)

SOAP intermediaries

- Active intermediary nodes are distinguished by the type of processing they perform above and beyond forwarding-related functions.
- An active intermediary is not required to limit its processing logic to the rules and instructions provided in the header blocks of a message it receives.
- It can alter existing header blocks, insert new ones, and execute a variety of supporting actions.

MESSAGING (WITH SOAP)

SOAP intermediaries



Different types of SOAP nodes involved with processing a message.

MESSAGING (WITH SOAP)

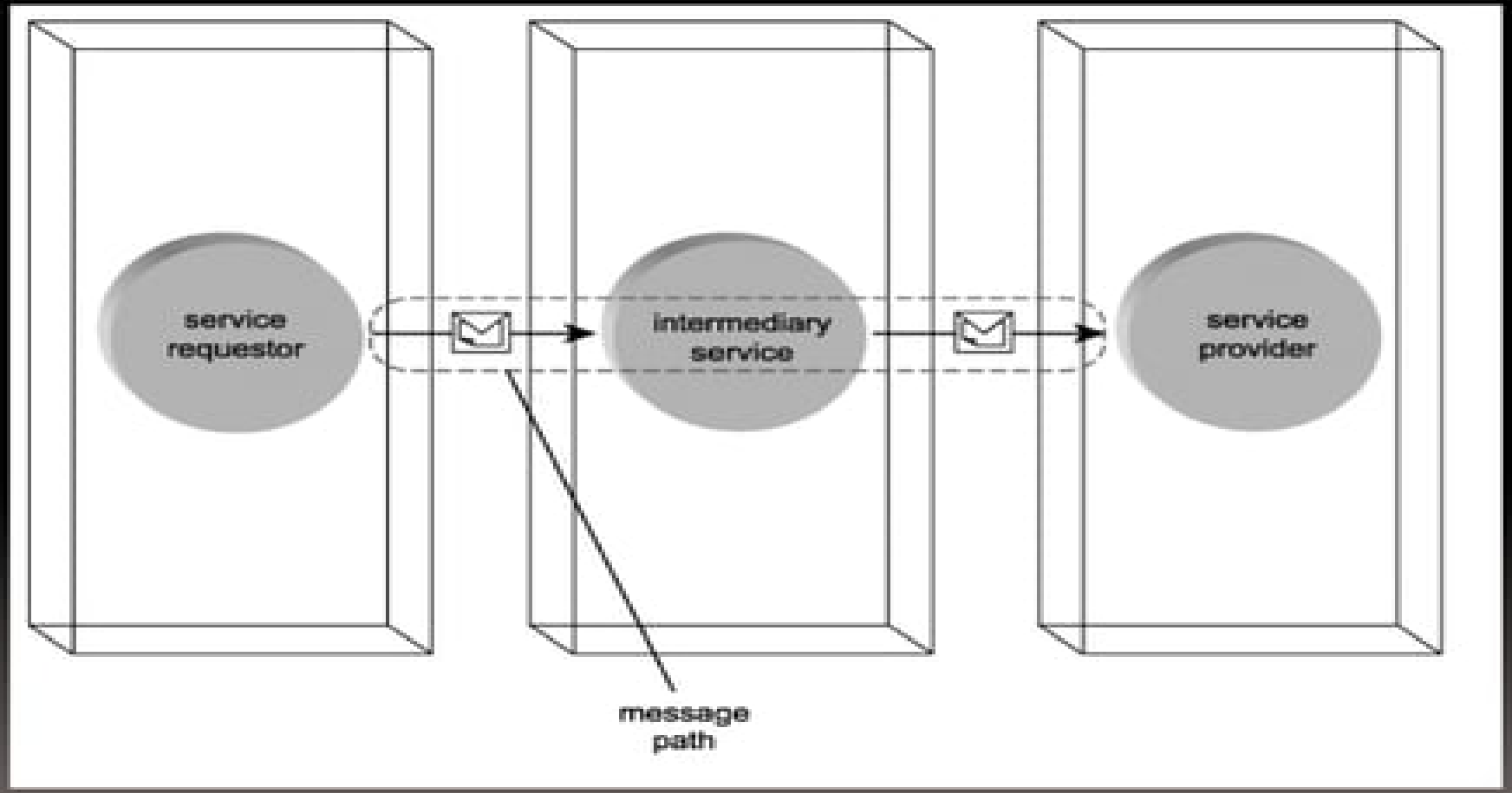
Message paths

- A message path refers to the route taken by a message from when it is first sent until it arrives at its ultimate destination.
- A message path consists of at least one initial sender, one ultimate receiver, and zero or more intermediaries .
- Mapping and modeling message paths becomes an increasingly important exercise in SOAs, as the amount of intermediary services tends to grow along with the expansion of a service-oriented solution.
- Design considerations relating to the path a message is required to travel often center around performance, security, context management, and reliable messaging concerns
- A message path is sometimes not predetermined.
- The use of header blocks processed by intermediaries can dynamically determine the path of a message. This may be the result of routing logic, workflow logic, or environmental conditions

MESSAGING (WITH SOAP)

Message paths

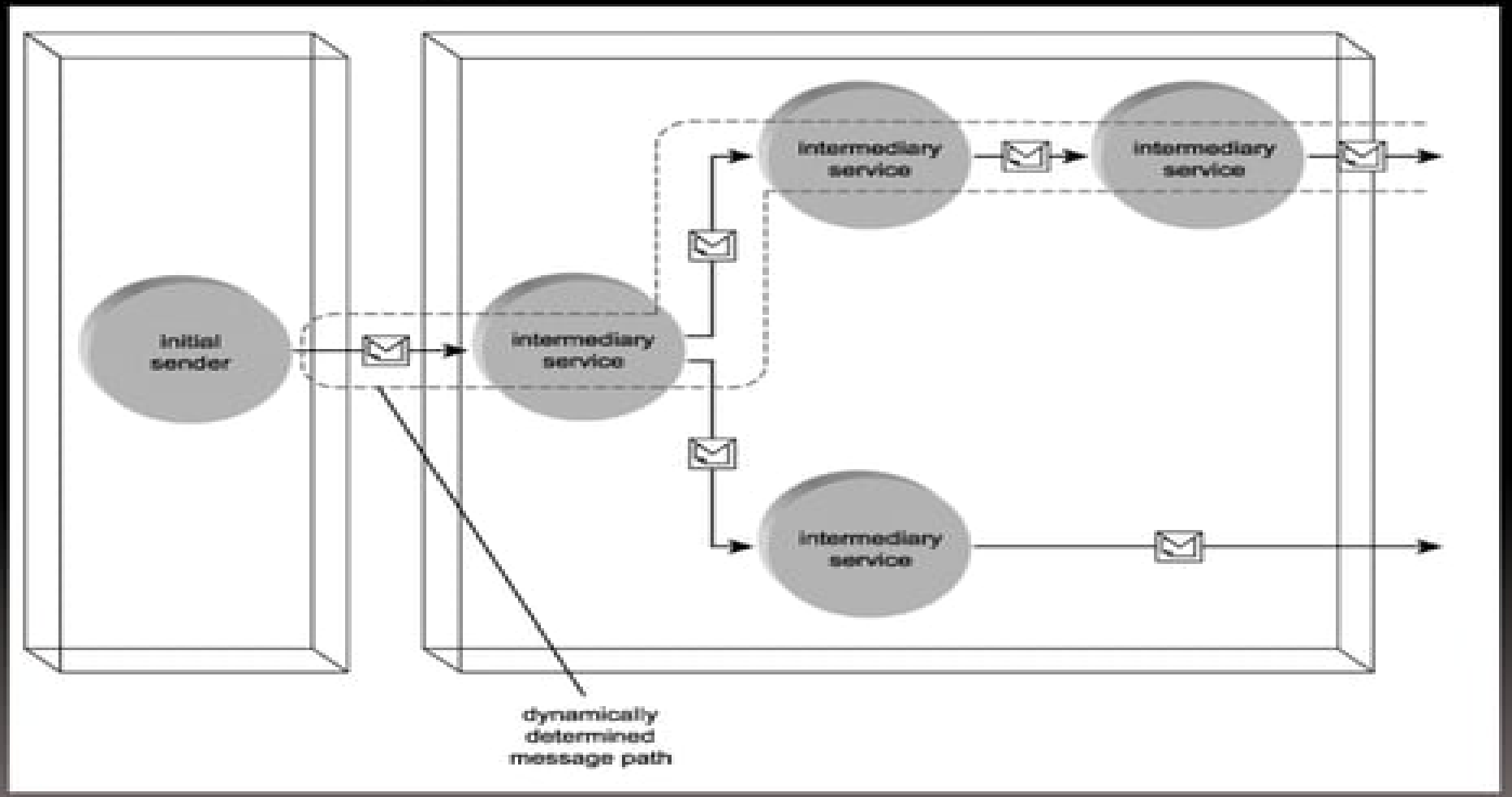
A message path consisting of three Web services



MESSAGING (WITH SOAP)

Message paths

A message path determined at runtime



MESSAGING (WITH SOAP)

Message paths

- When used within the context of SOAP nodes, it is referred to as a SOAP message path.
- While a message path in abstract can be purely logical, the SOAP node perspective is always focused on the actual physical transport route.
- A SOAP message path is comprised of a series of SOAP nodes, beginning with the initial SOAP sender and ending with the ultimate SOAP receiver.
- Every node refers to a physical installation of SOAP software, each with its own physical address