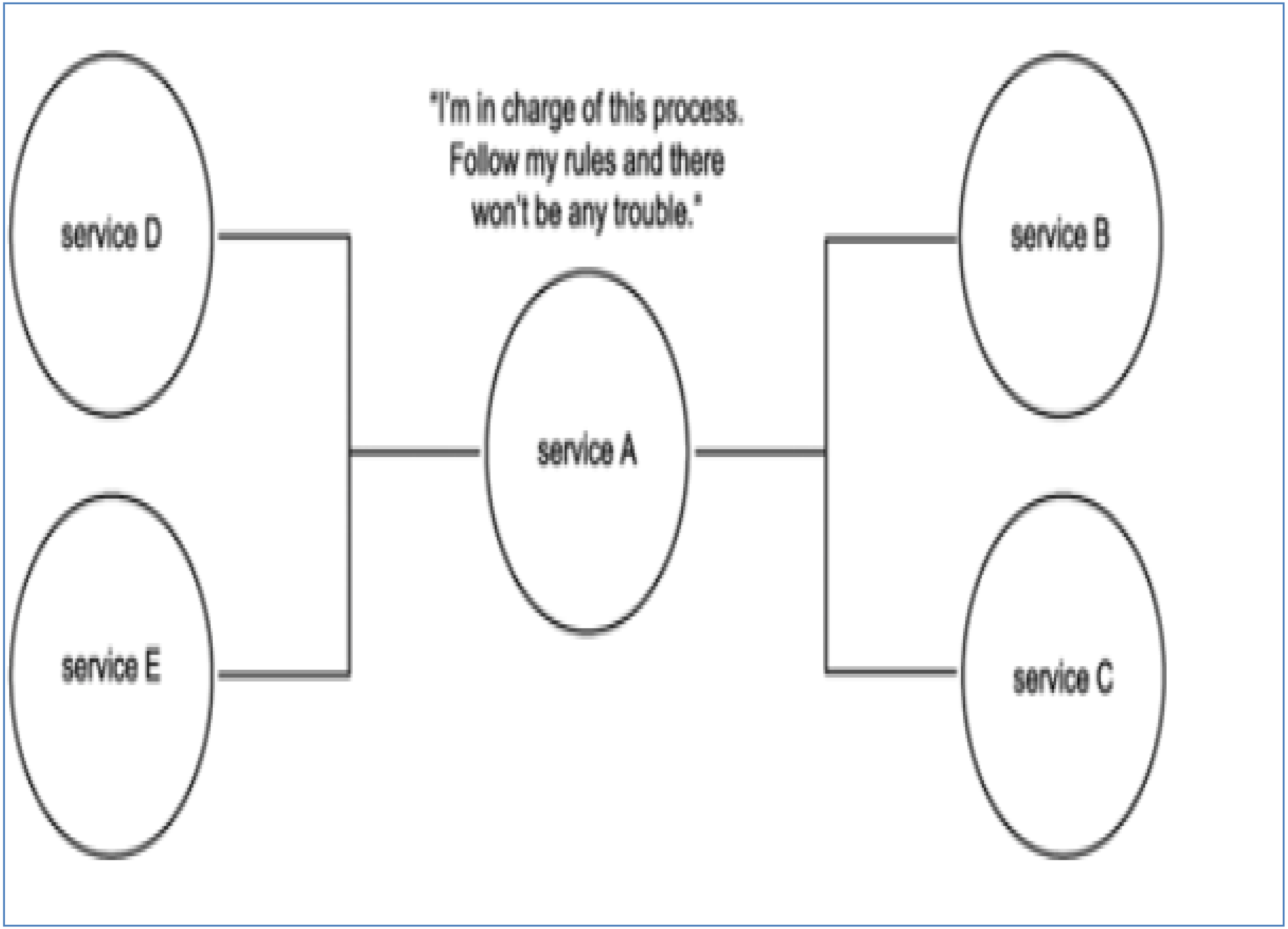
Orchestration

- Orchestration has following applications

1. To provide centrally controlled workflow logic for interoperability – For legacy systems
2. To act as a service which represents business logic in a standard venue – For SOA

- WS-BPEL (Business Process Execution Language) standardizes orchestration


"I'm in charge of this process. Follow my rules and there won't be any trouble."
service D
service E
service A
service B
service C

Business Protocols and Process Definition

- **Business Protocols:**
  - Define the rules , conditions and events by which the participants can interoperate to complete a business task
  - Part of Orchestration

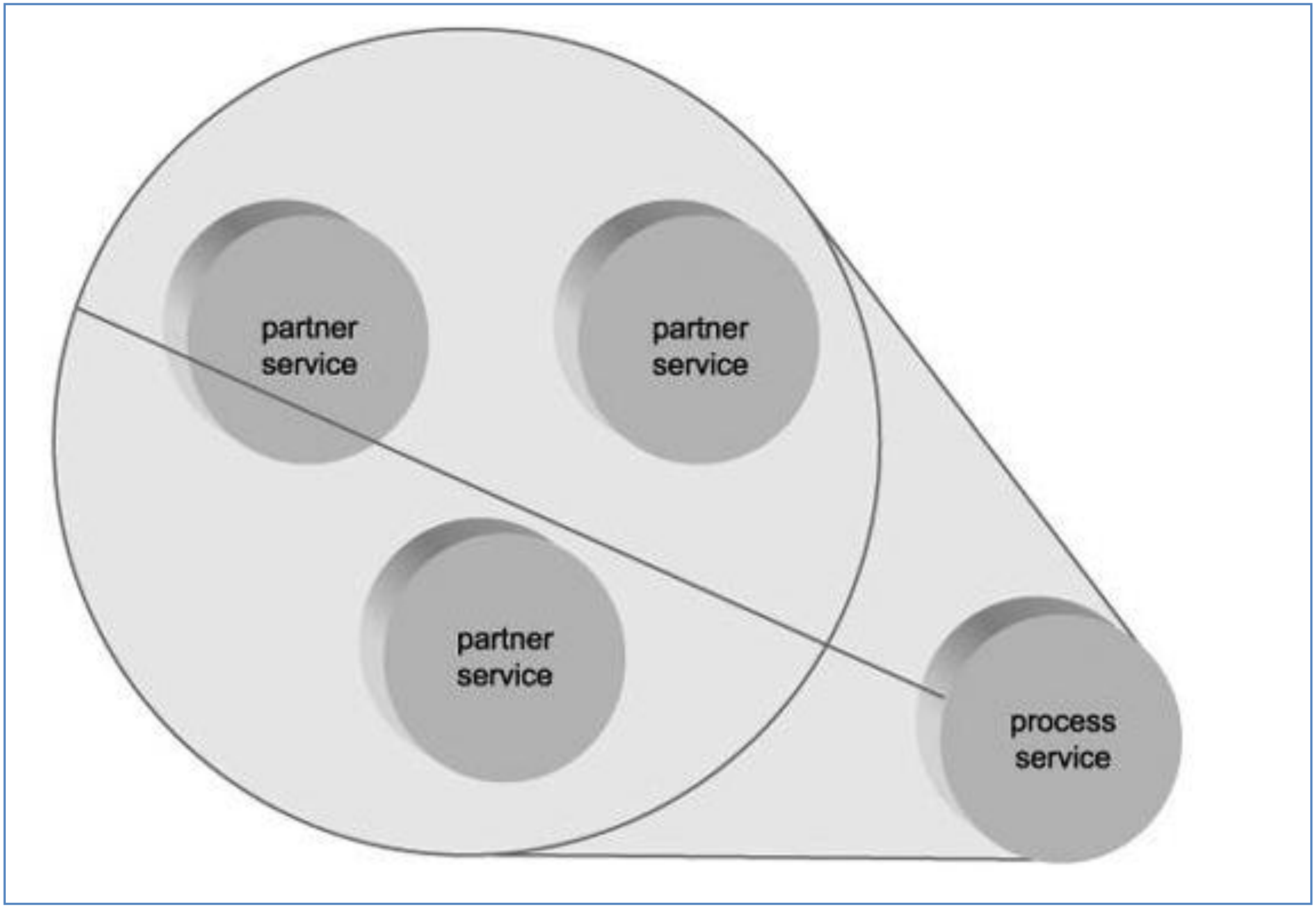
- **Process Definition:**
  - The details of the workflow logic are contained within process definition

Process Service and Partner Services

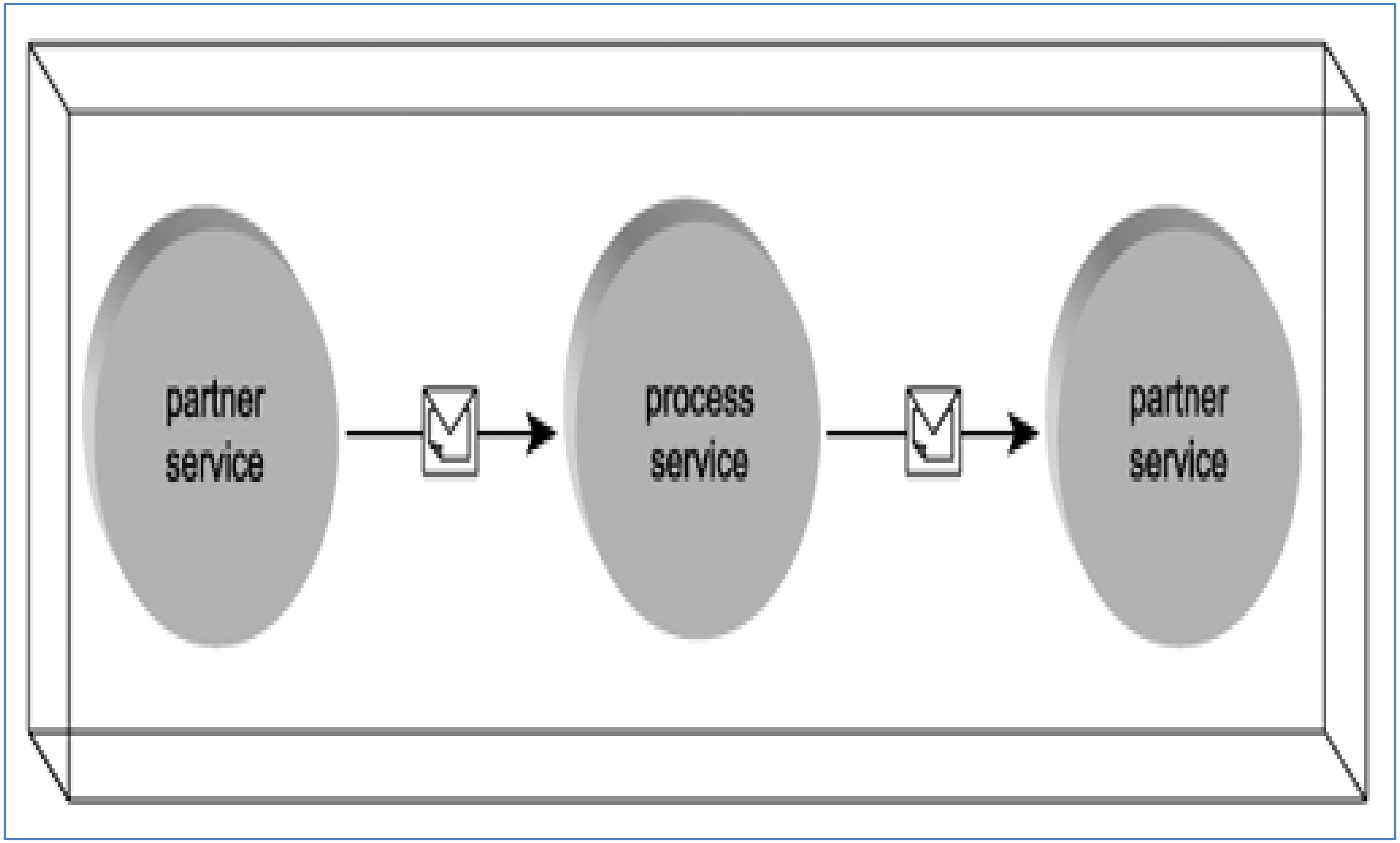
- Process definition (process service) contains process participants' details

- Process service can invoke partner services and it can also be invoked by partner services

Process Services and Partner Services



Process Service and Partner Services



Basic Activities and Structured Activities

- WS-BPEL breaks down workflow logic into a series of predefined primitive activities.

- **Basic activities** (receive, invoke, reply, throw, wait) represent fundamental workflow actions
  - which can be assembled using the logic supplied by **structured activities** (sequence, switch, while, flow).

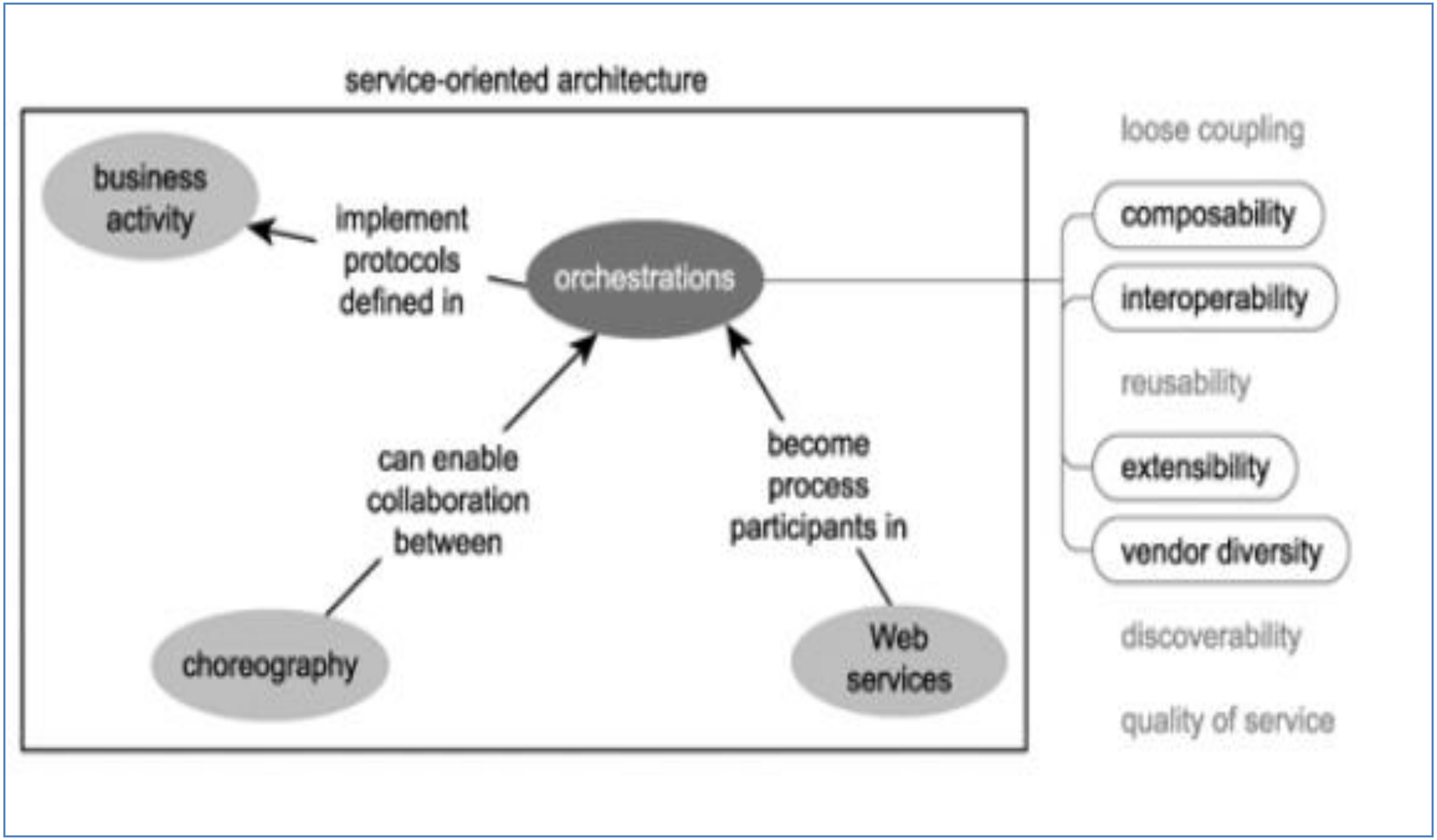
Sequences, Flows, and Links

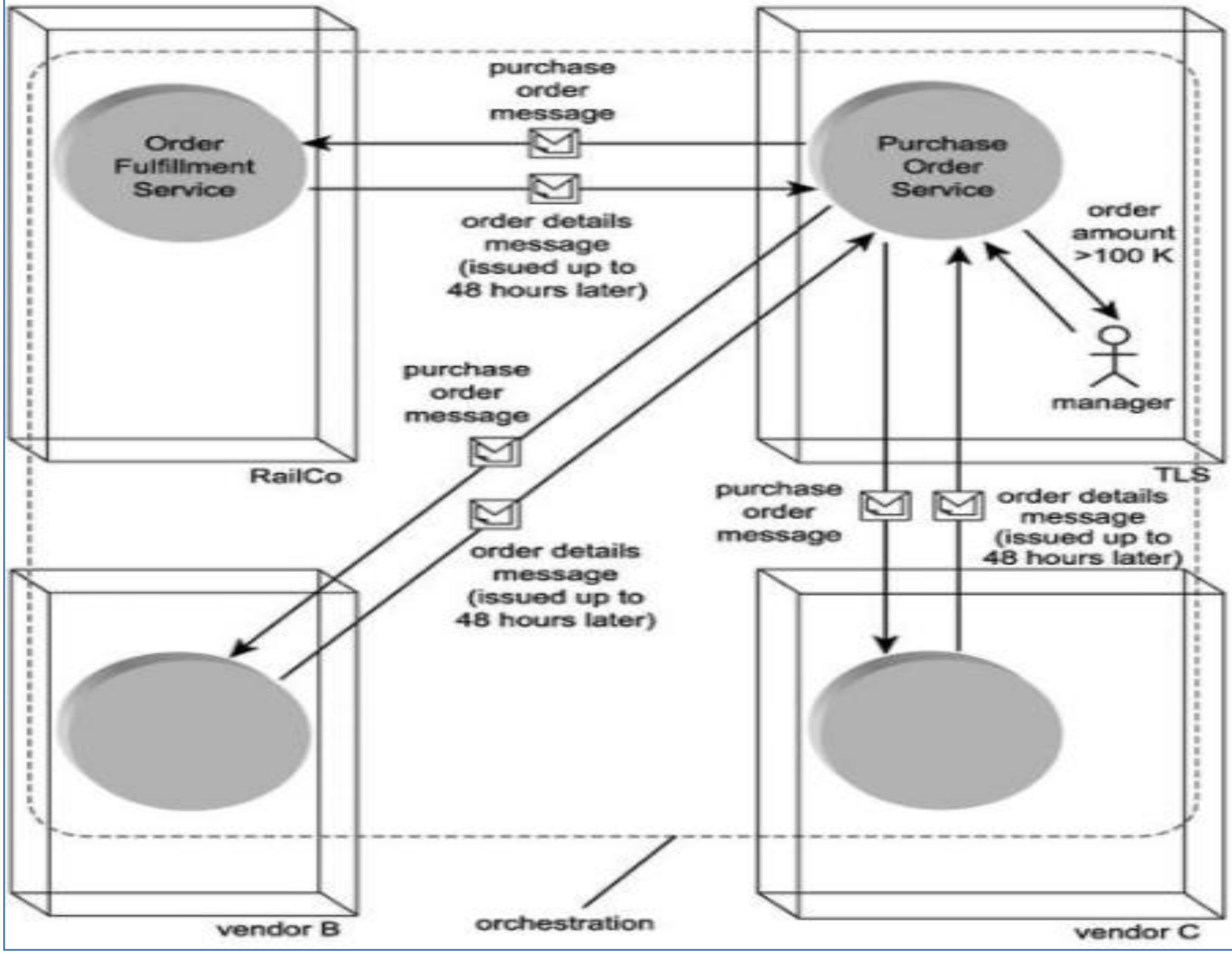
- A **sequence** aligns groups of related activities into a list that determines a sequential execution order.
- **Flows** also contain groups of related activities, but they introduce different execution requirements.
  - Flows ensures a form of synchronization among application logic residing in individual flows.
- **Links** are used to establish formal dependencies between activities that are part of flows.

Orchestration and Others

- **Orchestration and Activities:**
  - A single orchestration can be classified as a complex, long running activity
- **Orchestration and Coordination:**
  - Orchestration can use WS- Business Activity coordination type

Orchestration and SOA

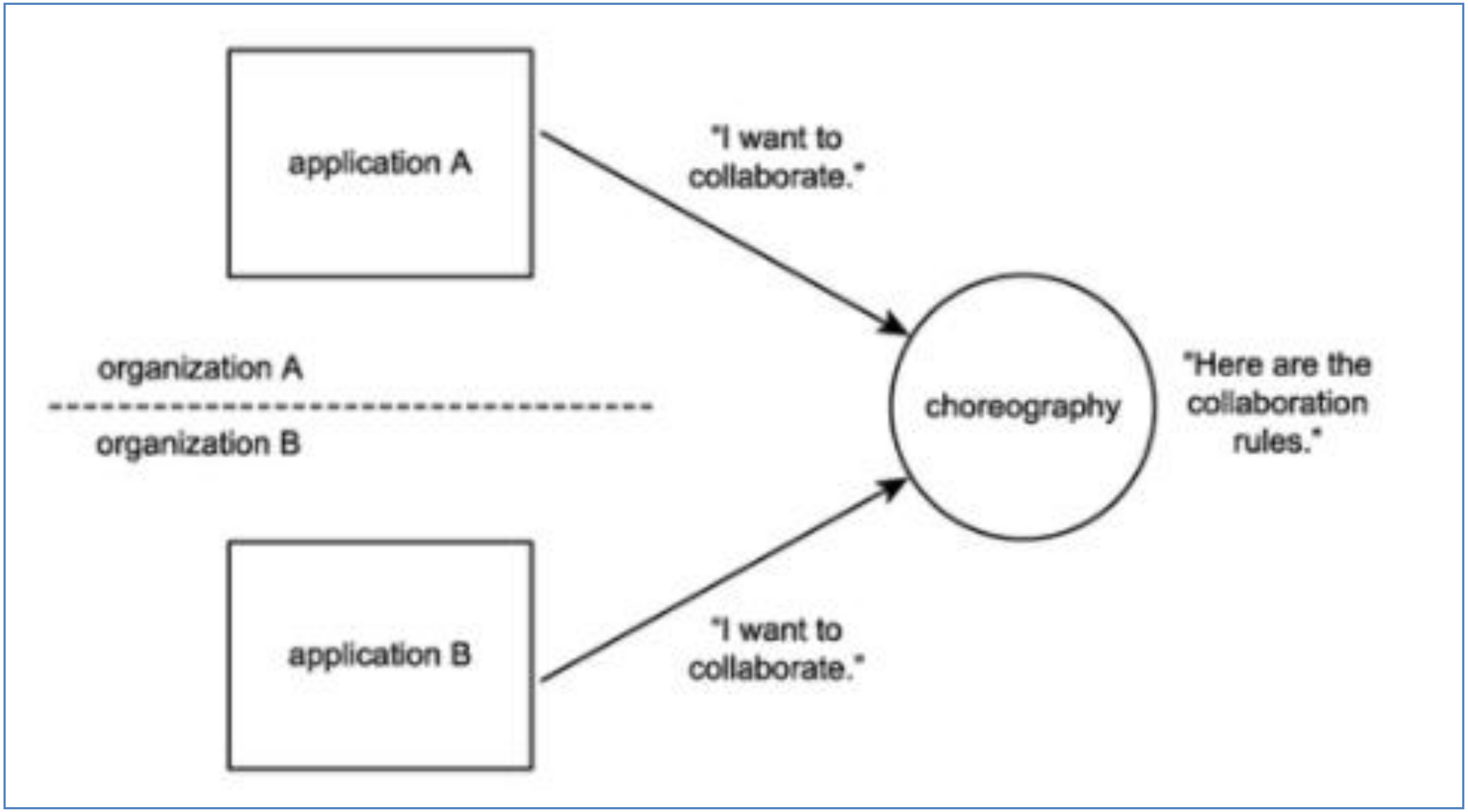



purchase order message
Order Fulfillment Service
Purchase Order Service
order details message (issued up to 48 hours later)
order amount >100 K
manager
purchase order message
RailCo
TLS
purchase order message
order details message (issued up to 48 hours later)
order details message (issued up to 48 hours later)
vendor B
orchestration
vendor C

So far,

- An orchestration expresses a body of business process logic that is typically owned by a single organization.
- An orchestration establishes a business protocol that formally defines a business process definition.
- The workflow logic within an orchestration is broken down into a series of basic and structured activities that can be organized into sequences and flows.
- Orchestration has been called the "heart of SOA," as it establishes a means of centralizing and controlling a great deal of inter and intra-application logic through a standardized service model.

Choreography



Choreography

- When different organizations interoperate via services, common collaboration is needed to work together

- WS-CDL (Choreography Description Language) allows information exchange between multiple organizations with public collaboration

Collaboration

- An important characteristic of choreographies is that they are intended for public message exchanges.
- No one entity (organization) necessarily controls the collaboration logic.
- Choreographies therefore provide the potential for establishing universal interoperability patterns for common inter-organization business tasks.

Roles and Participants

- Within any given choreography, a Web service assumes one of a number of predefined roles.
- Related services are grouped accordingly, categorized as participants (services).

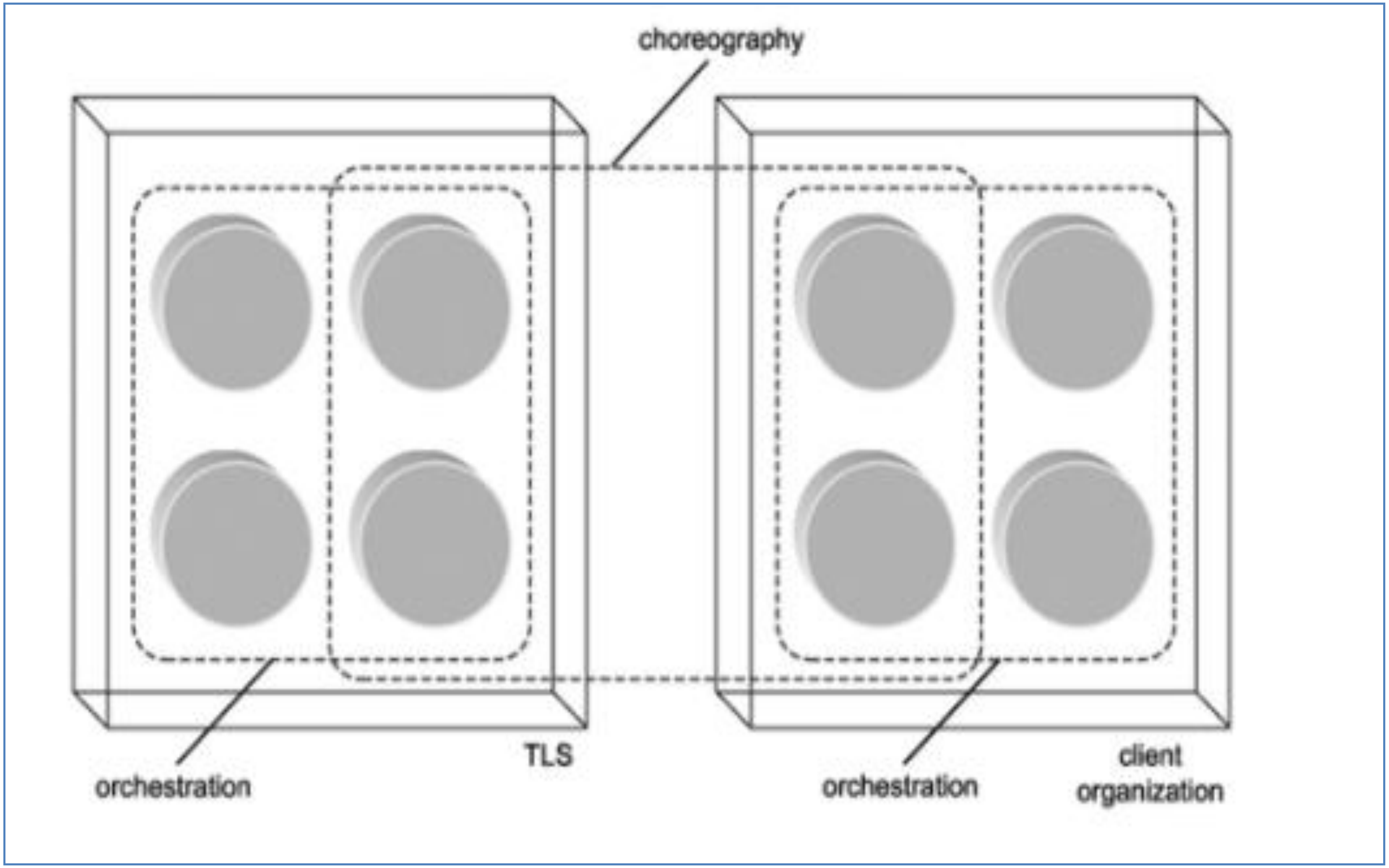
Relationships and Channels

- Each potential exchange between two roles in a choreography is defined individually as a relationship.
- Every relationship consequently consists of exactly two roles.
- Channels define the characteristics of the message exchange between two specific roles.

Interactions and Work Units

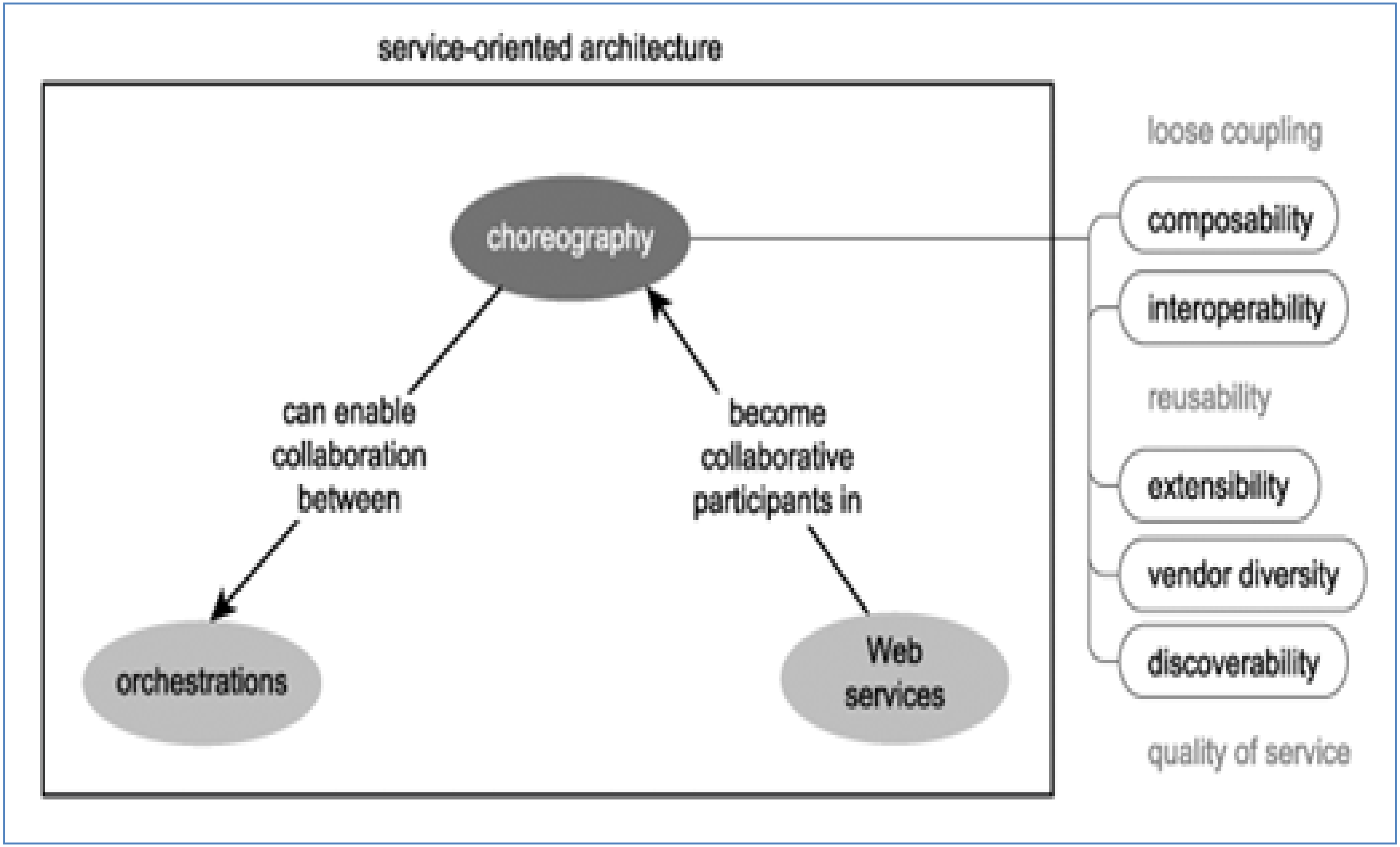
- The actual logic behind a message exchange is encapsulated within an interaction.
- *“Interactions are the fundamental building blocks of choreographies because the completion of an interaction represents actual progress within a choreography.”*
- Related to interactions are work units. These impose rules and constraints that must be adhered to for an interaction to successfully complete.

Orchestrations and Choreographies



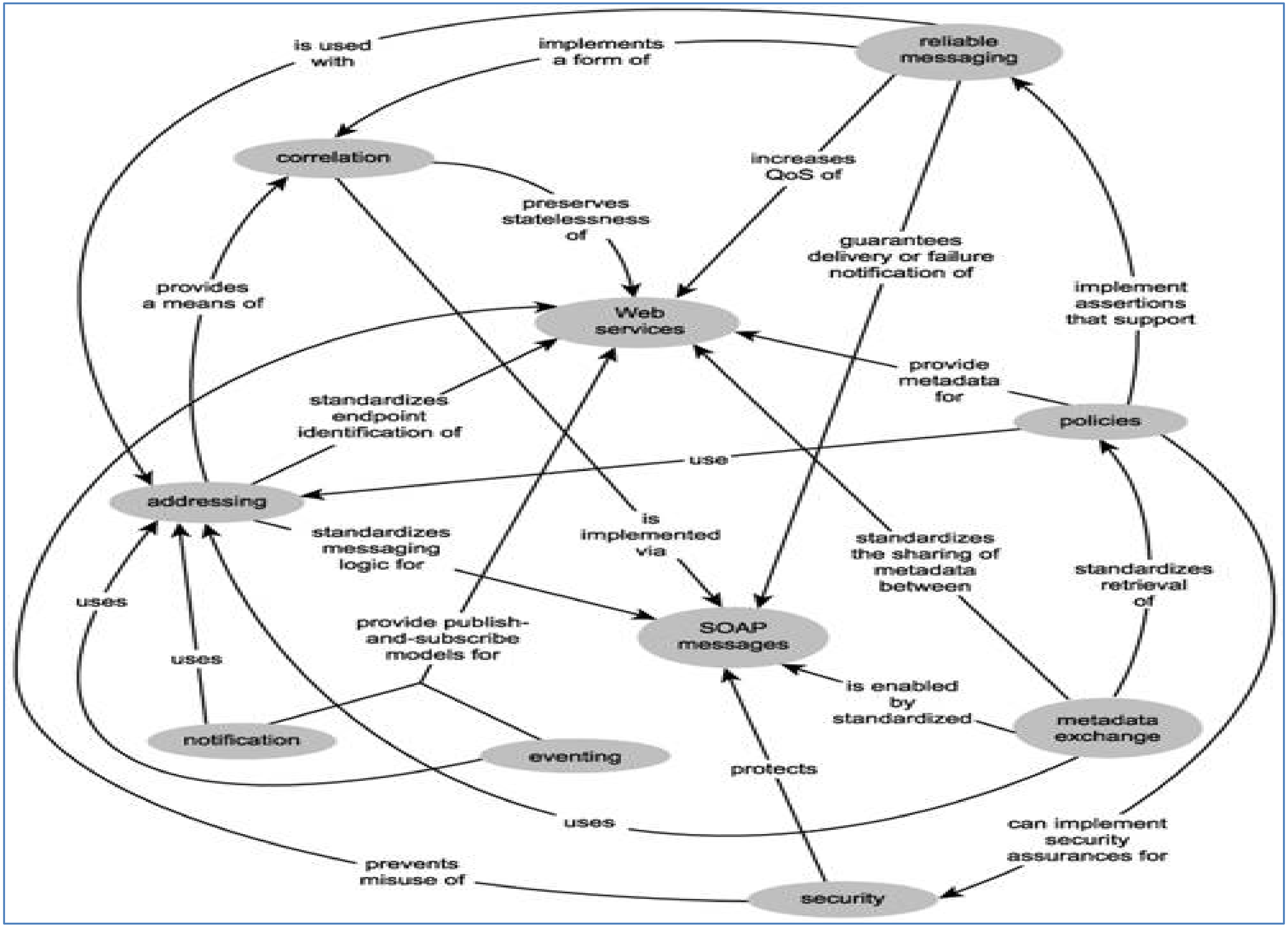
Orchestrations	Choreographies
An orchestration expresses organization-specific business workflow.	A choreography, on the other hand, is not necessarily owned by a single entity.
An organization owns and controls the logic behind an orchestration, even if that logic involves interaction with external business partners.	It acts as a community interchange pattern used for collaborative purposes by services from different provider entities
An orchestration is based on a model where the composition logic is executed and controlled in a centralized manner.	A choreography typically assumes that there is no single owner of collaboration logic.

Choreography and SOA



So far,

- A choreography is a complex activity comprised of a service composition and a series of MEPs.
- Choreographies consist of multiple participants that can assume different roles and that have different relationships.
- Choreographies are reusable, composable, and can be modularized.
- The concept of choreography extends the SOA vision to standardize cross-organization collaboration.

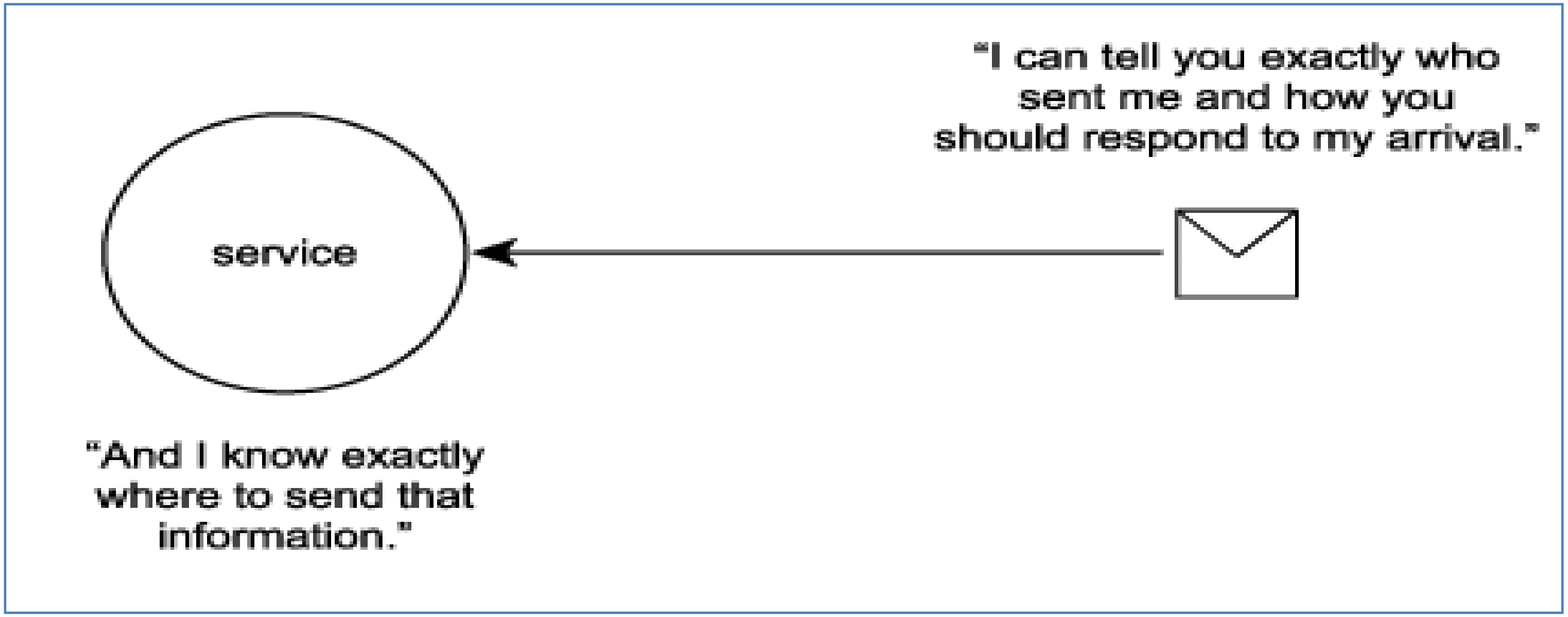

reliable messaging
is used with
implements a form of
correlation
preserves statelessness of
increases QoS of
guarantees delivery or failure notification of
provides a means of
implement assertions that support
Web services
provide metadata for
standardizes endpoint identification of
policies
use
addressing
is implemented via
standardizes messaging logic for
standardizes the sharing of metadata between
standardizes retrieval of
uses
provide publish-and-subscribe models for
SOAP messages
uses
is enabled by standardized
notification
eventing
metadata exchange
protects
uses
can implement security assurances for
prevents misuse of
security

WS-Addressing

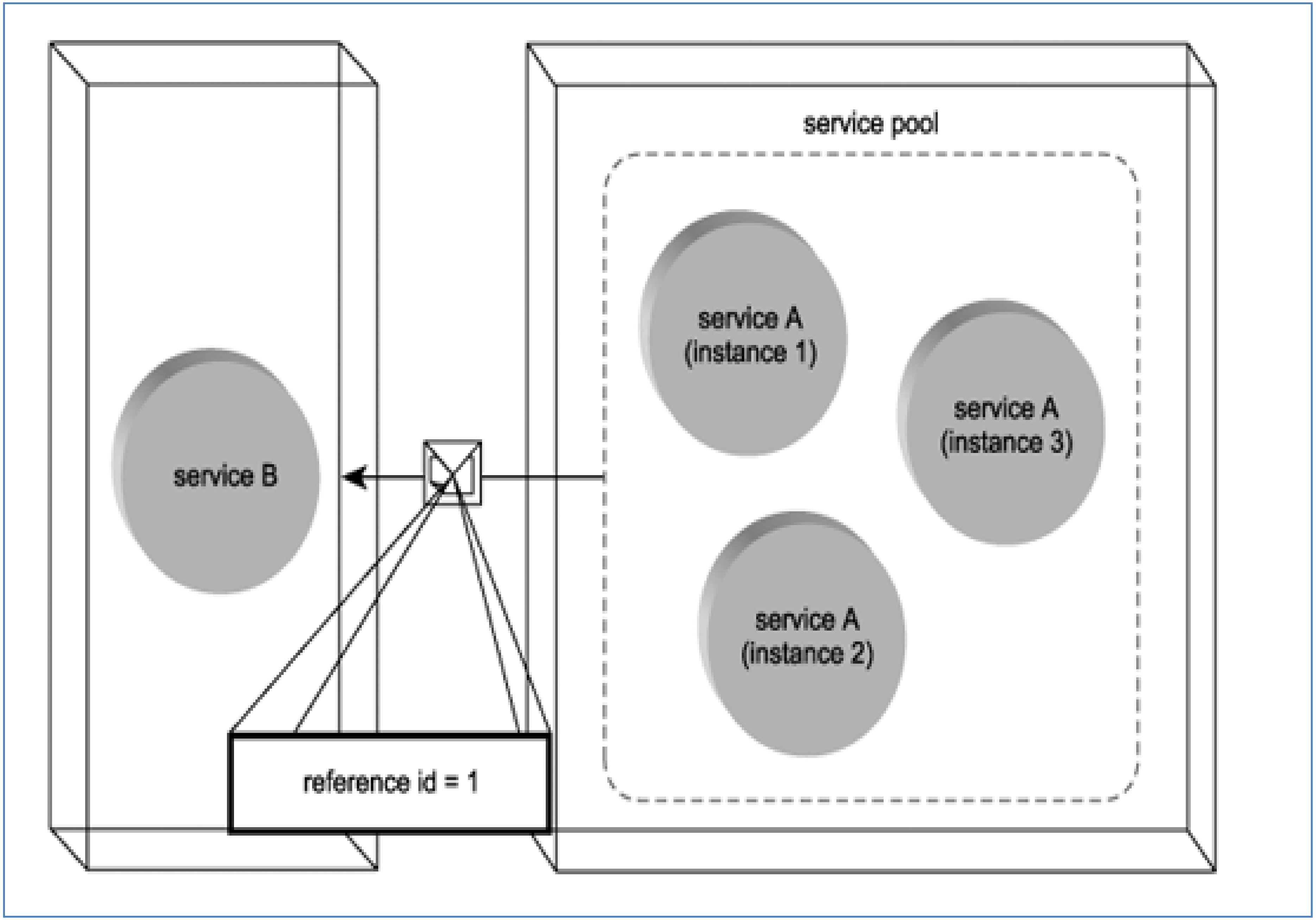
- Addressing for SOAP is like waybill for Shopping
  - Where is it coming from
  - The destination address
  - The specific person who is supposed to receive
  - Where it should go if it cannot be delivered as planned

WS-Addressing

- The WS-Addressing specification implements different addressing features by providing two types of SOAP headers


"I can tell you exactly who sent me and how you should respond to my arrival."
service
"And I know exactly where to send that information."

- What is required for a service requestor to contact a service provider ?
- What if, though, the service requestor needs to send a message to a specific instance of a service provider?


service B
service pool
service A (instance 1)
service A (instance 3)
service A (instance 2)
reference id = 1

Endpoint Reference

- The concept of addressing introduces the endpoint reference,
  - "An extension used primarily to provide identifiers that pinpoint a particular instance of a service."
- The endpoint reference is expected to be almost always dynamically generated and can contain a set of supplementary properties.

Endpoint Reference

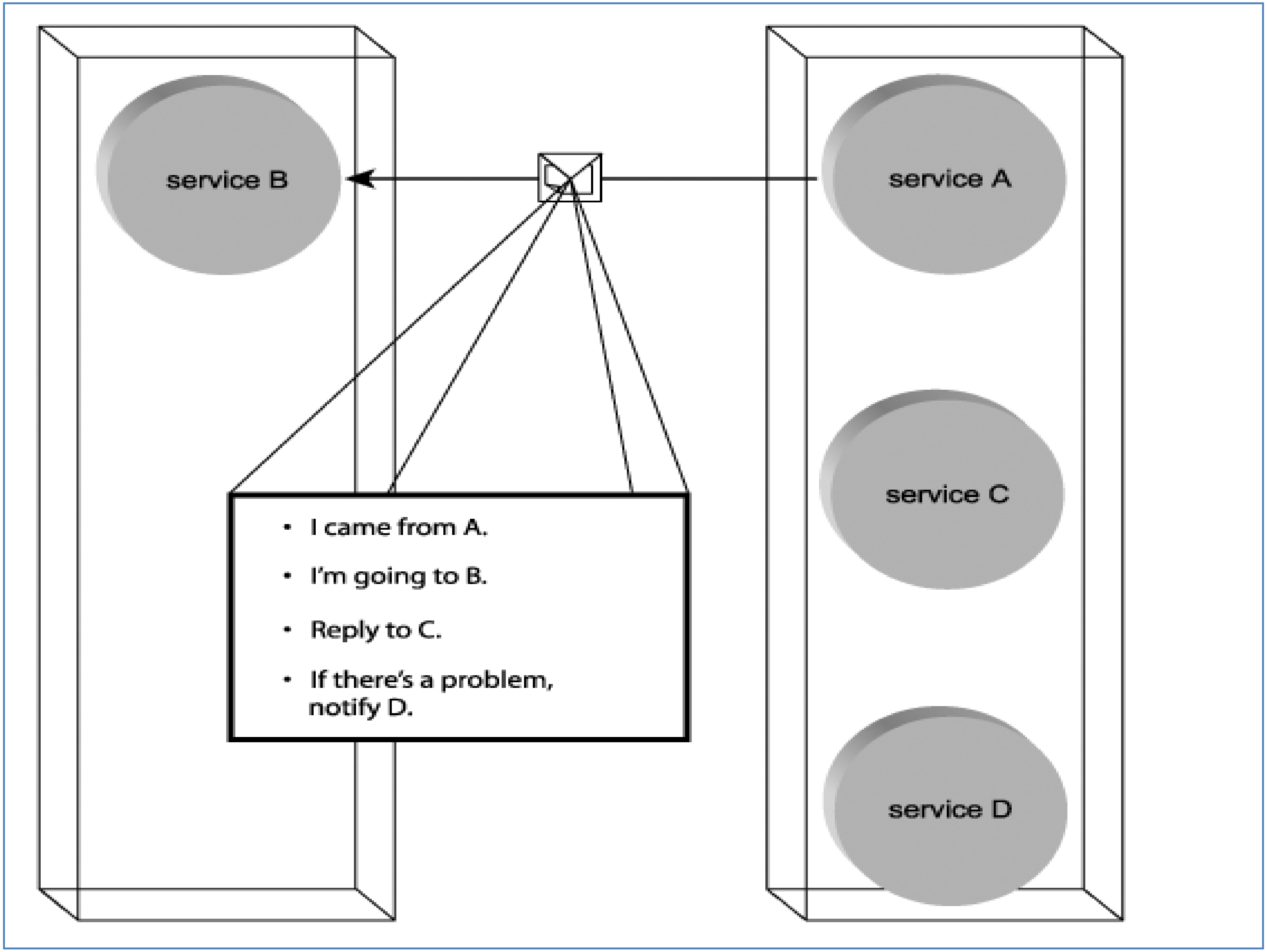
An endpoint reference consists of the following parts:

- **Address:** The URL of the Web service.
- **Reference properties:** A set of property values associated with the Web service instance. (Reference ID)
- **Reference parameters:** A set of parameter values that can be used to interact with a specific service instance.
- **Service port type:** Provides the exact location of service description details required for a reply.
- **Policy:** A WS-Policy compliant policy that provides rules and behavior information relevant to the current service interaction

Message Information Headers

- MEPs can limit the service interaction scenarios within which they participate.
- In sophisticated service-oriented solutions, services often require the flexibility to break a fixed pattern.

- The collection of standardized headers that establish message exchange-related characteristics within the messages themselves are MI headers.


service B
service A
service C
service D
• I came from A.
• I'm going to B.
• Reply to C.
• If there's a problem, notify D.

Message Information Headers

The MI headers provided by WS-Addressing include:

- **Destination:** The to address
- **Source endpoint:** An endpoint reference to the Web service that generated the message.
- **Reply endpoint:** The address where reply should be sent.
- **Fault endpoint:** The address to which a fault notification should be sent.
- **Message id:** A value that uniquely identifies the message or the retransmission of the message
- **Relationship:** Used in request-response scenarios, contains the message id of the related message to which a message is replying
- **Action:** A URI value that indicates the message's overall purpose

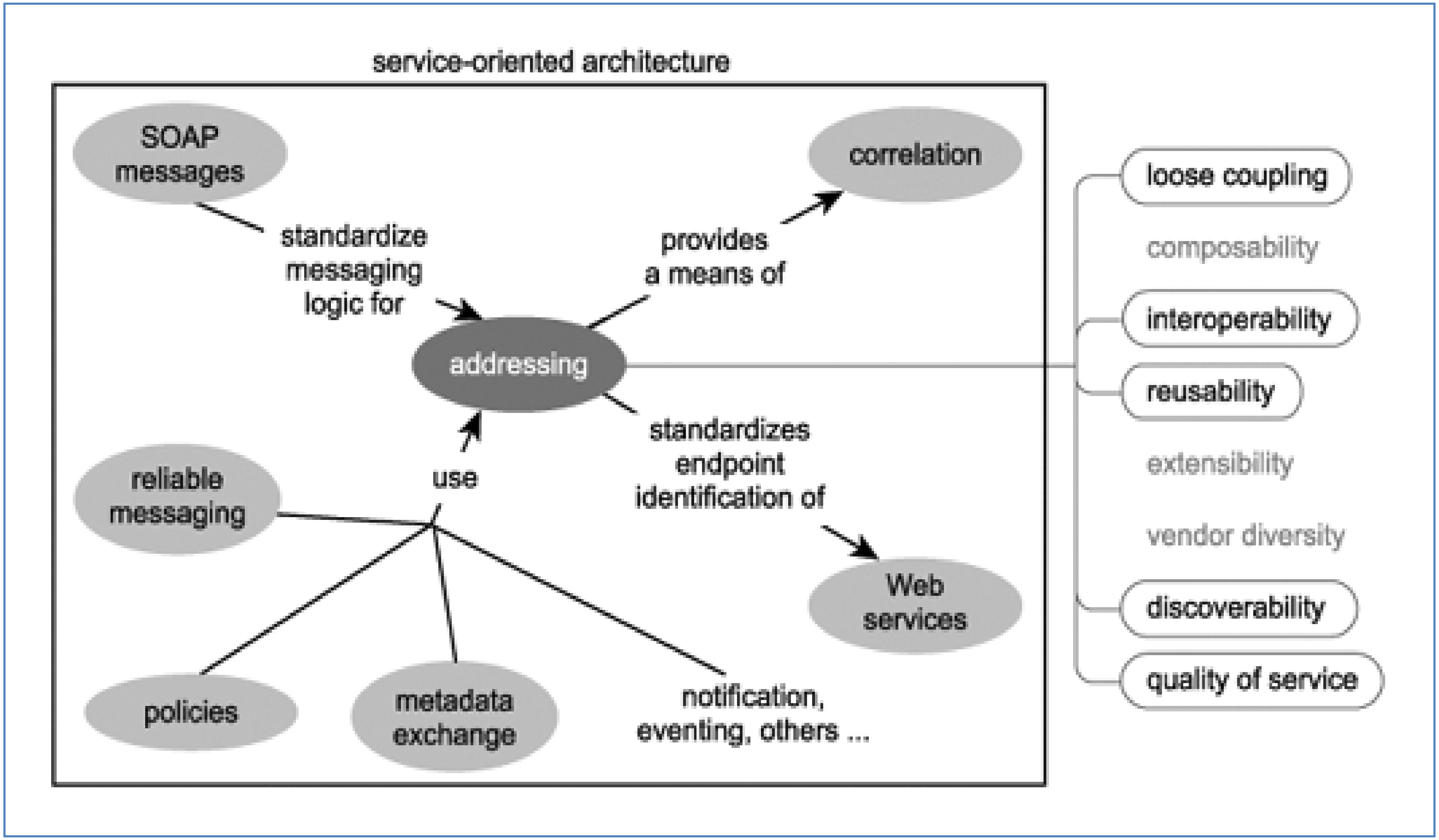
Message Information Headers

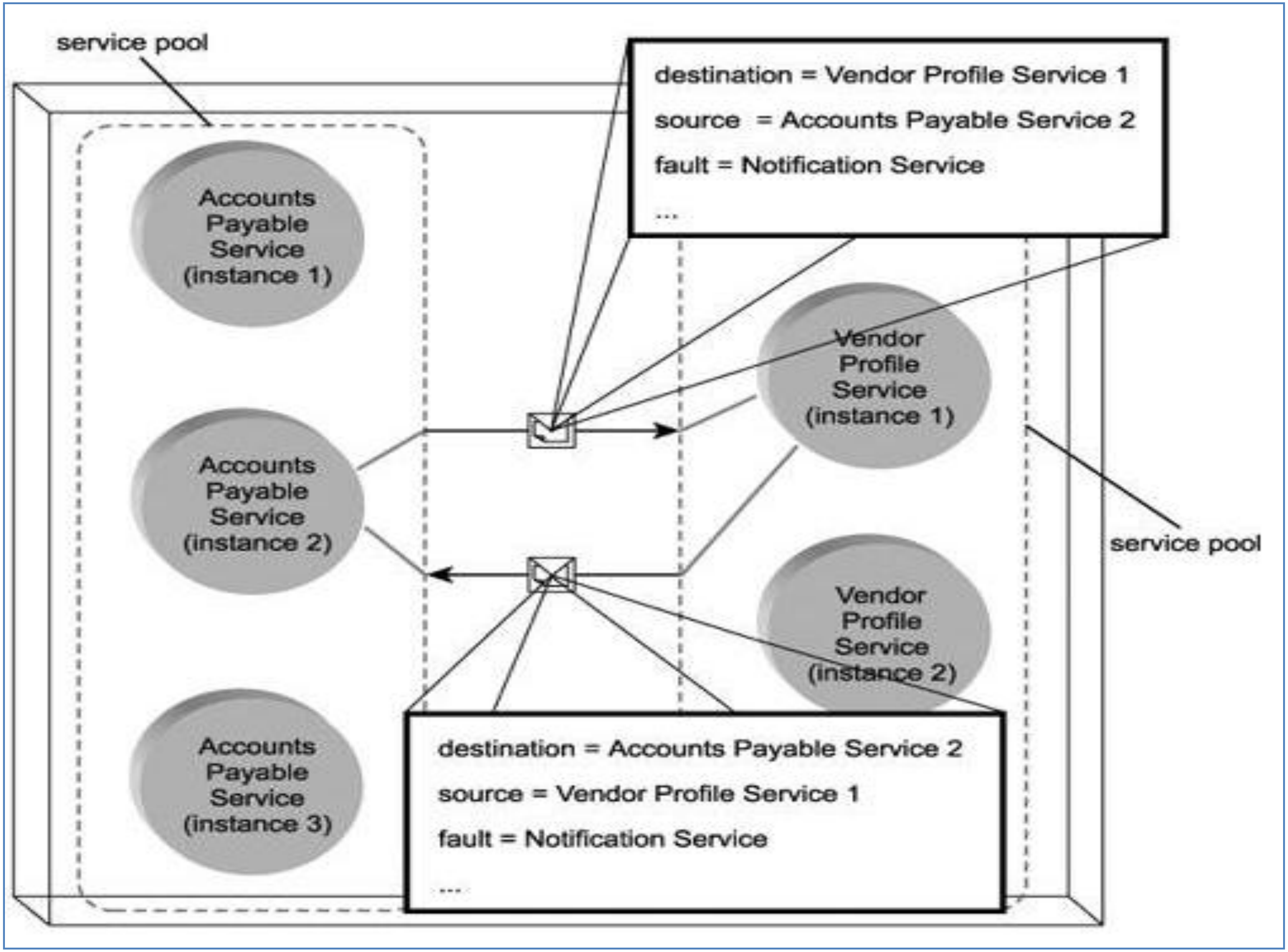
- Outfitting a SOAP message with these headers further increases its position as an independent unit of communication.
- Using MI headers, SOAP messages now can contain detailed information that defines the messaging interaction behavior of the service.
- The net result is standardized support for the use of unpredictable and highly flexible message exchanges, dynamically creatable and therefore adaptive and responsive to runtime conditions.

Addressing and Transport Protocol Independence

- In earlier days, once the message was sent, the message handling was left up to the transport layer protocols
- This approach is good for developers, but it has limitations
- WS-Addressing headers remove this protocol dependence
- SOAP message itself is in charge of its own destiny

Addressing and SOA




service pool
destination = Vendor Profile Service 1
source = Accounts Payable Service 2
fault = Notification Service
...
Accounts Payable Service (instance 1)
Vendor Profile Service (instance 1)
Accounts Payable Service (instance 2)
service pool
Vendor Profile Service (instance 2)
Accounts Payable Service (instance 3)
destination = Accounts Payable Service 2
source = Vendor Profile Service 1
fault = Notification Service
...

So far, WS-Addressing

- Addressing extensions, as implemented by the WS-Addressing specification, introduce two important concepts: <u>endpoint references and message information headers</u>.
- Endpoint references provide a standardized means of identifying a specific instance of a Web service.
- Message information headers add message exchange properties to a specific message, conveying interaction semantics to recipient services.
- Though simple in comparison to other WS-* specifications, WS-Addressing inserts a powerful layer of messaging autonomy within a service-oriented architecture.

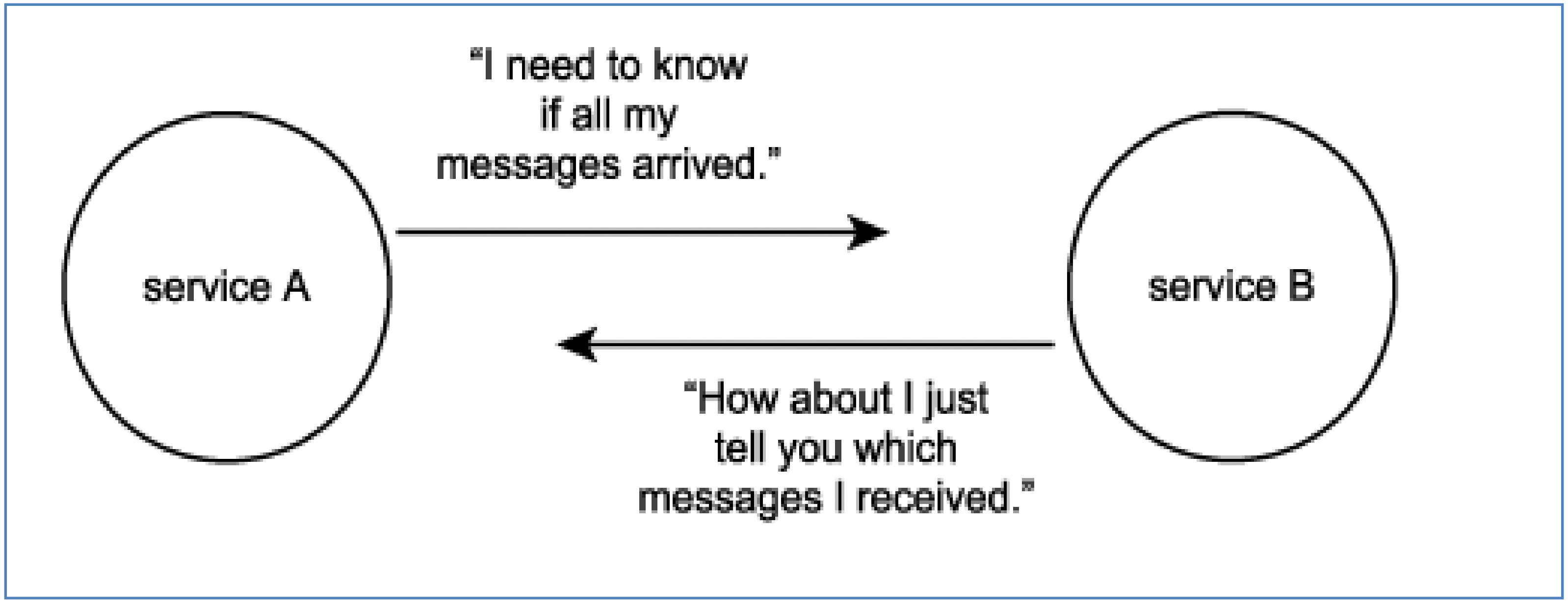
What is the side effect from benefits of loosely coupled messaging framework?

WS-Reliable Messaging

- The benefits of a loosely coupled messaging framework come at the cost of a loss of control over the actual communications process.
- After a Web service transmits a message, it has no immediate way of knowing:
  - whether the message successfully arrived at its intended destination
  - whether the message failed to arrive and therefore requires a retransmission
  - whether a series of messages arrived in the sequence they were intended to

WS-Reliable messaging

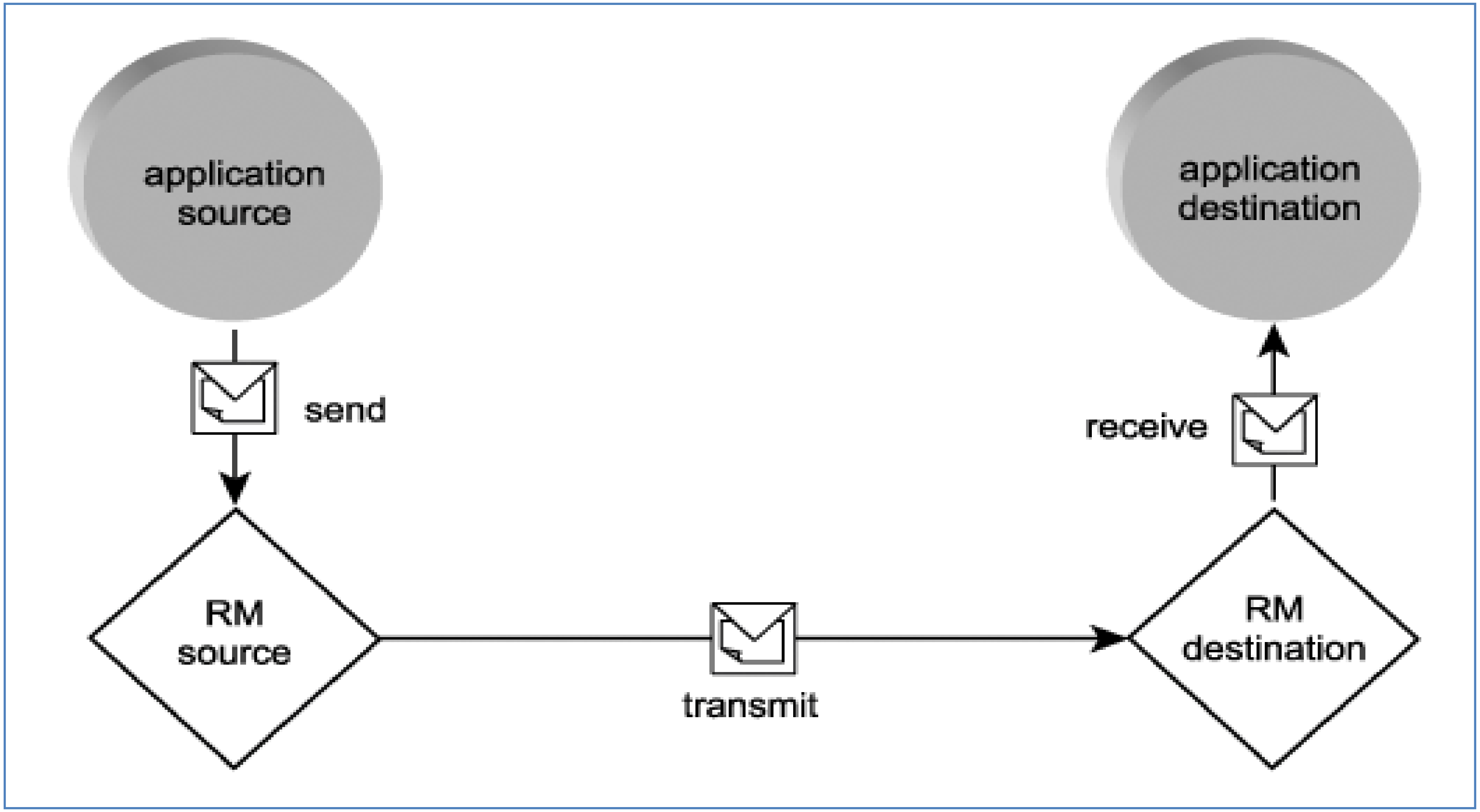
- Reliable messaging addresses certain concerns by establishing a measure of quality assurance that can be applied to other activity management frameworks .


"I need to know if all my messages arrived."
service A
service B
"How about I just tell you which messages I received."

WS-Reliable messaging

- WS-ReliableMessaging provides a framework capable of guaranteeing:
  - that service providers will be notified of the success or failure of message transmissions
  - that messages sent with specific sequence-related rules will arrive as intended (or generate a failure condition)
- Reliable messaging does not employ a coordinator service to keep track of the state of an activity; instead, all reliability rules are implemented as SOAP headers within the messages themselves.

RM Source, RM Destination, Application Source, and Application Destination



WS-Reliable messaging

- An application source is the service or application logic that sends the message to the RM source
- RM Source is the physical processor or node that performs the actual wire transmission.
- RM destination represents the target processor or node that receives the message and subsequently delivers it to the application destination

WS-Reliable messaging

- WS-ReliableMessaging makes a distinction between the parts of a solution that are responsible for initiating a message transmission and those that actually perform the transmission.

- It further assigns specific descriptions to the terms "send," "transmit," "receive," and "deliver," as they relate differently to these solution parts.

Sequence

- A sequence establishes the order in which messages should be delivered.
- Each message that is part of a sequence is labeled with a message number that identifies the position of the message within the sequence.
- The final message in a sequence is further tagged with a last message identifier.

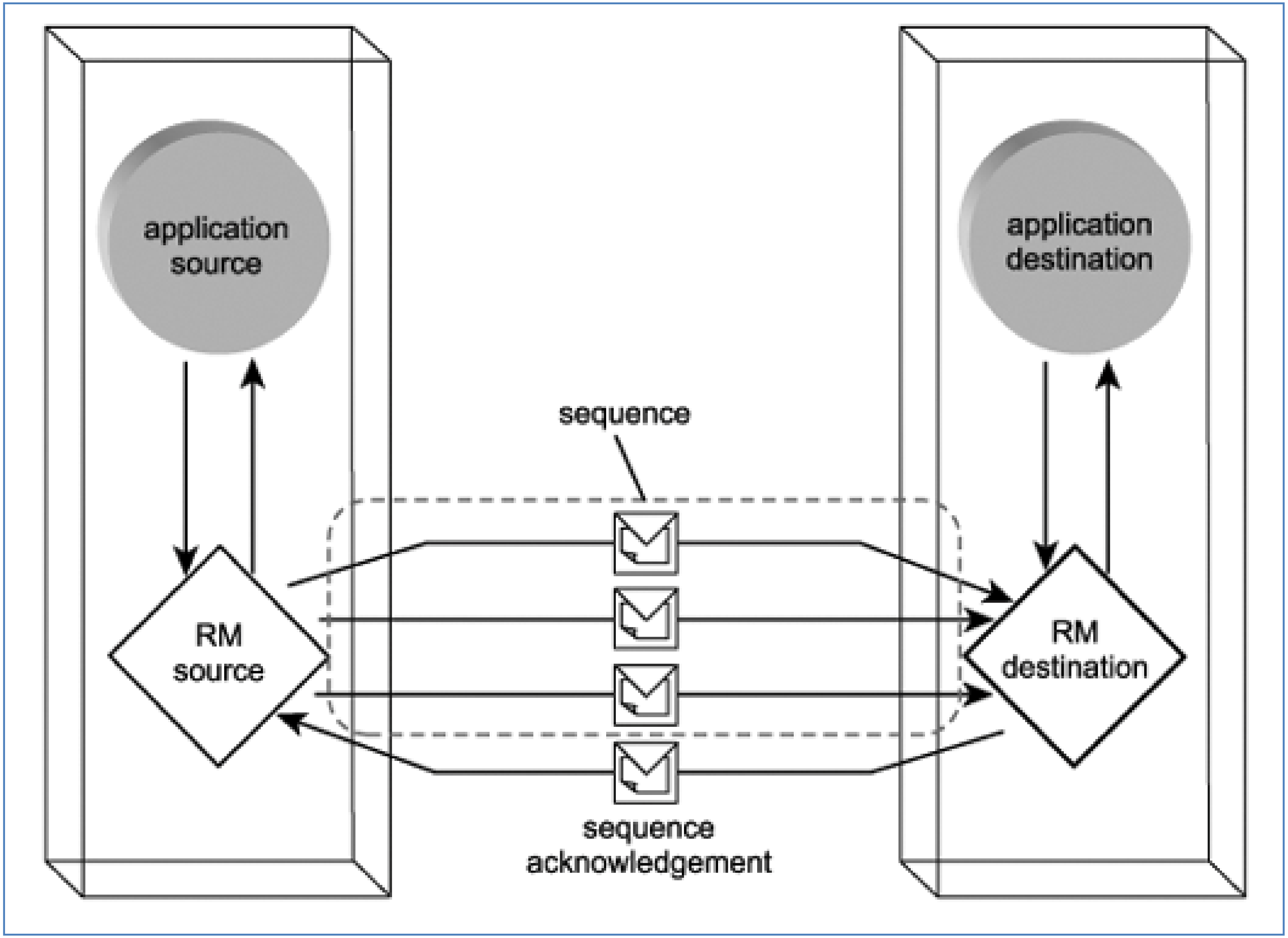
Sequence Acknowledgement

- A core part of the reliable messaging framework is a notification system used to communicate conditions from the RM destination to the RM source.
- Upon receipt of the message containing the last message identifier, the RM destination issues a sequence acknowledgement.
- The acknowledgement message indicates to the RM source which messages were received.

Acknowledgements

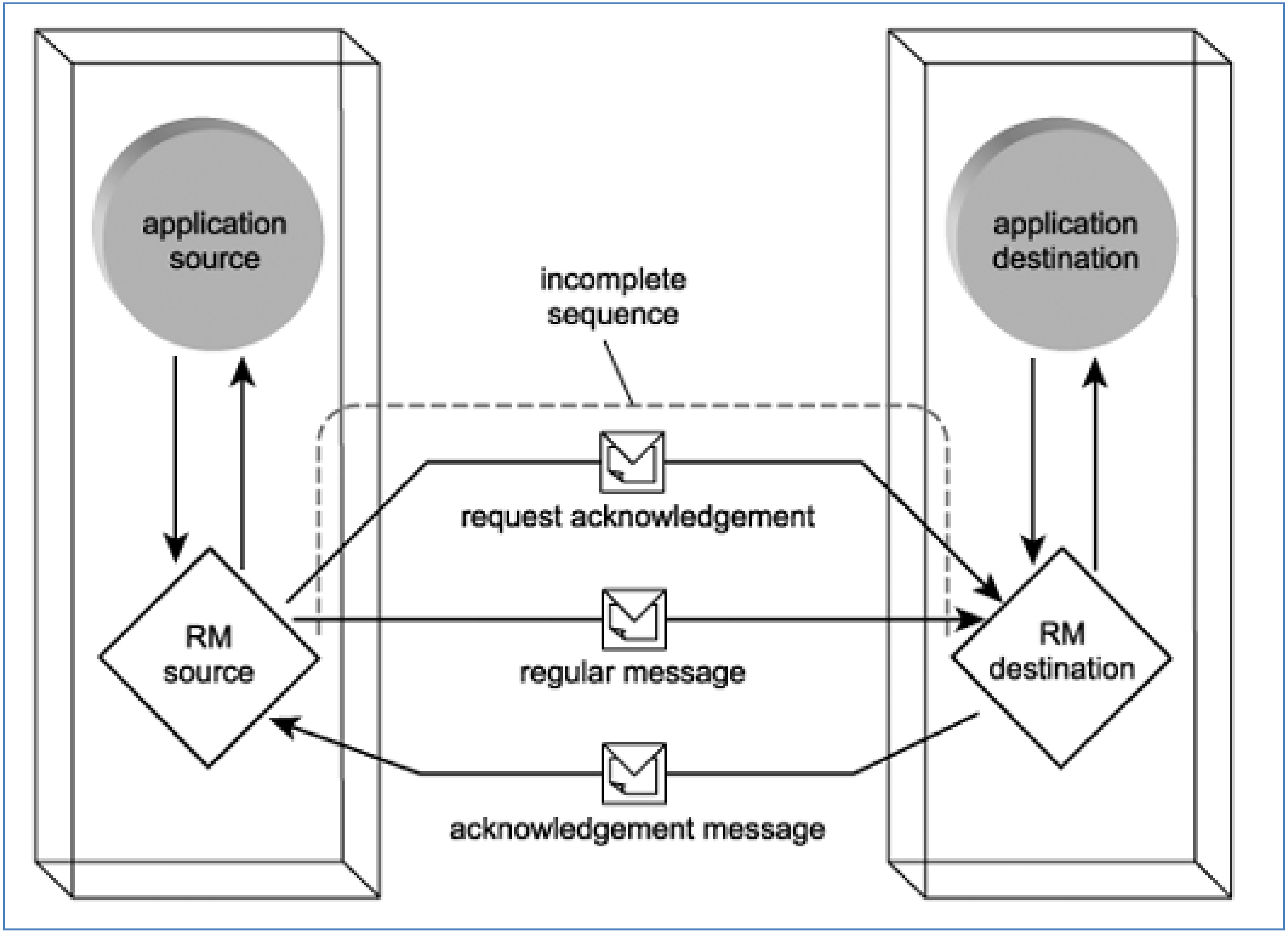
- It is up to the RM source to determine if the messages received are equal to the original messages transmitted.

- Missing messages can be retransmitted


application source
application destination
sequence
RM source
RM destination
sequence acknowledgement

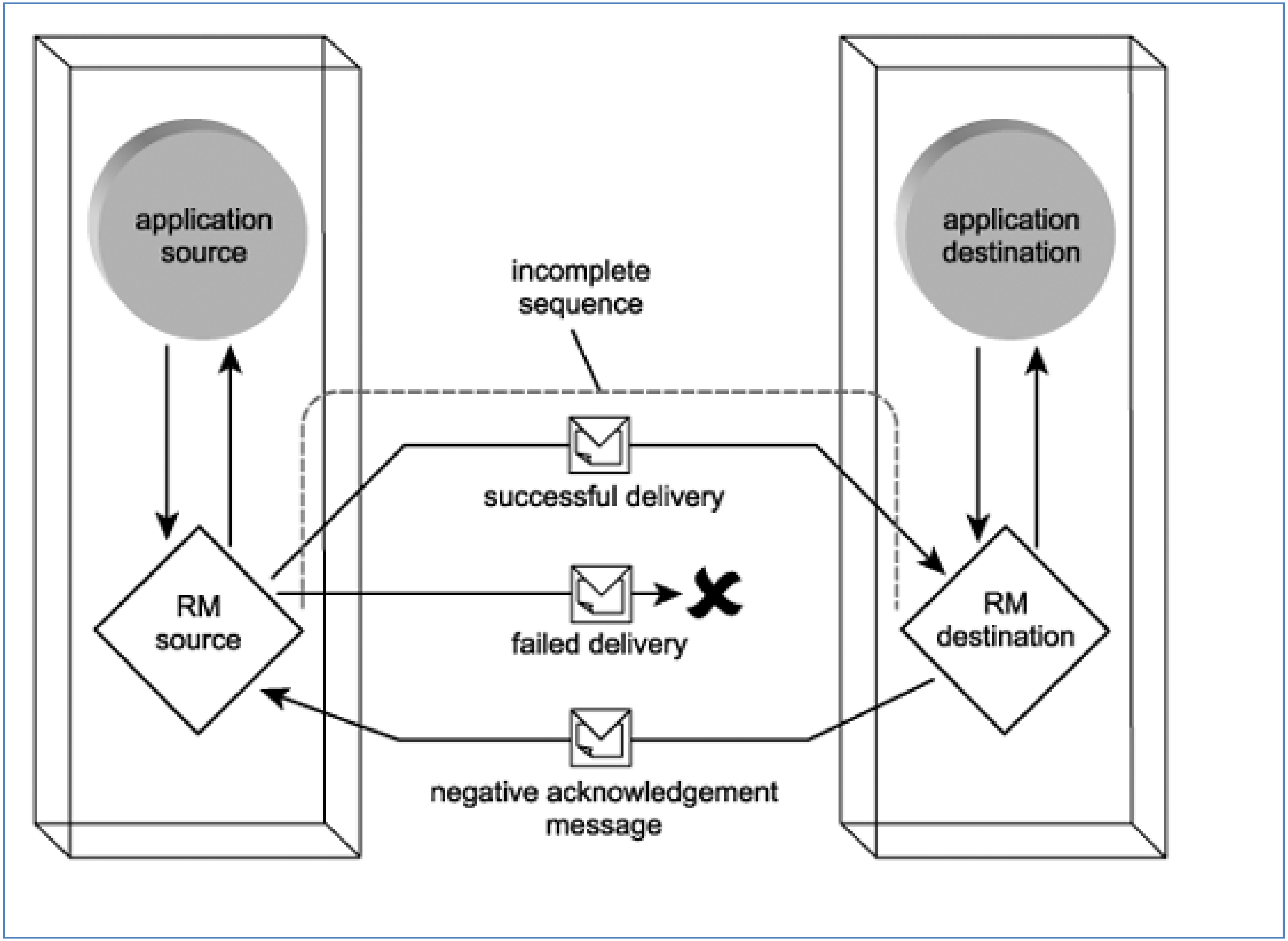
Request Acknowledgement

- RM source can request additional acknowledgement at any time by issuing request acknowledgement to RM destination


application source
application destination
incomplete sequence
request acknowledgement
RM source
regular message
RM destination
acknowledgement message

Negative Acknowledgement

- Negative Acknowledgement can be issued by RM destination to immediately indicate RM source that a failure has occurred

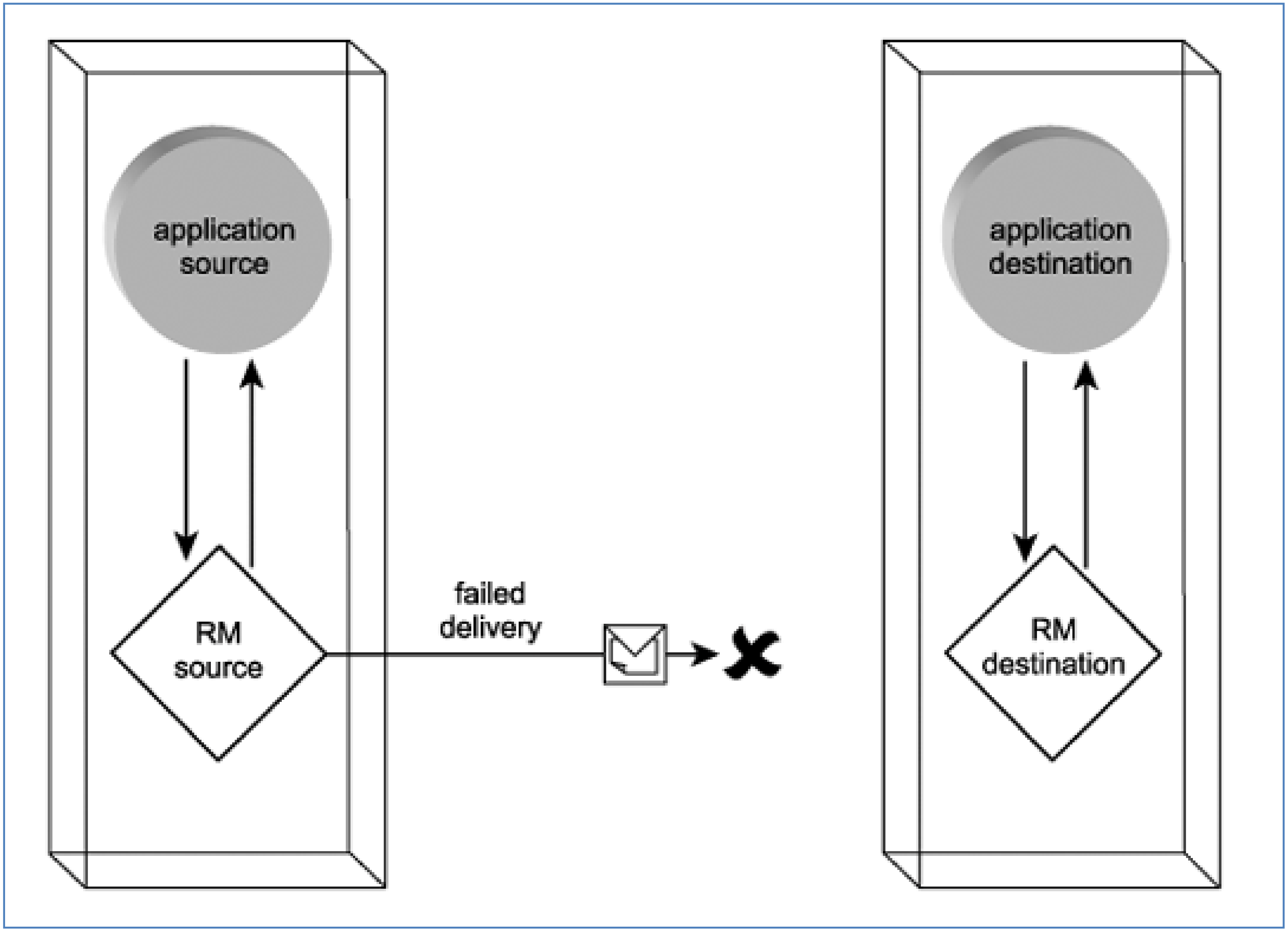

application
source
application
destination
incomplete
sequence
successful delivery
RM
source
failed delivery
RM
destination
negative acknowledgement
message

Delivery assurances

- The nature of a sequence is determined by a set of reliability rules known as delivery assurances.
- Delivery assurances are predefined message delivery patterns that establish a set of reliability policies.
  - AtMostOnce
  - AtLeastOnce
  - ExactlyOnce
  - Inorder

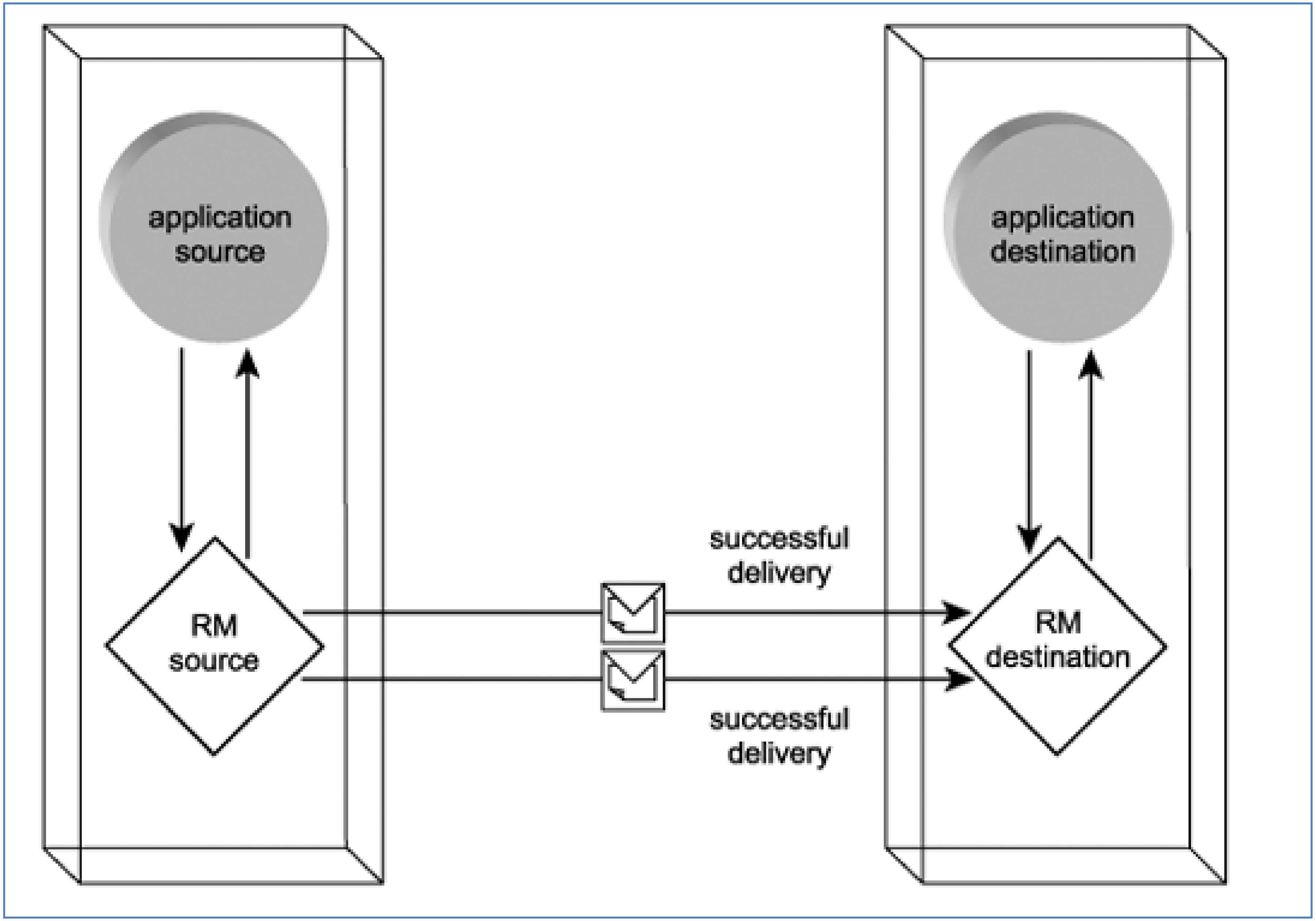
Delivery assurances

- **AtMostOnce:**
  - Ensures delivery of one or zero message
  - Error will be generated if more than one (duplicate) messages are delivered


application source
application destination
RM source
failed delivery
RM destination

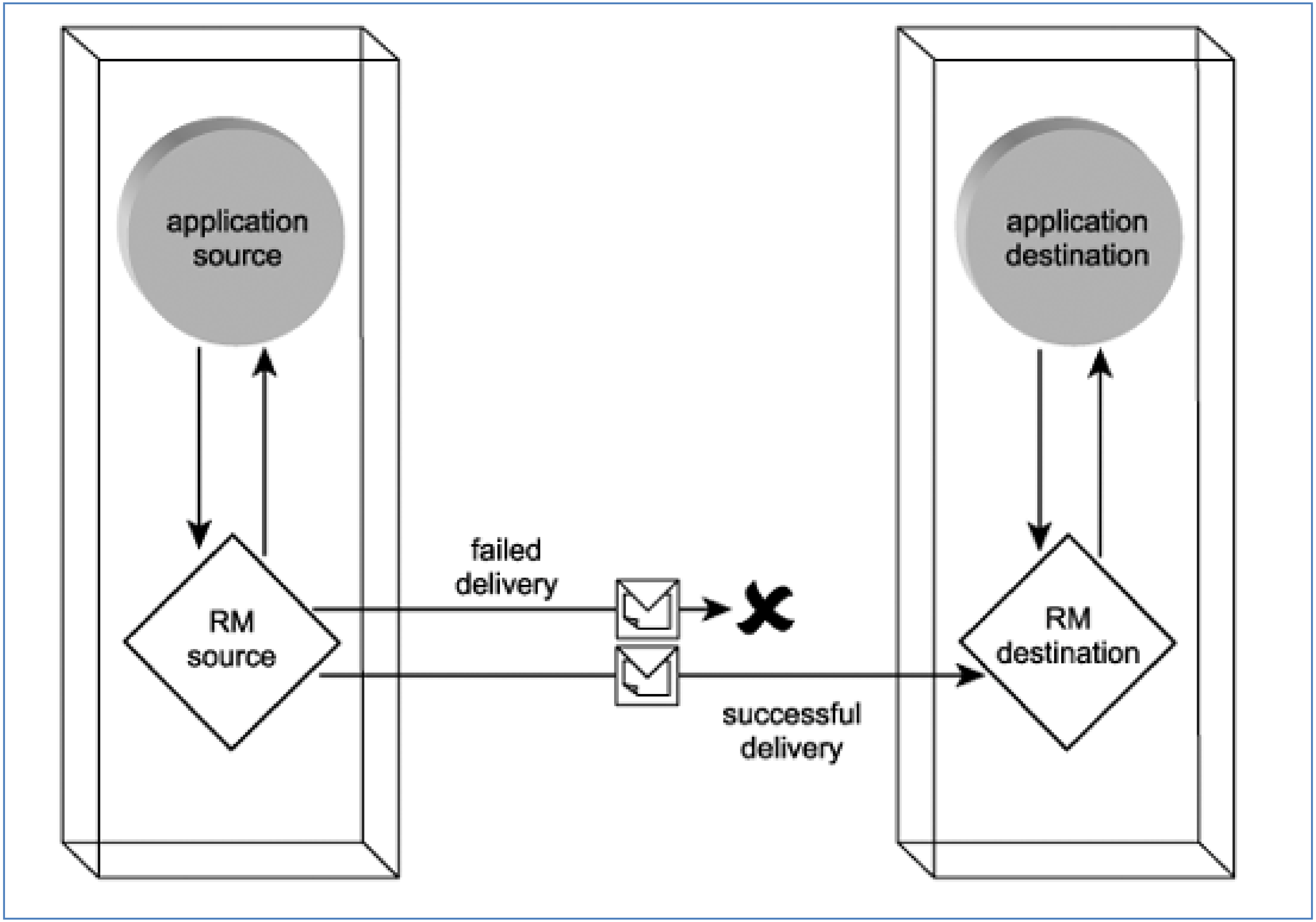
Delivery assurances

- **AtLeastOnce:**
  - Message can be delivered one or more time
  - Error will be generated if no messages are delivered


application source
application destination
successful delivery
RM source
RM destination
successful delivery

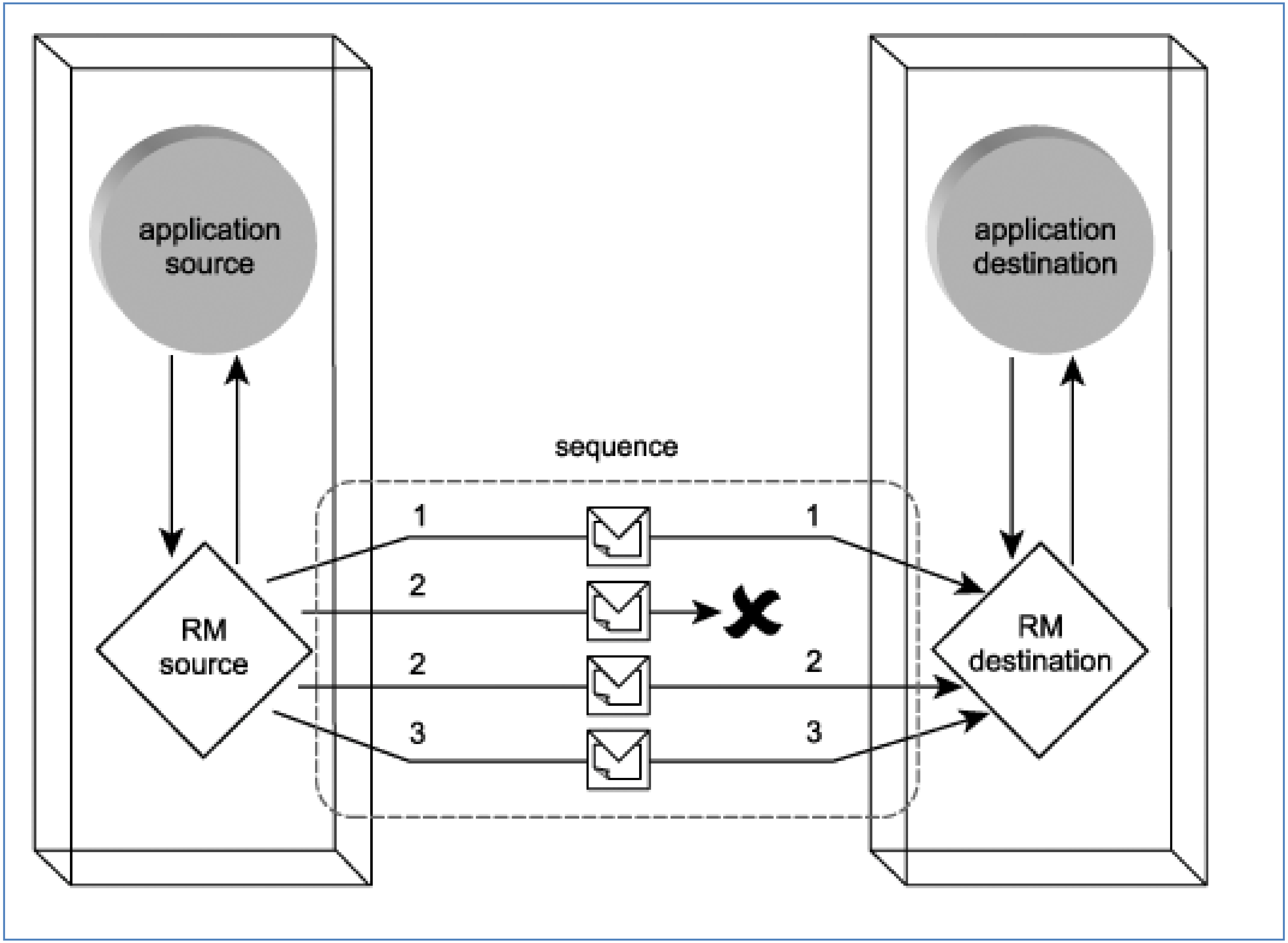
Delivery assurances

- **ExactlyOnce:**
  - Message will be delivered once
  - Error will be generated if zero or more than one (duplicate) messages are delivered


application source
application destination
failed delivery
RM source
RM destination
successful delivery

Delivery assurances

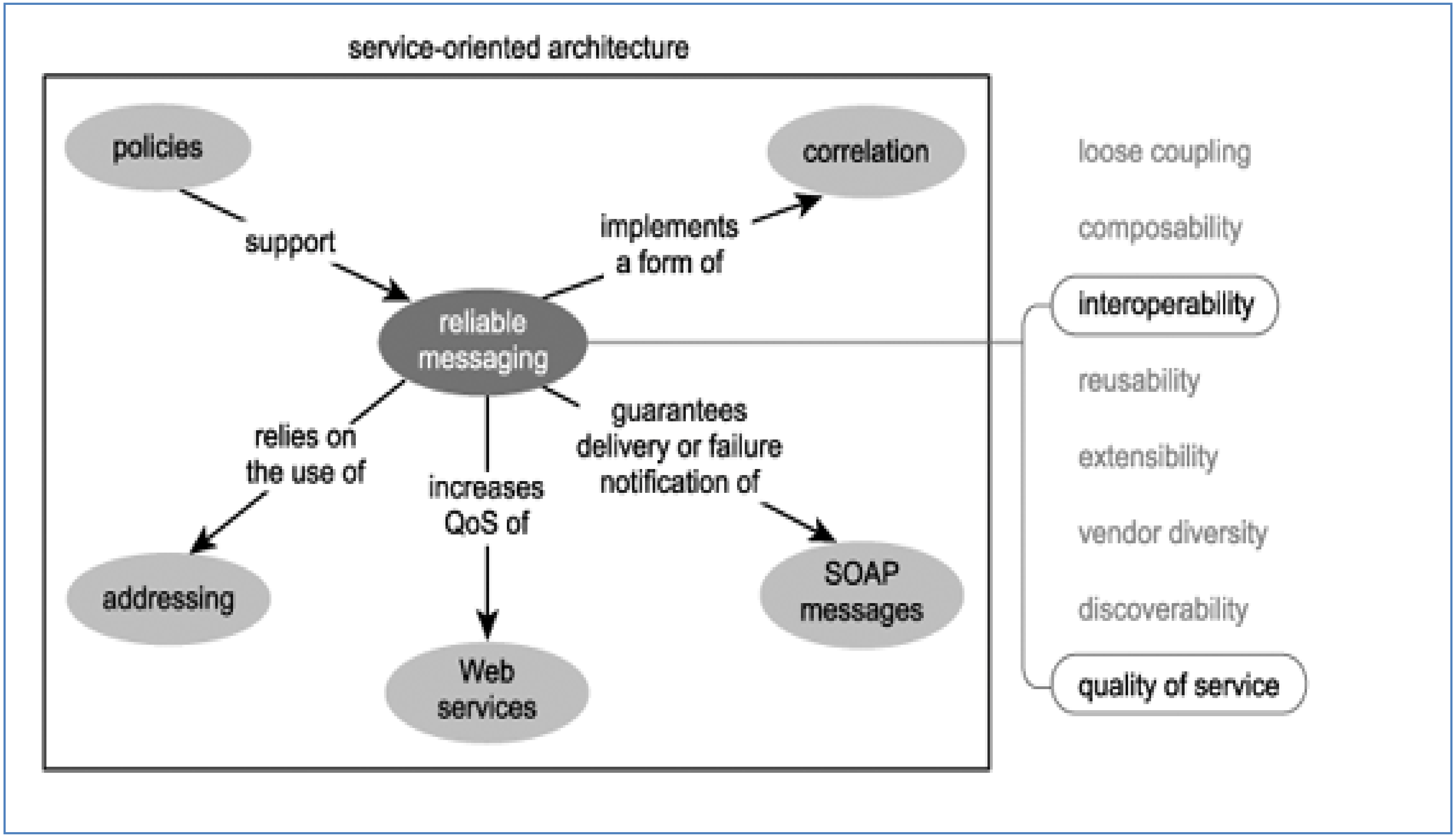
- **Inorder:**
  - Messages are delivered in specific sequence
  - Error will be generated if messages are delivered out of order

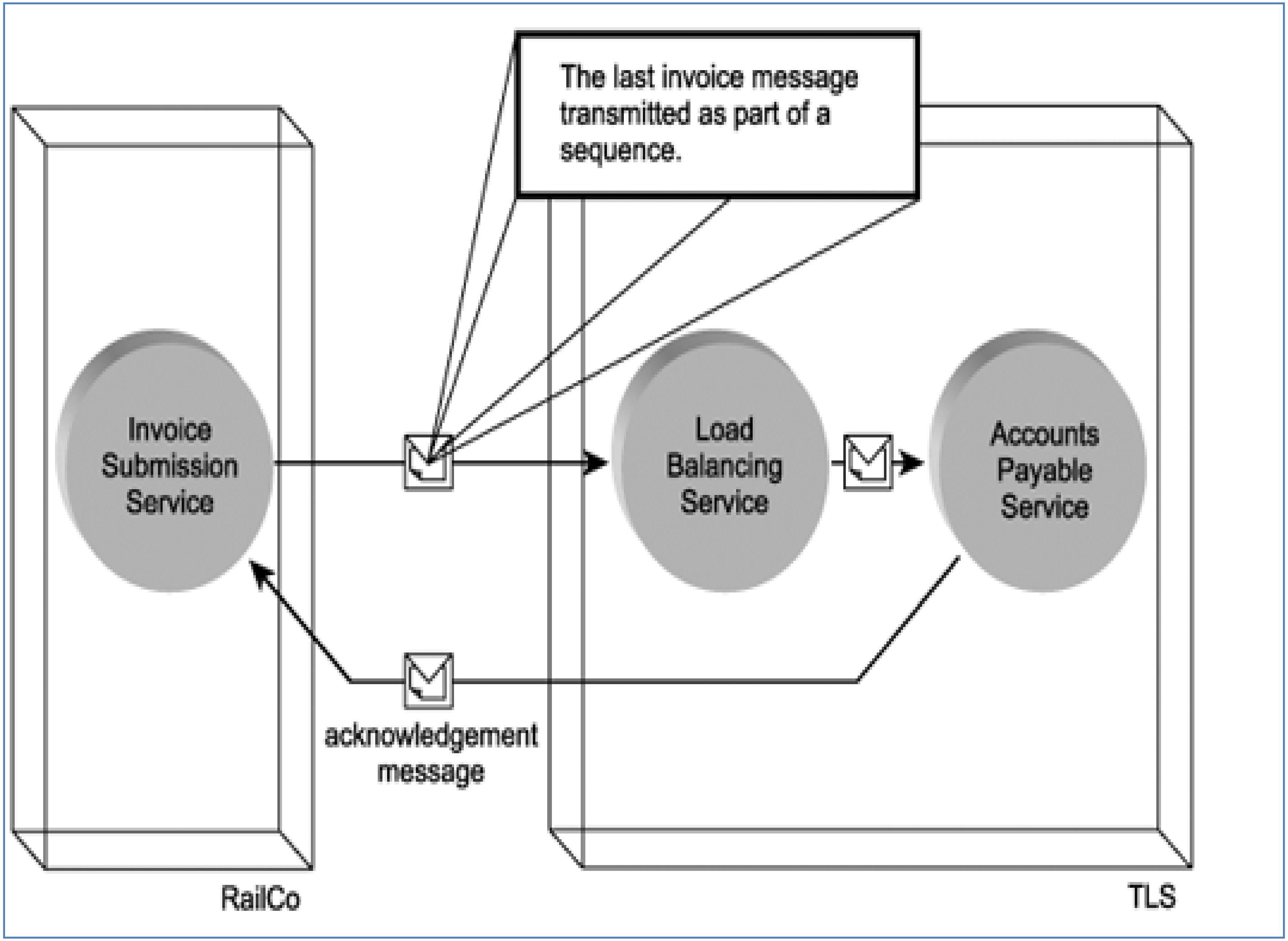

application source
application destination
sequence
1
1
2
2
2
3
3
RM source
RM destination

Reliable messaging and addressing

- WS-Addressing is closely tied to the WS-ReliableMessaging framework.
- In fact, it's interesting to note that the rules around the use of the WS-Addressing message id header were altered specifically to accommodate the WS-ReliableMessaging specification.
  - Message with same ID can be generated if retransmission is required

Reliable messaging and SOA




The last invoice message transmitted as part of a sequence.
Invoice Submission Service
Load Balancing Service
Accounts Payable Service
acknowledgement message
RailCo
TLS

So far, WS-Reliable messaging

- WS-ReliableMessaging establishes a framework that guarantees the delivery of a SOAP message or the reporting of a failure condition.
- The key parts of this framework are a notification system based on the delivery of acknowledgement messages and a series of delivery assurances that provide policies comprised of reliability rules.
- WS-ReliableMessaging is closely associated with the WS-Addressing and WS-Policy specifications.
- Reliable messaging significantly increases SOA's quality of service level and broadens its interoperability potential.