Orchestration and Choreography

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

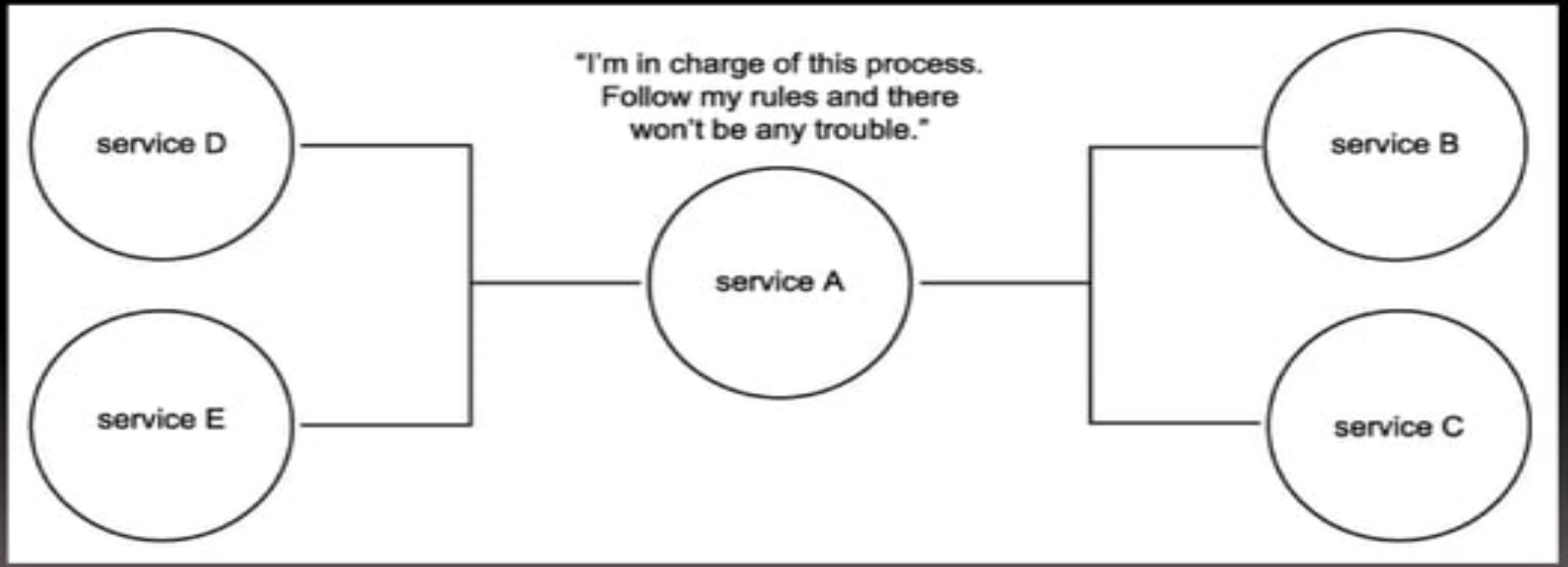
- A centrally controlled set of workflow logic facilitates interoperability between two or more different applications.
- A common implementation of orchestration is the hub-and-spoke model that allows multiple external participants to interface with a central orchestration engine.
- One of the driving requirements behind the creation of these solutions was to accommodate the merging of large business processes.
- With orchestration, different processes can be connected without having to redevelop the solutions that originally automated the processes individually.
- Orchestration bridges this gap by introducing new workflow logic.
- use of orchestration can significantly reduce the complexity of solution environments.
- Workflow logic is abstracted and more easily maintained than when embedded within individual solution components.

Orchestration

- Through the use of extensions that allow for business process logic to be expressed via services, orchestration can represent and express business logic in a standardized, services-based venue.
- When building service-oriented solutions, this provides an extremely attractive means of housing and controlling the logic representing the process being automated.
- Orchestration further leverages the intrinsic interoperability sought by service designs by providing potential integration endpoints into processes.
- Building upon orchestration logic standardizes process representation across an organization, while addressing the goal of enterprise federation and promoting service-orientation.

Orchestration

An orchestration controls almost every facet of a complex activity



WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

Business protocols and process definition

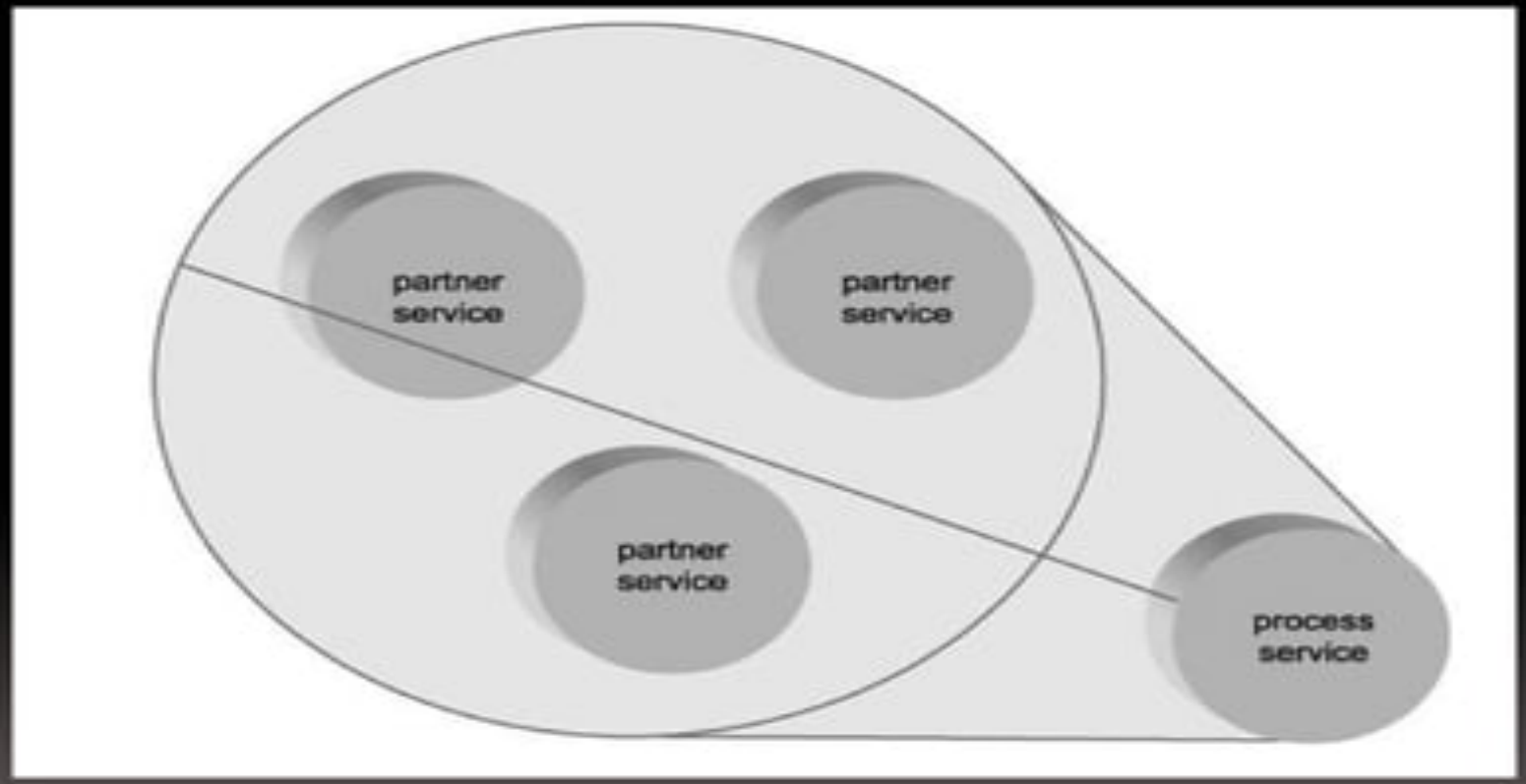
- The workflow logic that comprises an orchestration can consist of numerous business rules, conditions, and events.
- Collectively, these parts of an orchestration establish a business protocol that defines how participants can interoperate to achieve the completion of a business task.
- The details of the workflow logic encapsulated and expressed by an orchestration are contained within a process definition.

Process services and partner services

- Identified and described within a process definition are the allowable process participants.
- First, the process itself is represented as a service, resulting in a process service (which happens to be another one of our service models)

Orchestration

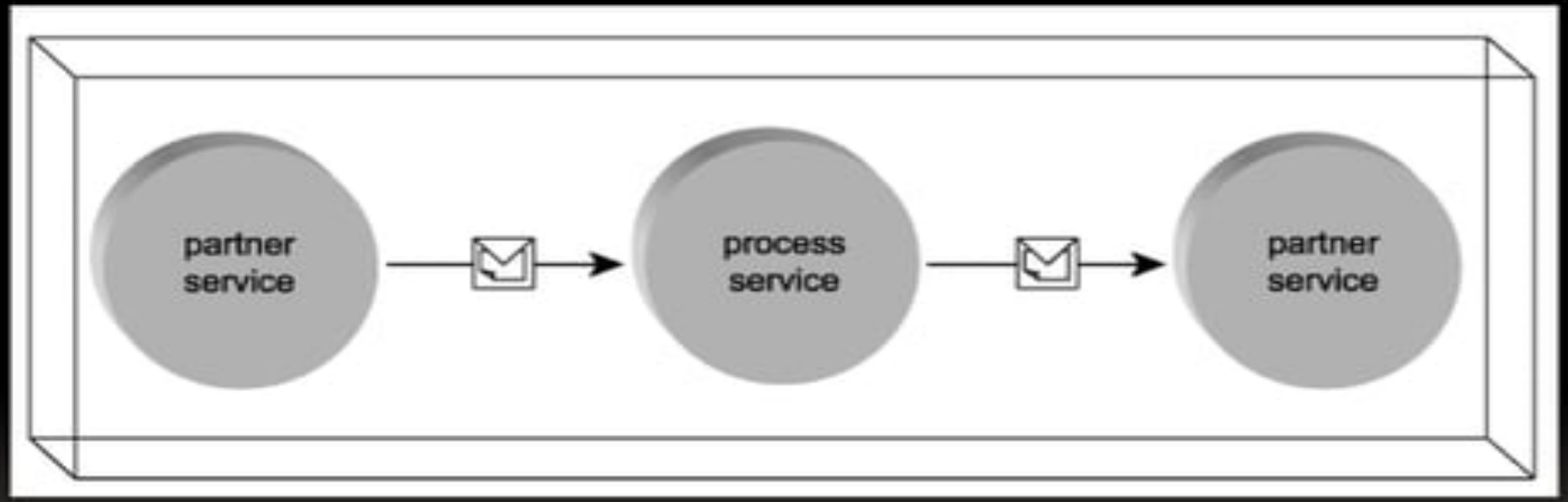
A process service coordinating and exposing functionality from three partner services



WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

The process service, after first being invoked by a partner service, then invokes another partner service



WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

Basic activities and structured activities

Web Services Business Process Execution Language (WS-BPEL),

- 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, pick).
- How these activities can be used to express actual business process logic

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

Sequences, flows, and links

- Basic and structured activities can be organized so that the order in which they execute is predefined.
- A sequence aligns groups of related activities into a list that determines a sequential execution order.
- Sequences are especially useful when one piece of application logic is dependent on the outcome of another.
- Flows also contain groups of related activities, but they introduce different execution requirements.
- Pieces of application logic can execute concurrently within a flow, meaning that there is not necessarily a requirement for one set of activities to wait before another finishes.
- However, the flow itself does not finish until all encapsulated activities have completed processing.
- This ensures a form of synchronization among application logic residing in individual flows.

Orchestration

Sequences, flows, and links

- Links are used to establish formal dependencies between activities that are part of flows.
- Before an activity fully can complete, it must ensure that any requirements established in outgoing links first are met.
- Similarly, before any linked activity can begin, requirements contained within any incoming links first must be satisfied.
- Rules provided by links are also referred to as synchronization dependencies

Orchestration

Orchestration and SOA

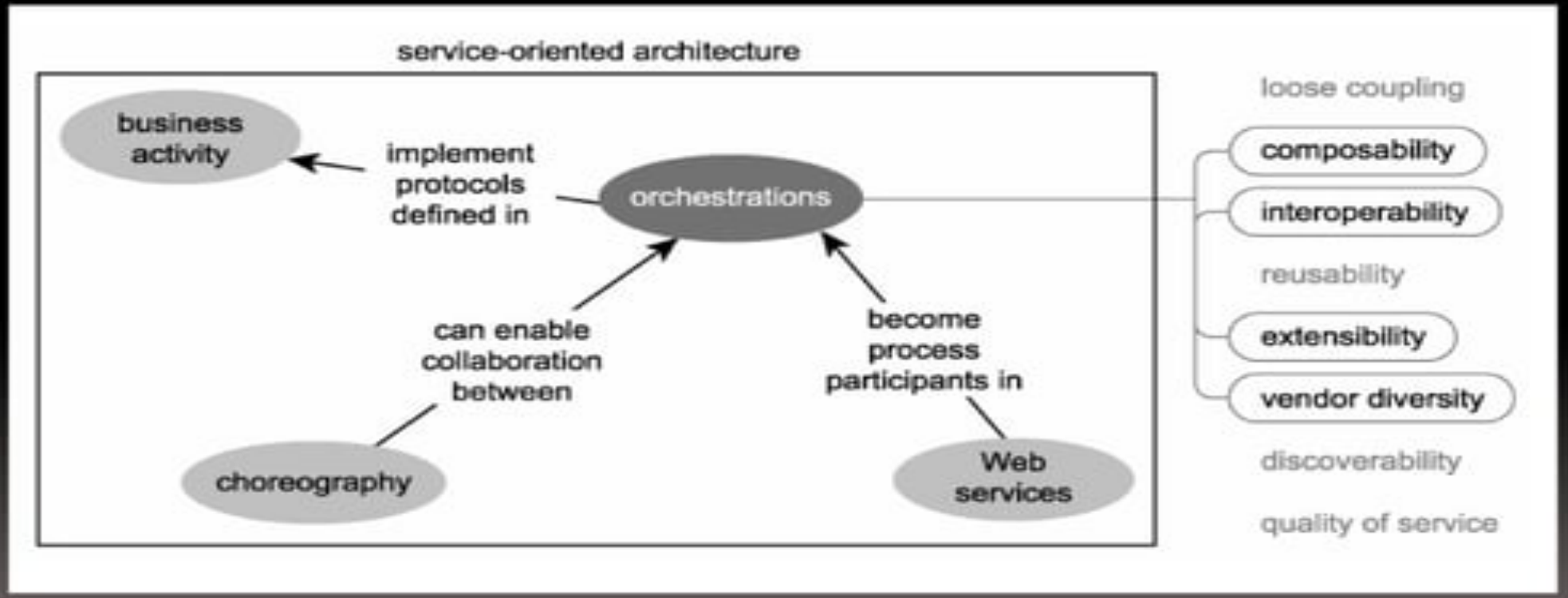
- Business process logic is at the root of automation solutions.
- Orchestration provides an automation model where process logic is centralized yet still extensible and composable .
- Through the use of orchestrations, service-oriented solution environments become inherently extensible and adaptive.
- Orchestrations themselves typically establish a common point of integration for other applications, which makes an implemented orchestration a key integration enabler.

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Orchestration

Orchestration and SOA

Orchestration relating to other parts of SOA



Orchestration

In Plain English

After successfully washing several cars together, Chuck, Bob, Jim, and I decide to start our own company. We formalize the steps in our car washing process so that we can handle different types of cars with different cleaning requirements.

Our process is therefore affected by the following new requirements:

- We decide to hire extra help during peak hours. This introduces up to two additional members that join our team.
- Because we have no venture capital for this business, we make an arrangement with a local gas station. In exchange for using a portion of their lot for our car washing operation, we agree to help out with the gas pumping duties during their peak hours.

Our simple car washing process now has become significantly more complicated. The process is no longer fixed in that it can change at any given time as a result of various conditions and events.

- When our extra workers arrive, the task allocation of the entire team is altered.
- When gas station personnel need extra help, we are obligated to send one or more of our car washing team members to assist them.

These examples relate to predictable conditions that occur on a daily basis. Our operation is further affected by some constraints:

- If our cash flow falls below a certain amount, we are unable to afford part-time workers.
- If it rains, all work is suspended (also leading to reduced cash flow).

These constraints introduce conditions that are less common, but which we always need to take into consideration. To accommodate these potential situations, we come up with a plan that maps out our expanded process and provides alternative processes for dealing with both common and uncommon conditions.

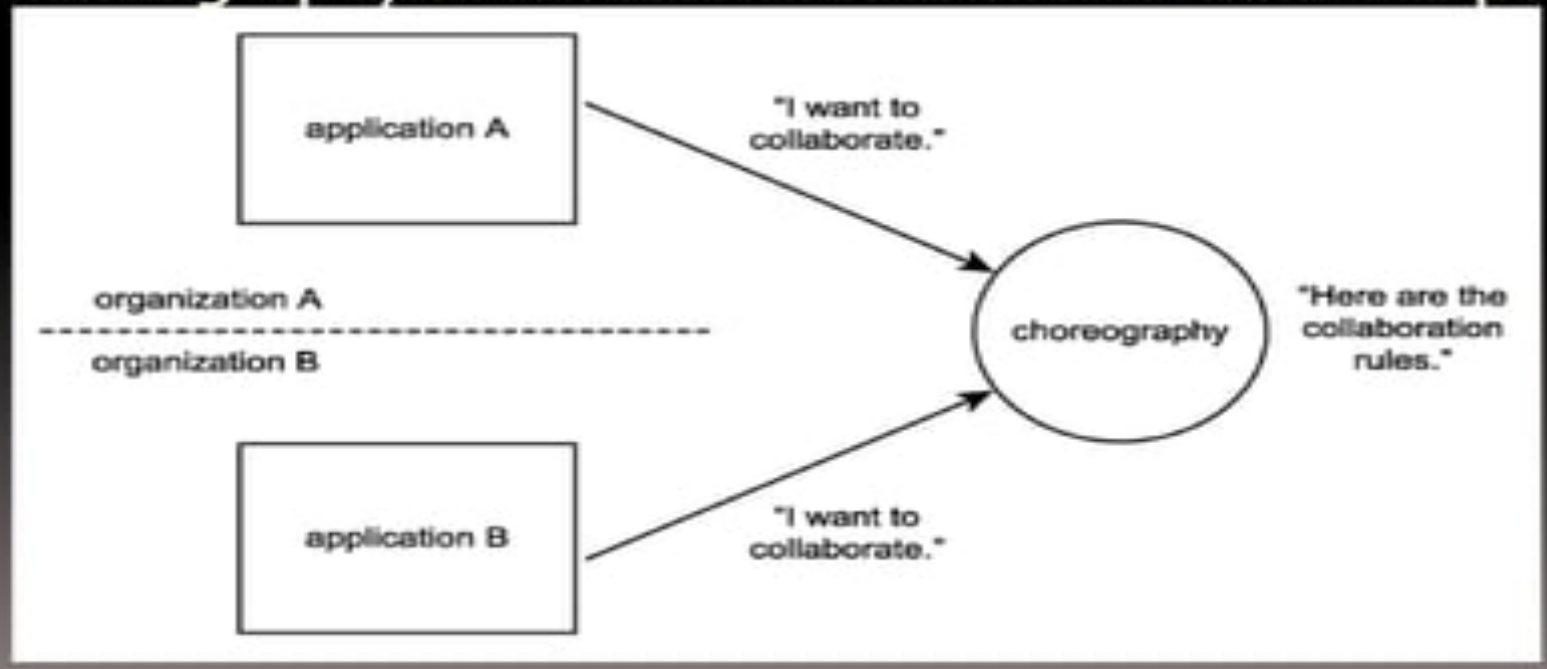
This plan is essentially a workflow that joins individual steps with processes and sub-processes partitioned by decision points. This elaborate workflow incorporates our original process with the gas station's process and the extended process resulting from the arrival of our part-time workers. This workflow is essentially an orchestration that manages the individual process requirements and related resources, participants, events, business rules, and activities.

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Choreography

- multiple services from different organizations need to work together to achieve a common goal
- The Web Services Choreography Description Language (WS-CDL) is one of several specifications that attempts to organize information exchange between multiple organizations (or even multiple applications within organizations), with an emphasis on public collaboration

A choreography enables collaboration between its participants



Choreography

In Plain English

After a few months in operation, our little car washing enterprise achieves some success. With our flexible and adaptive system, we have been able to efficiently wash enough cars to make some profit. Once word in the car washing circles gets around, we are contacted by a nearby car washing company.

Even though this team of car washers is located only a kilometer away, they are not considered competitors. We positioned ourselves at a gas station located at the off ramp of a highway, and they are on the other side. Our customers originate from North-bound traffic, whereas theirs come from cars heading South. As a result, we have different peak hours corresponding directly to the traffic patterns of the highway. Our volume peaks during the morning rush hours, whereas theirs peaks in the afternoon.

It is suggested to us that we could form a partnership whereby we pool our respective resources (workers) to allow each of our companies to maximize the potential of each rush hour period. This form of collaboration appeals to us, as so far we've never been able to wash as many cars as we could at peak times. When customers entering the gas station grounds see a line up to our car wash, they often change their minds and drive away.

We decide to join forces with the other team. However, this arrangement soon affects our original business process. We now have to introduce a process that imposes new conditions and constraints. At the same time, though, we want to protect our existing system because it has been successful. After discussing these issues with our new partner, we come to an agreement that results in a flexible collaboration process.

A choreography is essentially a collaboration process designed to allow organizations to interact in an environment that is not owned by any one partner.

Choreography

Collaboration

- An important characteristic of choreographies is that they are intended for public message exchanges.
- The goal is to establish a kind of organized collaboration between services representing different service entities, only 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. Roles can be bound to WSDL definitions, and those related are grouped accordingly, categorized as participants (services).

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Choreography

Relationships and channels

- Every action that is mapped out within a choreography can be broken down into a series of message exchanges between two services.
- Each potential exchange between two roles in a choreography is therefore defined individually as a relationship.
- Every relationship consequently consists of exactly two roles.
- Now it is defined who can talk with each other and requires establishing the nature of the conversation.
- Channels do exactly that by defining the characteristics of the message exchange between two specific roles.

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Choreography

Relationships and channels

- To facilitate more complex exchanges involving multiple participants, channel information can actually be passed around in a message.
- This allows one service to send another the information required for it to be communicated with by other services.
- This is a significant feature of the WS-CDL specification, as it fosters dynamic discovery and increases the number of potential participants within large-scale collaborative tasks.

Choreography

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 which impose rules and constraints that must be adhered to for an interaction to successfully complete.

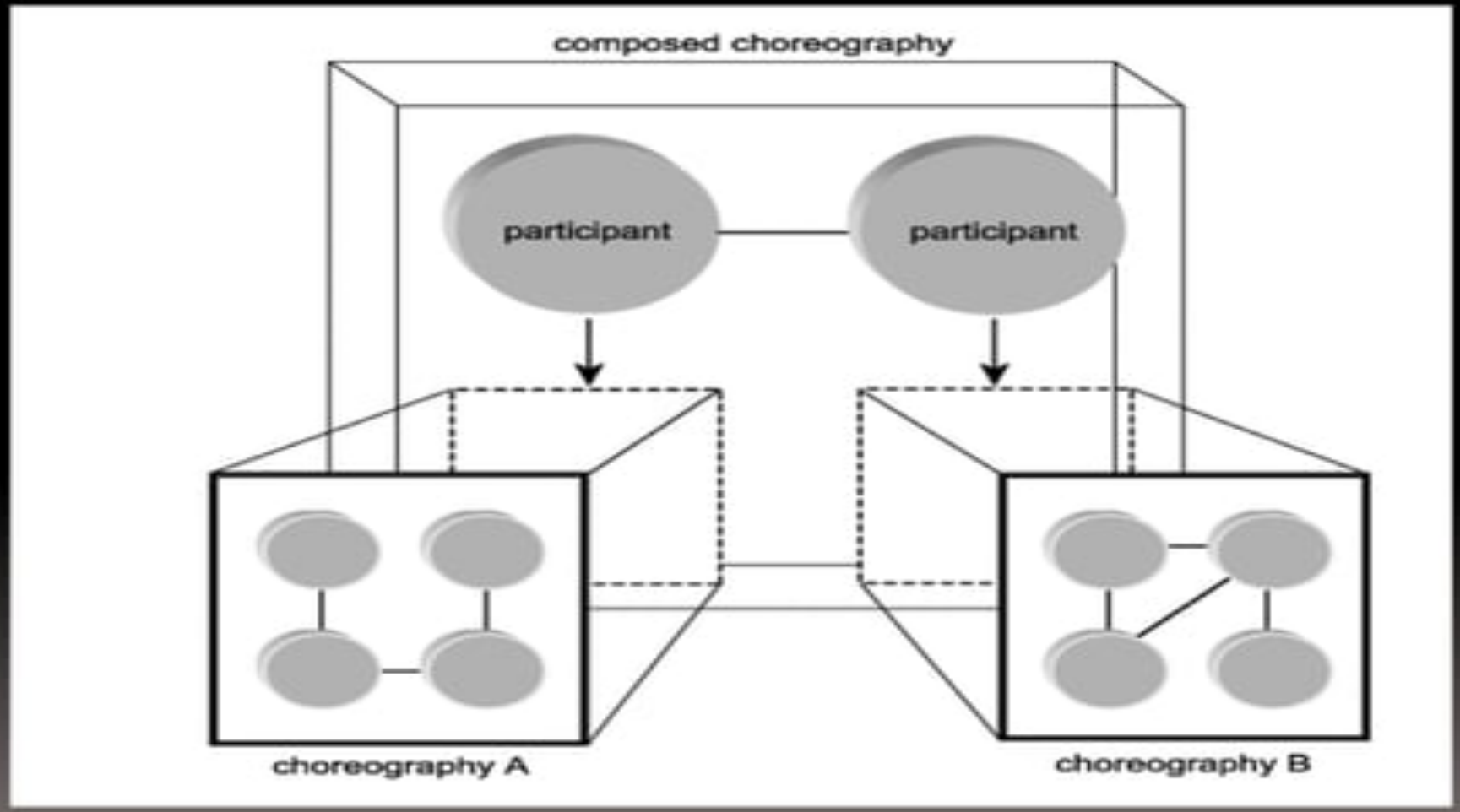
Choreography

Reusability, composability, and modularity

- Each choreography can be designed in a reusable manner, allowing it to be applied to different business tasks comprised of the same fundamental actions.
- using an import facility, a choreography can be assembled from independent modules.
- These modules can represent distinct sub-tasks and can be reused by numerous different parent choreographies

Choreography

A choreography composed of two smaller choreographies



Choreography

Orchestrations and choreographies

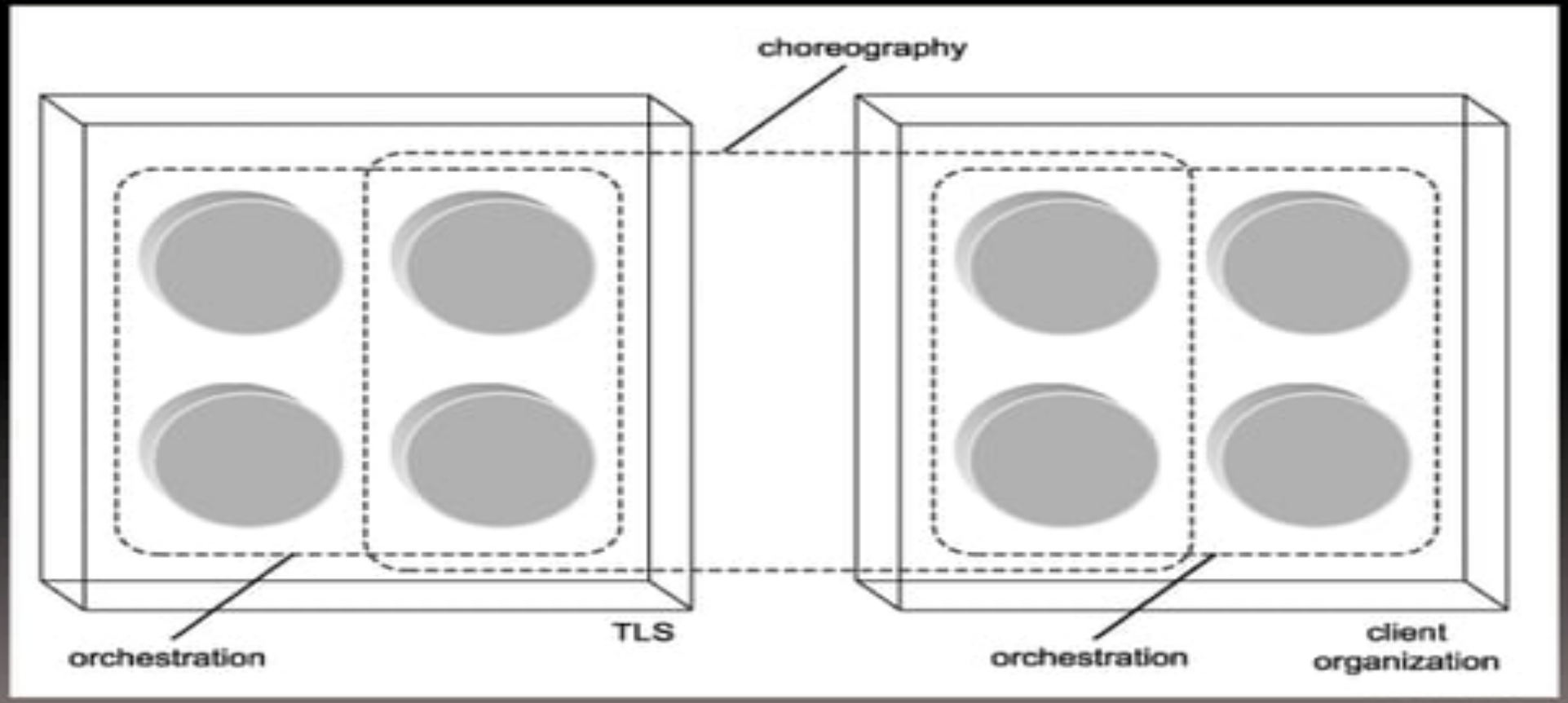
- While both represent complex message interchange patterns, there is a common distinction that separates the terms "orchestration" and "choreography."
- An orchestration expresses organization-specific business workflow.
- This means that an organization owns and controls the logic behind an orchestration, even if that logic involves interaction with external business partners.
- A choreography, on the other hand, is not necessarily owned by a single entity.
- It acts as a community interchange pattern used for collaborative purposes by services from different provider entities

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Choreography

Orchestrations and choreographies

A choreography enabling collaboration between two different orchestrations



Choreography

Orchestrations and choreographies

- 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.
- One overlap between the current orchestration and choreography extensions is the fact that orchestrations can be designed to include multi-organization participants.
- An orchestration can effectively establish cross-enterprise activities in a similar manner as a choreography.
- primary distinction is the fact that an orchestration is generally owned and operated by a single organization

Choreography

Choreography and SOA

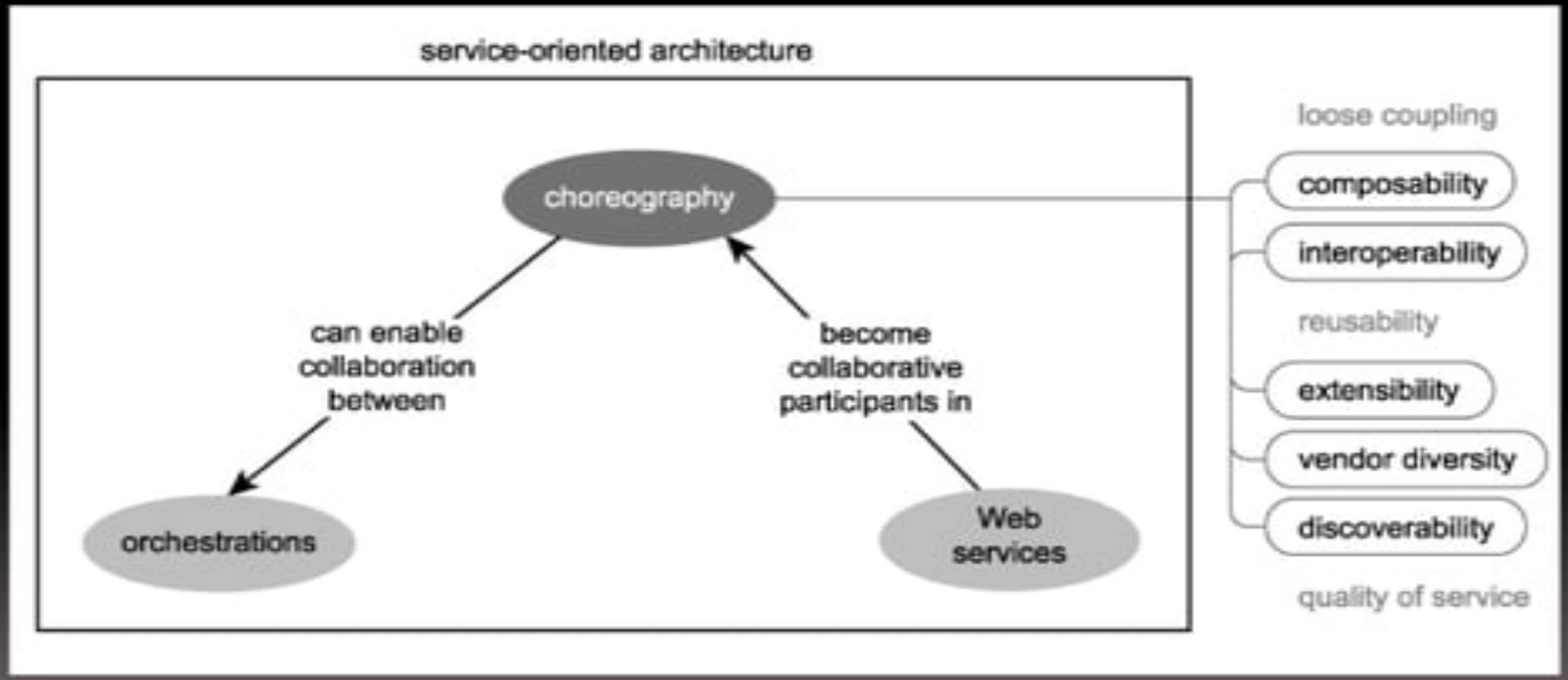
- Two services within a single organization, each exposing a simple function, can interact via a basic MEP to complete a simple task.
- Two services belonging to different organizations, each exposing functionality from entire enterprise business solutions, can interact via a basic choreography to complete a more complex task.
- Both scenarios involve two services, and both scenarios support SOA implementations.
- Choreography therefore can assist in the realization of SOA across organization boundaries .
- While it natively supports composability, reusability, and extensibility, choreography also can increase organizational agility and discovery.
- Organizations are able to join into multiple online collaborations, which can dynamically extend or even alter related business processes that integrate with the choreographies.

WEB SERVICES AND CONTEMPORARY SOA (PART I: ACTIVITY MANAGEMENT AND COMPOSITION)

Choreography

Choreography and SOA

Choreography relating to other parts of SOA.

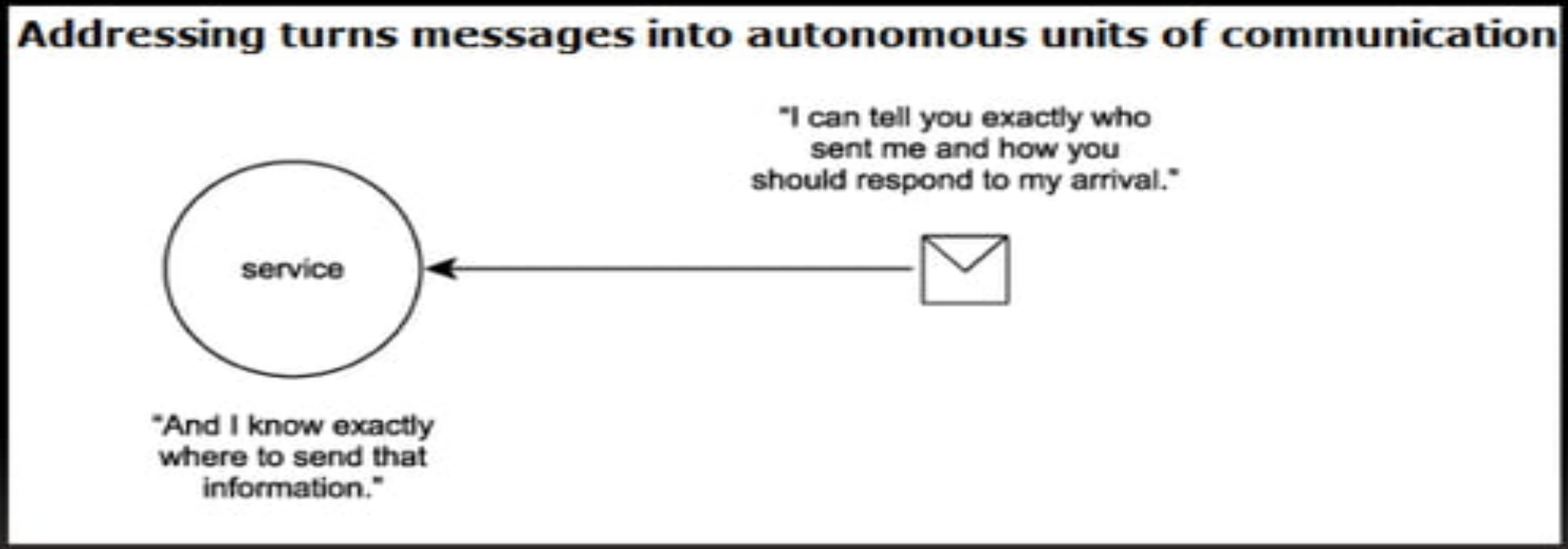


WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Addressing

- What addressing brings to SOAP messaging is much like what a waybill brings to the shipping process.
- Regardless of which ports, warehouses, or delivery stations a package passes through en route to its ultimate destination, with a waybill attached to it, everyone it comes into contact with knows:
  - where it's coming from
  - the address of where it's supposed to go
  - the specific person at the address who is supposed to receive it
  - where it should go if it can't be delivered as planned
- The WS-Addressing specification implements these addressing features by providing two types of SOAP headers .
- Addressing extensions are integral to SOA's underlying messaging mechanics.
- Many other WS-* specifications implicitly rely on the use of WS-Addressing

Addressing


Addressing turns messages into autonomous units of communication
"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."

Addressing

In Plain English

As our car washing company grows, so do the administration duties. Every week Chuck reviews the mail and takes care of necessary paperwork. This week he receives two letters: one from our insurance company and the other from the tax office.

The first letter includes our renewed insurance policy statement, along with an invoice for another year of coverage. The "from" address on this letter is simply the name and location of the insurance company's head office. The enclosed statement contains a letter written by our account representative, outlining some of the changes in this year's policy and requesting that we mail our check directly to him. Chuck therefore encloses our payment in an envelope with a "to" address that includes an "attention" line stating that this letter should be delivered directly to the account representative.

The next letter contains another bill. This time, it's a tax statement accompanied by a letter of instruction and two return envelopes. According to the instructions, we are to use the first envelope (addressed to the A/R office) to mail a check if we are paying the full amount owing. If we cannot make a full payment, we need to use the second envelope (addressed to the collections department) to send whatever funds we have.

These scenarios, in their own crude way, demonstrate the fundamental concepts of endpoint references and message information headers, which are explained in the following sections.

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Addressing

Endpoint references

- For a service requestor to contact a service provider, the provider's WSDL definition is needed.
- This document supplies the requestor with an address at which the provider can be contacted.
- What if, though, the service requestor needs to send a message to a specific instance of a service provider? In this case, the address provided by the WSDL is not sufficient.
- Traditional Web applications had different ways of managing and communicating session identifiers.
- The most common approach was to append the identifier as a query string parameter to the end of a URL.
- While easy to develop, this technique resulted in application designs that lacked security and were non-standardized.

Addressing

Endpoint references

- The concept of addressing introduces the endpoint reference, an extension used primarily to provide identifiers that pinpoint a particular instance of a service (as well as supplementary service metadata).

- The endpoint reference is expected to be almost always dynamically generated and can contain a set of supplementary properties.

Addressing- Endpoint references

An endpoint reference consists of the following parts:

- address
  - The URL of the Web service.

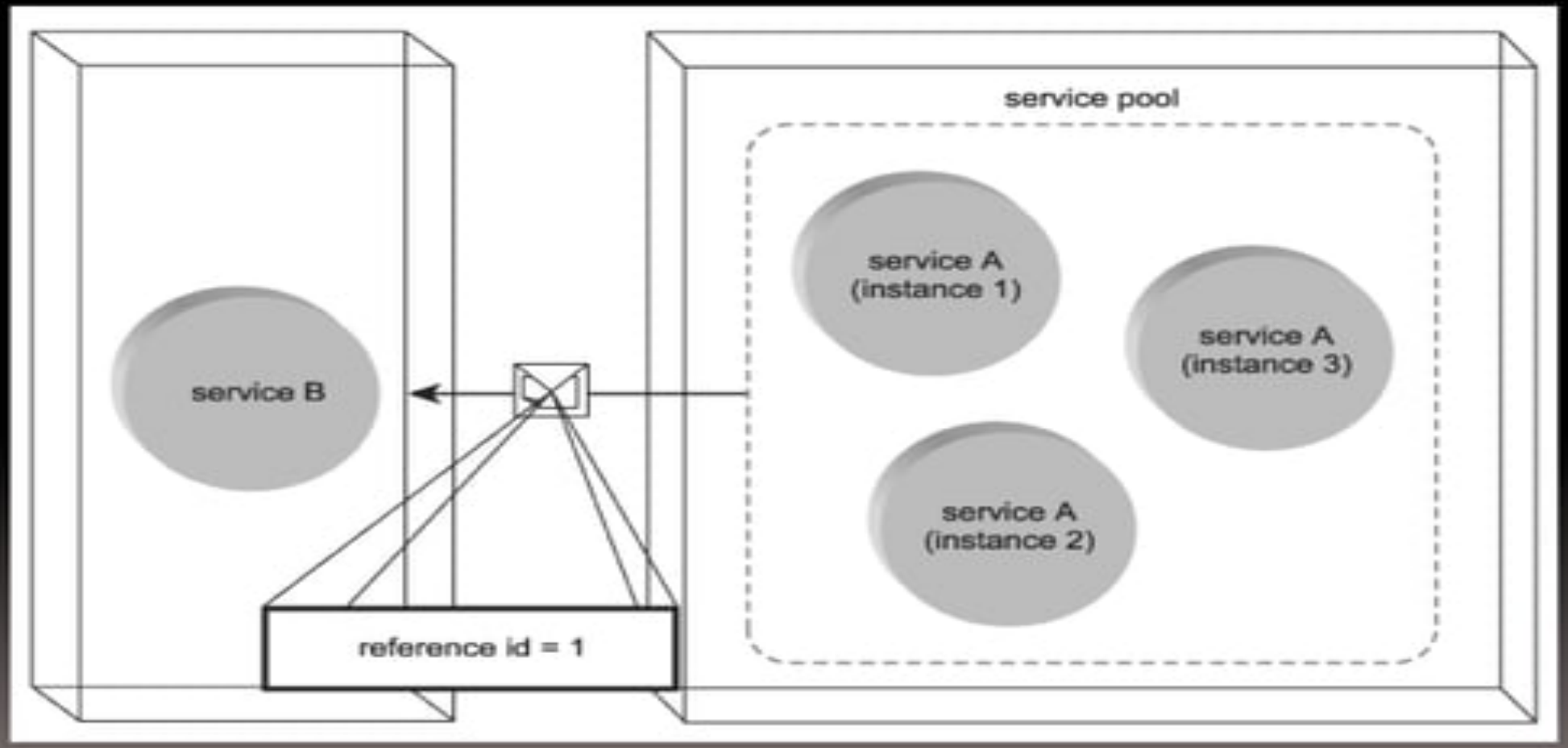
reference properties

- A set of property values associated with the Web service instance. (In Plain English example, the "attention" line used in the first scenario is representative of the reference ID property.)

- reference parameters
  - A set of parameter values that can be used to further interact with a specific service instance.
- service port type and port type
  - Specific service interface information giving the recipient of the message 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

Addressing- Endpoint references

A SOAP message containing a reference to the instance of the service that sent it



Addressing- Endpoint references

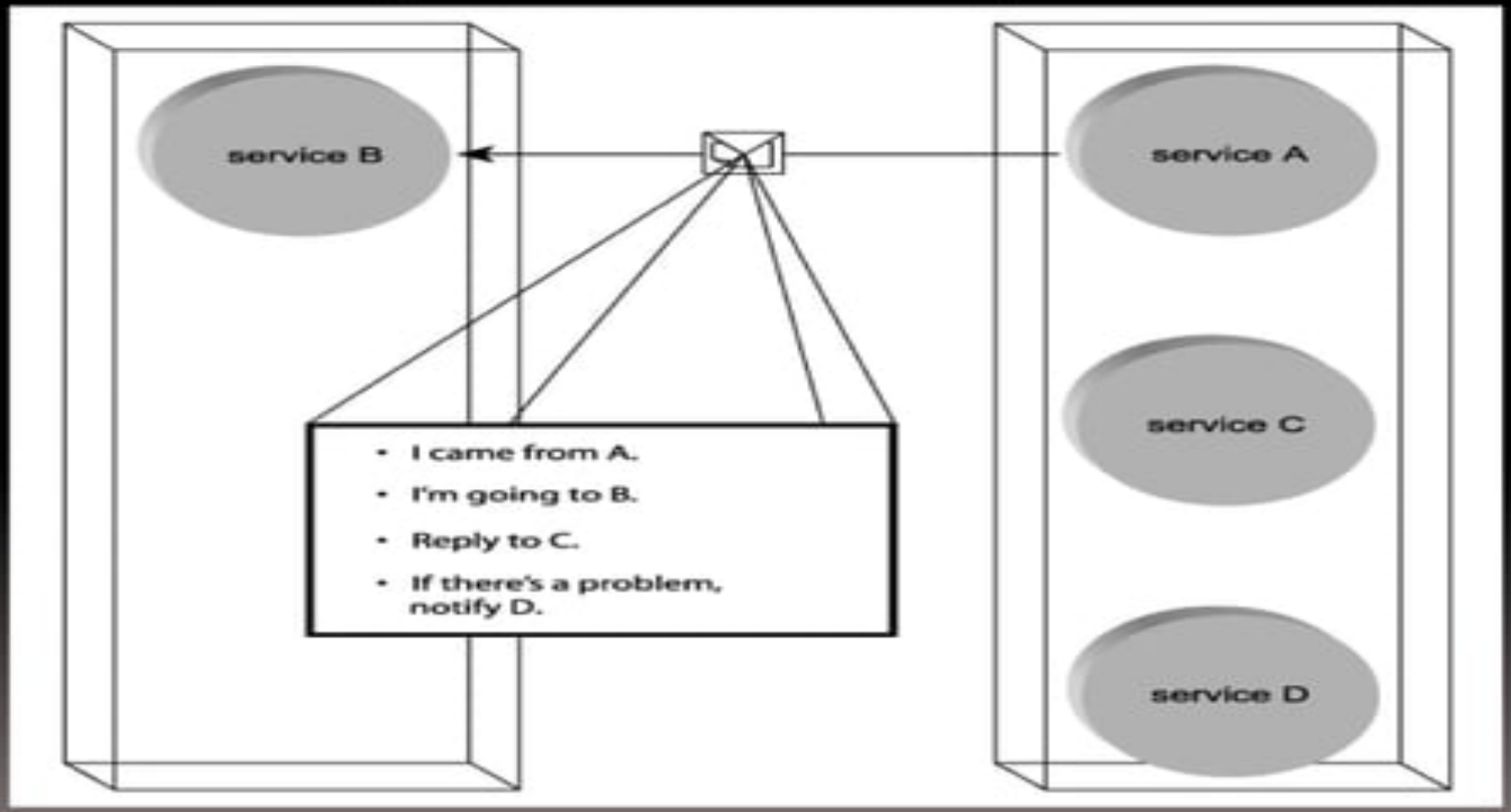
Message information headers

- MEPs have predictable characteristics that can ease the manner in which Web services are designed but also can limit the service interaction scenarios within which they participate.
- In service-oriented solutions, services often require the flexibility to break a fixed pattern.
- For example, they may want to dynamically determine the nature of a message exchange.
- The extensions provided by WS-Addressing were broadened to include new SOAP headers that establish message exchange-related characteristics within the messages themselves.
- This collection of standardized headers is known as the message information (or MI) headers

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Addressing- Endpoint references

A SOAP message with message information headers specifying exactly how the recipient service should respond to its arrival



Addressing- Endpoint references

The MI headers provided by WS-Addressing include:

- destination

  The address to which the message is being sent.

- source endpoint

  An endpoint reference to the Web service that generated the message.

- reply endpoint

  This important header allows a message to dictate to which address its reply should be sent.

- fault endpoint

  Further extending the messaging flexibility is this header, which gives a message the ability to set 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 (this header is required when using the reply endpoint header).

Addressing- Endpoint references

The MI headers provided by WS-Addressing include:

- relationship

  Most commonly used in request-response scenarios, this header contains the message id of the related message to which a message is replying (this header also is required within the reply message).

- action

  A URI value that indicates the message's overall purpose (the equivalent of the standard SOAP HTTP action value).

  [WS-Addressing specification provides an anonymous URI that allows MI headers to intentionally contain an invalid address]

Addressing- Endpoint references

- 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 in receipt of the message.
- 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.

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

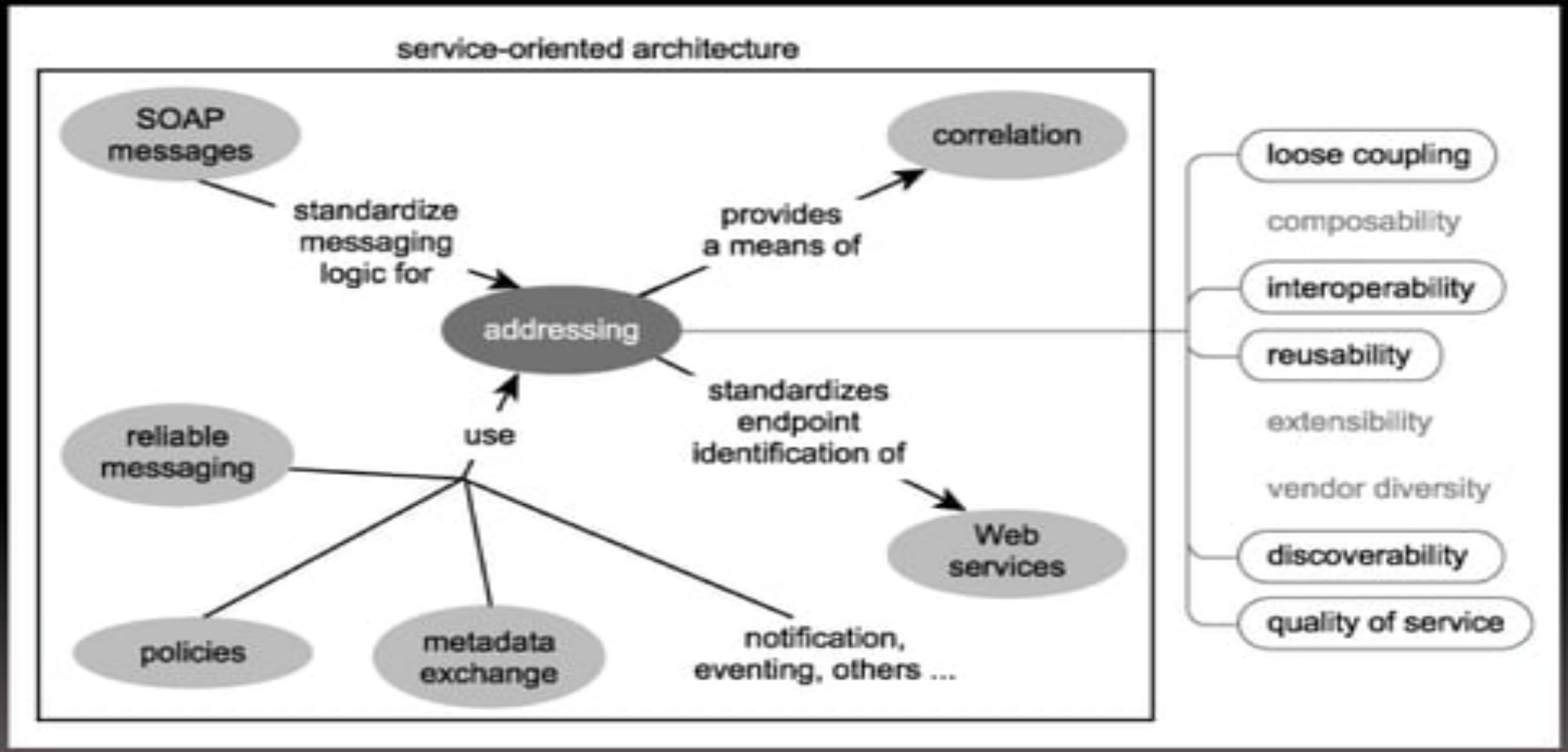
Addressing- Endpoint references

Addressing and SOA

- Addressing achieves an important low-level, transport standardization within SOA, further promoting open standards that establish a level of transport technology independence .
- The use of endpoint references and MI headers deepens the intelligence embedded into SOAP messages, increasing message-level autonomy.
- Empowering a message with the ability to self-direct its payload, as well as the ability to dictate how services receiving the message should behave, significantly increases the potential for Web services to be intrinsically interoperable.
- It places the task-specific logic into the message and promotes a highly reusable and generic service design standard that also facilitates the discovery of additional service metadata.

Addressing- Endpoint references

Addressing relating to other parts of SOA



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

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

Reliable messaging

WS-ReliableMessaging provides a framework capable of guaranteeing:

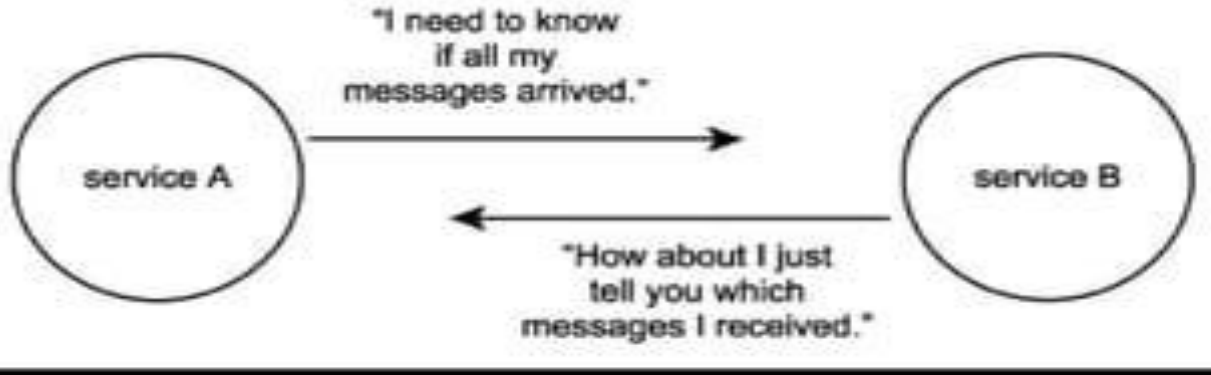
- that service providers will be notified of the success or failure of message transmissions
- that messages sent with specific sequence-related rules will arrive as intended (or generate a failure condition)

- Although the extensions introduced by reliable messaging govern aspects of service activities, the WS-ReliableMessaging specification is different from the activity management specifications .
- 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.

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Reliable messaging

Reliable messaging provides a guaranteed notification of delivery success or failure.



In Plain English

In the last chapter's Choreography section we explained how our car wash had formed an alliance with the car wash located on the other side of the highway. Part of our arrangement was to share part-time workers during peak hours.

One of the workers that joined our team is named George. Though good at his job, George has a bad memory. When we request that workers from the other side walk over to help us out, we always are warned when one of those workers is George.

The walk from the other gas station is about one kilometer. Sometimes George forgets the way and gets lost. We therefore put a system in place where we agree to call the other company to tell them how many workers have arrived. If it's not equal to the number of workers they actually sent, it's usually because George has gone missing again.

Our system of calling the other company to acknowledge the receipt of the workers and to report any missing workers builds an element of reliability into our resource sharing arrangement.

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

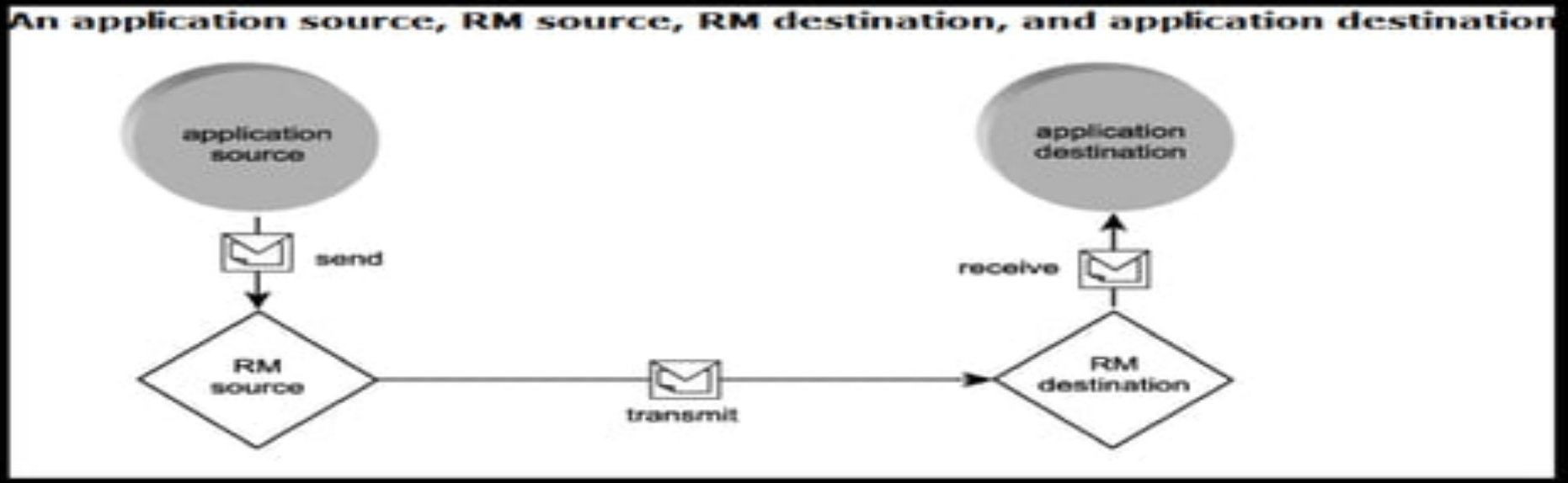
Reliable messaging

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

- 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.
- These differentiations are necessary to abstract the reliable messaging framework from the overall SOA.
- An application source is the service or application logic that sends the message to the RM source, 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

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Reliable messaging



Sequences

- 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

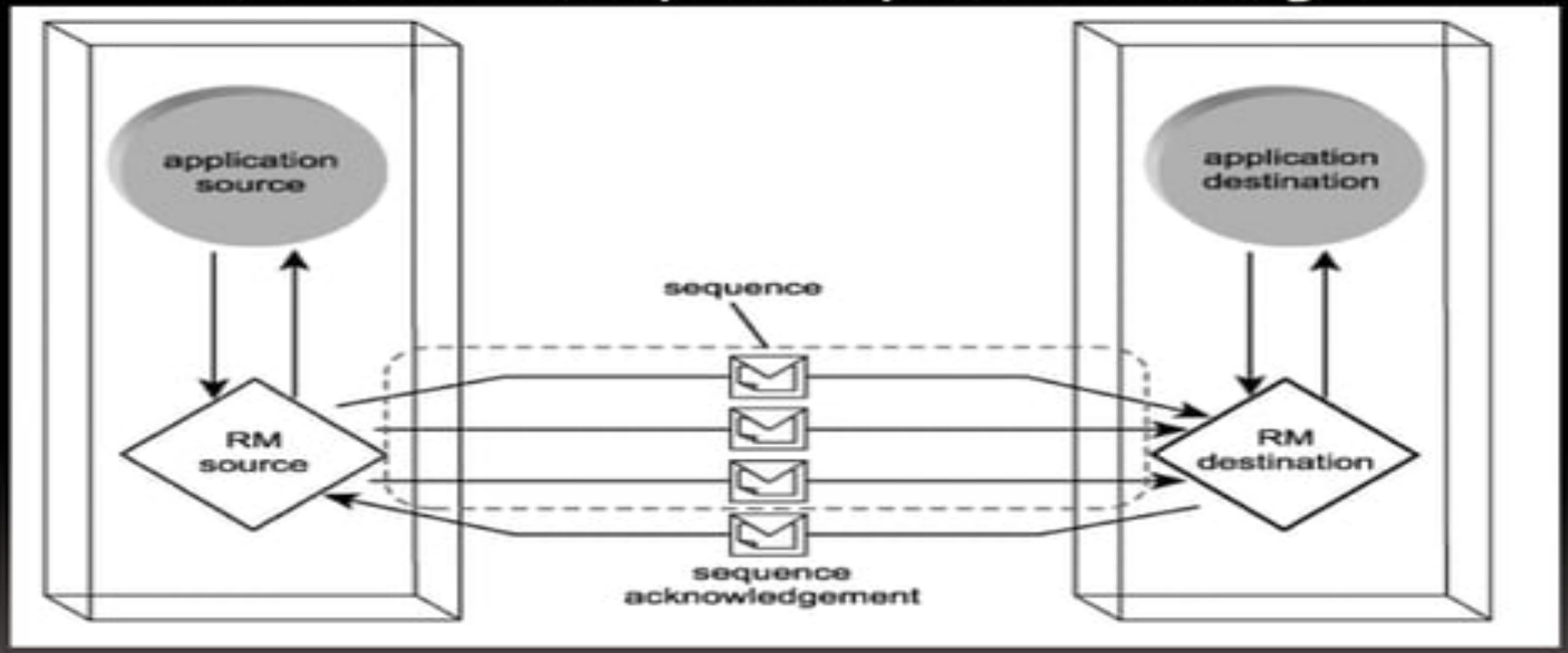
Reliable messaging

Acknowledgements

- 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.
- It is up to the RM source to determine if the messages received are equal to the original messages transmitted.
- The RM source may retransmit any of the missing messages, depending on the delivery assurance used

Reliable messaging-Acknowledgements

A sequence acknowledgement sent by the RM destination after the successful delivery of a sequence of messages

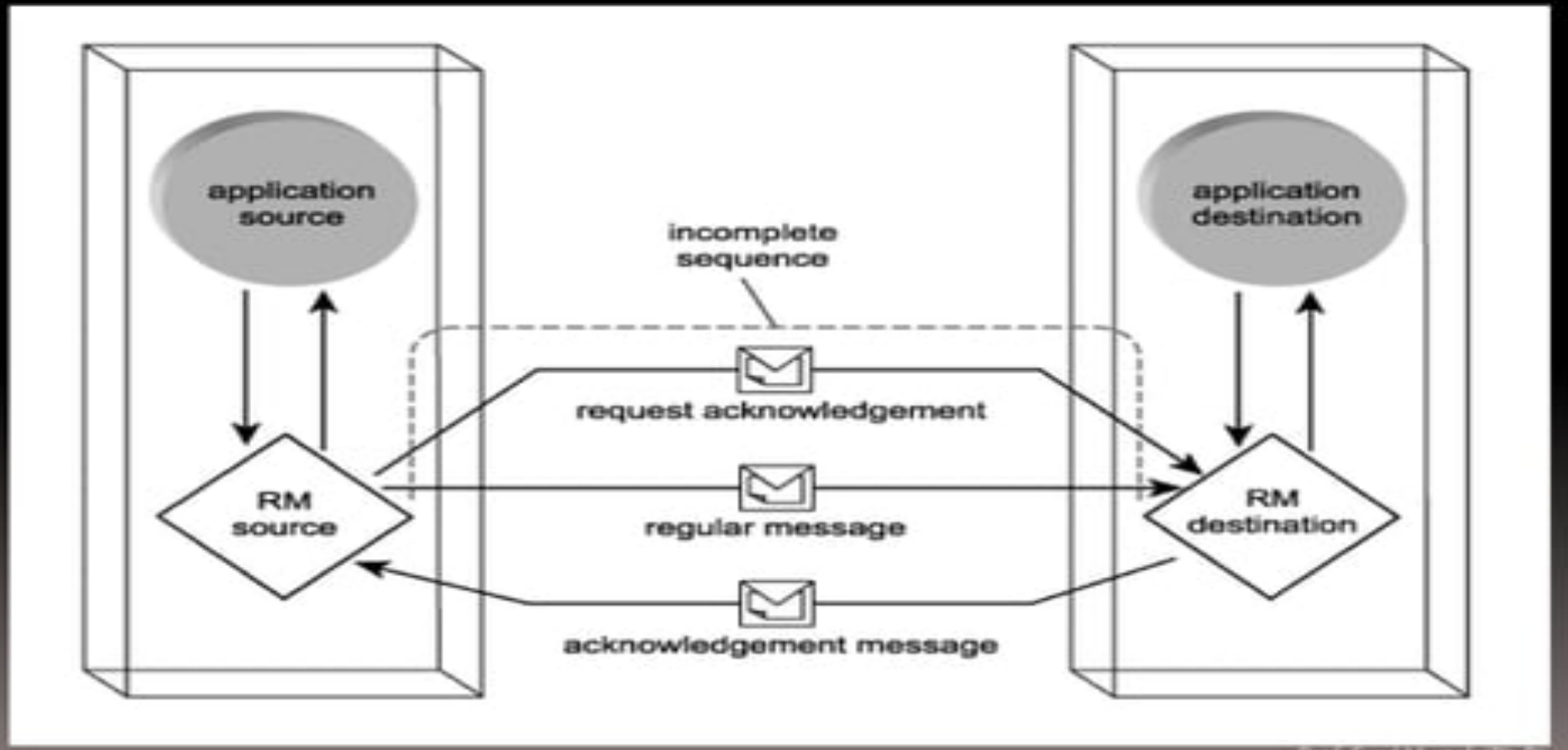


Reliable messaging-Acknowledgements

- An RM source does not need to wait until the RM destination receives the last message before receiving an acknowledgement.
- RM sources can request that additional acknowledgements be transmitted at any time by issuing request acknowledgements to RM destinations
- Additionally, RM destinations have the option of transmitting negative acknowledgements that immediately indicate to the RM source that a failure condition has occurred

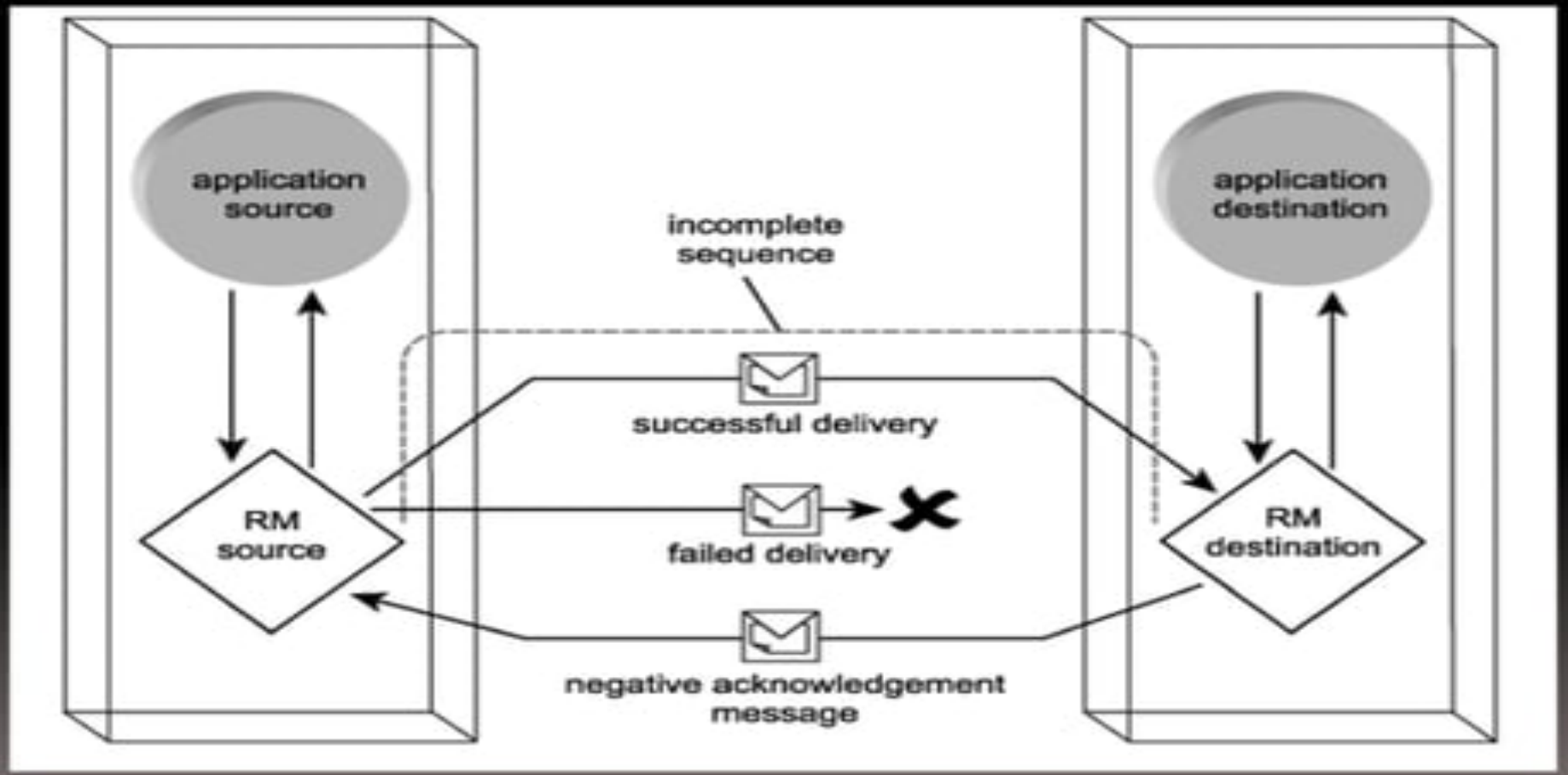
Reliable messaging-Acknowledgements

A request acknowledgement sent by the RM source to the RM destination, indicating that the RM source would like to receive an acknowledgement message before the sequence completes



Reliable messaging-Acknowledgements

A negative acknowledgement sent by the RM destination to the RM source, indicating a failed delivery prior to the completion of the sequence



WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

Reliable messaging-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.

The following delivery assurances are supported:

The AtMostOnce delivery assurance

- promises the delivery of one or zero messages. If more than one of the same message is delivered, an error condition occurs

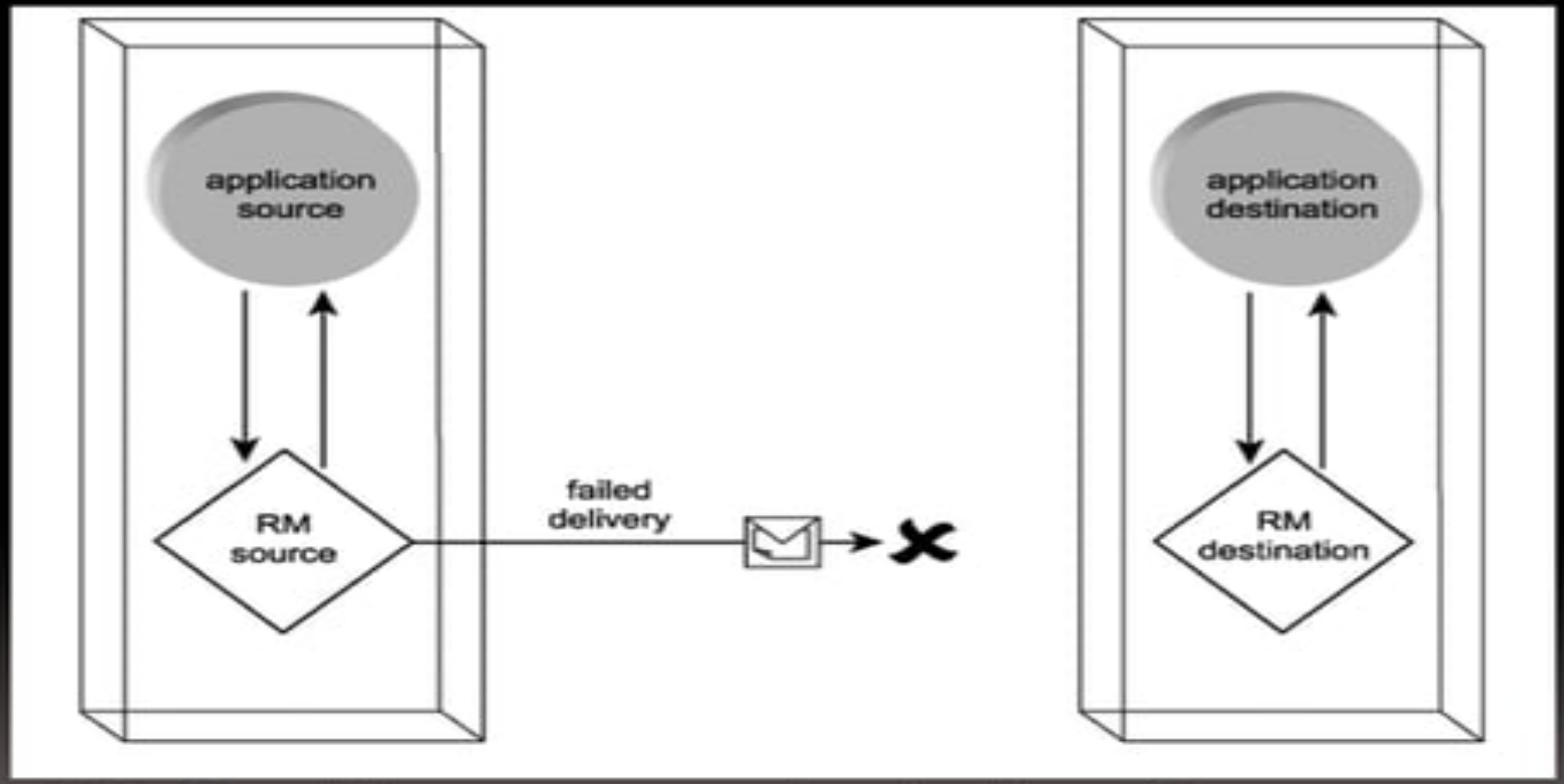
The ExactlyOnce delivery assurance

- guarantees that a message only will be delivered once. An error is raised if zero or duplicate messages are delivered

The InOrder delivery assurance

- is used to ensure that messages are delivered in a specific sequence .The delivery of messages out of sequence triggers an error. This delivery assurance can be combined with any of the previously described assurances.

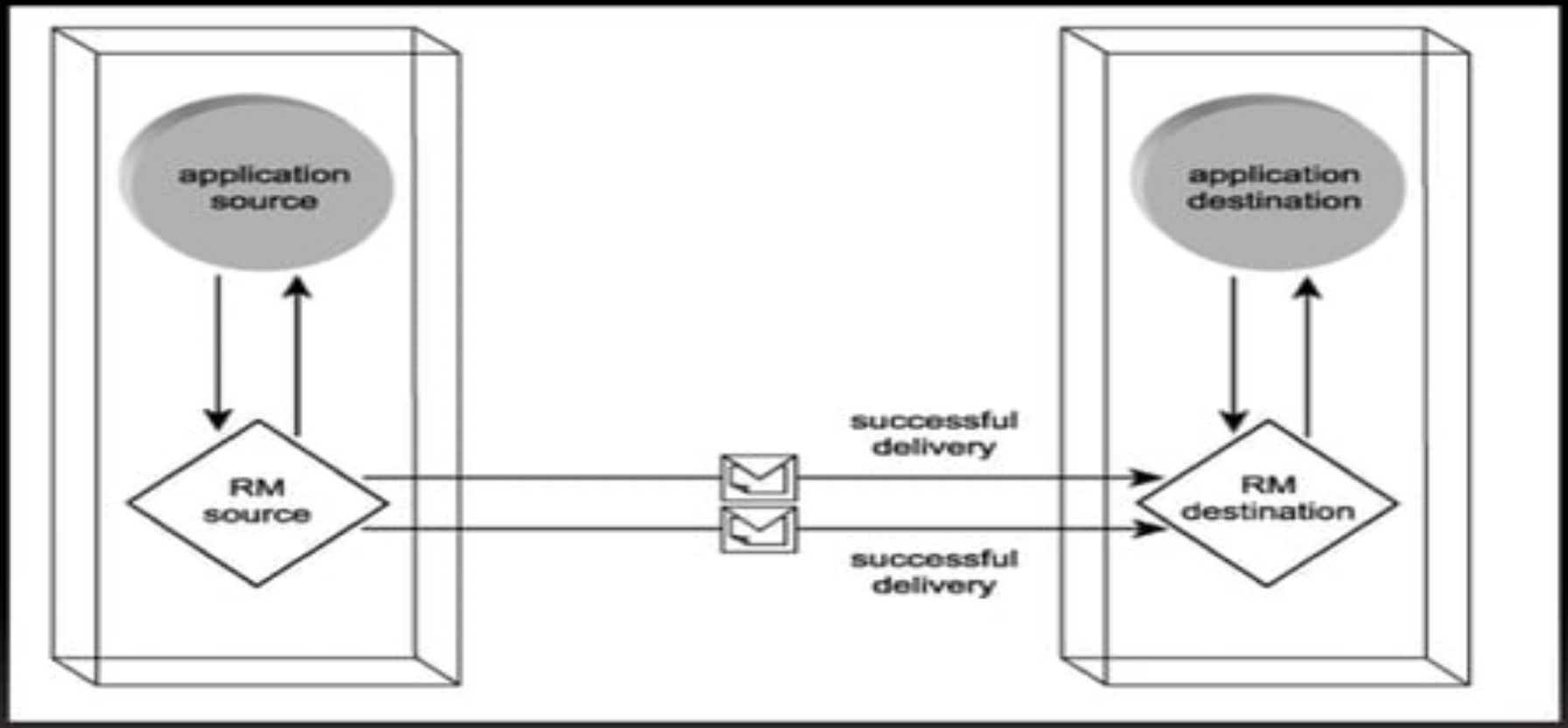
The AtMostOnce delivery assurance



The AtMostOnce delivery assurance promises the delivery of one or zero messages.

If more than one of the same message is delivered, an error condition occurs

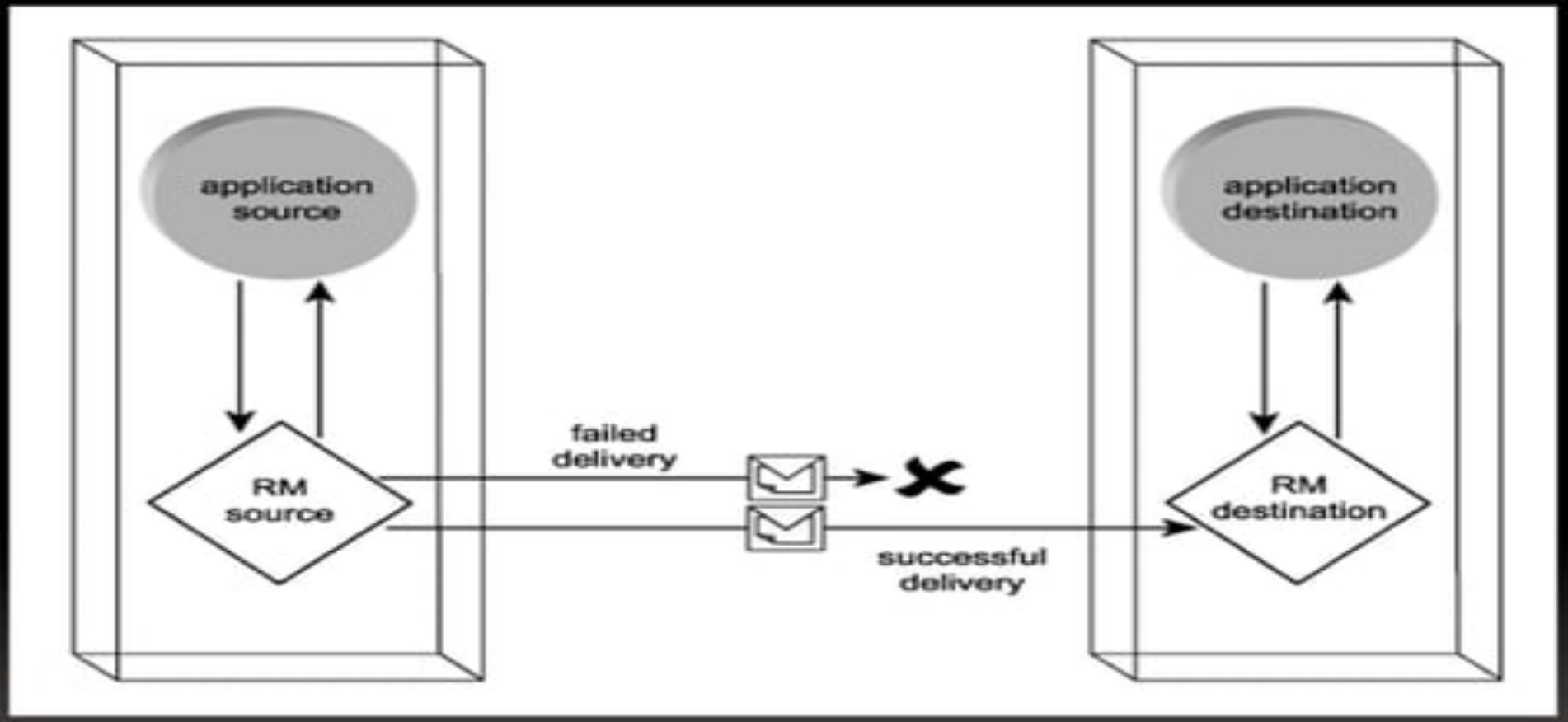
The AtLeastOnce delivery assurance



The AtLeastOnce delivery assurance allows a message to be delivered once or several times.

The delivery of zero messages creates an error condition

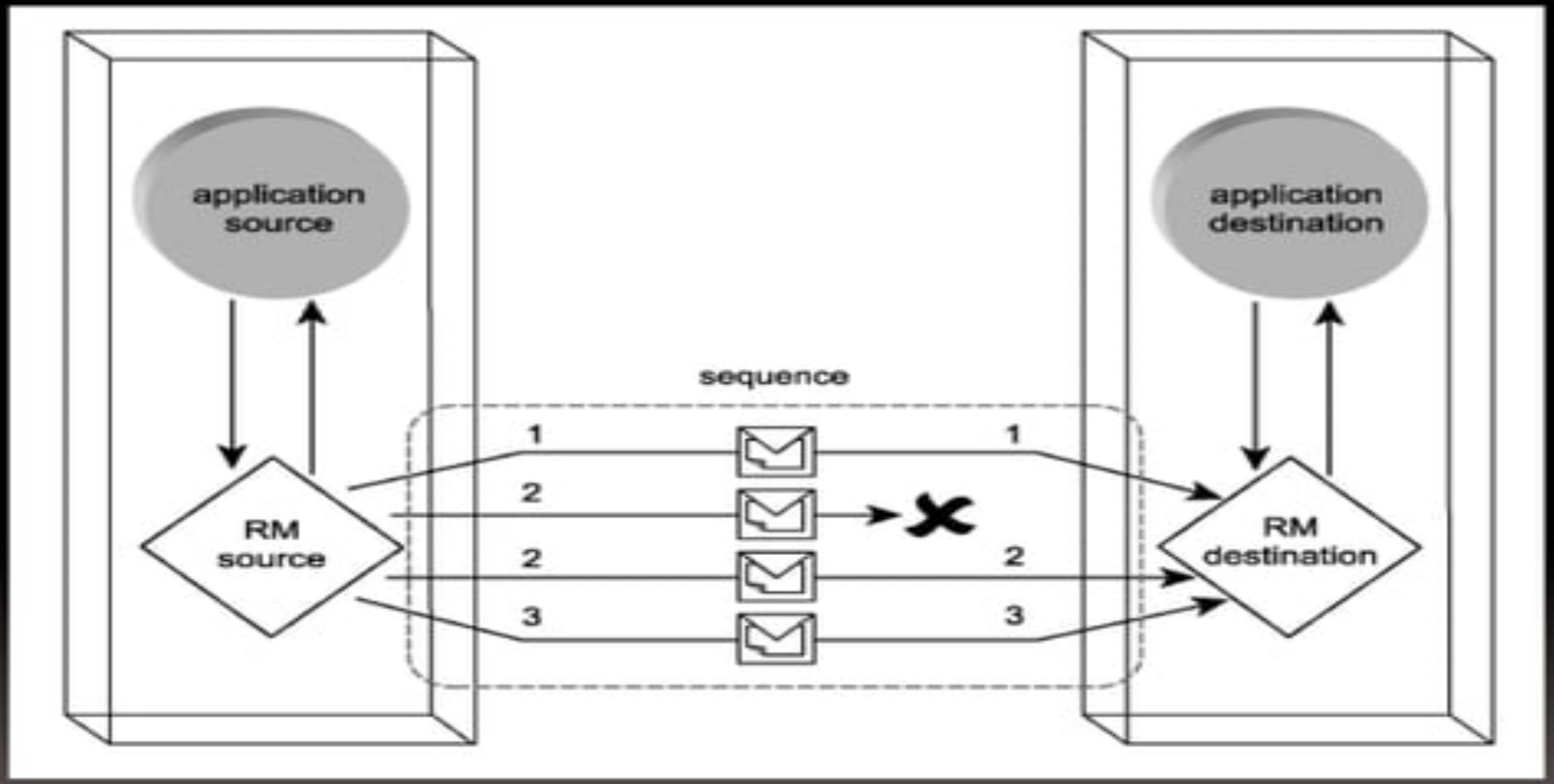
The ExactlyOnce delivery assurance



The ExactlyOnce delivery assurance guarantees that a message only will be delivered once.

An error is raised if zero or duplicate messages are delivered

The InOrder delivery assurance



The InOrder delivery assurance is used to ensure that messages are delivered in a specific sequence . The delivery of messages out of sequence triggers an error

WEB SERVICES AND CONTEMPORARY SOA
(PART II: ADVANCED MESSAGING, METADATA, AND SECURITY)

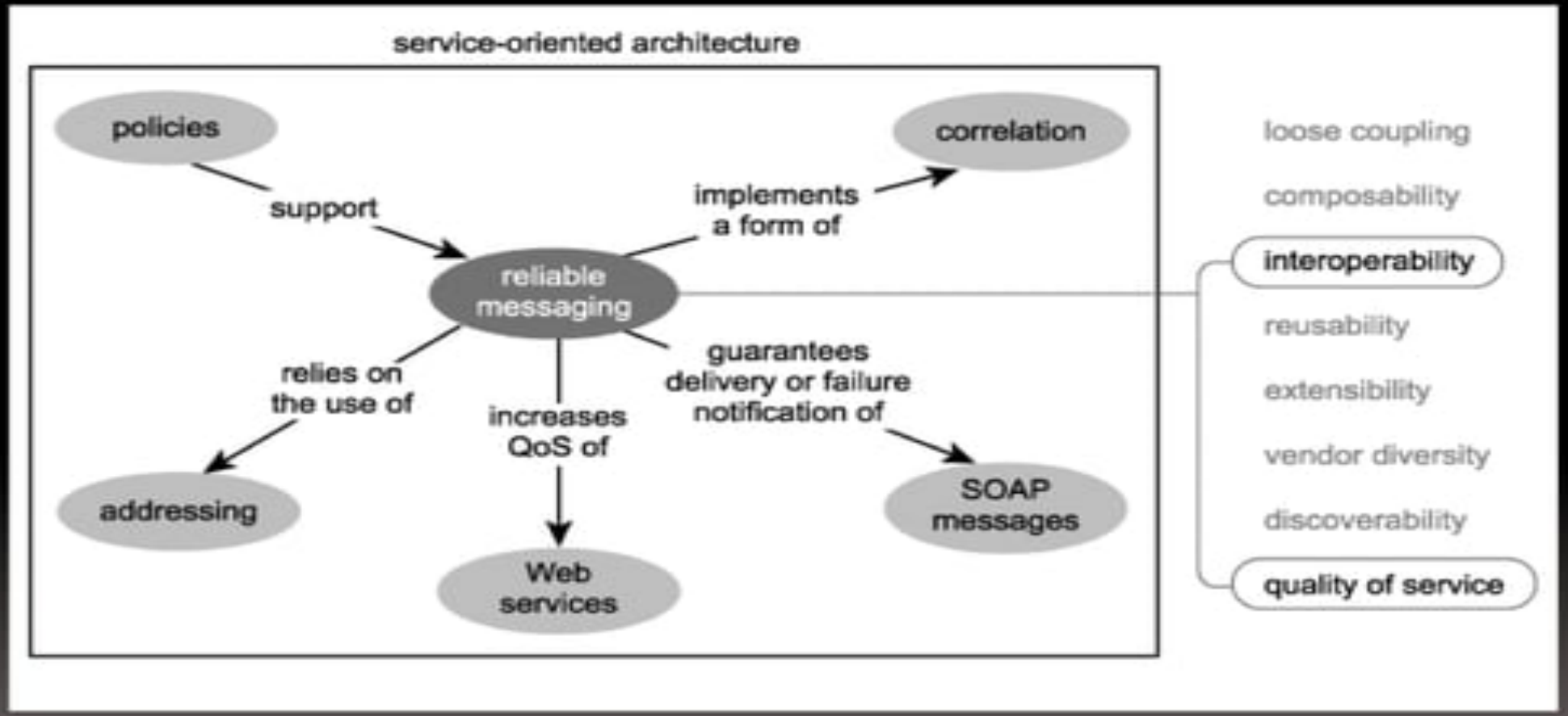
Reliable messaging

Reliable messaging and SOA

- Reliable messaging brings to service-oriented solutions a tangible quality of service .
- It introduces a flexible system that guarantees the delivery of message sequences supported by comprehensive fault reporting.
- This elevates the robustness of SOAP messaging implementations and eliminates the reliability concerns most often associated with any messaging frameworks.

Reliable messaging

Reliable messaging relating to other parts of SOA

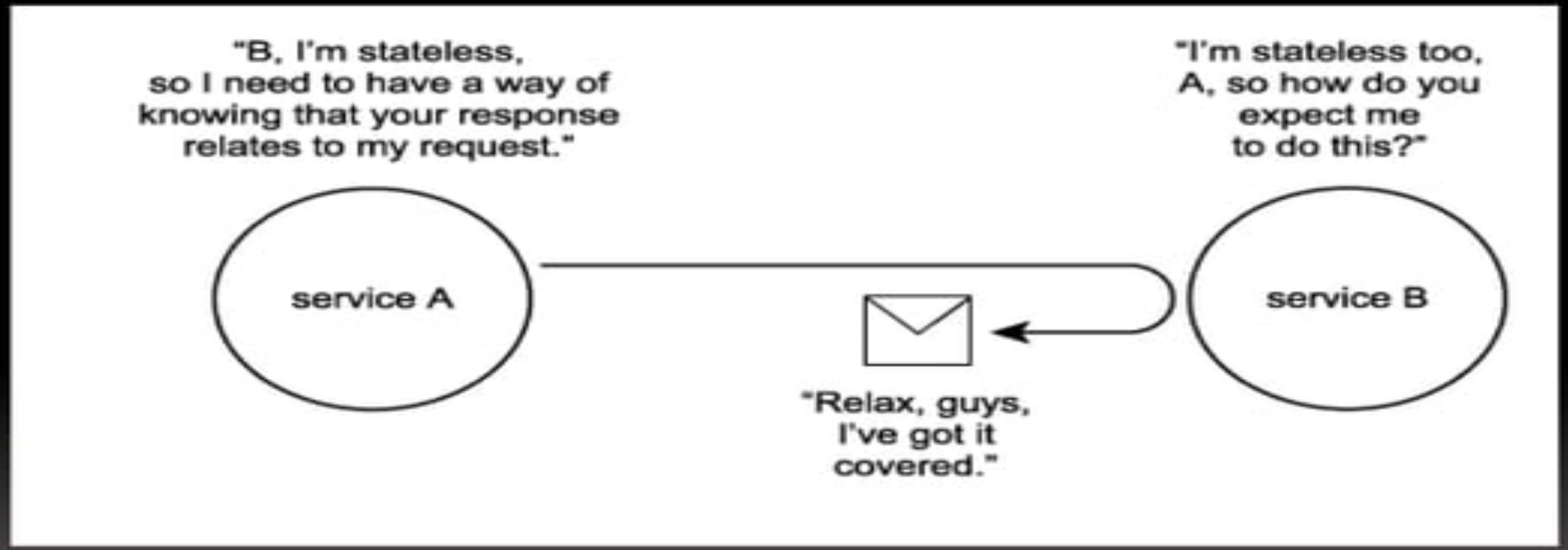


Correlation

- One of the fundamental requirements for exchanging information via Web services is the ability to persist context and state across the delivery of multiple messages.
- Because a service-oriented communications framework is inherently loosely coupled, there is no intrinsic mechanism for associating messages exchanged under a common context or as part of a common activity.
- Even the execution of a simple request-response message exchange pattern provides no built-in means of automatically associating the response message with the original request.
- Correlation addresses this issue by requiring that related messages contain some common value that services can identify to establish their relationship with each other or with the overall task they are participating in .

Correlation

Correlation places the association of message exchanges into the hands of the message itself



Correlation

In Plain English

- To encourage repeat business, we introduce a promotion where, after ten visits to our car wash, your eleventh visit is free.
- We implement this promotion through the use of a punch card.
- Every time a customer drives in, we punch the driver's card.
- This card associates the current visit with all of the previous visits.
- Essentially, the punch card provides us with a form of correlation.
- Without it we would have a very hard time remembering which customers had visited us before.

Correlation

Correlation in abstract

- The technology that enabled tightly bound communication between components, databases, and legacy applications typically established an active connection that persisted for the duration of a given business activity (or longer).
- Because the connection remained active, context was inherently present, and correlation between individual transmissions of data was intrinsically managed by the technology protocol itself.
- When one stateless service sends a message to another, it loses control of the message and preserves no context of the activity in which the message is participating.
- It is up to the message to introduce the concept of correlation to provide services with the ability to associate a message with others.
- This is achieved by embedding a value in the message that is propagated to all related messages.

Correlation

Correlation in abstract

- When a service processes a message and locates this value, it can establish a form of context, in that it can be used to associate this message with others.
- The nature of the context can vary.
- For example, a message could be part of a simple exchange activity, an atomic transaction, or a long running orchestration

Correlation in MEPs and activities

- Because they are generic and non-business-specific in nature, MEPs and activities have no predefined notion of correlation.
- They are simple, conceptual building blocks incorporated and assembled by either custom-developed solutions that employ custom correlation identifiers and related processing logic or by specifications that impose proprietary forms of correlation

Correlation

Correlation in coordination

- The context management framework provided by WS-Coordination establishes a sophisticated mechanism for propagating identifiers and context information between services.
- A separate activation service is responsible for creating new activities and subsequently generating and distributing corresponding context data. S
- ervices can forward this information to others that can use it to register for participation in the activity.

- While context management uses a correlation-type identifier to uniquely represent a specific activity context, it goes well beyond correlation features to provide a comprehensive context management framework that can be leveraged through activity protocols, such as those supplied by the WS-AtomicTransaction and WS-BusinessActivity extensions.

Correlation

Correlation in orchestration

- WS-BPEL orchestrations need to concern themselves with the correlation of messages between process and partner services.
- This involves the added complexity of representing specific process instances within the correlation data.
- Further complicating this scenario is the fact that a single message may participate in multiple contexts, each identified by a separate correlation value.
- To facilitate these requirements, the WS-BPEL specification defines specific syntax that allows for the creation of extensible correlation sets.
- These message properties can be dynamically added, deleted, and altered to reflect a wide variety of message exchange scenarios and environments.

Correlation in addressing

WS-Addressing's message id and relationship MI headers provide inherent correlation abilities, which can be leveraged by many composition and messaging extensions.

Correlation

Correlation in reliable messaging

- Every message that participates in a WS-ReliableMessaging sequence carries sequence information with it.
- This data consists of a sequence identifier that represents the series of messages required to follow the messaging rules of the sequence, along with a message identifier that identifies how the current message fits into the overall sequence.
- As a whole, this information can be considered correlation-related.
- However, its primary purpose is to support the enforcement of reliability rules.

Correlation and SOA

- Correlation is a key contributor to preserving service autonomy and statelessness.