I'm not robot	reCAPTCHA

Continue

```
Partition standardized business data from proprietary legacy data into body and header sections 3. Service Design Patterns 4. State data cached to support the activities, thereby decreasing scalability. Service Inventory Patterns >Domain
 Inventory Service Normalization Dronk Analysis) 113. Relationships 129. Example 237. 103. Reliability - Acknowledgement 3. Application Sample Message processing logic that is a natural part of any
Web service implementation actually consists of a series of system (and perhaps custom) service agents that collectively carry out necessary runtime actions. > When runtime activities that span multiple services fail, parent business task is incomplete and actions performed and changes made up to that point may affect integrity of underlying solution
 and architecture. Organizational business domains not represented by separate IT departments or groups, increased up-front analysis Required prior to the delivery of WS-Policy definitions 88. Formal analysis and governance processes can help avoid these situations.
may risk the reliability of all compositions in which it participates if an unexpected error condition occurs. 130. Solution > Dynamically translate one data format into another. 61. Application > Service A has multiple service contracts as well as a redundant implementation,
allowing this service to facilitate a wide range of consumer programs. In these environments, global and domain-level policies can also be established via service agents that act as policy enforcement points 90. > If agnostic services are not reused, redundant functionality can be delivered in other services, > Result problems associated with inventory
denormalization, service ownership and governance 117. Impact Runtime Performance overhead Format conversion will need to be executed every time interaction between disparate format sources is required Protocol Bridging is a pattern only used out of necessity. Modern replication technology allows for the runtime transformation of database
schemas, enable the replicated database to be tuned for individual service architectures while performance and reliability is acceptable. Governance Issue Service agents will need to be owned and maintained by a separate group that needs to understand the inventory-wide impacts of any changes made to agent logic, An agent versioning system
will be further required to address these challenges. Asynchronous Queue Intelligent routing options to facilitate various runtime conditions 2. Application > Inventory domain boundaries need to be carefully established. Real World Sample > Application > Impact >
Relationships [543] 195. A variety of service agents are part of the Accounts service accounts service accounts service account service accounts service accounts service accounts service account service accounts service account
service design. > Event management program is introduced, allowing the service consumer to set itself up as a subscriber to events associated with a service that assumes role of publisher. (metadata with message) 204. Impact Notification broadcast can't be predicted Consumer must always be available to receive notification message
transmissions. WS-Reliable Messaging standard (A) Persistent repository (B) 62. Problem > How can the misuse of redundant service consumer
programs. Relationships 223. Create a standardized legacy function in a standardized manner. 189. Course Calendar Exam 1: Fundamental SOA & Service-Oriented Computing Exam 2: SOA Design & Architecture Exam 4: Advanced SOA Design & Architecture Exam 5:
SOA Design & Architecture Lab 3. Service composition logic consists of a service invocations; each invocation enlisting a service to carry out a segment of the overall parent business process logic Larger business processes can be very complex, especially when having to incorporate numerous "what if" conditions via compensation and
exception handling sub-processes. It then periodically attempts retransmission 54. Tracking messages at runtime 3. Reliability - Acknowledgement 1. Policy Centralization > Problem > Solution > Real World Sample > Application > Problem > Solution > Real World Sample > Application > Problem > Solution > Real World Sample > Application > Problem > Solution > Real World Sample > Application > Real World Sample > Application > Real World Sample > 
subject to change, resulting in the release of a second version 151. - An error is raised if zero or duplicate message are delivered, an error condition occurs - However situations may arise when a message will never reach a
destination. Service Design Patterns \triangleright Contract Centralization \triangleright Concurrent Contract \triangleright Concurre
Solution > Protocol bridging technology is introduced to enable communication between different communication protocols > Dynamic conversion between protocols > Dynamic conversion between protocols at runtime 34. > The usage patterns of shared resources accessed by the service are changed, resulting in changes to the established service behavior that end up negatively affecting
existing service consumers. These domains can still form the basis of service inventory boundaries but require cooperation across IT departments. Persist message in queue & Reliable Messaging: to make sure that message have been send to destination 80. 145. Application > A central service can be established to provide an official access point for
the creation, modification, retrieval, and application of business rules. 128. Relationships 186. Impact Extra operational-related governance effort to keep all redundant implementations in synch. Reliability - Delivery Assurance 4. 17 Official Endpoint > Introduction [711] 139. (Add up and reduce overall service scalability) (Capture a snapshot of a
resource prior to making changes to it) 247. Content-Based Routing (Commonly Used) 45. > when services are delivered by multiple project teams, risk that services are delivered by multiple project teams, risk that services are delivered by multiple project teams.
 Message undergoing various processing steps carried out by an ESB in order to successfully transform and deliver the message contents to their destination Some message transmissions may not require all of this functionality 9. Solution > Services can have their own dedicated databases with replication to shared data sources. Relationships 256.
 Relationships 162. can be applied in direct support of Standardized Service Contract. Real World Sample Customer Service facade 146. Problem 52. > Web services and REST servi
136. The service is subjected to decomposition, as per Service Decomposition 144. 222. Real World Sample Domain Store (Internal Delivery, International Delivery, International Delivery) OR Enterprise work for different Industries 104. These business areas would then establish the basis for business domains. 84. 19 Redundant Implementation Problem Solution Internal Delivery Solution Internal Deliv
Real World Sample > Application > Impact > Relationships [345] 155. > Service logic can be deployed in isolation to increase service autonomy, but services be delivered to maximize recomposition when enterprise-wide Standardization,
Governance is not possible? Application > A reliability framework is typically comprised of infrastructure and intermediary processing logic capable of: 1. Impact Performance overhead Each centralized policy effectively adds a layer of runtime processing and service dependency. Design complexity Introduce a layer in distributed environment
 Relationships - part of ESB, Broker 82. Relationships - part of ESB Middle-tier platforms can provide policy features supported by runtime agents that carry out policy compliance checking. (challenging in contract design to accommodates different consumers types) > A service's contract may not be suitable for or applicable to all potential service
consumers. Real World Sample Parcel Metadata 206. It may be required to supplement the service logic with additional functions that compensate for any potential negative effects these behavioral changes may have on consumers. Service Façade is often used for this purpose. Solution > Service composition is supplemented with compensating
logic. Create the schema definitions by applying design standards to ensure consistency and normalization. > Modern runtime platforms have native rules repositories and processing logic that is made accessible via a set of system service agents and APIs. > This allows any service to interface with business rules-related logic without having to
compose a separate service > Centralized rule service sare most often classified as members of the utility service layer 96. > Service communication cannot be guaranteed when using unreliable environment. While all of this processing can also be located within the core service
logic, it may be desirable to isolate it into façade components when the processing requirements are exclusive to specific contracts. Real World Sample Subscribe to Youtube Channel Notification Bell 78. Relationships - part of Orchestration 224. Can also reduce scalability of overall service inventory 251. Service Broker Queuing Mechanism enables
asynchronous message exchanges 3. Impact Memory Consumption Because of the requirement for each service to preserve its original state until it is notified to rollback or commit its changes. Relationships - part of ESB, Broker 67. Application Common types of logic that tend to reside within a service façade component include: 1. Implementation
environment is a large enterprise without strong executive sponsorship and wide-spread support for the SOA initiative Organization is incapable of changing the complexion of its IT departments in support of a more centralized governance model. Solution > Multiple contracts can be created for a single service, each target a specific type of
consumer. AtMostOnce delivery assurance - Promises the delivery of one or zero messages. > Wrapper service extracts, encapsulates, and possibly eliminates legacy technical details from contract to Implementation coupling technical contract to Implementation coupling technical details from contract.
(like , .NET , Java) > As a result, the utilization and evolution of services is inhibited because they can only be used by consumer programs compatible with their technology > Bridging and transformation products generally results in increased integration effort and reduced number of potential service consumers (reduced reuse potential). Behavior
Correction: Façade logic is used compensate for changes in the behavior of the core service behavior to which established consumers have become familiar. Application > Up-front analysis effort is required to establish a schema layer
independent of and in support of the service layer > Separate schema layer is taken into account subsequent to the completion of the service implementation is designated as the official contact point for consumers, but it is further supported by one or more backup implementations that are used
in case of failure or unavailability. Real World Sample Separation of Concerns like entity, utility abstraction (Care about Functional context) 220. Introduction Multifaceted runtime conversion features that enable integration systems. 11 Rules Centralization Problem Solution Real World Sample Problem Real World Sample Real World Samp
[216] 92. Establishing an single enterprise service inventory may be impractical or even unrealistic. May jeopardize the success of an SOA adoption as a whole. Solution Global or domain-specific policies can be isolated and applied to multiple services. Relationships 115. 28 Process
Centralization \triangleright Problem \triangleright Solution \triangleright Real World Sample \triangleright Application \triangleright Impact \triangleright Relationships [193] 225. \triangleright WSDL documents is extended with individual WS-Policy definitions. \triangleright The larger and more complex a service composition is, the more difficult it is to expect and design for all possible runtime scenarios especially in asynchronous.
messaging-based. This can place the burden of runtime activity management onto the services themselves. > Select schemas that exist as physically separate parts of the services are commonly implemented as individual Web services. > Data format transformation logic can exist within a
service's internal logic, within a service Agent > Middleware will often provide out-of-the-box services and agents that perform a variety of data format conversions. Relationships - part of Orchestration 241. 205. Extended Enterprise Service Bus Core Patterns in Left Side Common Extend Base patterns in Right Side 10. Replace legacy-specific data
with standardized business data 2. > Service implementations can be equipped with dedicated data source. Problem 42. > Allows each contract to be extended and governed individually > Also provides the option of exposing a subset of the
service capabilities to specific consumers 182. Any change to a centralized schema can affect numerous service contracts. Some perform general processing of all data while others are specific to input or output data flow. 137. Described by the service boundaries is abstracted into a distinct functional context associated with
the task service model > Increase organizational agility because it is the parent business change and implemented independently from the service capabilities that utilize them to represent the structure and typing
of message content > schema architecture can be established and standardized somewhat separately from the parent service layer. Issuing acknowledgements for individual or sequences of message > The repository used for guaranteed delivery may provide the option to store messages in memory or on disk so as to act as a back-up mechanism for
when message transmissions fail > Allows admins to track status of messages and trace causes behind unresolved delivery problems. Solution > Services can be grouped into manageable, domain-specific service inventories are created for one enterprise > Each of which can be independently standardized, governed, and
owned. Orchestration Design Patterns 5. Impact Performance overhead adds a layer of processing and associated performance overhead with the additional service invocation \triangleright Relationships \triangleright Reliability
(Acknowledgment - Assurance) [592] 60. Problem > How can direct consumer-to-implementation coupling be avoided? - The delivery of zero messages creates an error condition 72. Real World Sample Nurse Event-Driven Agent of (Doctor check up service) (Temperature, pressure, length, weight) measurements 198. Eliminate input and output
data contain highly proprietary characteristics like [error code can be replaced by SOAP Fault] 2. Real World Sample Concurrent Edition contracts to accommodate consumers concerns] 183. Introduction > Official Endpoint supports the goal of establishing a federated layer of service endpoints 141. Part
of ESB Broker-related features are a fundamental part of ESB platforms. Problem > How can a service facilitate multi-consumer coupling requirements and abstraction concerns at the same time? 193. Problem > How can a service state data be persisted for extended periods without consuming service runtime resources? Relationships 122. Impact
Composition complexity not reduced Although the perceived size of the composition may be reduced, The actual complexity of the composition itself does not decrease. 77. 69. 226. Transformation Patterns used only out of necessity. 37. High contract-to-logic coupling resulting in high implementation coupling by service consumers. Problem >
How can service interact with programs that communicate with different data formats? > An event-driven messaging framework is implemented as an extension to the service inventory. This abstraction helps protect the core service inventory. This abstraction helps protect the core service inventory.
Pattern used when legacy systems need to act as service consumers or when legacy logic needs to be encapsulated by services > A protocol bridging layer is composed of a set of adapters that act as on-/off-ramps for a given transport protocol. 185. Impact Expensive orchestration technology Introduce orchestration technology can be expensive and
disruptive. Broker Logic : - Façade logic carries out transformation logic as per patterns associated with Service Broker - This may be required when a single unit of core service logic is used together with multiple service contracts, as per Concurrent Contracts 3. 106. Reliability - Delivery Assurance 1. Solution > Reusable services can be deployed
via redundant implementations or with failover support. 153. Add New service (or capability) will typically contain logic that performs transformation between its standardized contract and the native legacy interface, accomplished by eliminating and
encapsulating technical information 191. Solution > A service can exchange messages with its consumers via an intermediary buffer (remaining temporally decoupled) > A queue is introduced as an intermediary buffer receives request messages and then forwards them on behalf of the service consumers 53. Solution > Message contents can be
supplemented with activity-specific metadata that can be interpreted and processed separately at runtime. Increase Complexity of service-oriented solutions Relative to the size of their service compositions The use of various delivery rules (including those based on group message delivery via sequences), it may not be possible to wrap services using
reliability features into atomic transactions, as per Atomic Services, leading to redundancy and governance challenges > A global change to either rule will therefore impact multiple services 93. Impact Performance overhead, Design and development effort Due to
Addition of façade component Although some performance overhead is expected, it is generally minor as long as façade and core service components are located on the same physical server. Service logic that is triggered by a predictable event can be isolated into a separate program especially designed for automatic invocation upon the occurrence
of the event. 231. 160. "To Be continue with SOA 258. Solution > The service implementation > The servi
and governed 173. 172. Real World Sample Orchestra centrally governed by maestro 228. Impact Runtime performance overhead Format conversion will need to be executed every time interaction between disparate format sources is required Design complexity.
Introduce a new layer in an already distributed environment. 15 Schema Centralization > Real World Sample > Application > Impact > Relationships [200] 123. Application > Impact > Relationships [200] 123. Application > Impact > Relationships [200] 123. Application > Impact 
required for the dedicated service databases Difficult to manage Excess of replication channels can be difficult to manage Excess of replication protocols or different versions of the same
protocol cannot exchange data. Relationships 89. Problem > How can a service express its capabilities independently of its implementation? > Mapping logic needs to be developed and deployed to perform transformation at runtime > XSLT style sheets represents the most common application of this pattern > Only applied when other patterns cannot
 be realized to their full potential \triangleright Ensure "transformation avoidance" as you can throughout service-orientation \triangleright This pattern as a last resort to solving interoperability problems. \triangleright The service implementation is upgraded or refactored, resulting in changes to the core business logic in order to accommodate the new and/or improved
composition logic via specialized tools provided by orchestration platform. Application > A custom service contract and required service consumer to be developed to represent the proprietary legacy interface represent in two ways: 1. 147. S90.08 Advanced SOA Design & Architecture 4. > Consistent form of loose coupling with all service consumer
programs 134. ESB Design Patterns 2. 3 Data Model Transformation \triangleright Real World Sample \triangleright Application \triangleright Impact \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Problem \triangleright Solution \triangleright Real World Sample \triangleright Application \triangleright Impact \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Problem \triangleright Solution \triangleright Real World Sample \triangleright Application \triangleright Impact \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Problem \triangleright Solution \triangleright Real World Sample \triangleright Application \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Application \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample \triangleright Real World Sample \triangleright Relationships [671] 15. 6 Intermediate Routing \triangleright Real World Sample Sample \triangleright Real World Sample Samp
requirements Additional hardware and administration \triangleright Real World Sample \triangleright Application \triangleright Relationships [242] 233. 19. \triangleright Persistent connections between a service
and its consumer place various types of state and context data about current service activity into memory, allowing routines within service capabilities to access this information as required. - There may be resistance to giving up control of solution designs if teams are forced to include existing agnostic services or produce new services that need to
be reusable. Impact Open-ended Compensation logic - Left up to the designer, it can vary in quality and effectiveness. 26 Orchestration \triangleright Introduction \triangleright Introduction \triangleright Compensation for reduced
resource locking and memory consumption > As long as there is no firm requirement for compensations to restore a runtime activity back to its original state, exception conditions can be handled gracefully without create risk for the integrity of the composition 252. InOrder delivery assurance - Ensure that messages are delivered in a specific
sequence - Delivery of messages out of sequence triggers an error - Can be combined with any of the previously described assurances 74. 152. Application > Compensation logic is pre-defined and implemented as part of the parent composition controller logic or via individual "undo" service capabilities. > Service agents can be designed to
automatically respond to predefined conditions without invocation via a published contract, service agent typically lightweight, generally contain common utility-centric processing logic, commonly deployed to facilitate inter-service communication via a published contract, service agent typically lightweight, generally contain common utility-centric processing logic, commonly deployed to facilitate inter-service communication via a published contract, service agent typically lightweight, generally contain common utility-centric processing logic.
such as authentication, authorization, logging, audit, and load balancing. Application > Contract implementation coupling to service consumers. Solution > Business rules can be physically abstracted into a dedicated part of the architecture under management of specialized rules enginess.
and platforms. Allocating the individual parts of the process into appropriate new or existing conceptual service databases that provide fully replicated data sets on behalf of a central databases become the contact point for service databases requiring
only a subset of the replicated information. Dedicated database is provided for state deferral purposes Database located on same physical server as services that utilizing it, so as to minimize runtime performance overhead associated with writing and retrieval of the data Database is provided for state deferral purposes as to minimize runtime performance overhead associated with writing and retrieval of the data Database is provided for state deferral purposes.
database. \triangleright Consumer programs and services connect to a broker that provide bridging protocol \triangleright Off-the-shelf products provided by the broker or a third-party vendor 36. 199. Relationships - Part of Orchestration 23. Relationships 81. Problem \triangleright How can non-agnostic process logic be separated and governed independently? \triangleright When a highly reused
service becomes unexpectedly unavailable, it will risk all of its service consumers. Relationships 179. A graphical front-end tool allowing users to express and maintain business rule logic and avoids redundancy. Impact Additional performance and guarantee requirements Reliable Messaging
 entities and information sets likely to be processed by services in this inventory. Solution > Access to reusable functionality is limited to official agnostic services > The level to which the repeated ROI of services can be realized 118. Problem > How can
abstracted business process logic be centrally governed? Application > Queuing technology needs to be incorporated into the surrounding architecture > In modern ESB platforms, the use of a queue can be completely transparent (neither consumer nor service may know that a queue was involved in a data exchange) > The queuing framework can be
supported by intelligent service agents that detect when a queue is required and intercept message broadcasts to multiple consumers, as per Event-Driven Messaging 56. Introduction - Dedicated to the effective maintenance and execution of
parent business process logic - Support complex service composition logic that can result in long-running runtime activities Business Process Execution - Separating non-agnostic logic from agnostic logic fr
Real World Sample Data Packet Router 44. Relationships 21. Problem > How can service interoperate when using different data models for the same type of data? Application One service façade component abstracts a shared database (resource A) whereas another abstracts a legacy system (resource B). Application > A transaction management
system is made part of the inventory architecture and then used by those service compositions that require rollback features. 21 Decoupled Contract \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401] 171. This service abstracts the legacy application \triangleright Relationships [401
27 Process Abstraction > Problem > Solution > Real World Sample > Application > Impact > Relationships [182] 216. Problem > Domain service inventories are an appropriate alternative when: Enterprise does not have an established, global data models and creating them is
considered unrealistic. 1:1 Routing \triangleright Routing agent directly wired to a single physical service instances or redundant se
Relationships [123] 101. Relationships 66. Real World Sample Postal Reliability Index Measures the level of postal operational efficiency. Introduction Environment designed to include any interconnectivity between services Intermediate layer of processing Base Patterns that Comprise ESB Incompatible Translation (Data Model, Format, Protocol) 1
 Request acknowledgement sent by RM source to RM destination, indicating that RM source would like to receive an acknowledgement message before the sequence completes.. To fully apply Service Normalization requires that this process be carried out iteratively, once for every business process that is associated with the scope of the service
 inventory Typically part of a top-down delivery effort. Application \triangleright Messaging technology used for service communication needs to support message document \triangleright Many types of messaging metadata have been standardized through the emergence of WS-^*
extensions that define industry standard SOAP header blocks, like \triangleright WS-Addressing \triangleright WS-MetadataExchange \triangleright WS-Policy 207. Relationships – part of ESB , Broker 59. 2 Service Broker \triangleright Introduction \triangleright Nested Patterns of Service Broker \triangleright Core Part of ESB [707] 11. \triangleright Service (acting as publisher) automatically sends the event details
to the event management program, which then broadcasts an event notification to all of the consumers registered as subscribers of the event. Can lead to complex service activities that are difficult to design principle on a broad scale.
Guaranteeing message delivery during failure conditions via the use of a persistence store 2. Relationships 154. When new systems or resources are introduced into a service architecture, 1. Impact Tightee to control and monitor. Impact Tightee to control and monitor.
architectural dependencies upon Rules services Accessing centralized business rules , the overall service inventory needs to be designed with an emphasis on service boundary alignment > Services are collectively modeled before their individual physical
contracts are created, ensure that it does not overlap with other services > The result is a service inventory with a higher degree of functional normalization 111. Solution > Runtime service activities can be wrapped in a transaction with rollback feature that resets all actions and changes if the parent business task cannot be successfully completed
Application > A shared or dedicated repository is made available as part of the inventory or service architecture. 127. ** Schema Centralization Mandates that schema definitions be shared across multiple contracts.
service consumers. Relationships 209. 188. Problem > How can business rules be abstracted and centrally governed? As long as Process Abstraction was used to define a layer of task services, The proper separation of agnostic and non agnostic logic will exist to allow for the non-agnostic (process-specific) logic to be cleanly migrated to a centra
Organizational business areas represented by specific IT departments or groups. Problem > How can wrapper services with non-standard contracts be prevented from spreading indirect consumer-to-implementation coupling? This logic differs from atomic transactions in that it does not require services to maintain the original state or lock
resources for the duration of the transaction. Relationships Legacy Wrapper is a therefore frequently applied together with Enterprise Service Bus in order to leverage the broker capabilities natively provided by ESB platforms. 102. > Traditional MEPs, consumer need to continually poll the service to find out whether such an event had occurred (and
while allowing the core service logic to evolve independently? > Policy definitions can be embedded within or linked to WSDL documents 87. The need for duplicate schema data models emerges, leading to the definition of service compositions can be embedded within or linked to WSDL documents 87.
Transaction 213. Increase infrastructure requirements Host and run necessary middleware can increase the size and overall operational costs of the IT environment as a whole 230. Impact Runtime Performance overhead Because of interpretation and processing of different types of messaging headers Design complexity Complex message paths that
agents and over reliance on service agents can further tie an inventory architecture to proprietary vendor technology. Problem > Bypass the service contract and simply access its underlying logic directly using native protocols > Service ends up with numerous
tight dependencies 133. Introduction - Enables orchestration to support long-running service activities - Provide state management repository that can be leveraged as a state deferral mechanism. Relationships - part of ESB 49. May even result in inaccurate semantic meanings being applied to message contents. Relationships - part of ESB, Broker
the delivery method, Agnostic and non-agnostic logic is grouped together in each service like traditional silo based applications 217. Relationships 248. 64. Re
infrastructure with redundancy features. Problem > How can the reliability of a service be increased? It imposes governance complexity when expertise associated with business processes lie with different individuals 3. (they can be modified and evolved from a single location) 94. > High reuse services or
providing critical functionality can be deployed with multiple implementation to guarantee high availability > Having redundant implementation for down. Application > service contract can be kept free of implementation details, consumers can
avoid binding to them (Logic to contract coupling) 177. Features that comply with industry standards related to business process logic expression and execution, such as WS-BPEL 229. As with any service, a task service can be comprised of multiple capabilities, each of which represents a separate process or task. This API still represents a technical
contract, but not the "official" service entry point. 169. > In Large enterprise environment, the context established by each service logic. This can result in geographical-based domains. 239. > In Large enterprise environment, the context established by each service logic.
boundary will usually not be exclusive to one body of data. > Abstracted into separate policy definition documents or service agents that are part of an inventory-wide policy enforcement framework > Positioned to apply to multiple services, thereby reducing redundancy and providing centralized policy governance 85. Application > Ensure services
level autonomy applied - Common Steps include : 1. Solution > State data can be temporarily written to and then later retrieved from a dedicated state repository, the service is able to transition to a stateless condition during pauses in the activity, thereby temporarily freeing system resources > This
prevent services from having to unnecessarily keep state data in memory for extended periods. Application > Eliminating Technical Information: 1. Create the WSDL definitions using the standard schemas wherever appropriate and supplementing the contract with any required service-specific schemas.
that needs to reside within services, further decreases quantity of services (or services (or services E and G. Solution \triangleright Two services business processes can be deployed and governed from a central location. 235. Solution
\triangleright Contract Centralization establishes a design standard that positions the service contract \triangleright Relationships [421] 180. Reliability - Acknowledgement 2. \triangleright Moving from RPC-based toward a messaging-based
removes this. Problem > How can a service exchange data with consumers that use different communication > Real World Sample > Application > Impact > Relationships [136] 116. Impact Data standardization > Relationships [136] 116. Im
design standards. Application > Middleware platforms provide necessary orchestration technologies to apply this pattern. 70. > service façade logic is primarily responsible for providing supplemental, intermediate processing logic in support of the core service logic. Conceptually modeled prior to delivery Can result in separate analysis project that
needs to be completed before any service can actually be built to guarantee inventory-wide normalization (as part of inventory services have been defined, allowing for the definition of task services that comprise this layer. Solution
\triangleright The consumer establishes itself as a subscriber of the service. The only rule is that these processes be related to a common overarching functional context \triangleright The inventory architecture is equipped with a reliability framework that
tracks and temporarily persists message transmissions and issues positive and negative acknowledgements to communicate successful and failed transmissions to message senders. Real World Sample Migrate Contact Information from Samsung to iPhone 18. 168. Process Centralization 212. 149. The service, in turn, automatically issues notifications and issues positive and negative acknowledgements to communicate successful and failed transmissions to message senders.
of relevant events to this and any of its subscribers. Orchestration > Process Abstraction > Relationships 108. 14. Solution > Event-driven logic can be deferred to event-driven programs that don't require explicit invocation
AtomicTransaction standards provide an industry standard mechanism to support transaction propagation across Web service is positioned to define the transaction boundary and manage transaction activity 246. Application > Specialized implementation of Service Agent (Routing
centric agents) > Often provided by middleware and are a fundamental component of ESB products like > Determines a message's path based on its contents (inside or metadata) > Model complex business processes (Recompose services on the fly) > Access business rules engine to accurately assess message destinations 1. Inventory Design
Patterns 3. 1. Problem \triangleright How can service contracts be designed to avoid redundant data representation? Real World Example Unconscious mind State Delegation \triangleright Relationships [623] 242. \triangleright Runtime platforms
messaging middleware, and ESB products commonly provide the necessary infrastructure for message processing and tracing capabilities, along with service agents that supply complex event processing, filtering, and correlation 79. > This approach can leads to numerous unnecessary service invocations and data exchanges, also introduce delays as
to when the consumer receives the event information because it may be only able to check for the event at predetermined polling intervals. Because each program contains embedded business process logic, it results in a physically decentralized architecture, distributed across multiple locations > When changes come, Ability to efficiently extend,
streamline, or even combine business process logic is inhibited because the underlying logic of each affected task service needs to be revisited, opened up, and changed, as required. Some governance overhead can also be expected, due to the increased amount of components per service Creating façade components results in an increased amount of
physical logic decomposition. 143. Service may be incompatible with resources it needs to access due to different data format (service consumer may use different data format (service may have been standardized to send and receive XML-formatted data but is required to also retrieve data from a legacy environment that only supports
the CSV format 25. 197. Problem > How can service consumers be automatically notified of runtime service events? > The message routing logic embedded within the logic of each service in a composition if there is unexpected factors can lead to unexpected system failure like: 1. It makes it difficult to apply Non-Agnostic Context, thereby reducing
the chances of successfully abstracting single-purpose cross-entity logic into legal services. Relationships - part of ESB, Broker 22. Services delivered by different times can be built using different tommunication technologies common scenario when design standards are not prevalent in an enterprise Leading to missed
opportunities to reuse and recompose services for new purposes 33. Enterprise Service Bus \triangleright Service Broker \triangleright Data Model Transformation \triangleright Protocol Bridging \triangleright Intermediate Routing \triangleright Asynchronous Queue \triangleright Reliable Messaging \triangleright Event-Driven Messaging \triangleright Rules Centralization \triangleright Policy Centralization 11 6. 218.
Problem > How can a service inventory avoid redundant service logic? Identifying and decomposing a business process logic is abstracted and then physically centralized into an environment that provides a native state management
repository. Negative acknowledgement sent by the RM destination to the RM source, indicating a failed delivery prior to the completion of the sequence. Impact Runtime Performance overhead Increased due to The addition of required write and read functionality, can negatively affect performance overhead Increase Design complexity Leading to more
development effort and expense For service activities with strict real-time performance requirements, this state deferral option needs to be carefully assessed. Remote offices, each with its own IT department and development center. Real World Sample Quiet Library Policy 86. Impact Service Utilization Deficiencies within contract technology
 Allows a message to be delivered once or several times. Problem > How can composition runtime exceptions be consistently accommodated without requiring services to lock resources? Performance Overhead Cross-domain data exchange impact the development and design effort of corresponding service compositions often lead to the creation of
redundant services across domains. Nested Patterns of Service Broker Convert Data between different incompatible schema structure Data Model Transformation Protocols 
 architecture so that policies can be separately defined and associated with services and then validated, enforced, and even audited at runtime > WS-Policy framework includes a separate WS-Policy framework includes a separate WS-Policy framework includes a separate with services and then validated, enforced, and even audited by
requiring the involvement of reusable services as part of their development projects. 125. > Logic can be extended with compensating routines that also respond to exception conditions with pre-determined exception handling logic to respond to exception conditions with pre-determined exception conditions with the pre-determ
service that was invoked up until that point. Intermediate Routing 8. Raise Cultural Issues Introduce Logic Centralization into organization that does not sponsor reuse raise issues with people and IT departments affected by service delivery projects 121. 91. Validating that no two service boundaries overlap. Contract-Specific Requirements - Service
facades are coupled to contracts to accommodate different types of service consumers, - This can include special security, reliability, and activity management processing requirements. Impact Runtime performance overhead Every time services with disparate schemas need to exchange data Development effort Development of the mapping logic
Design complexity Into a service composition and the service inventory as a whole 20. > Consumer programs designed to access underlying service resources using different entry points > Different implementation dependencies that inhibit service from evolving in response to change. Application > This pattern establishes a distinction between the
official published service contract and other parts of the service capabilities and resources via separate technical endpoints > For example, a consumer could interface with an underlying service capabilities and resources via a central
contract requires also on-going standardization effort Resistance to centralization is common Tactical requirements, such as time-to-market priorities and budget restrictions, can motivate some project teams to simply disregard this pattern altogether. Problem > This approach has several reflections: 1. 150. Relationships 38. Application > Agnostic
services need to be properly designed and governed, and their use must be enforced via enterprise standards > Make service more discoverable by applying Metadata Centralization > if a new capability needed by a project team falls within the boundary of an existing service, corresponding functionality needs to be added to that service instead of
ending up elsewhere 120. Application > A messaging framework is implemented capable of supporting the publish-and-subscribe MEP and associated complex event processing and tracking. Solution > A data transformation technology can be used to convert data between various schema structures at runtime > Enable non-standardized messaging
framework, to dynamically overcome difference between schemas used by a service contract and messages transmitted to that contract. Real World Sample Login Schema for Google Services 126. Vendor Platform's support for relevant contract
and related communications technologies deficiencies on service inventory Standardizing on a technology decoupled contract design can then impose any deficiencies on the service inventory as a whole 178. When business process logic is distributed across independent service implementations, it can be problematic to extend and evolve. - Unlike
atomic transactions that are governed by specific rules, and can vary in its actual effectiveness Bloat the service contract with an undo capability for each existing capability that alters the service implementation state 255. Relationships 138.
Complete a service inventory blueprint to establish a conceptual representation \triangleright Relationships 202. 13 Service Normalization \triangleright Relationships 202. 14 Service Normalization \triangleright Relationships 202. 15 Service Normalization Normalization \triangleright Relationships 202. 15 Service Normalization Normalization Normalization Nor
Relationships [131] 109. Services within a particular composition register themselves as part of a transaction prior to completing their changes System Queries all services to ask whether their functions were carried out successfully If any one service responds negatively A Rollback command is issued. Cannot combined as part of Atomic Services
Transaction Publisher/subscriber availability issues can arise Messages are typically issued via the one-way MEP, which does not require an acknowledgement response from consumer. 31 Compensating Service Transaction \triangleright Relationships [631] 250. The previous can raise serious
reliability issues that can be addressed through application of Asynchronous Queuing: when the consumer not available 2. 12. Impact Additional design and development effort Places logic into services that could otherwise be located into other types of service consumer programs An organization needs to be prepared to
implement and support service compositions in order for this pattern to be effectively applied. Common Policies that apply to multiple services can introduce redundancy and inconsistency within service logic and contracts Leads to bloated policy content and increases the governance burden required to ensure that common policies are kept in
synch over time. Impact Performance overhead add overhead compared to direct service communication Security Concerns Who will process a message containing sensitive data Difficult Management of dynamic routing paths Centralized routing paths Centralize
a common means of defining compensating processes as part of the overall parent composition member • composition m
interaction and composition > Canonical Schema has not been successfully applied > Schema incompatibilities can impose interoperability challenges 16. Thanks! SOA Architecture Certification You can find me at: @MohamedZekus eng.mohamedZekus eng
destination) 3. Nature of Webservice adapter contains embedded, implementation-specific (and sometimes technology-specific) details. Solution A dedicated parent business process service layer is established to support governance independence and the positioning of task services as potential enterprise resources. Application New services
contracts can be accommodated by repeating this pattern to introduce new façade components, thereby potentially shielding the core service logic. Application 2. Solution > If the target
service is unavailable, the queue acts as temporary storage and retains the message. SOA Design Pattern S90-08A 2. 10. Real World Sample Legacy Wrapper of old building ( 190 ( معنا المعالمة المعادية) may only require read access to data, which can be fulfilled by a one-way data
replication channel. 28. Reliability - Delivery Assurance 2. > For example, service contract may need to incorporate special processing extensions (such as policy assertions) not supported by all consumer types is challenging both from design and governance perspectives and can ultimately lead
to security concerns and constraints that limit the service's overall effectiveness 181. > This design pattern is especially relevant to agnostic services that are heavily reused and need to facilitate multiple compositions. Problem
> How can event-driven logic be separated and governed independently? > Façade logic is inserted into the service contract, the service logic, and the underlying service implementation. Real World Sample Network Rules in Insurance
Companies 95. Persistent connections are no longer available. Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Problem Solution Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit to Juice Blender 27. 18 Service Façade Real World Sample Transform from PDF to Excel Format Fruit Transform from PDF to Exce
throughout the inventory as they are revised and evolved over time. 140. > When a service capability requires that consumer and service be designed to
process activity-specific data at runtime? 25 Messaging Metadata > Problem > Solution > Real World Sample Compensation Plan B, no full rollback of plan A If any step in A fails, there will be backup
plan B compensation For case of Fault Exceptions 253. 159. Solution > Real World Sample > Application > Impact > Relationships [409] 132. Application > Web services represent most
popular means of applying this design pattern, as they force service contract to be expressed in separate description documents using industry standard meta languages (WSDL-WADL) > A service's technical interface is physically separated and subject to relevant service-orientation design principles > A common risk associated with expressing
service contracts using Web service technologies is the established approach of auto-generating Web service contract description documents (WSDL, XML Schema, and WS-Policy definitions) via modern development tools (Negative coupling type) (This technique can result in implementation coupling) 175. > separate façade can actually be created
for each new contract > Often additional contracts are not considered necessary until well after the deployment of the service and its original contract 184. Problem > How can service may be impacted, 2. Relationships 98. > Most
services, though, end up requiring both read and update abilities, which leads to the need for two-way replication 167. Real World Sample Restaurant Asynchronous Launch Queue Take Order Number, Then Receive Order 55. By deferring common logic to service agents. Problem > How can service communicate reliably when implemented in an
unreliable environment? Relationships - part of Orchestration 232. Application > This pattern is ideally applied together with Service Façade to support new contracts as required. Real World Sample Redundant Failover camera Battery Backup Redundant sim card Implementation for type of customers (Internet - calls) 158. Sequence
acknowledgement sent by the RM destination after the successful delivery of a sequence of message paths can be dynamically determined
through the use of intermediary routing logic \triangleright multi-purpose routing logic can be abstracted as separate part of architecture (event-driven service agents) \triangleright Intercept messages and dynamically determine their paths 43. Real World Sample Driving license Contract Driving School Driving Test ** Driving License Issue Illegal immigration Visa Issue
Legal Country Ports ** Country Entry 135. Problem - design-time, runtime challenges > Single body of service logic move business rules associated with
business services into a separate location Performance Requirement increase due to the need for business rules to be retrieved or applied at runtime Caching mechanisms can reduce of this impact to an extend 97, 196. Introduction Two Patterns Work together to form Official Endpoint Logic Centralization Contract Centralization - Asks designers to
build consumer programs that only invoke Particular services when specific types of information processing are required - Does not address how this logic is to be accessed - Asks designers to build consumer programs that access a service only via its published contract - Does not indicate which services should be accessed for specific purposes.
Application > The event-driven logic is implemented as a service agent—a program with no published contract that is capable of intercepting and processing messages at runtime. 4 Data Format Transformation > Problem > Solution > Real World Sample > Application > Impact > Relationships [681] 24. State Repository - Supports long-running
processes by allowing parent composition logic to be supplemented with compensation sub processes that address exception conditions. > The most isolated service will still need to interact with some central database to access or update business data > These repositories shared with various parts of enterprise include legacy applications, on the composition is a supplemented with various parts of enterprise include legacy applications.
locking and performance constraints that will inhibit their individual autonomy 164. Example A single global policy (A) is established, thereby replacing the redundant policy definitions entirely. Dependency on Carrying out that logic via the composition of other
services. The formation of a task service layer is the result of repeatedly applying Service Encapsulation and Non-Agnostic Context to this limitation is not required. Policies Conflict Risk of conflict between centralized and
service-specific policies. > Wrapping composition in an atomic transaction can tie up too many resources, for the duration of the transaction thereby negatively affecting performance and scalability. 7 Asynchronous Queue > Problem > Solution > Real World Sample > Application > Impact > Relationships [582] 50. > Some complex compositions
introduce extended periods of processing during which this data is not required > Cached data continues to be stored in memory and consumes runtime resources 234. 148. 176. Different service contracts often need to express capabilities that process similar business documents, resulting in redundant schema content that is difficult to govern.
Denormalization of the inventory - Inability to establish service capabilities as the official endpoints for bodies of agnostic logic - Services with overlapping can become out of synch, providing same functions in different ways 110. 201. Relationships This pattern can also be applied after a service inventory has already been established. A back-end
middleware runtime environment capable of hosting orchestrated task services and the corresponding collection of business process definitions created with the front-end tool 3. Relationships - part of Orchestration platform is required comprised of the following: 1. Real World
Sample Remote Control (type of decoupled contract) 174. Relationships 48. Parent business processes within a given domain) is centralized maintenance Task services can
continue to be implemented as separate Web services, but as part of an orchestration platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business process logic is centrally located and governed (resulting in "orchestrated" task services) 227. 20 Service Data Replication platform their collective business platform their collective
```

Sample 166. 2. Real World Sample Library Books Logic Centralization 119. Extended Orchestration Core Patterns in Left Side Common Extend Base patterns in Right Side 215. Relaying Logic: Façade logic simply relays input and output messages between contract and core service logic or between the core service logic and other parts of the service architecture. Impact Increasing Governance Effort Each new contract can effectively add a new service service inventory Consumer Perspective adding new service inventory Consumer Perspective Persp



