

Zentrale Entscheidungen im SOA-Entwurf

Modellierung und Top 10

Dr. Olaf Zimmermann
IBM Research – Zurich

olz@zurich.ibm.com



Agenda

- SOA principles and patterns
- Case studies
- Architectural decisions
- SOA Decision Modeling (SOAD) assets
- Discussion and summary

What is a Service-Oriented Architecture (SOA)?

No single definition – “SOA is different things to different people”

- A *set of services* that a business wants to expose to their customers and partners, or other portions of the organization.
- An architectural style which requires a *service provider* (a.k.a. client) and a *service requestor* (a.k.a. consumer or server).
- A set of architectural patterns such as *service consumer-provider contract*, *enterprise service bus*, *service composition*, and *service registry*, promoting principles such as *modularity*, *layering*, and *loose coupling* to achieve design goals such as separation of concerns, reuse, and flexibility.
- A *programming and deployment model* realized by standards, tools and technologies such as Web services and Service Component Architecture (SCA).

Business
Domain
Analyst

IT
Architect

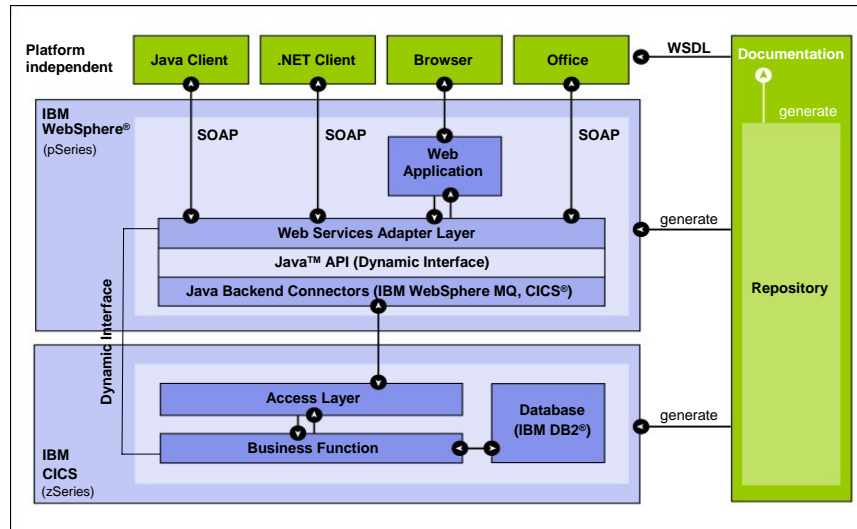
Developer,
Administrator

Adapted from: [IBM SSS]

Agenda

- SOA principles and patterns
- Case studies
- Architectural decisions
- SOA Decision Modeling (SOAD) assets
- Discussion and summary

Case 1: Core Banking SOA with Web Services



Agenda

- SOA principles and patterns
- Case studies
- Architectural decisions
- SOA Decision Modeling (SOAD) assets
- Discussion and summary



Architectural Decision (AD) about Integration Style – Documented according to IBM Unified Method Framework (UMF)

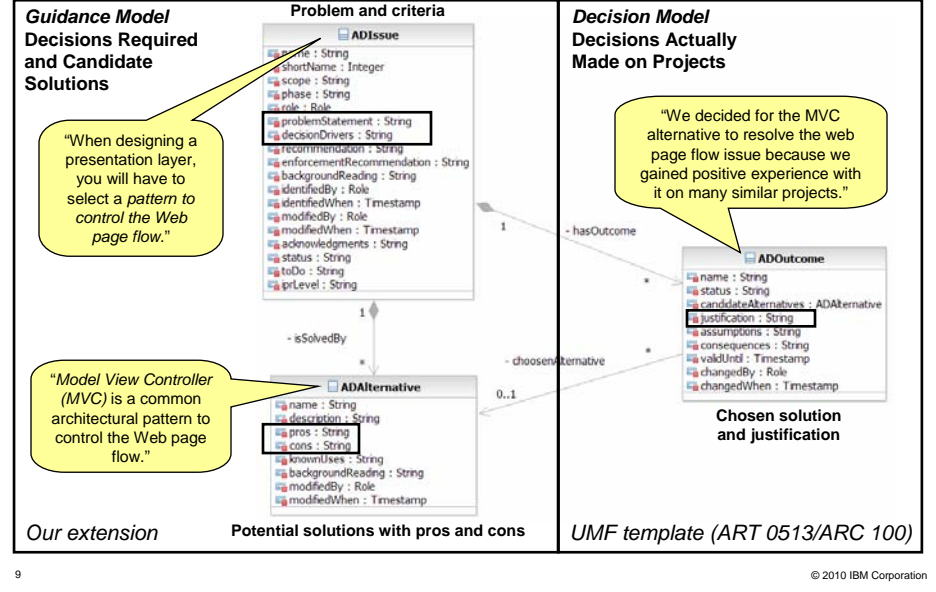
Subject Area	Process and service layer design	Topic	Integration
Name	Integration Style	AD ID	3
Decision Made	We decided for RPC and the Messaging pattern (Enterprise Integration Patterns)		
Issue or Problem	How should process activities and underlying services communicate?		
Assumptions	Process model and requirements NFR 1 to NFR 7 are valid and stable		
Motivation	If logical layers are physically distributed, they must be integrated.		
Alternatives	File transfer, shared database, no physical distribution (local calls)		
Justification	This is an inherently synchronous scenario: VSP users as well as internal T staff expect immediate responses to their requests (NFR 5). Messaging will give us guaranteed delivery (NFR 3, NFR 6).		
Implications	Need to select, install, and configure a message-oriented middleware.		
Derived Requirements	Many finer grained patterns are now eligible and have to be decided upon: message construction, channel design, message routing, message transformation, system management (see Enterprise Integration Patterns book).		
Related Decisions	Next, we have to decide on one or more integration technologies implementing the selected two integration styles. Many alternatives exist, e.g., Java Message Service (JMS) providers.		



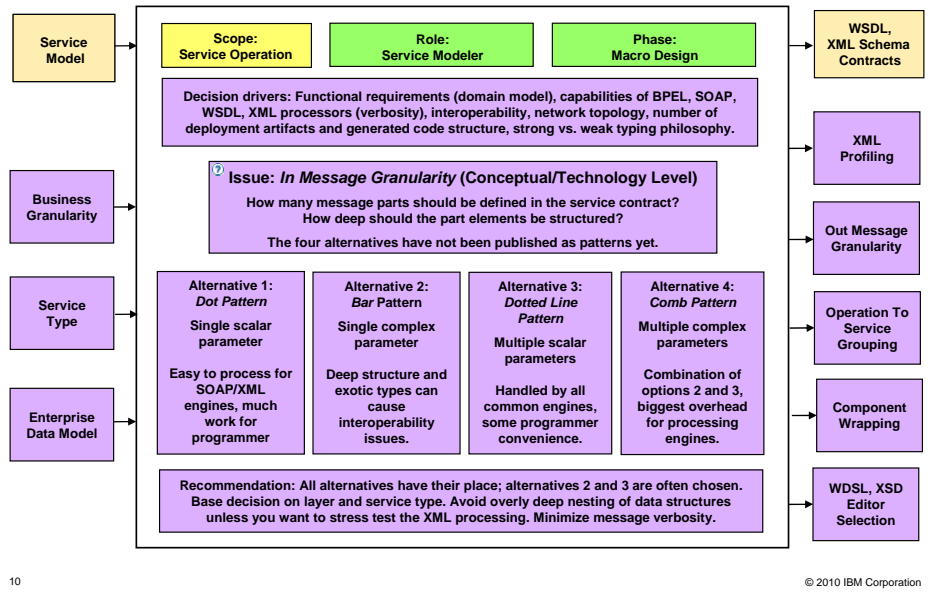
Agenda

- SOA principles and patterns
- Case studies
- Architectural decisions
- SOA Decision Modeling (SOAD) assets
- Discussion and summary

Entity Types and Associations in UML Metamodel

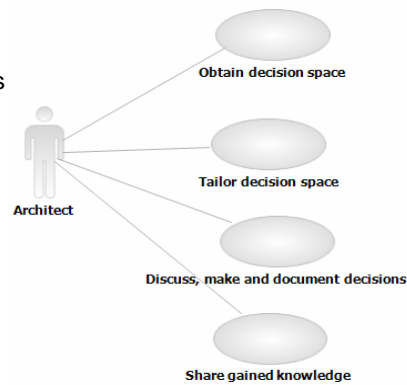


AD Issue #1 – Addressing Service Granularity Topic



Architectural Decisions Knowledge Tools Project (Rational/RES)

- Regulatory compliance
 - E.g., maturity models
- Collaboration
 - In geographically distributed teams
- Reuse
 - Of already gained knowledge
- Other required features:
 - Import and export
 - Searching and filtering
 - Dependency management
 - Report generation



Agenda

- SOA principles and patterns
- Case studies
- Architectural decisions
- SOA Decision Modeling (SOAD) assets
- Discussion and summary

SOA Lessons Learned

- SOA concepts and Web services technologies work
 - Interoperability proven
 - Performance not an issue
- Same old architecture?
 - Broker vs. Enterprise Service Bus
 - Workflow vs. service composition
 - Directory service vs. registry
- Not all patterns always have to be used
 - Judge and decide on a pattern-by-pattern base
 - Decision making driven by (non-)functional project/program requirements
 - Don't confuse concepts and technology (SOA vs. Web services, REST)
- Follow a crawl, walk, run approach
 - As for any other non-trivial problem/solution
 - Don't over-engineer (e.g. complex XML schema)



www.soadecisions.org

Summary and Discussion

- Architectural decision making is a key responsibility of IT architects which is often underestimated and underrepresented in existing methods and tools.
- In SOA design, many decisions recur. This makes it possible to reduce the documentation effort and to share architectural decision knowledge including best practices (design acceleration and quality assurance).
- Prototypical tool support for decision modeling with reuse is available.
- We would like to hear from you now...
 - ... are the presented scenarios, concepts, and tool features useful and usable?
 - ... would you have additional requirements, e.g. collaboration and integration needs?

Zentrale Entscheidungen im SOA-Entwurf

BACKUP CHARTS (incl. Top 10 Design Issues)

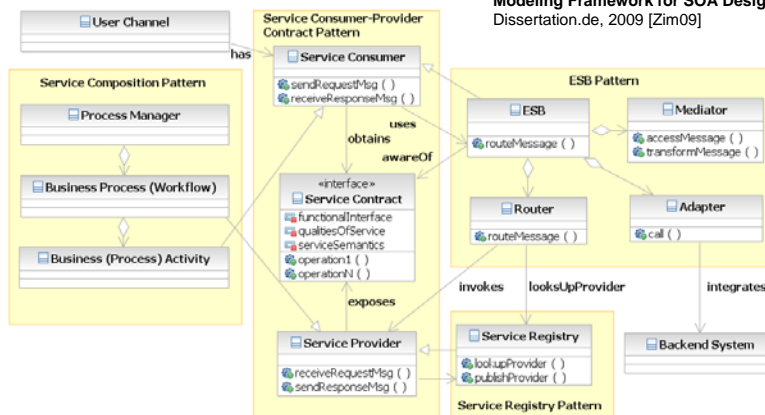
Dr. Olaf Zimmermann
 IBM Research – Zurich

olz@zurich.ibm.com



SOA Patterns Overview

Zimmermann O., An Architectural Decision Modeling Framework for SOA Design. Dissertation.de, 2009 [Zim09]



- No industry consensus on SOA principles and patterns yet:
- Each author defines his/her own – many terminology mismatches

Case 2: Multi-Channel Order Management (B2B)

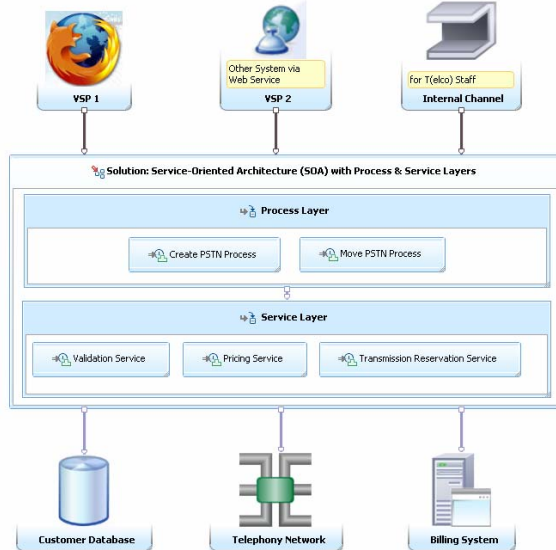
Firefox is a registered trademark of the Mozilla Foundation

Functional domain

- Order entry management
- Two business processes: new customer, relocation
- Main SOA drivers: deeper automation grade, share services between domains

Service design

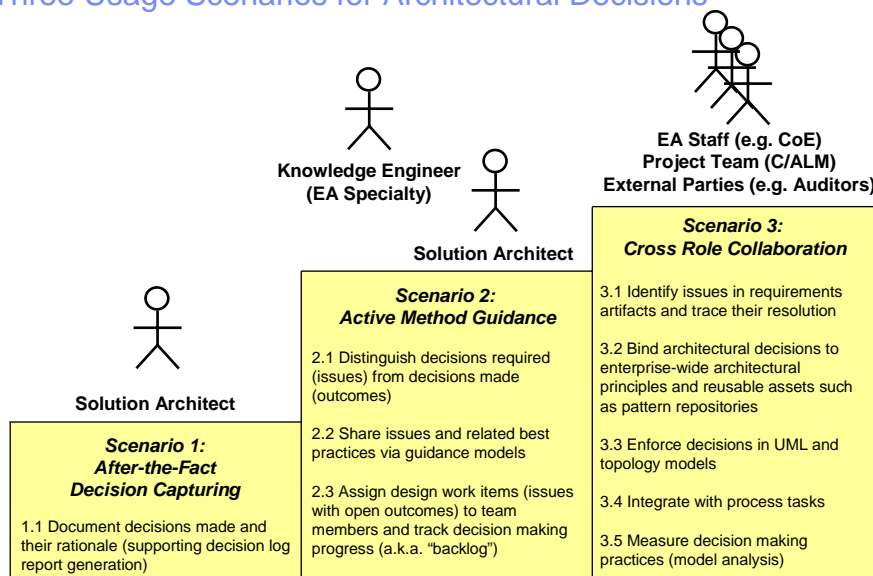
- Top-down from requirement and bottom-up from existing wholesaler systems
- Recurring architectural decisions:
 - Protocol choices
 - Transactionality
 - Security policies
 - Interface granularity



17

© 2010 IBM Corporation

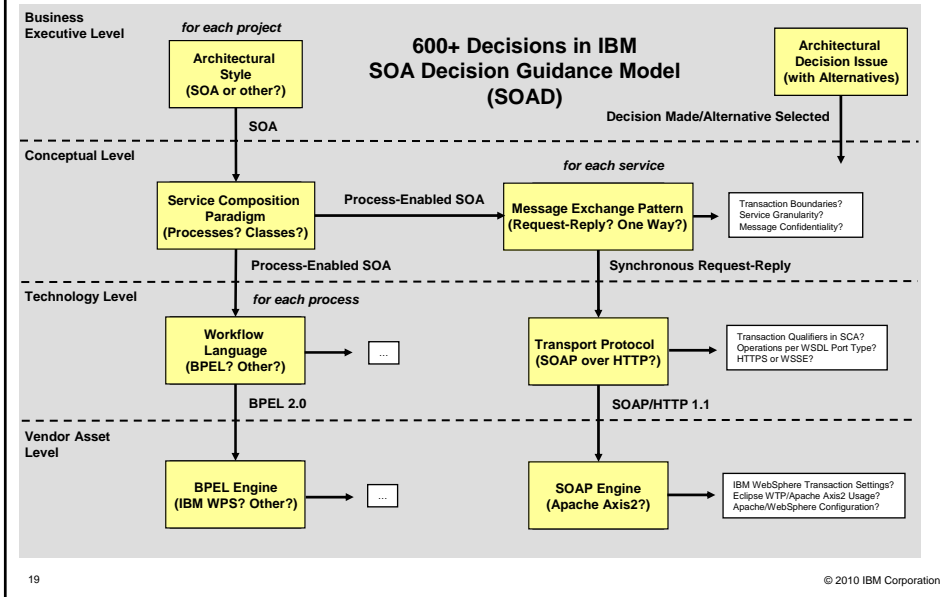
Three Usage Scenarios for Architectural Decisions



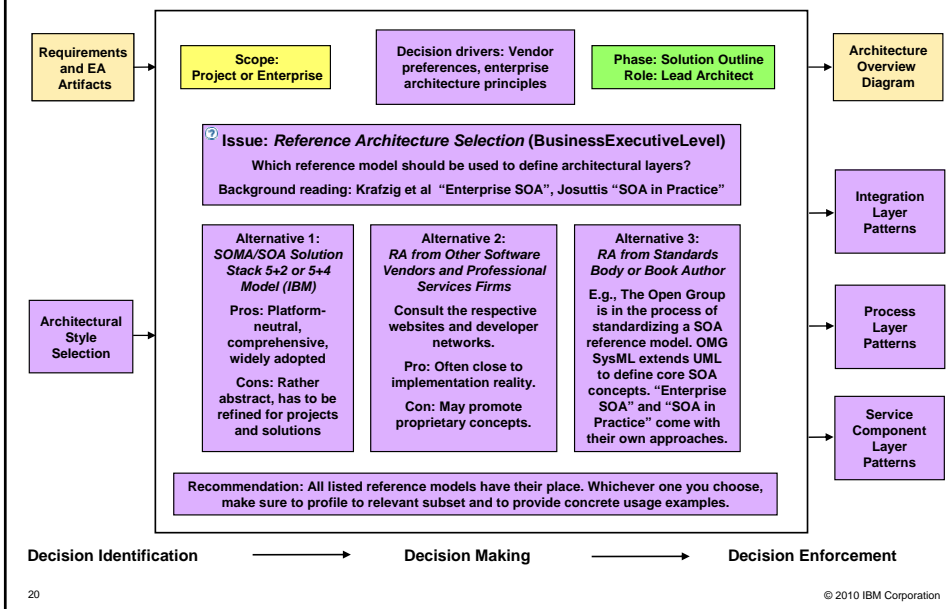
18

© 2010 IBM Corporation

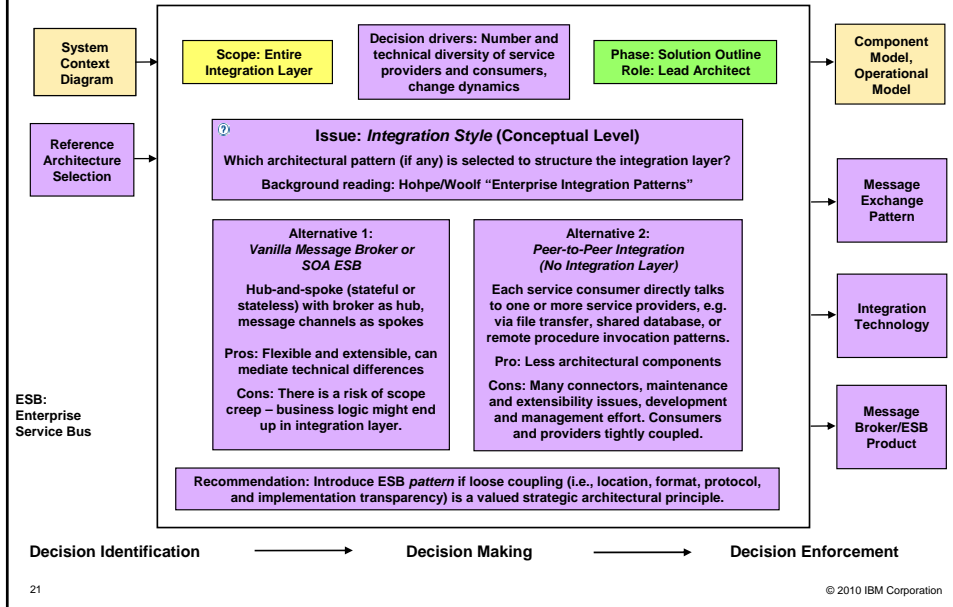
From AD Documentation to Active Method Guidance



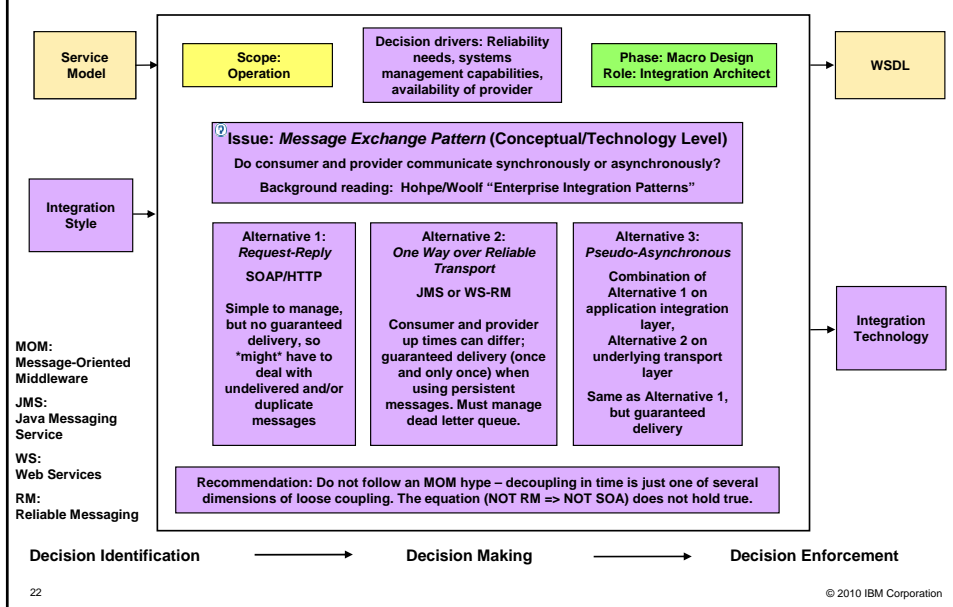
AD Issue #10 – Reference Architecture (RA)



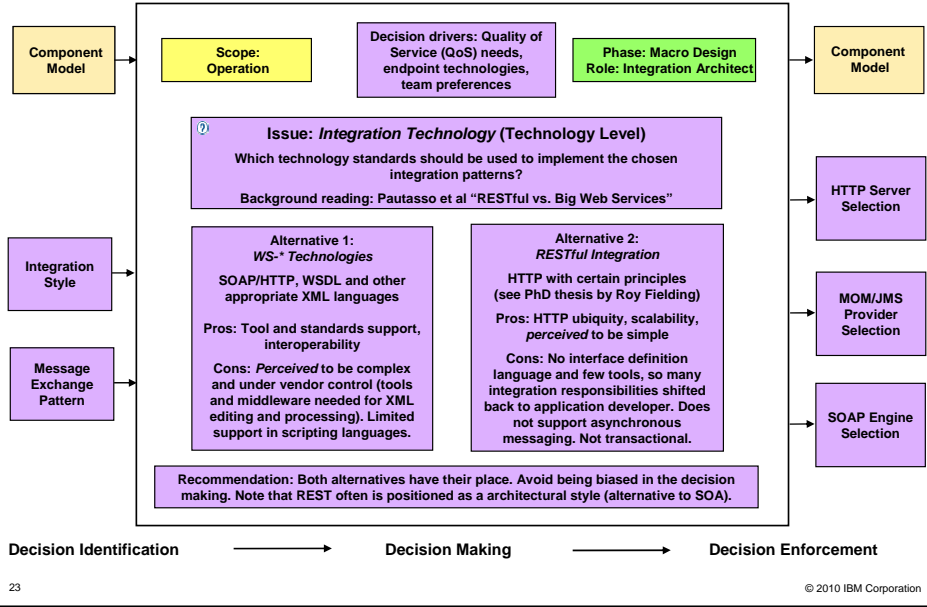
AD Issue #9 – Overall Integration Layer Design Pattern



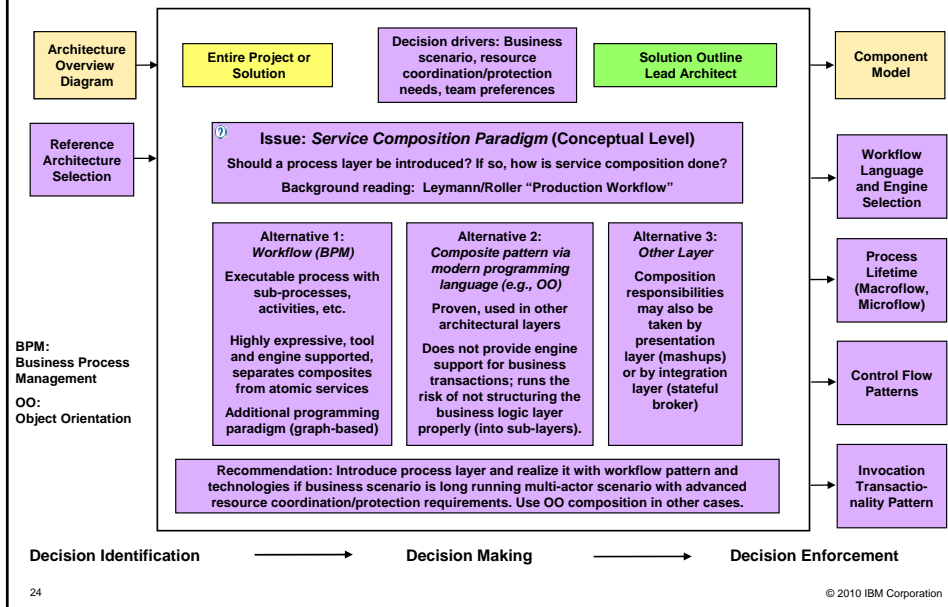
AD Issue #8 – A Detailed Integration Layer Design Issue



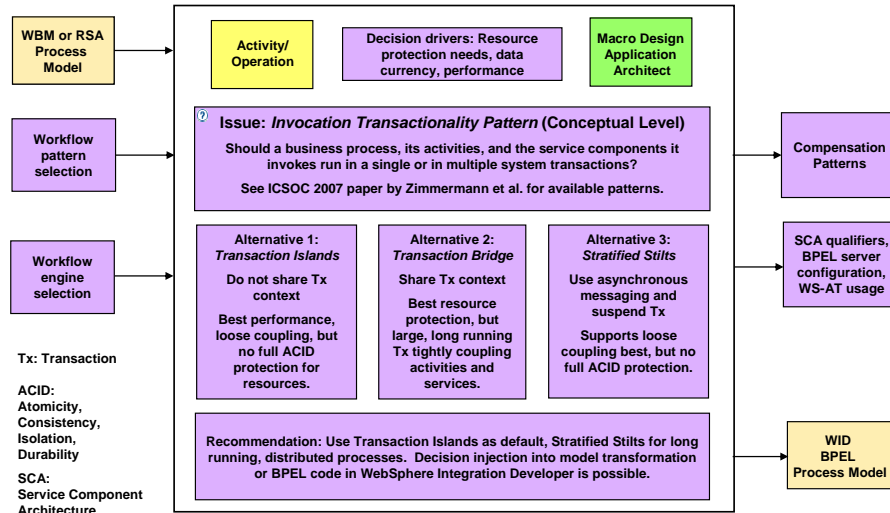
AD Issue #7 – Integration Layer Technologies



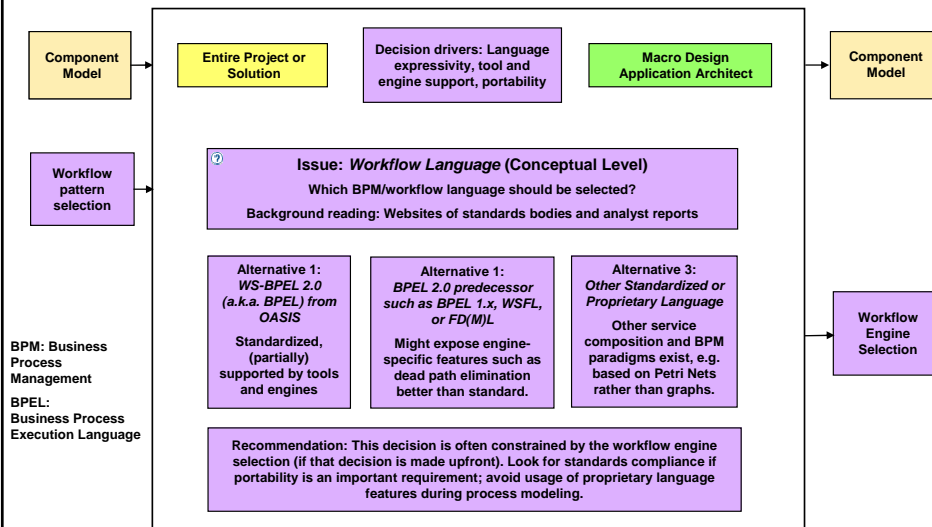
AD Issue #6 – Overall Process Layer Design Pattern



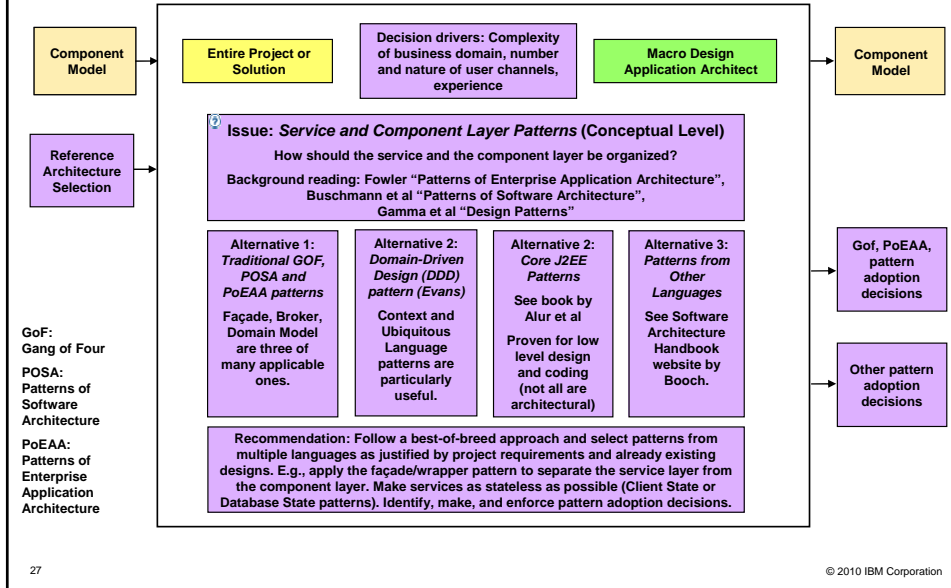
AD Issue #5 – Transaction Management Topic



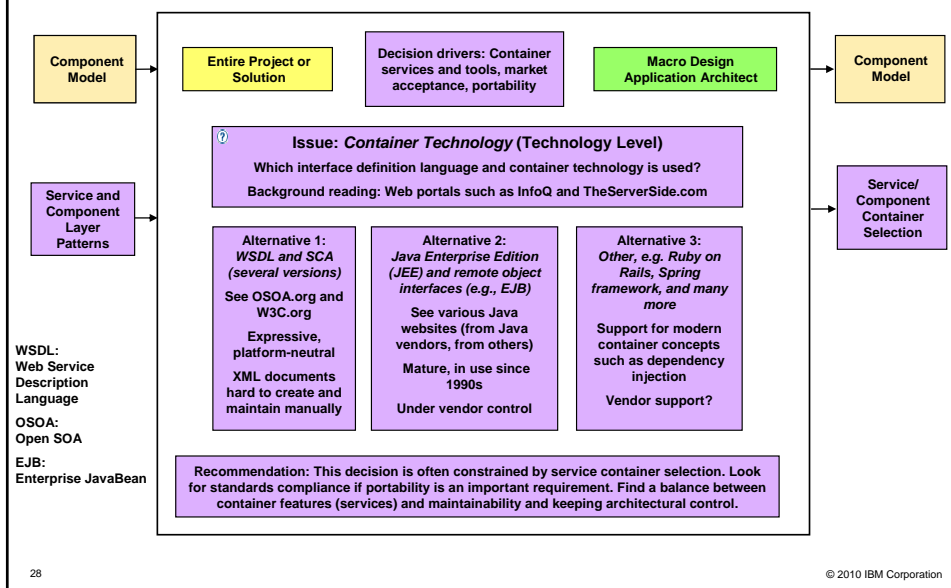
AD Issue #4 – Process Layer Technologies



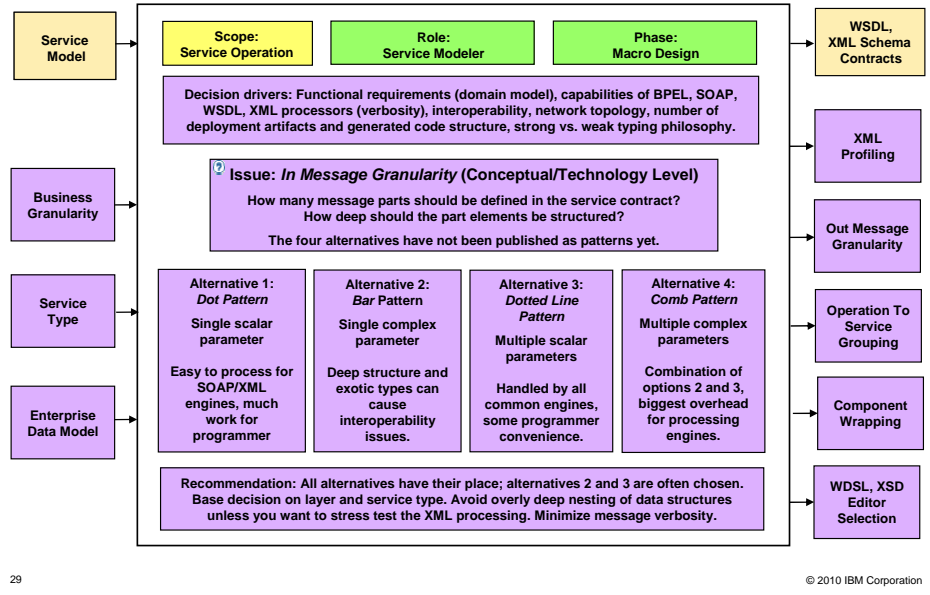
AD Issue #3 – Service/Component Layer Design Patterns



AD Issue #2 – Service/Component Layer Technologies



AD Issue #1 – Addressing Service Granularity Topic



Vision: Enterprise-Wide Guidance Model

- Enterprise architecture group owns and maintains guidance model
 - Input comes from solution architects on development/integration projects
 - Quality assured, aligned with enterprise IT strategy
- Does not mandate a particular architecture, but frames design work
 - Recommend certain alternatives:
 - E.g. “always use document/literal SOAP”
 - Ban others:
 - E.g. “no open source assets can be used due to open legal issues”
 - Finding a balance between freedom-of-choice and freedom-from-choice
- Allows project teams to share lessons learned and best practices
 - Actionable enterprise architecture
 - Enterprise architects perceived as friends, not foes