

VERTRAULICH

SOA in Großunternehmen

Universitätsvorlesung 1/3

Dieser Bericht ist ausschließlich für Mitarbeiter des Klienten bestimmt. Die Verteilung, Zitierung und Vervielfältigung – auch auszugsweise – zum Zwecke der Weitergabe an Dritte ist nur mit vorheriger schriftlicher Zustimmung von McKinsey & Company, Inc., gestattet.

Die hier zusammengefassten Texte und Grafiken wurden von McKinsey & Company, Inc., im Rahmen einer Präsentation eingesetzt; sie stellen keine vollständige Dokumentation der Veranstaltung dar.

Blockvorlesung

SOA in Großunternehmen

Teilnehmerbeschreibung

Die Vorlesung kann von Studenten der Informatik und Wirtschaftsinformatik besucht werden mit einem technischem Grundverständnis der IT und Grundlagenwissen zu Applikationen. Betriebswirtschaftlicher Hintergrund ist hilfreich aber nicht notwendig.

Dozentenbeschreibung

Dieter Dratwa ist ein Berater und Spezialist für 'IT Architektur' in McKinsey & Company. Basiert im Züricher Büro arbeitet er seit 2004 schwerpunktmäßig in den Sektoren Banken, Telekommunikation, Transport und öffentlicher Sektor. Zuvor arbeitete er 3 Jahre als IT Berater and Architekt bei Accenture.

Die Grundlagen zur Dozententätigkeit bilden Informatikdiplom der Universität Trier, Microsoft Trainer Zertifikat sowie HWK Trier Trainer für Wirtschaftsinformatiker.

Kontakt

 dratwa@ddratwa.de  +41 79 477 8124



McKinsey&Company



Eine typische Unternehmensstruktur

Probleme einer IT-Abteilung

SOA-Grundlagen

SOA-Laufzeitumgebung

SOA-Beispielanwendung

Das Deutsche Post World Net ist ein großes und globales Unternehmen



Schnelle Fakten - DHL

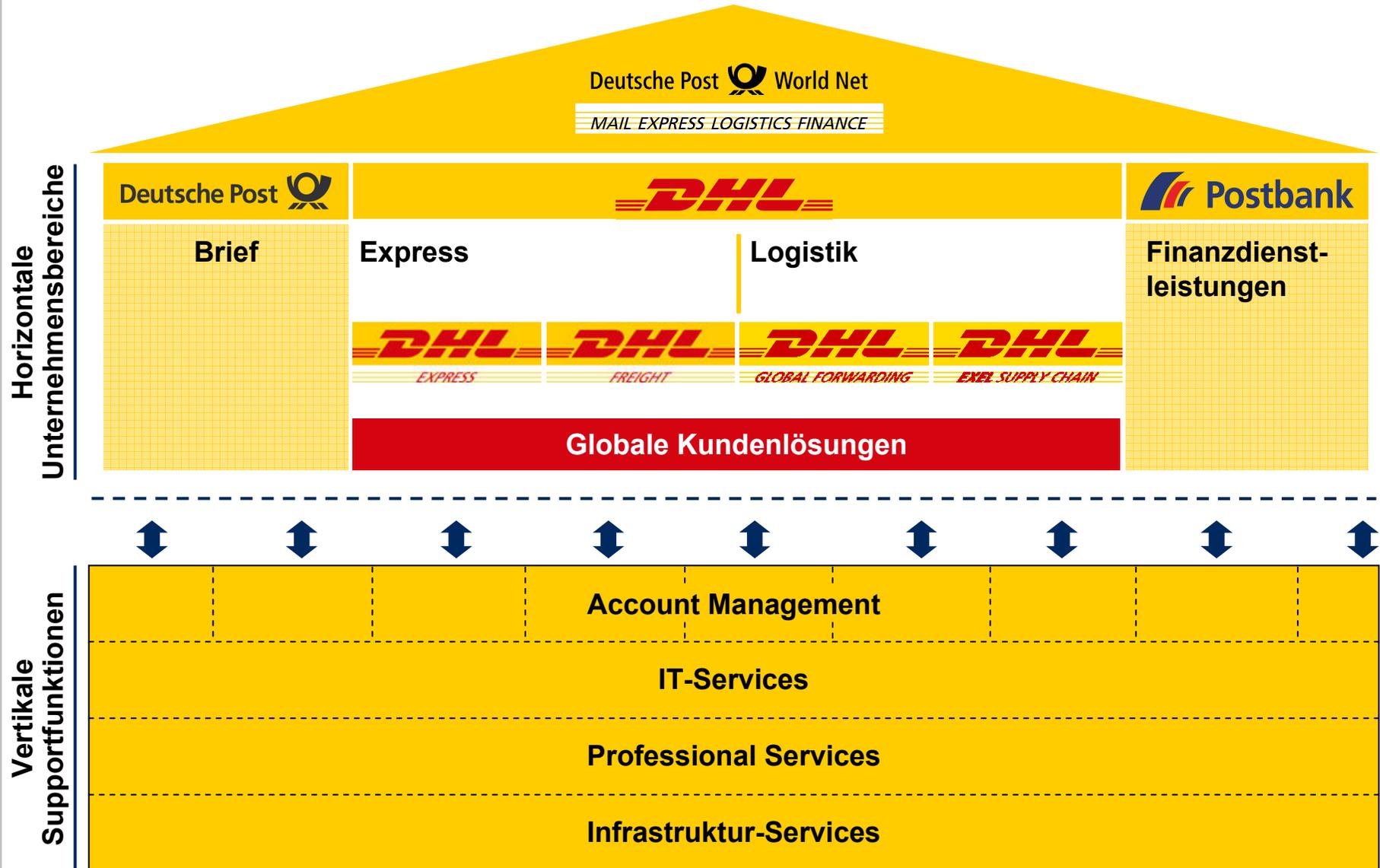
Das Deutsche Post World Net ist weltweiter Marktführer in der Logistikbranche. Deutsche Post, DHL und Postbank repräsentieren eine umfassende Palette integrierter Services für das Management von Post-, Waren-, Informations- und Finanzströmen.

Zahl der Mitarbeiter:	Ca. 285.000
Zahl der Niederlassungen:	Ca. 6.500
Zahl der Hubs**, Lager & Terminals:	Über 450
Zahl der Gateways:	240
Zahl der Flugzeuge*:	420
Zahl der Fahrzeuge:	76.200
Zahl der Länder & Territorien:	Mehr als 220
Sendungen pro Jahr:	Über 1,5 Mrd.
Versandziele:	120.000

* Incl. Leasingmaschinen

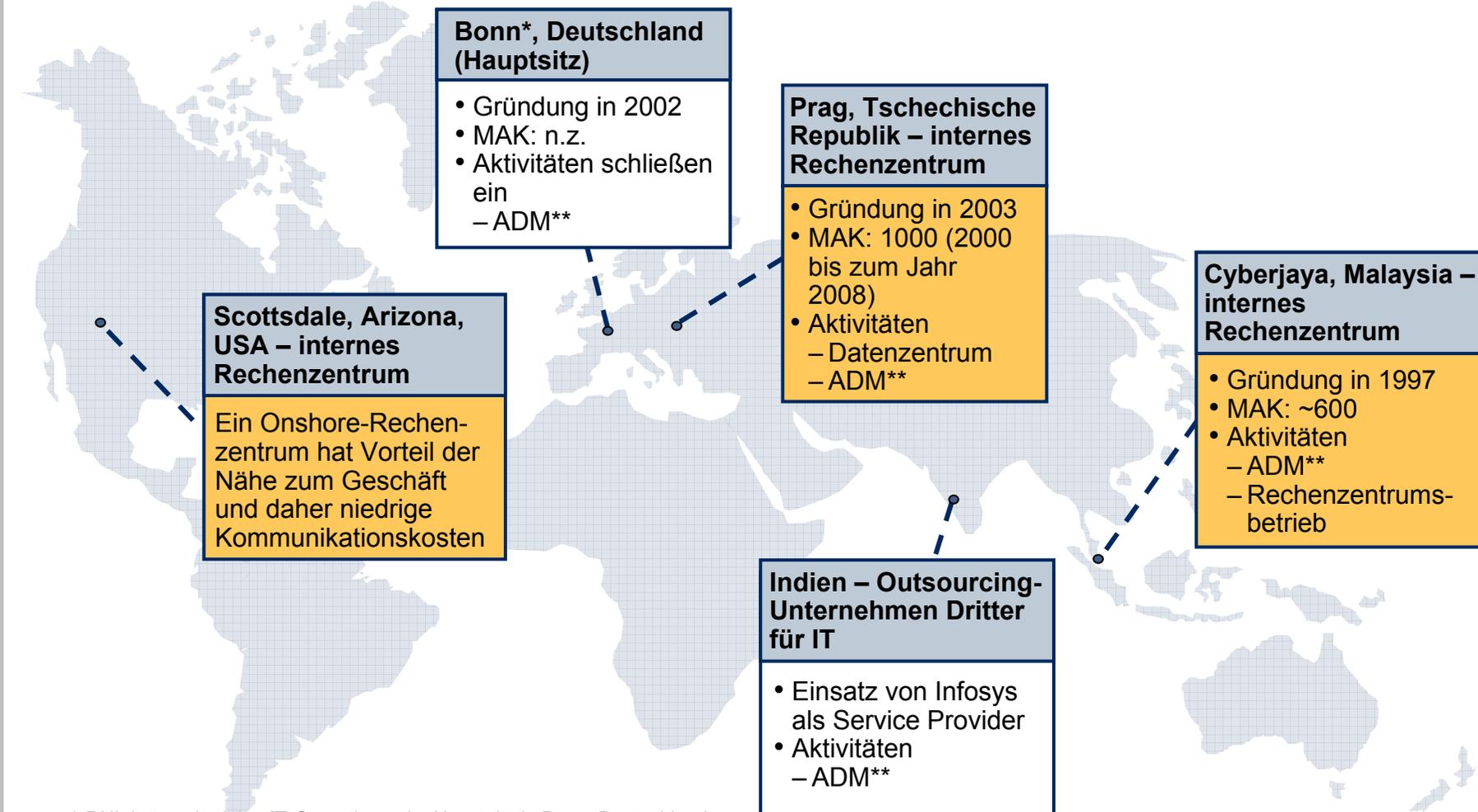
** Hauptumschlagplätze

Struktur von Deutsche Post World Net



Globales Betriebsmodell für DHL-IT

DHL verfolgt die "Follow-the-Sun"-Strategie mit 3 regionalen konzern-eigenen Zentren, wobei jedes während des Tages an das nächste übergibt

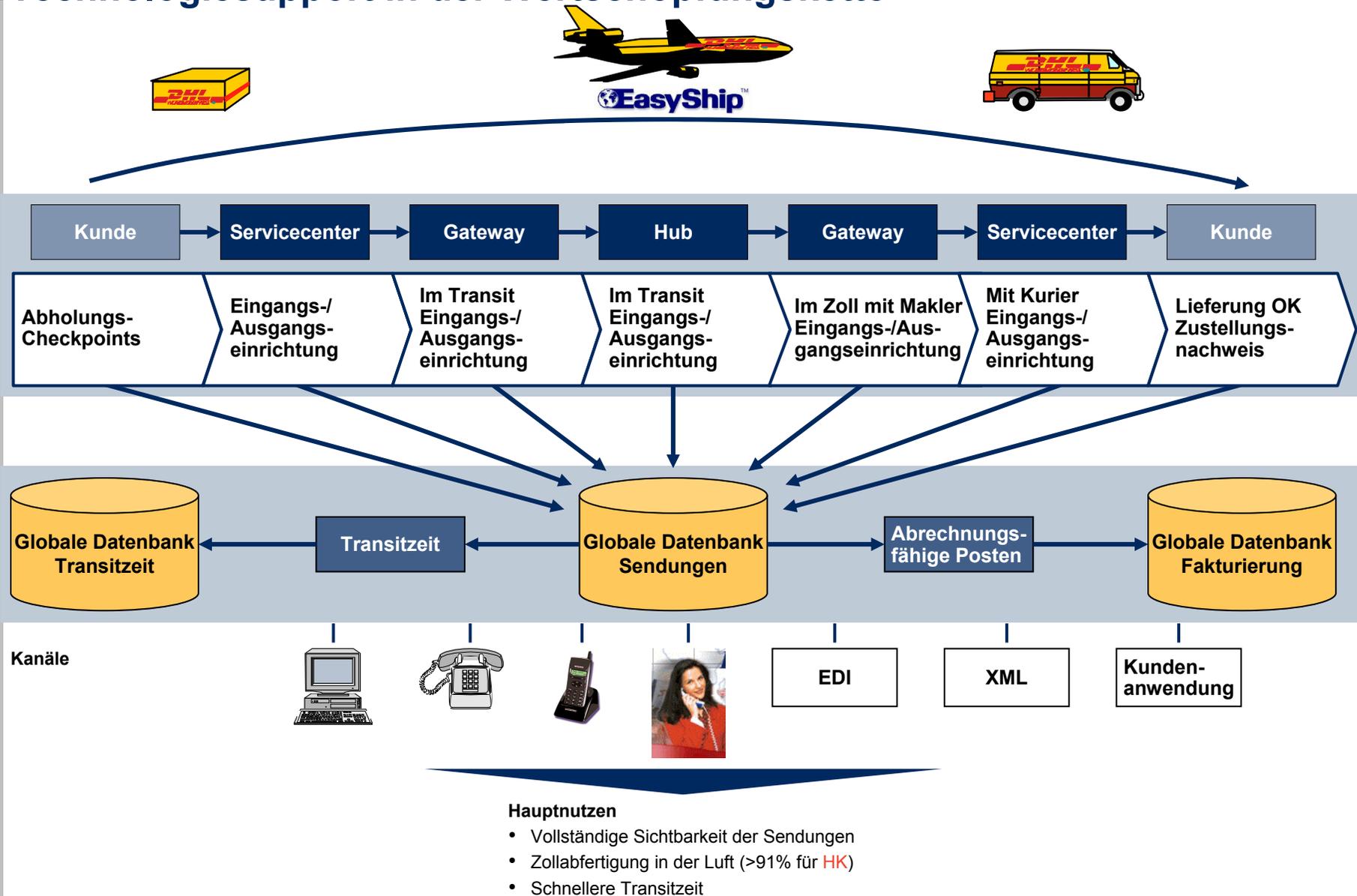


* DHL hat auch einige IT-Operationen im Hauptsitz in Bonn, Deutschland

** Anwendungsentwicklung und -wartung

Quelle: Web; Presseberichte, McKinsey zu IT

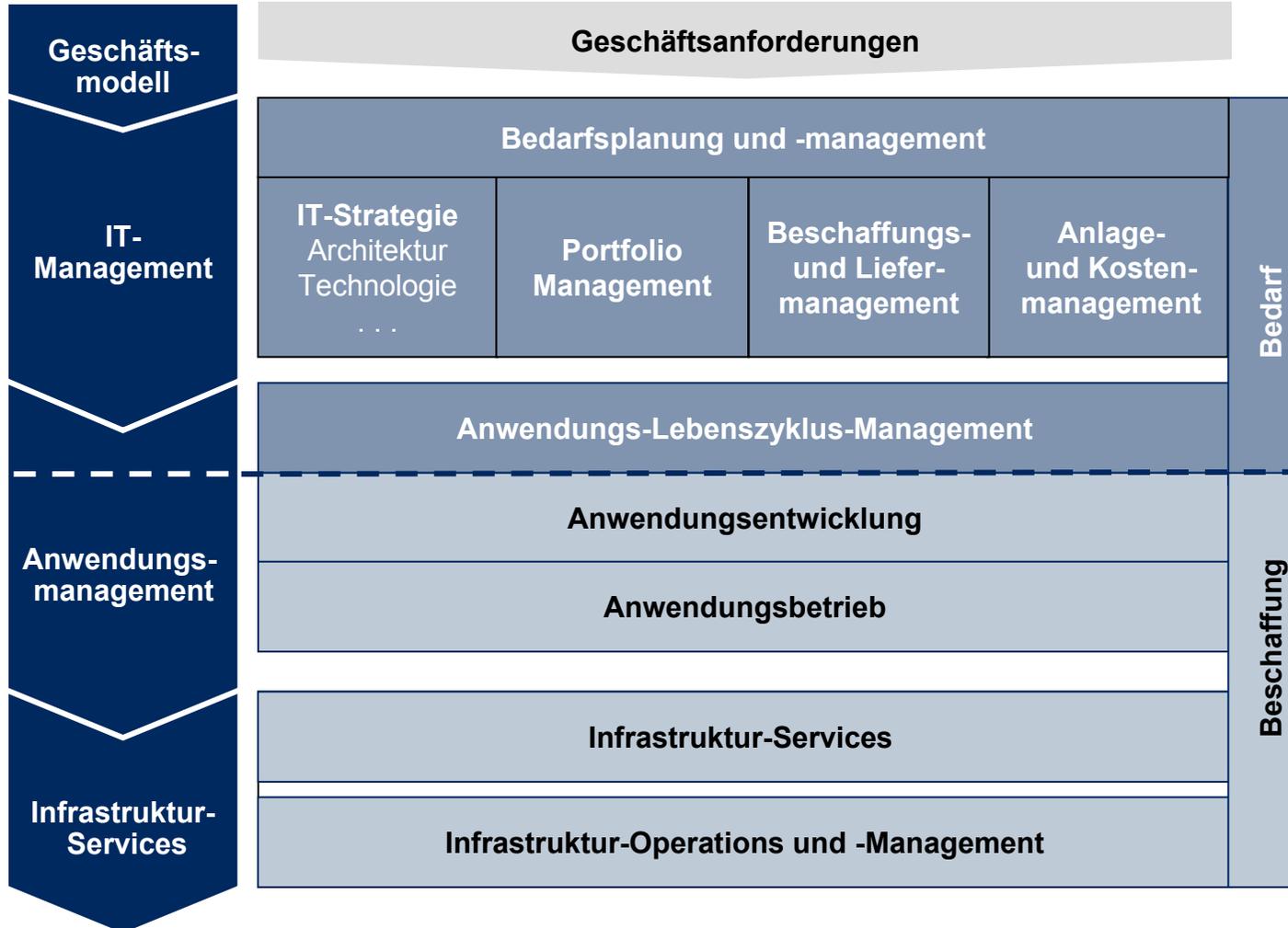
Technologiesupport in der Wertschöpfungskette



Trennung der Verantwortlichkeiten zwischen Bedarf und Beschaffung auf höchster Ebene

IT-Bedarf
 IT-Beschaffung

Wertschöpfungskette



Klare Trennung zwischen Bedarfs- und Bereitstellungsaufgaben ist Haupterfolgsfaktor

Das Organisationsmodell sollte die Lieferung von IT-Services an die Geschäftsbereiche unterstützen

	Geschäftsstruktur	Merkmale
Zentral	<pre> graph TD CEO[CEO] --- Marketing[Marketing] CEO --- Produktion[Produktion] CEO --- Operations[Operations] CEO --- IT[IT] CEO --- Buchhaltung[Buchhaltung] </pre>	<ul style="list-style-type: none"> • Begrenztes Produktportfolio mit ähnlichen Produktmerkmalen/ähnlichem Produkt-handling • IT-Impact basiert auf Automatisierung • Insgesamt niedriges IT-Investment
Förderal	<pre> graph TD CEO[CEO] --- GB1[GB* 1] CEO --- GB2[GB 2] CEO --- GB3[GB 3] CEO --- GB4[GB 4] CEO --- IT[IT] IT --- IT1[IT] IT --- IT2[IT] </pre>	<ul style="list-style-type: none"> • Großes Produktportfolio mit einzigartigen Produktmerkmalen/einzigartigem Produkt-handling • Hohe Zahl an "einzigartigen" Anwendungen • Geschäftsbereichsübergreifend wenige gemeinsame Anwendungen
Dezentral	<pre> graph TD CEO[CEO] --- GB1[GB 1] CEO --- GB2[GB 2] CEO --- GB3[GB 3] CEO --- GB4[GB 4] CEO --- Admin[Admin] GB1 --- IT1[IT] GB2 --- IT2[IT] GB3 --- IT3[IT] GB4 --- IT4[IT] Admin --- IT5[IT] </pre>	<ul style="list-style-type: none"> • Differenzierte Produktportfolios • Hoher Einkauf und Verkauf von Portfoliounternehmen
Outgesourct	<pre> graph TD CEO[CEO] --- GB1[GB 1] CEO --- GB2[GB 2] CEO --- Admin[Admin] CEO --- CEO_GB3[CEO/GB3] CEO_GB3 --- GB3[GB 3] </pre>	<ul style="list-style-type: none"> • Streben nach der Schaffung eines IT-spezifischen Geschäfts, das auf dem Markt konkurriert z.B. SABRE • IT wird als besondere "Initiative" behandelt, die geeignete und spezialisierte Ressourcen erfordert, z.B. Aufbau von E-Business

Weit verbreitete Modelle, die für die IT-Rolle gut funktionieren

DACH* Top 20 – Öffentlich gelistete Gesellschaften nach IT-Budget

Geschäftsjahr 2004, in Mio. EUR

Name	IT-Budget	Land	Branchenkategorie
Allianz AG	5.355	Deutschland	Versicherung
DaimlerChrysler AG	3.963		Automobilhersteller
Siemens AG	3.546		Hersteller
Zurich Financial Services AG	2.983	Schweiz	Versicherung
Münchener Rück AG	2.867	Deutschland	Versicherung
Volkswagen AG	2.482		Automobilhersteller
Deutsche Telekom AG	2.188		Telekommunikation
Credit Suisse Group	2.180	Schweiz	Bankwesen
Deutsche Post AG	2.158	Deutschland	Transportwesen
ThyssenKrupp AG	1.856		Metallverarbeitende Industrie
UBS AG	1.737	Schweiz	Bankwesen
Swiss Re	1.456	Deutschland	Versicherung
Metro AG	1.354		Lebensmittel
Nestle SA	1.349	Schweiz	Lebensmittel
Deutsche Bank AG	1.341	Deutschland	Bankwesen
E.ON AG	1.248		Elektrizität
Bayerische Motoren Werke AG	1.237		Automobilhersteller
Ergo Versicherungsgruppe AG	1.176	Schweiz	Versicherung
Novartis AG	1.136	Schweiz	Medizinische Verbrauchsgüter
RWE AG	1.055	Deutschland	Elektrizität

* DACH: Deutschland (D), Österreich (A), Schweiz (CH)

Quelle: Bloomberg

Inhalt



Eine typische Unternehmensstruktur

Probleme einer IT-Abteilung

SOA-Grundlagen

SOA-Laufzeitumgebung

SOA-Beispielanwendung

Hauptprobleme, die den CIO* beschäftigen

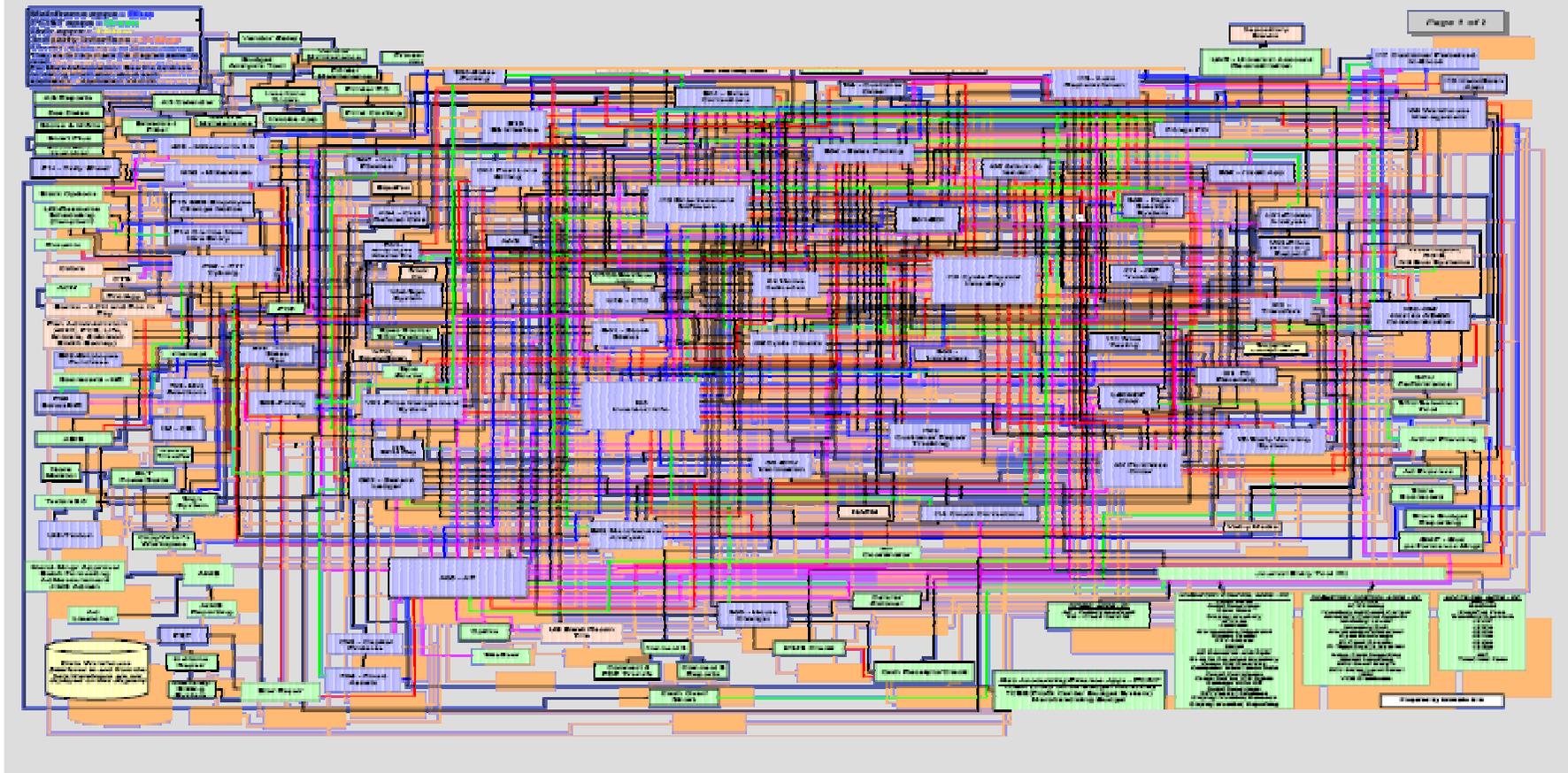
Einzelheiten auf den nächsten Seiten

- 
- IT-/GB-Strategieanpassung
 - Senkung der IT-Kosten
 - Maximierung der IT-Effizienz
 - Anwendungsintegration
 - Verwaltung von Legacy-Systemen

* Chief Information Officer = Leiter für Informationstechnologie
Quelle: McKinsey

Anwendungsintegration

Beispiel Anwendungslandschaft in Verbrauchsgüter Industrie

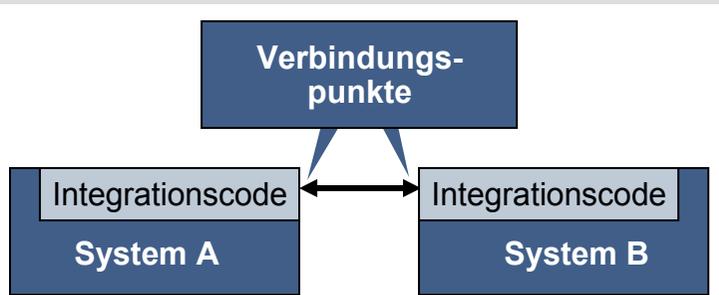


Quelle: McKinsey

Integrationsansätze (1/2)

Punkt-zu-Punkt

- Eine Anwendung interagiert direkt mit einer anderen Anwendung über einen Verbindungspunkt
- Der Integrationscode (Adapter) ist speziell für den Kunden für jede Anwendung geschrieben und ändert sich mit jeder neuen Systemergänzung
- Mehr als 2 Anwendungen führen zu einer Mesh-Architektur, die auch als N-Quadrat-Problem bezeichnet wird



Vorteile

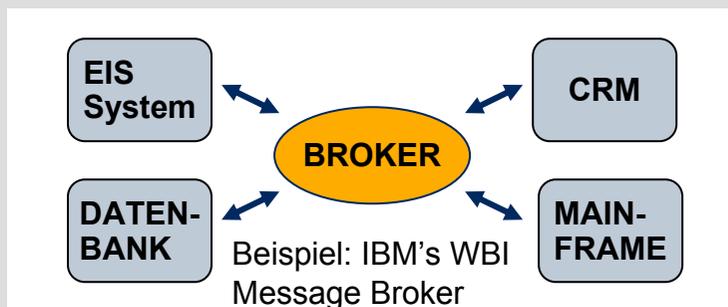
- Einfachheit und deshalb oft der erste Integrationsansatz

Nachteile

- Das N-Quadrat-Problem, dass für ein Update in einer einzigen Anwendung N Adapter geändert werden

Hub & Spoke

- Alle Anwendungen senden und empfangen Daten durch einen *zentralisierten Server* oder eine zentralisierte Maschine
- Daten von den Quellanwendungen werden durch die Nutzung von spezialisierten Adaptern extrahiert und an die Maschine gesendet.
- Die zentrale Maschine oder der zentrale Hub transformiert und routet die Daten zu einer oder mehreren Zielanwendungen basierend auf den Transformations- und Routingregeln, die in der Maschine vorgegeben sind



Vorteile

- Lose Kopplung
- Reduzierte Komplexität
- Ein Minimum an invasiver Programmierung

Nachteile

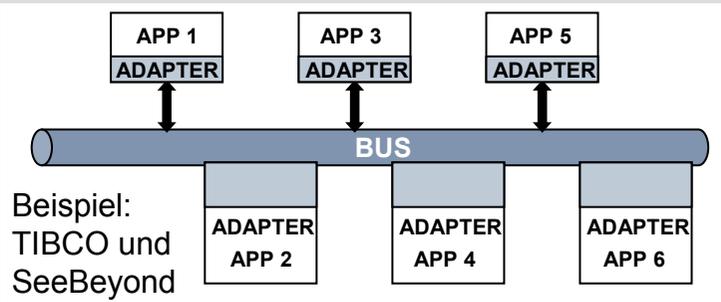
- Ein Engpass beim zentralen Broker beeinflusst die Performance
- Alle Anwendungsadapter sollten auf einen einzigen proprietären Hub abgestimmt sein

Quelle: McKinsey

Integrationsansätze (2/2)

Message-Bus-Architektur

- Alle Knoten sind in Reihe entlang eines *gemeinsamen Kommunikationsbackbones* (Bus) verbunden
- Die Quellanwendungen stellen dem Bus Daten direkt zur Verfügung. Diese Art der Bus-Architektur wird manchmal auch als "publish/subscribe" bezeichnet, bei dem die Quelle "Publisher"/"Producer" und die Ziel- oder Empfängeranwendung, die ein Ereignis empfängt, "Subscriber" oder "Consumer" genannt wird



Vorteile

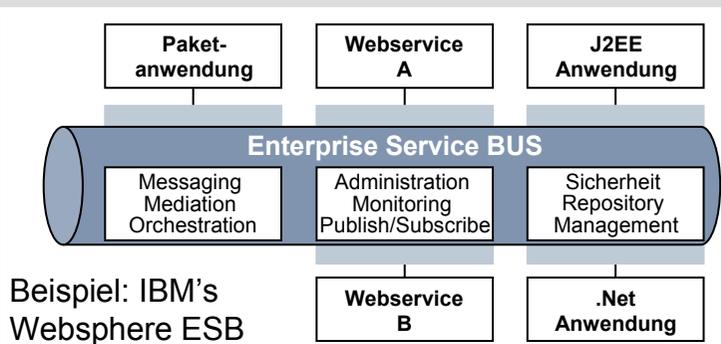
- Kein einzelner Störpunkt
- Skalierbare Architektur

Nachteile

- Jede Anwendung erfordert einen separaten Adapter. Nicht sinnvoll, wenn eine Anwendung Daten mit einer unvorhersagbaren Anzahl von Identitäten teilen muss

ESB Enterprise Service Bus

- Kombiniert die Vorteile von Hub-and-Spoke und Message-Bus. Nachrichten werden intelligent weitergeleitet, und verteilte Adapter wandeln die Daten in ein einheitliches logisches Modell (Föderation) um.
- Es basiert auf einem SOA-Framework, die verteilten Services das Teilen von Daten erlaubt
- Andere Services beinhalten Repository Services, Administration, Management etc.



Vorteile

- Es basiert auf SOA-Standards
- Leichtgewichtig und basiert nicht auf einem proprietären Code

Nachteile

- Die meisten ESB-Produkte stellen keine Service-Orchestration und -mediation (Umwandlung und Weiterleitung) zur Verfügung
- Keine inhaltliche Konsistenz bei den ESB-Produkten

Quelle: McKinsey

Semantische Integration

Unternehmen sehen sich mit dem Integrationsproblem konfrontiert, da immer mehr Prozesse Anwendungen umfassen (durch Gestaltung oder neue Anforderungen wie eine kundenzentrierte Sichtweise oder die Fähigkeit, Cross Selling zu betreiben). Ein Beispiel: Zwei Parteien müssen eine Integration durchführen, und sie sprechen über ein "Namens"-Feld. Eine Partei versteht darunter Vor- und Nachname zusammen in einem Textfeld, die andere Partei versteht darunter Nachname, da sie ein zusätzliches Feld mit der Bezeichnung "Vorname" hat. Semantisch gesehen sprechen die Parteien nicht über die gleiche Sache, selbst wenn sie in gleicher Weise bezeichnet wird.

Neben den Methoden der technischen Integration (**siehe vorherige Seite**) stellt ein **zugrunde liegendes** Problem der **semantischen Integration** die Hauptherausforderung dar. Die semantische Integration ist der **Harmonisierungsprozess** der Benutzung der Geschäftssemantik zur Automatisierung der Kommunikation zwischen Computersystemen. Diese Semantik konzentriert sich auf die Bedeutung von Daten.

Middleware ist die Schnittstellensoftware, die als eine wieder verwendbare, erweiterbare Reihe von Services und Funktionen betrachtet werden kann, die gemeinhin von vielen Anwendungen benötigt werden, um in einer Netzwerkumgebung gut zu funktionieren

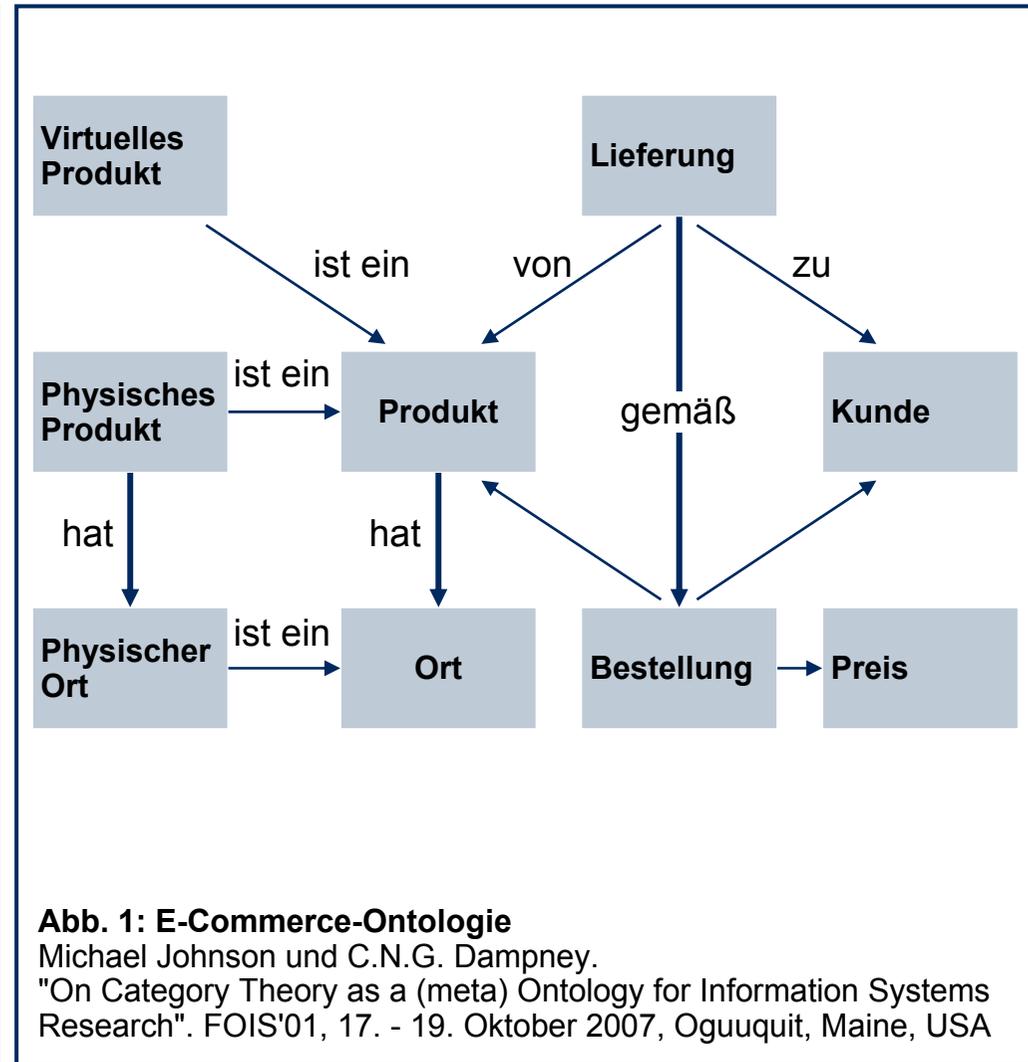


Abb. 1: E-Commerce-Ontologie

Michael Johnson und C.N.G. Dampney.

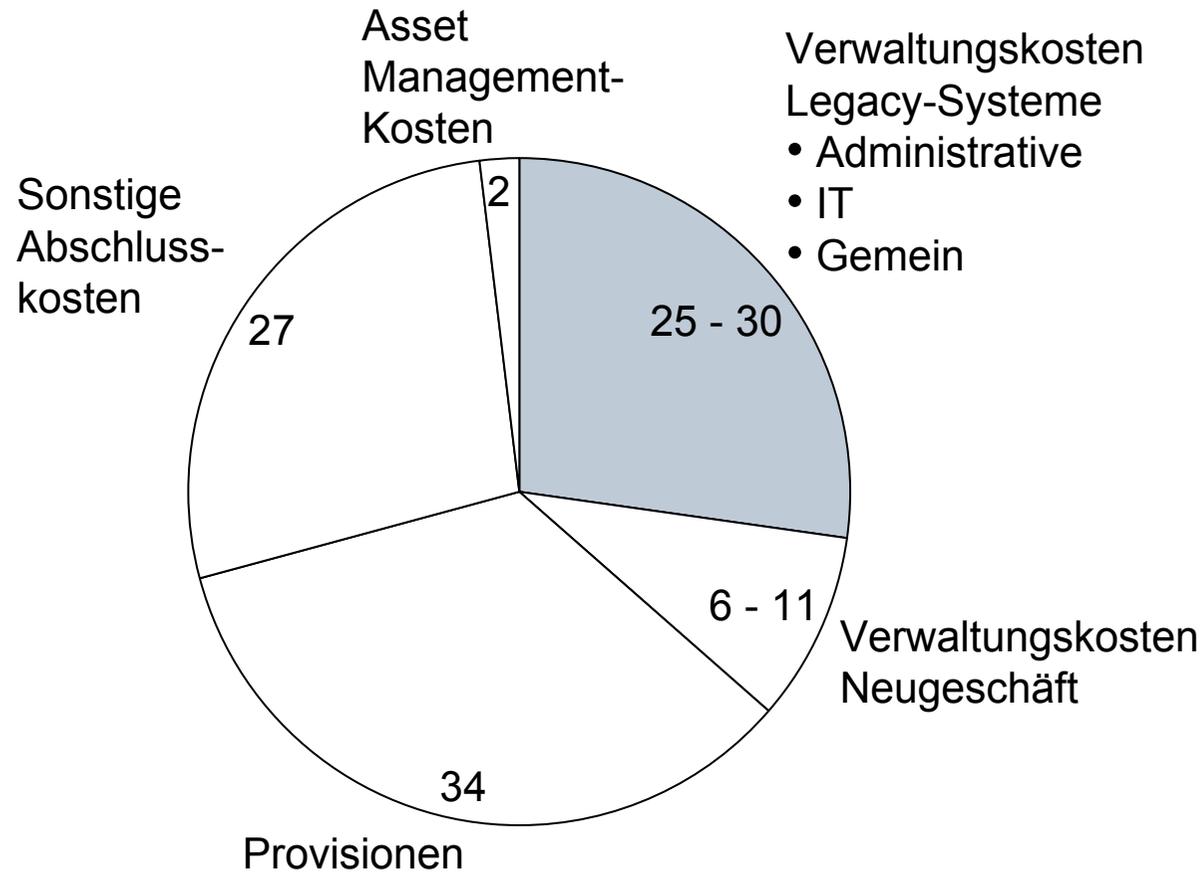
"On Category Theory as a (meta) Ontology for Information Systems Research". FOIS'01, 17. - 19. Oktober 2007, Ogunquit, Maine, USA

Ungefähr ein Viertel der gesamten Lebensversicherungskosten wird für die Verwaltung von Legacy-Systemen aufgewandt

BEISPIEL
VERSICHERUNG

INDIVIDUELLE AUFWENDUNGEN LEBEN

in Prozent

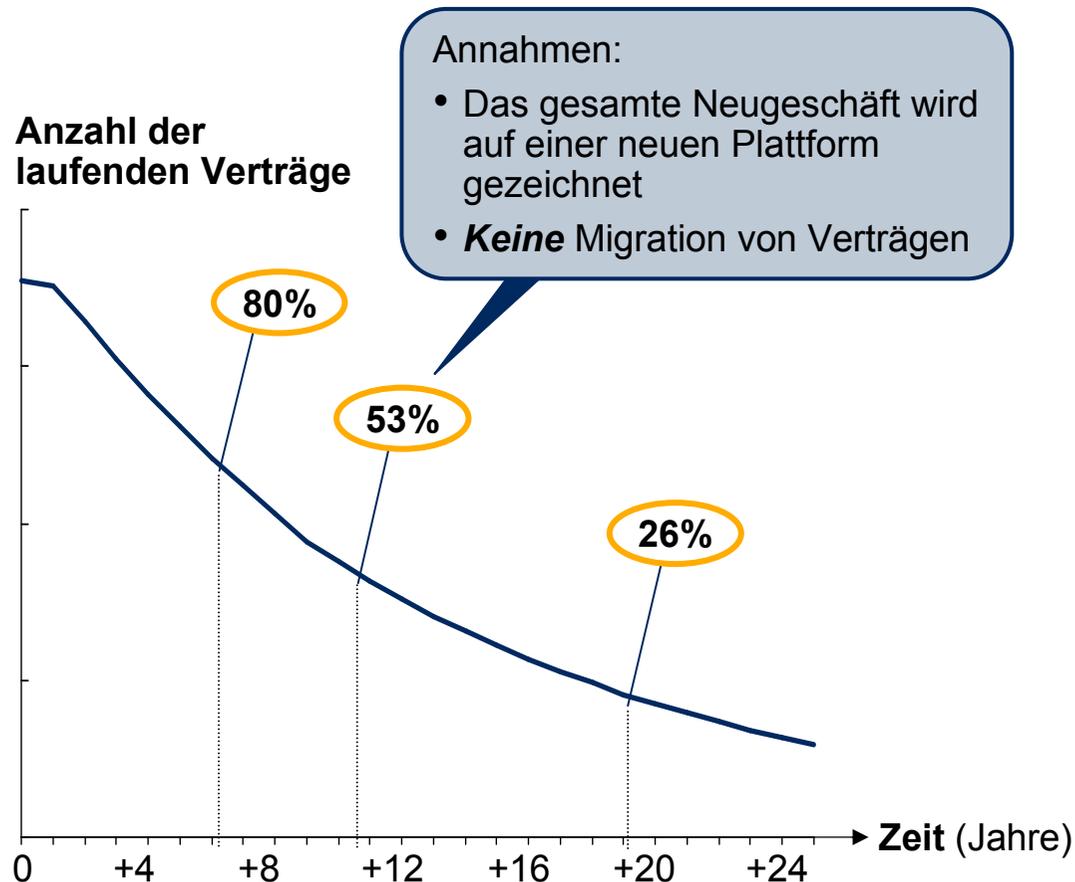


Quelle: McKinsey

Die Langlebigkeitsstruktur des Legacy-Portfolios als Ursprung bedeutender Management- und Kostenherausforderungen

BEISPIEL
VERSICHERUNG

ENTWICKLUNG EINER ANZAHL VON INDIVIDUELLEN LEBENSVERSICHERUNGS-
VERTRÄGEN, WENN KEIN NEUGESCHÄFT HINZUKOMMT



- Gravierendes **Managementproblem**
 - Halten von Mitarbeitern / Know-how
 - Schnittstellen zu Neugeschäft und Distribution
 - Komplexitätsmanagement
- Gravierendes **Kostenproblem**
 - Verdopplung der Fixkosten getrieben von der Anzahl von Plattformen
 - Subkritische Volumen
 - Komplexitätskosten

Quelle: McKinsey

Inhalt



Eine typische Unternehmensstruktur

Probleme einer IT-Abteilung

SOA-Grundlagen

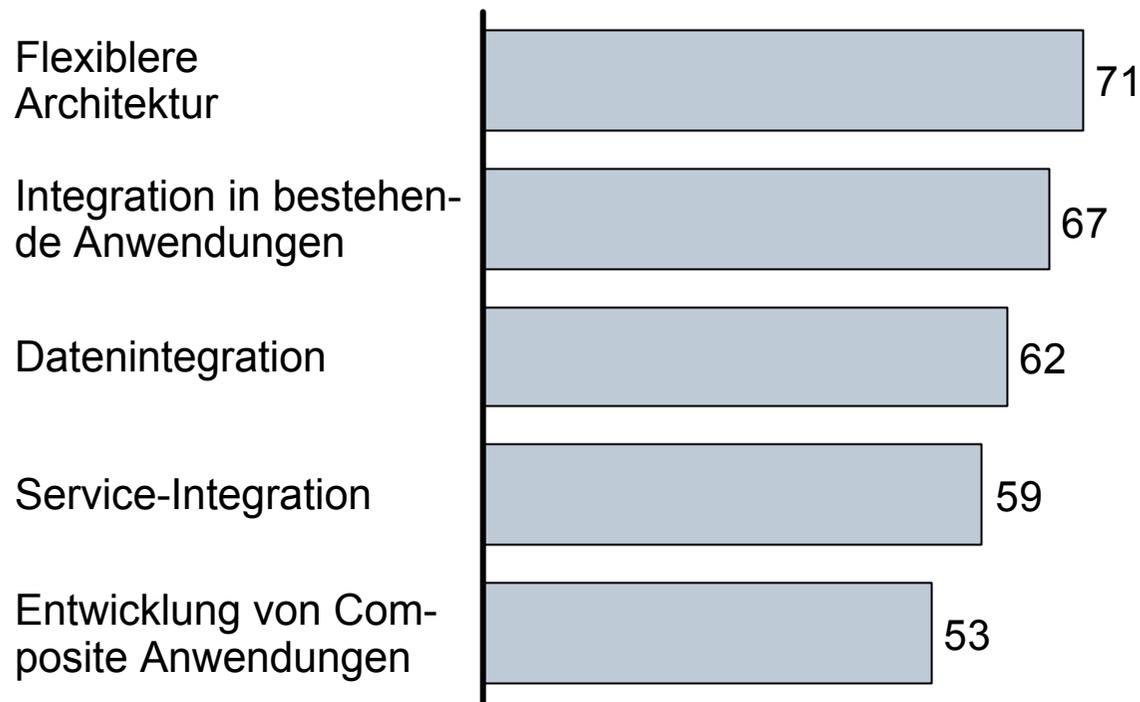
SOA-Laufzeitumgebung

SOA-Beispielanwendung

Gründe dafür, dass die IT zu SOA übergeht

in Prozent

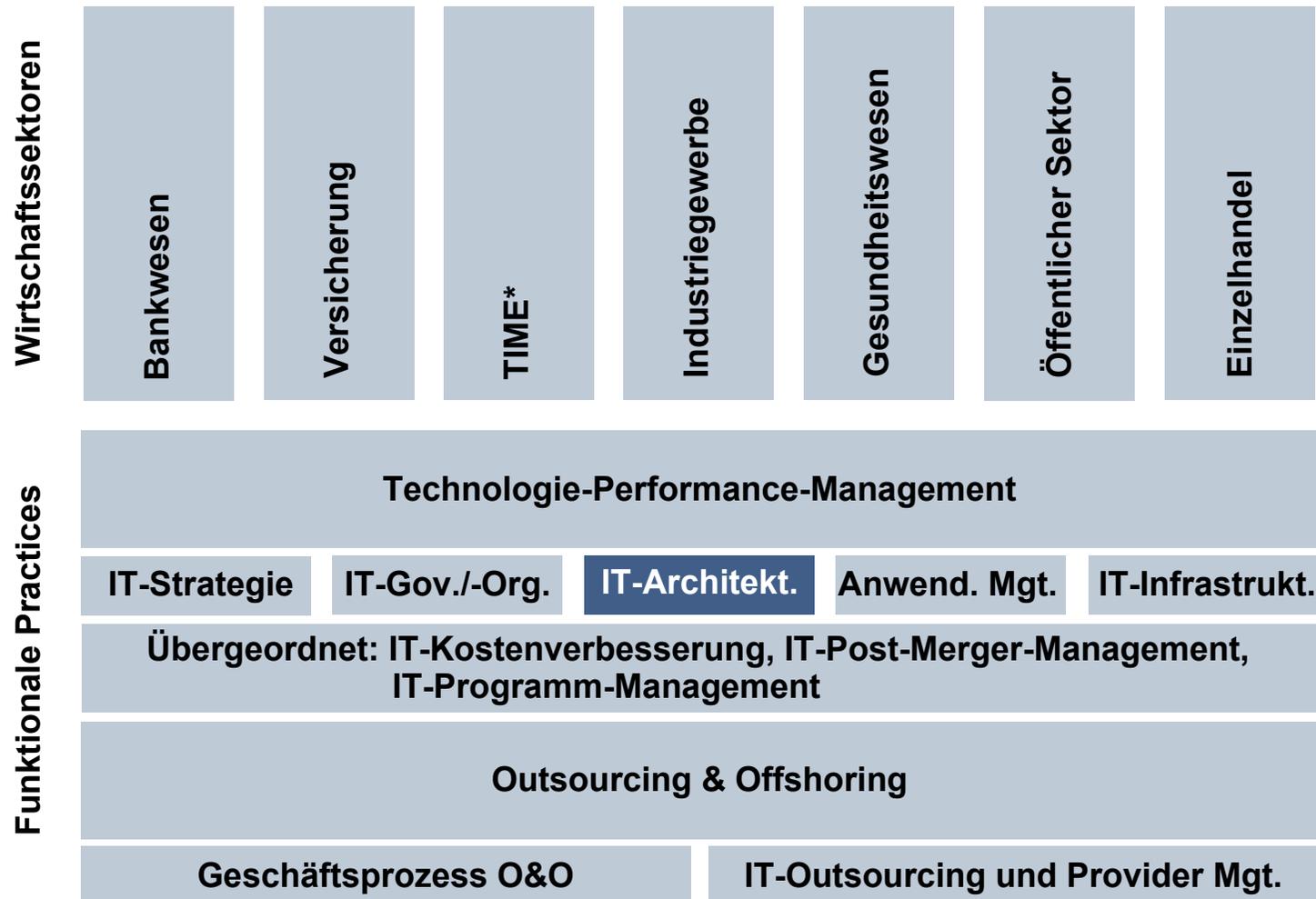
Top 5-IT-Bedürfnisse, denen Unternehmen mit SOA zu entsprechen versuchen



Die Stichprobe setzte sich aus 1.040 IT-Managern in Organisationen von 500 oder mehr zusammen
Quelle: Von IDG Research Services Group im Januar 2006 durchgeführte Studie.

SOA ist ein IT-Architekturthema

BEZIEHUNG ZU ANDEREN IT-DISZIPLINEN



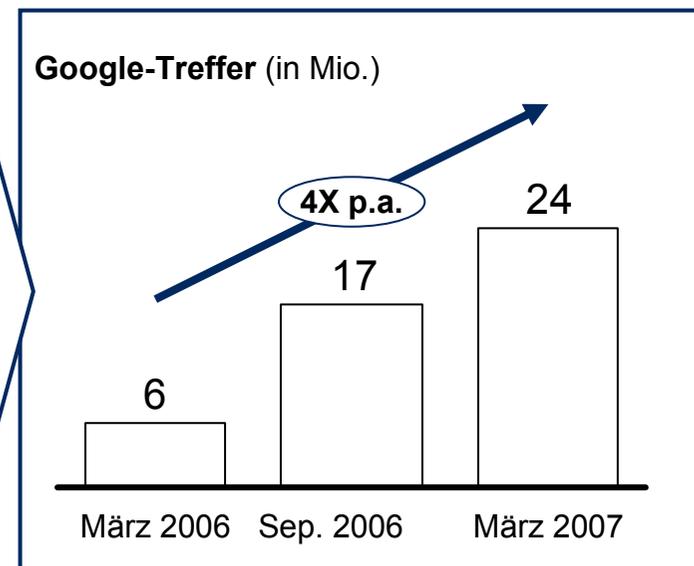
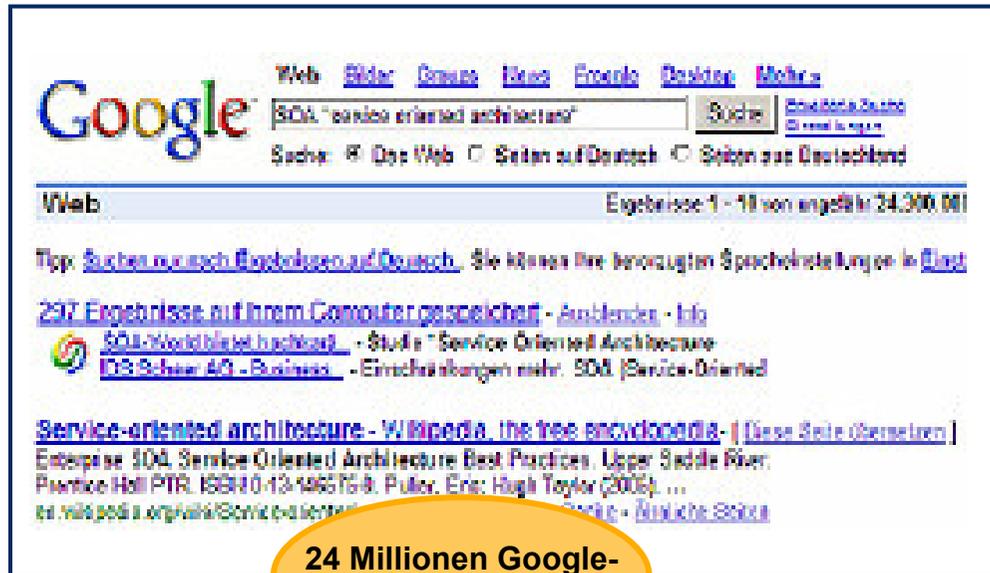
* = Telekommunikation, IT, Medien

Quelle: McKinsey

Das Interesse an SOA nimmt weiterhin schnell zu

Definition

Die **serviceorientierte Architektur (SOA)** ist ein Konzept der Unternehmensarchitektur, bei dem eine Sammlung lose gekoppelter Services als ein System interagiert. Eindeutig definierte Schnittstellen, die die Wiederverwendung von Anwendungen und Daten zulassen und eine gemeinsame Funktionalität bereitstellen, wobei die Entwicklereffizienz gesteigert wird und Integrationen beschleunigt werden.



Im Jahr 1996 erwähnte Gartner die SOA-Idee erstmalig

Im Jahr 2002 iterierte Gartner SOA-Konzepte, was in der Industrie breite Akzeptanz findet

Im Jahr 2007 schätzte Gartner, dass 60% der Unternehmen eine SOA-basierte Lösung einsetzen, und im Jahr 2008 werden es noch mehr sein



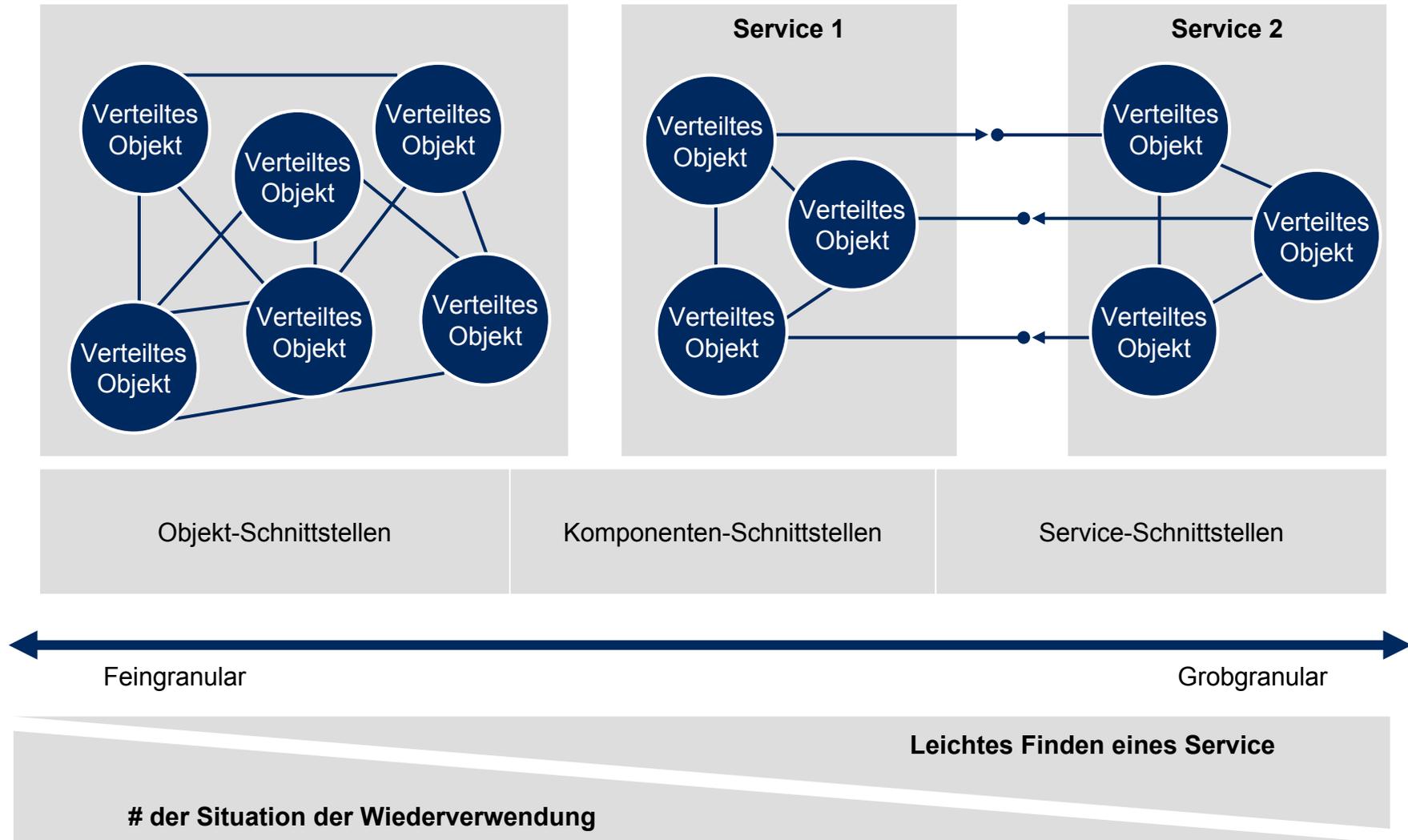
Quelle: Deutsche Post, Presseberichte

SOA-Prinzipien

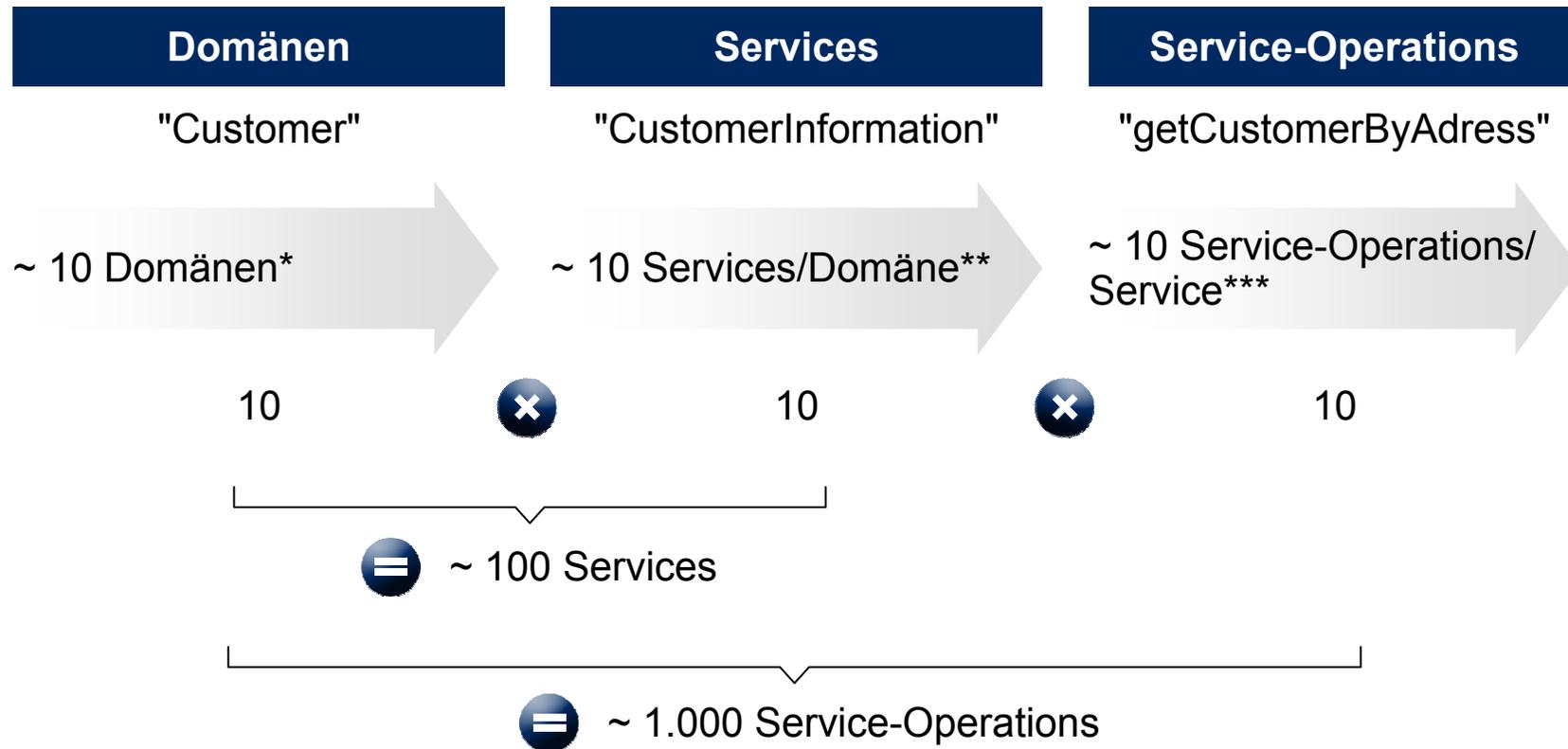
Prinzipien	Beschreibung	
	Designzeit	Laufzeit
1 Wiederverwendbarkeit	<ul style="list-style-type: none"> Services mit hoher Granularität reduzieren Redundanzen bei Daten und Logik 	<ul style="list-style-type: none"> Zustandslose Services sollten bereitgestellt werden, um ein hohes Niveau der Service-Wiederverwendbarkeit zu gewährleisten, da der Geschäftskontext nicht relevant ist
2 Technologie-unabhängigkeit	<ul style="list-style-type: none"> Eine Entkopplung von Service Provider und Service Consumer gewährleistet die Flexibilität von System und Betriebsmodell (z.B. Outsourcing, White Labeling) Services sollten für die Existenz in weit verteilten und heterogenen Rechnerumgebungen entwickelt werden 	<ul style="list-style-type: none"> Eine Vielfalt von Kommunikationsprotokollen sollte möglich sein, um Services ungeachtet der Technologie, die der Consumer oder Provider nutzt, aufzurufen
3 Interoperabilität	<ul style="list-style-type: none"> Die Services basieren auf offenen Standards (z.B. Webservices) 	<ul style="list-style-type: none"> Die Services können zusammenarbeiten, um verbundene Services zur Verfügung zu stellen
4 Lose Kopplung	<ul style="list-style-type: none"> Spezifikationen von Service Consumer und Service Provider werden unabhängig voneinander entwickelt 	<ul style="list-style-type: none"> Die Anbindung an ein spezifisches Kommunikationsprotokoll muss in der Laufzeit stattfinden
5 Autonomie	<ul style="list-style-type: none"> Ein atomarer Aufbau gewährleistet Integrität und Konsistenz von Transaktionen auf Geschäftsebene 	<ul style="list-style-type: none"> Die Services sind eigenständig und selbstbeschreibend mit einer bekannten Schnittstelle
6 Erweiterbarkeit	<ul style="list-style-type: none"> Composite Anwendungen sollten auf bestehende IT aufgebaut werden (z.B. Beibehaltung und Integration von Legacy-Systemen) 	<ul style="list-style-type: none"> Der Einsatz von Composite Anwendungen zur Erweiterung und individuellen Anpassung verhindert Einschränkungen durch Paketanwendungen

Quelle: McKinsey

Zur Maximierung der Wiederverwendbarkeit muss die Granularität der Services optimal ausgewogen sein



Erste Orientierung/Daumenregel für Service-Granularität – "Zehnerpotenz"



* Bereich ~ 8 - 20

** Bereich ~ 4 - 30

*** Bereich ~ 5 - 50

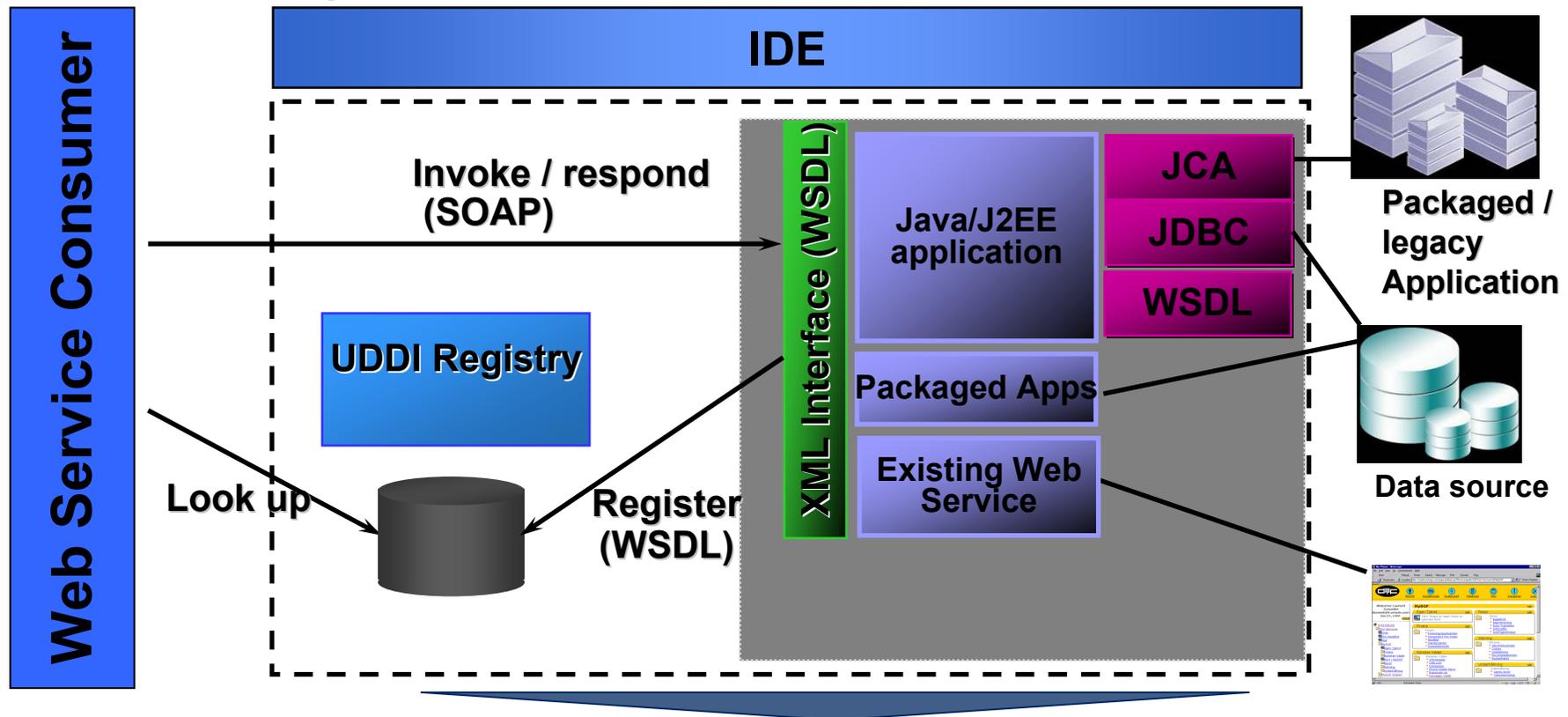
Quelle: Deutsche Post, SOA Days 2007 Business Conference, 29. - 30. März 2007

Erreichung von Interoperabilität durch Standards

HISTORIE DER TECHNISCHEN STANDARDS ZUR SERVICEORIENTIERUNG

Architektur	Payload	Adressierung	Protokoll	Interface
Corba	Common Data Representation (CDR)	Interoperable Object Reference (IOR)	IIOP (Binary)	Interface Definition Language (IDL)
DCOM	Network Data Representation (NDR)	OBJREF	ORPC	(MS)IDL
.NET Remoting	XML, Binary	URL	HTTP, TCP	.NET Interfaces
RMI	Serialisierbare Java Objekte	URL	Java Remote Method Protocol (JRMP)	Java Interfaces
Web Services	XML	URL	SOAP (Text-based)	WSDL

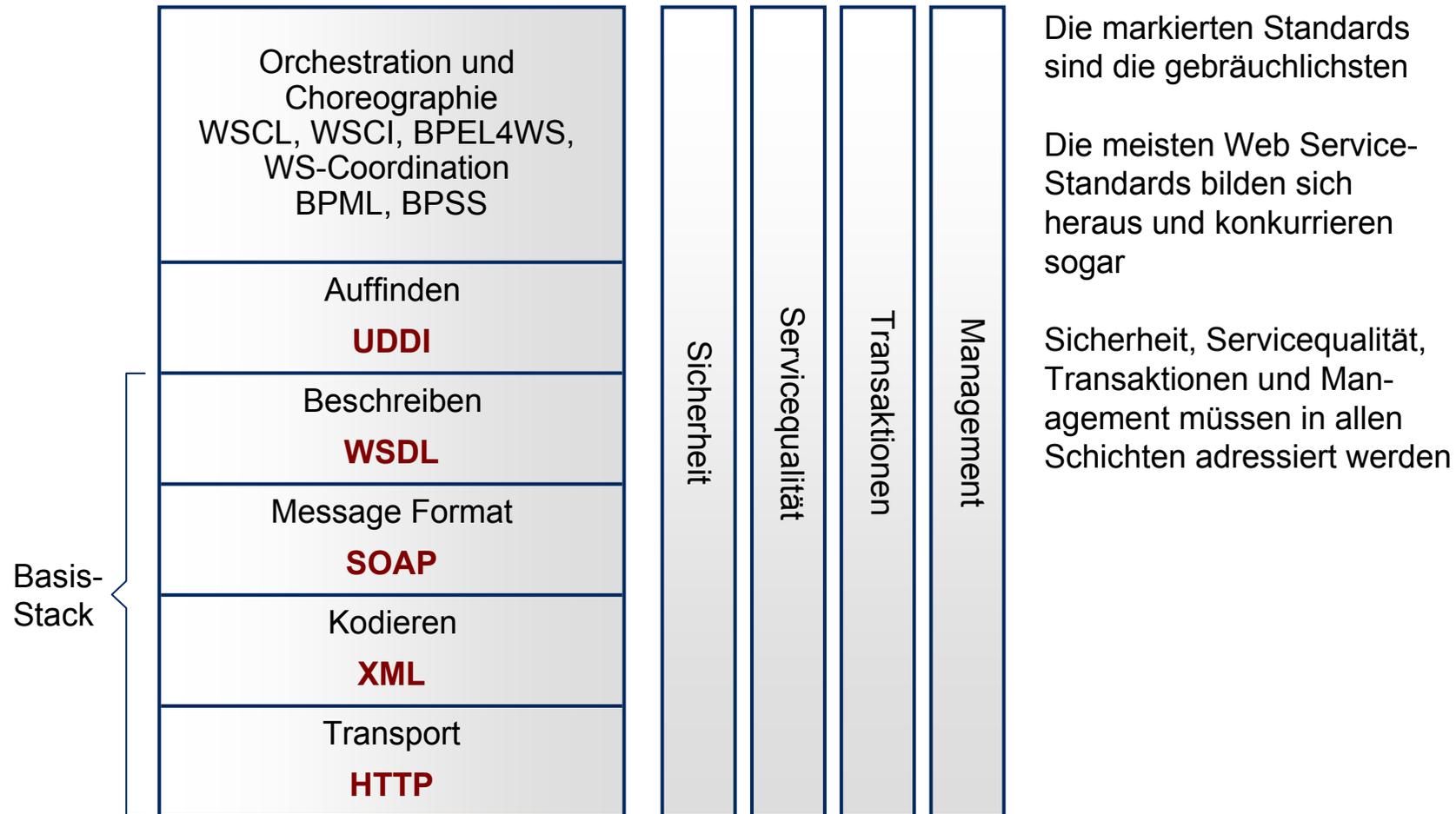
'Web Services' ist ein Mechanismus zur Implementierung eines SOA-basierten Systems



- Service-Schnittstellen werden mit Web Services Description Language (WSDL) beschrieben
- Daten werden mit SOAP über HTTP übermittelt
- UDDI wird optional als Directory Service eingesetzt

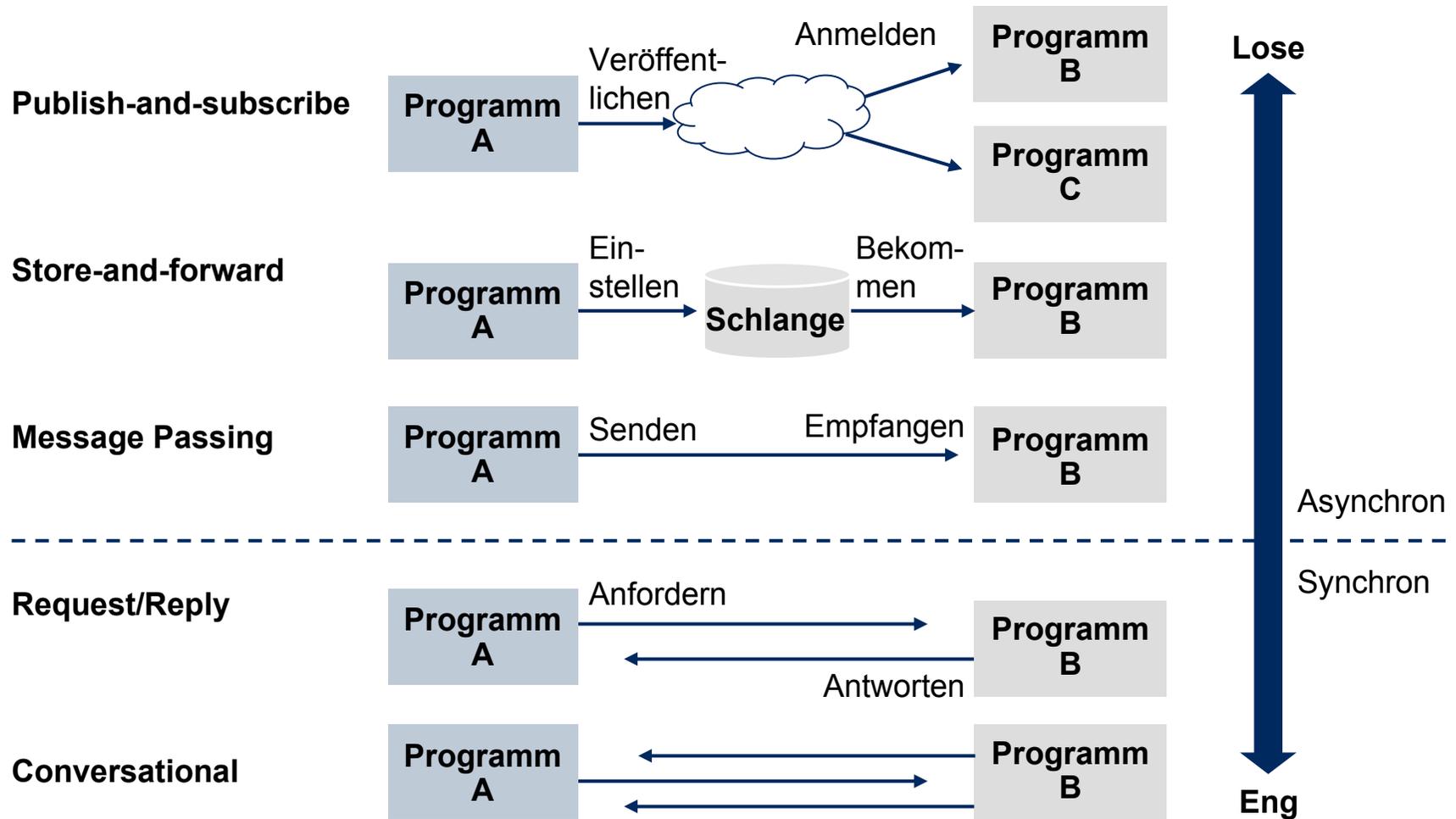
Da er der gebräuchlichste Mechanismus ist, wird er häufig mit SOA gleichgesetzt

Web Service Protokoll-Stack

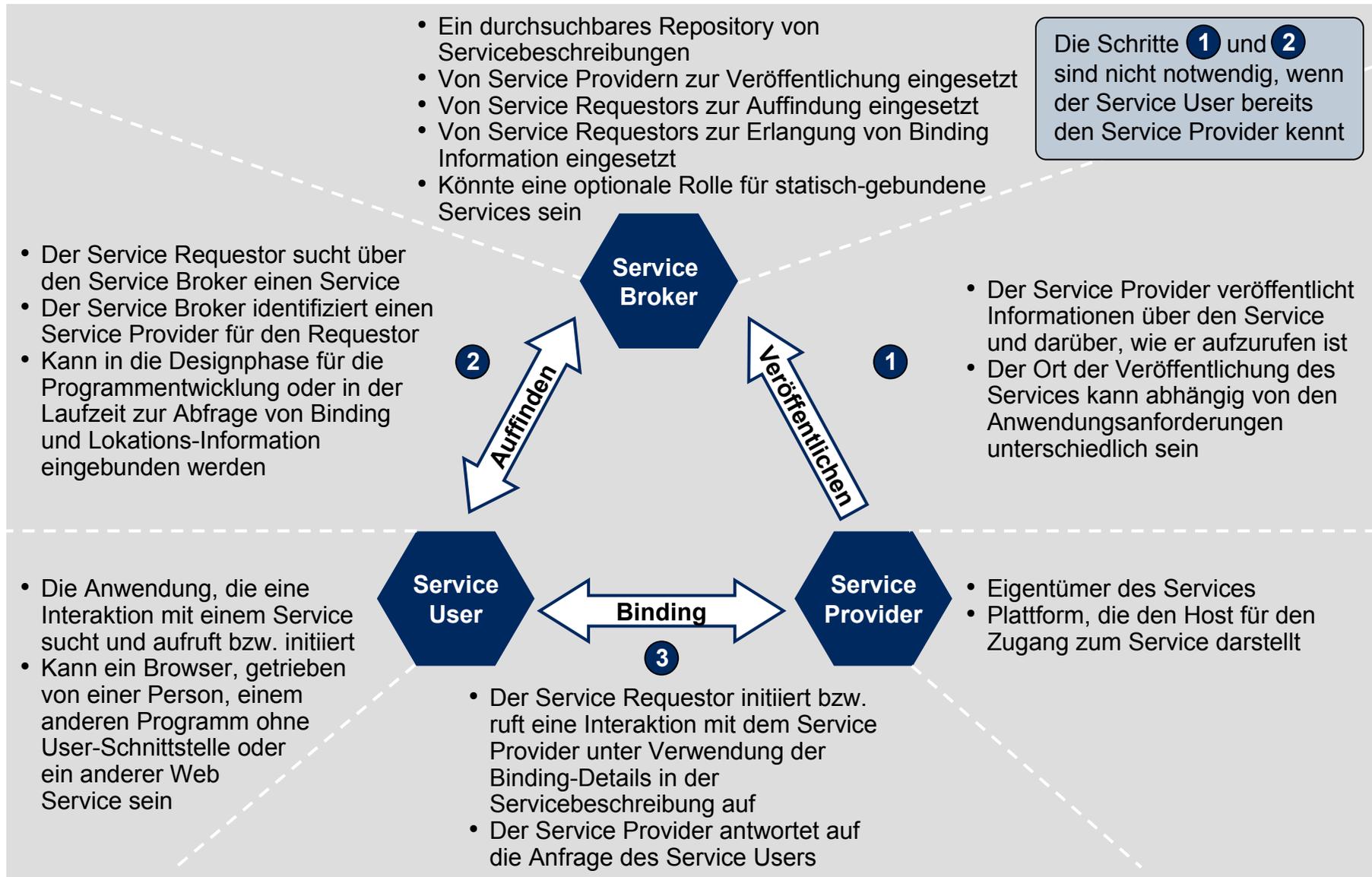


Quelle: Adaptiert aus "XML and Web Services Unleashed" SAMS Publishing

Wesentliche Paradigmen der Serviceorientierung sind die lose Kopplung und Asynchronität



Erreichung von loser Kopplung durch Indirektion



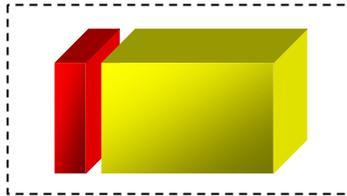
Quelle: Webservices.org

Service Typen

 WSDL interface
  Service

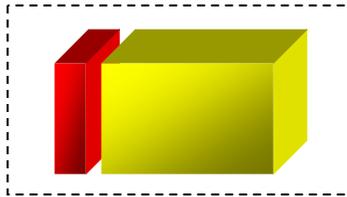
Business

Service



Technical

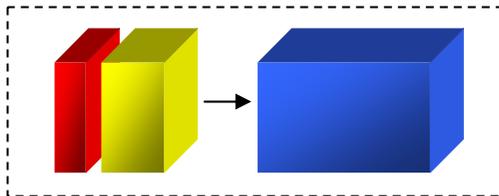
Service



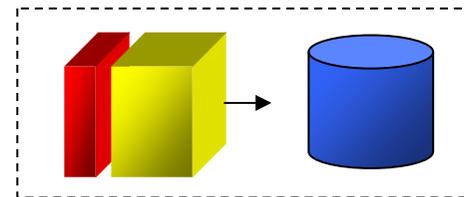
Service operation/
element



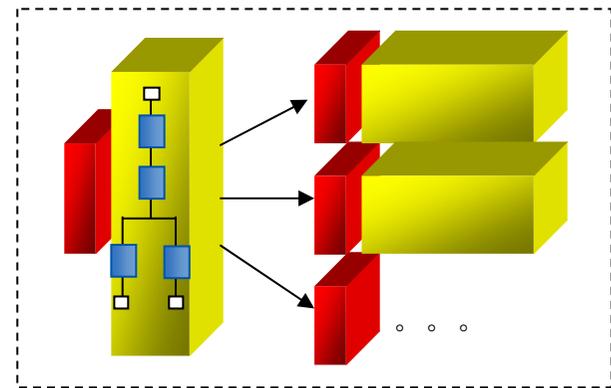
Packaged /
legacy
Application



Data
service

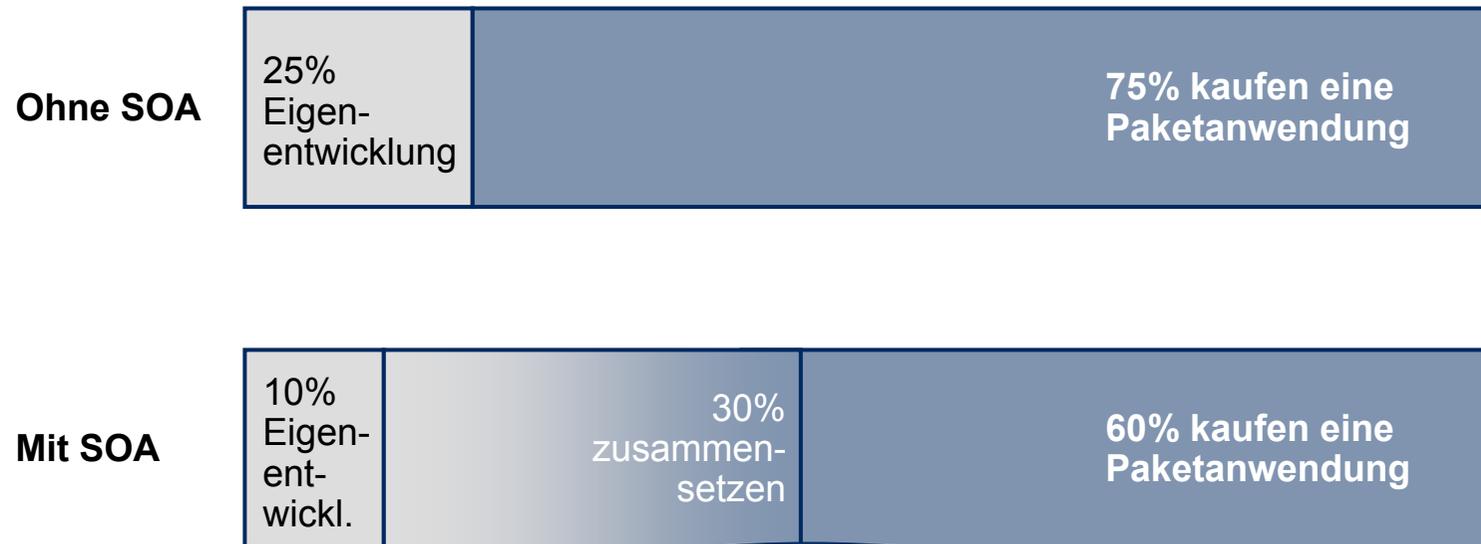


Composite
service



Wie spielt hier eine BPM hinein?

Paradigmaverschiebung der Anwendungsentwicklung beim Einsatz von SOA



Was sind andere Anzeichen für eine Verschiebung?

Inhalt



Eine typische Unternehmensstruktur

Probleme einer IT-Abteilung

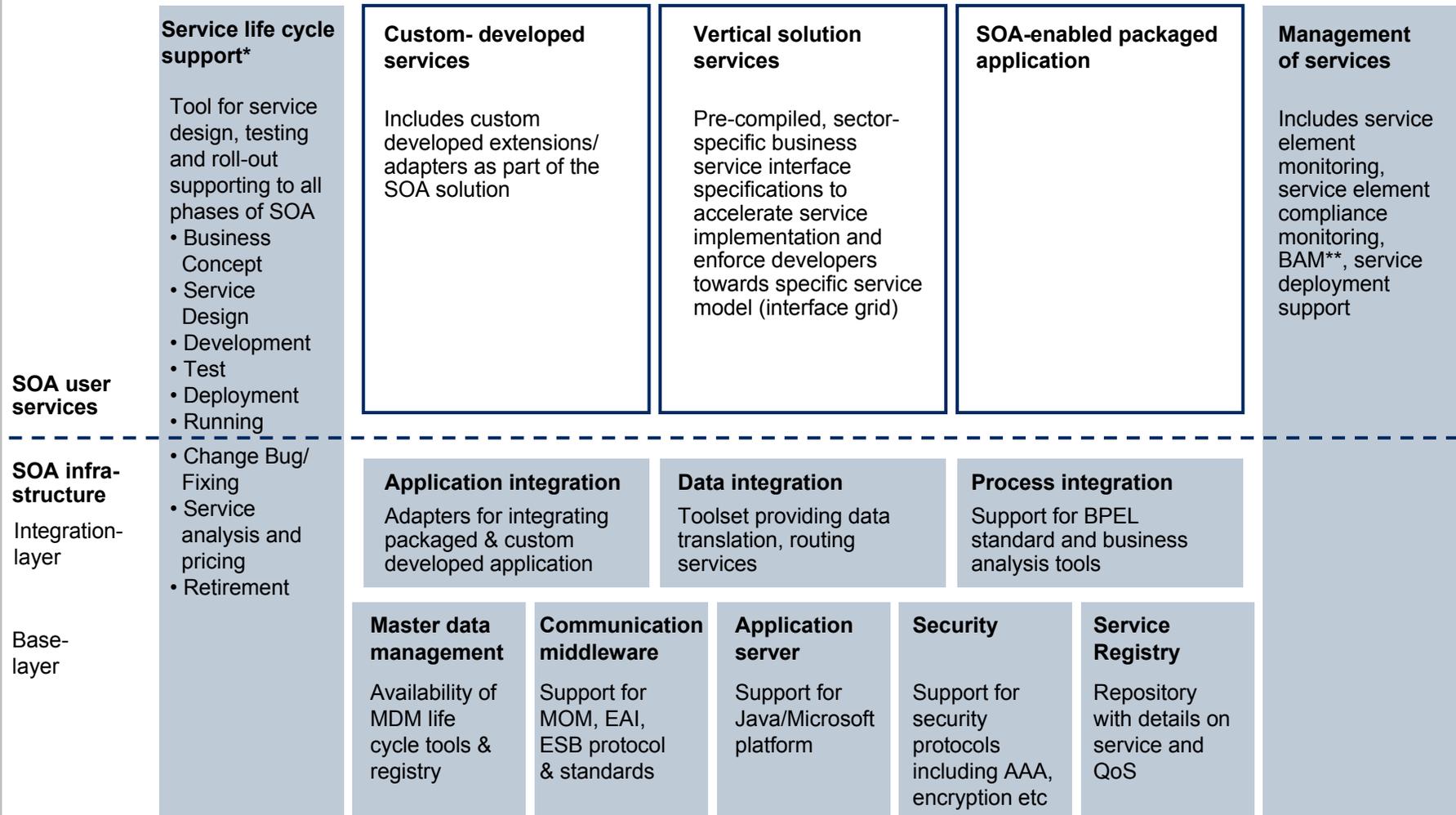
SOA-Grundlagen

SOA-Laufzeitumgebung

SOA-Beispielanwendung

Enterprise services run on a SOA infrastructure and are supported by life cycle and management tools

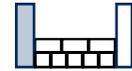
SOA products
Enterprise services



* E.g. Process design, service design, service repository, service testing, service provisioning

** Business activity monitoring

Source: McKinsey

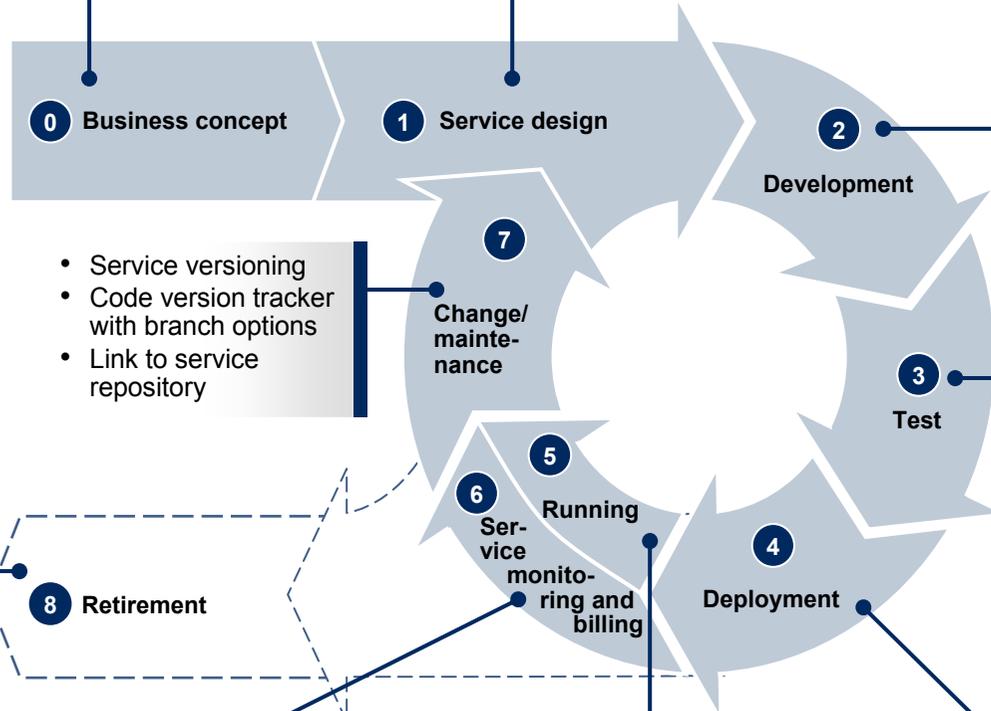


Service life cycle

- Define service cycle for business concept
- Business defines functional and non-functional requirements
- Define billing scheme and service pricing

- Business-driven service modeling* (policy, QoS)
- Reuse of existing services
- Graphical standard user interface built on service repository
- Combine services into new composite applications (like BPM)

- IDE-integrated automated creation of service framework based on service design (code generator and patterns)
- Syntax (programming language) and semantic code checks (programming/architecture guidelines, naming conventions)
- Programmers interface to explore and link services



- Service versioning
- Code version tracker with branch options
- Link to service repository

- Central test interface
- Automated unit tests
- Simulated integration tests (environment simulation)
- Performance tests (end-to-end)
- Flexibly configurable test cases
- Consolidate and analyze upcoming test errors
- Support for batch and scheduled execution times
- Generate human-readable test reports

- Phased service decommissioning
- Services disabling and discarding
- Link to service repository

- Logging of utilization/consumption of resources
- Exception handling built on exception repository
- Live view of "service network"-relations and interdependencies
- Monitoring and managing SLAs

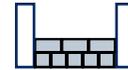
- Distributed execution
- Capacity management and load balancing
- Policy management
- Central service registry
- Invocation without knowledge of location or technology of target

- Deployment wizard with version control
- Hot deployment (live)
- Activate business rules
- Register and activate service policies built on service registry

* Input/output parameter and service specification

Source: McKinsey

SOA can be built based on existing integration technologies with some compromises



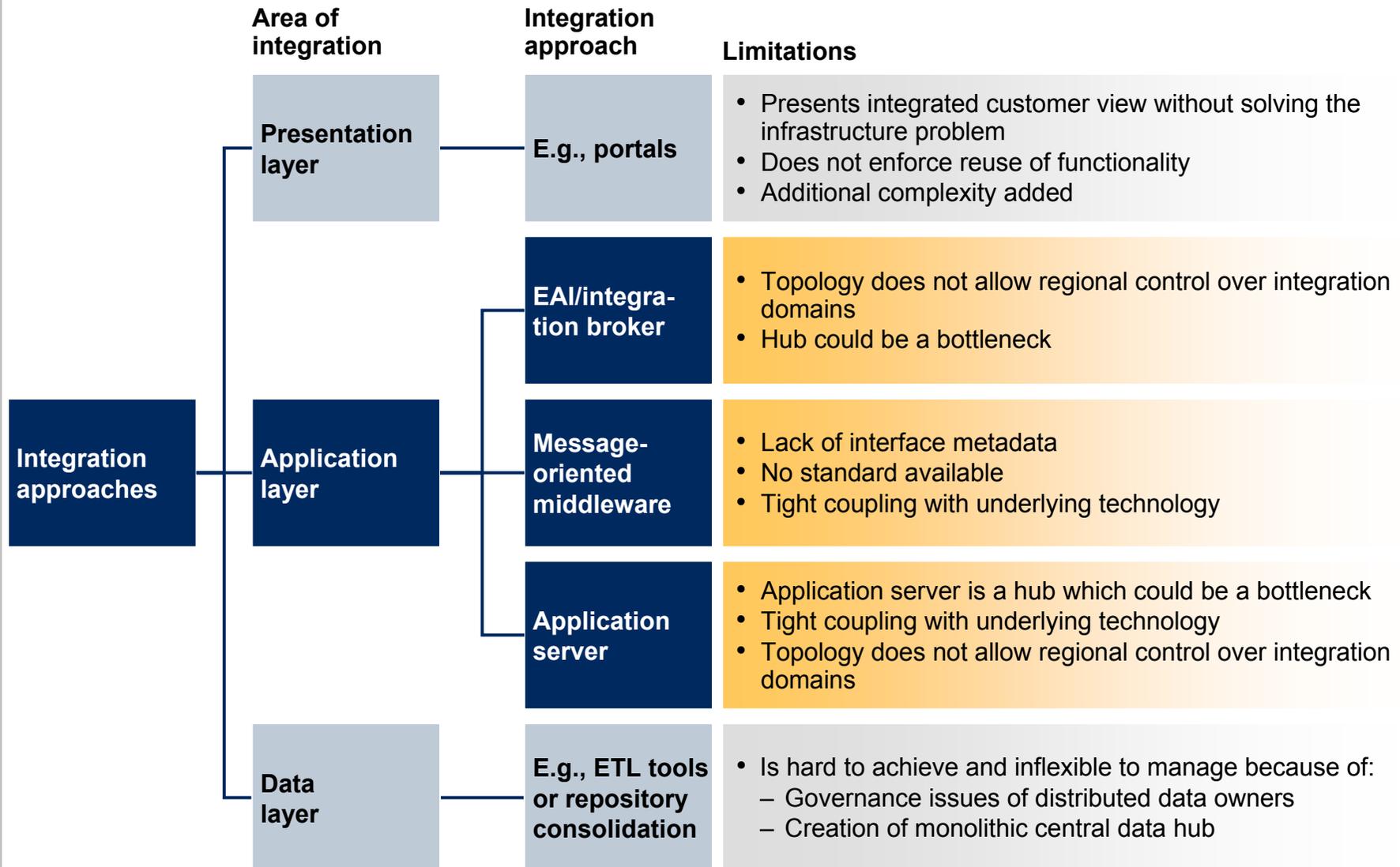
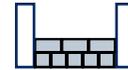
Existing integration technologies

	Integration broker	Message-oriented middleware	Application server
Integration capabilities to be used in a SOA	<ul style="list-style-type: none"> Integration/business process logic can be managed Wide range of technical and business adapters available 	<ul style="list-style-type: none"> Very mature and scalable technology Message-driven technology provides loose coupling of applications 	<ul style="list-style-type: none"> Includes MOM as well as ORB technologies
Potential compromises in a SOA	<ul style="list-style-type: none"> Hub does not allow distributed execution and could be a bottleneck Lack of interoperability between EAI tools 	<ul style="list-style-type: none"> Lack of metadata (for service repository) No standards available, therefore tight coupling to underlying technology 	<ul style="list-style-type: none"> Tight coupling to underlying technology in case of distributed execution Limited capabilities for managing integration/business logic

Source: McKinsey

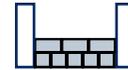
Current integration technologies have limitations – critical for SOA

 Focus of presentation



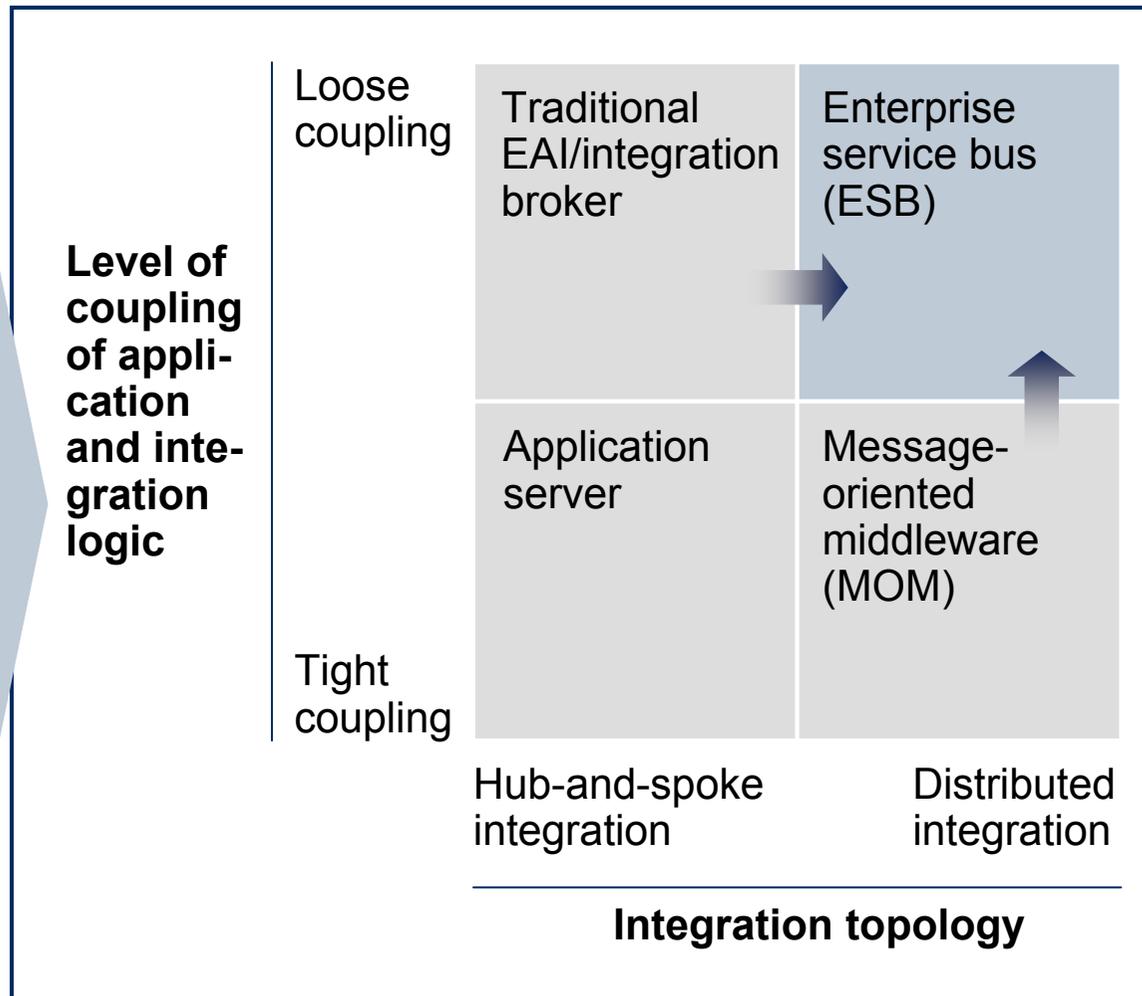
Source: McKinsey

A general architecture view highlights the movement toward a new technology – the enterprise service bus

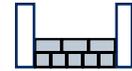


Integration technologies

- SOA requirements for integration
 - Distributed integration (a bus topology) in contrast to a hub-and-spoke integration
 - Loose coupling/separation of application and integration logic in contrast to intertwined/tight coupling
- Today, available technologies do not match these requirements and need to be extended to a new technology, enterprise service bus

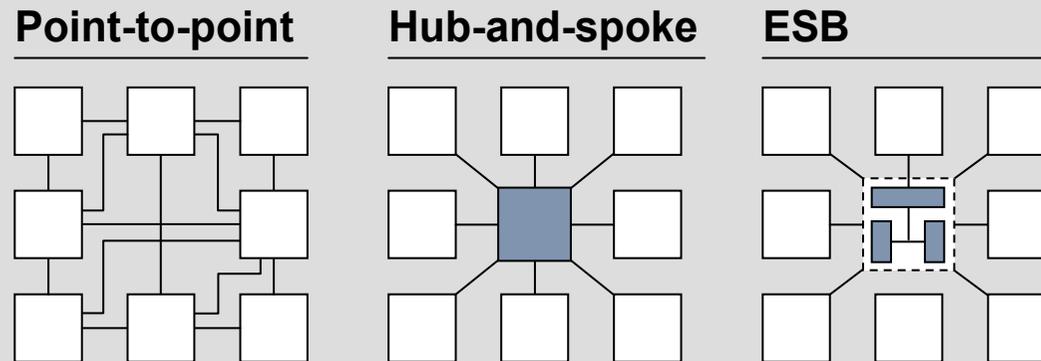


Source: McKinsey



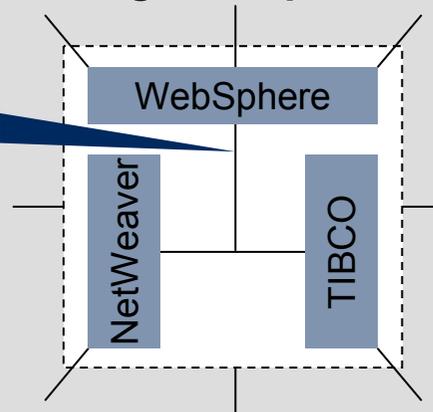
ESB is both an integration approach and a technology

ESB is a federated integration architecture approach beside point-to-point and hub-and-spoke

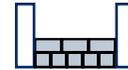


ESB is a set of software technology standards for the interoperability of different integration products

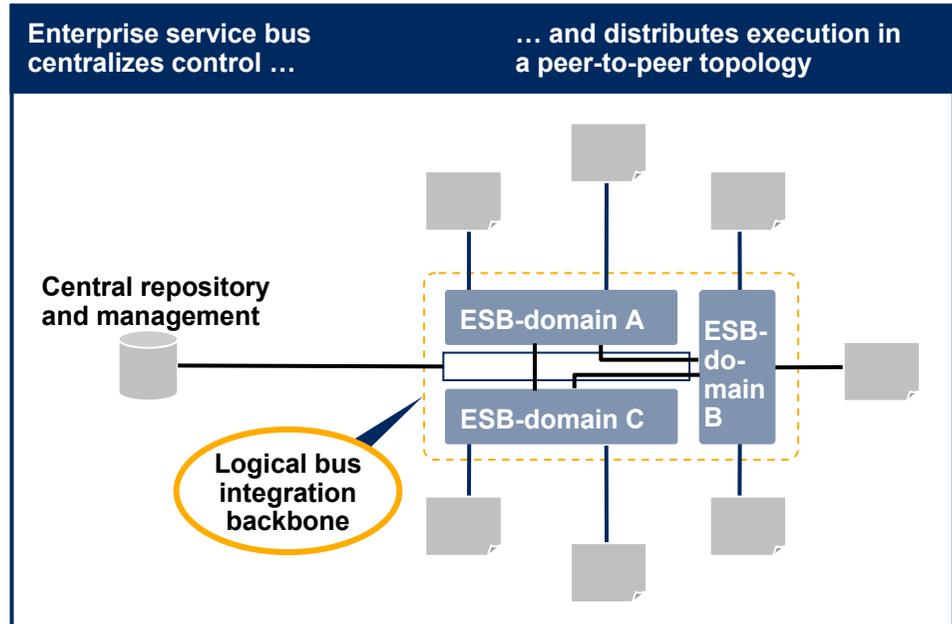
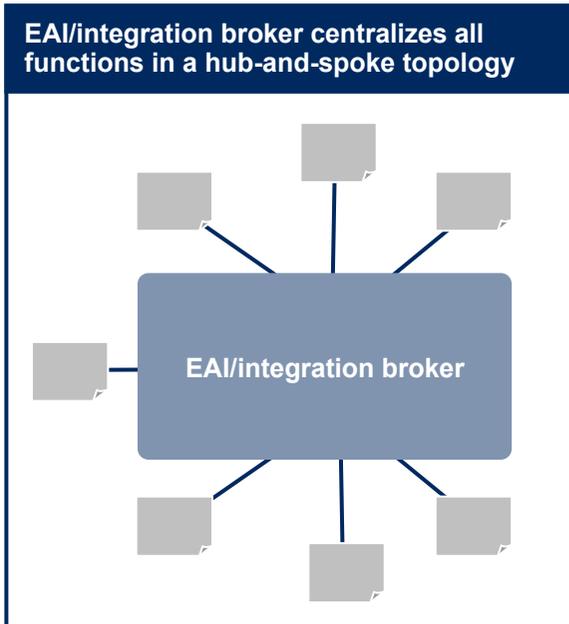
Integration products must comply with standards to build an ESB



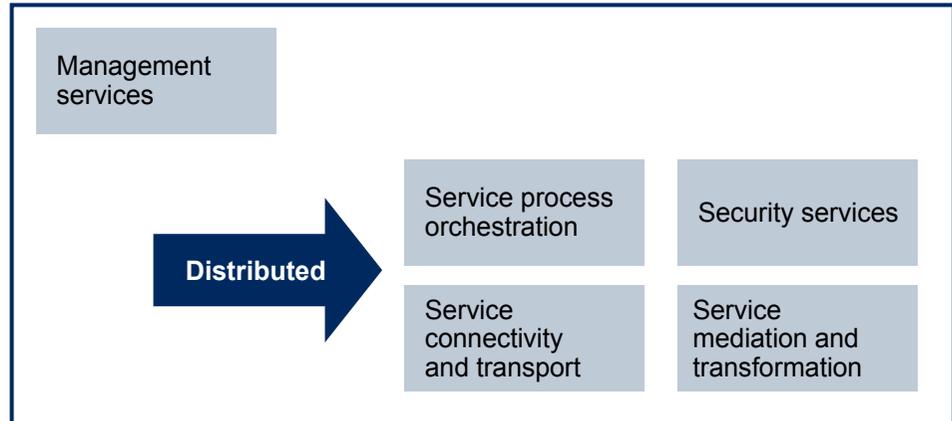
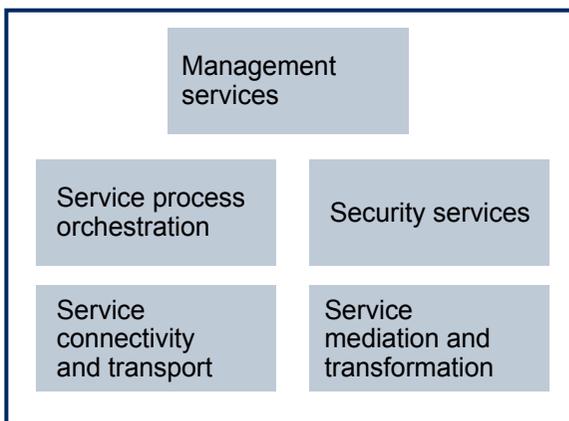
EAI centralizes all functionalities, whereas ESB overcomes some shortcomings by distributing execution



Architecture topology

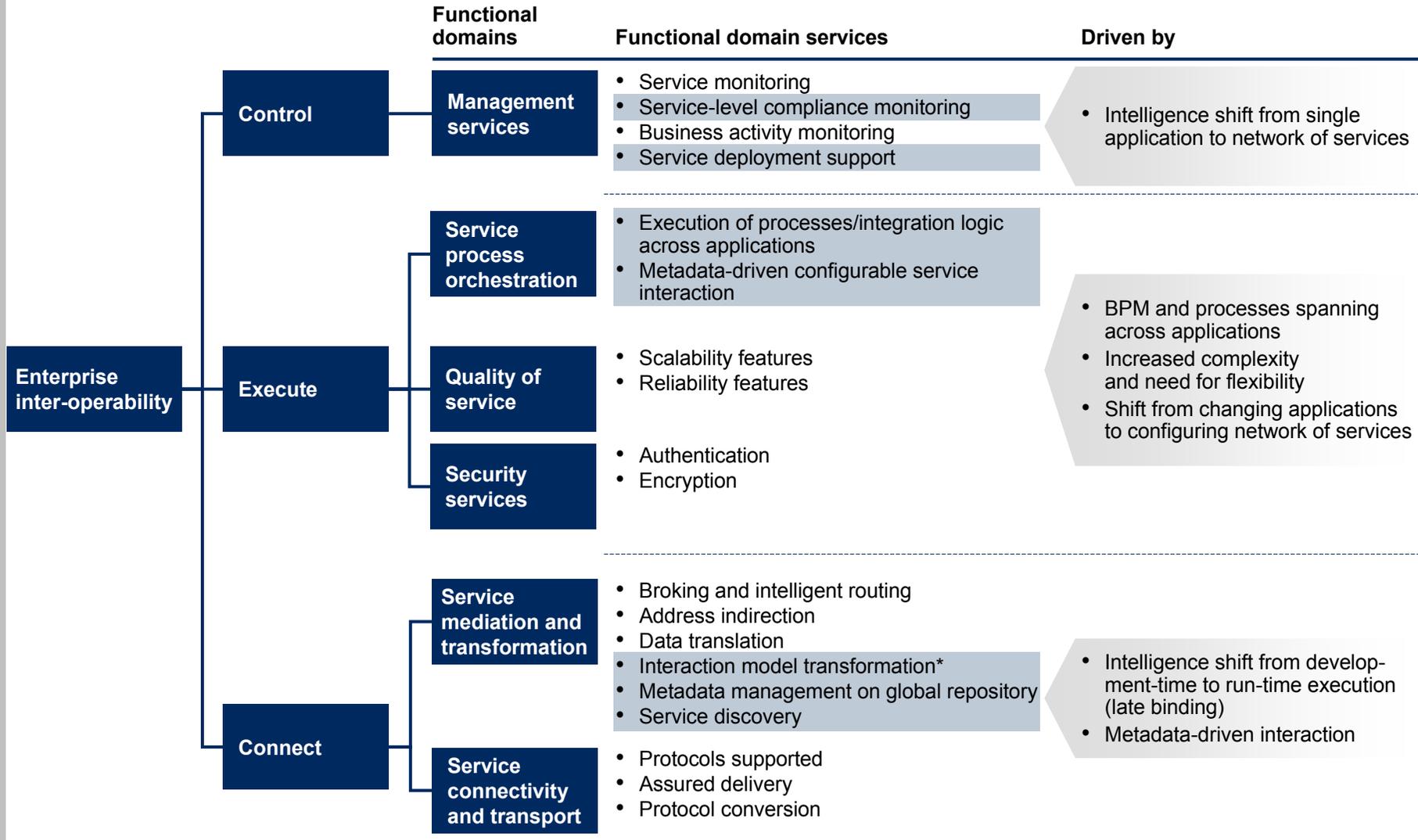


Functional domains



Source: McKinsey

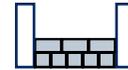
An enterprise service bus has 6 key capability areas



* E.g., transform synchronous SQL query to asynchronous message

Source: CBDI report "Modernizing Application Integration with SOA," 2005; Microsoft on ESB

Evaluation template for quality of integration products in the market

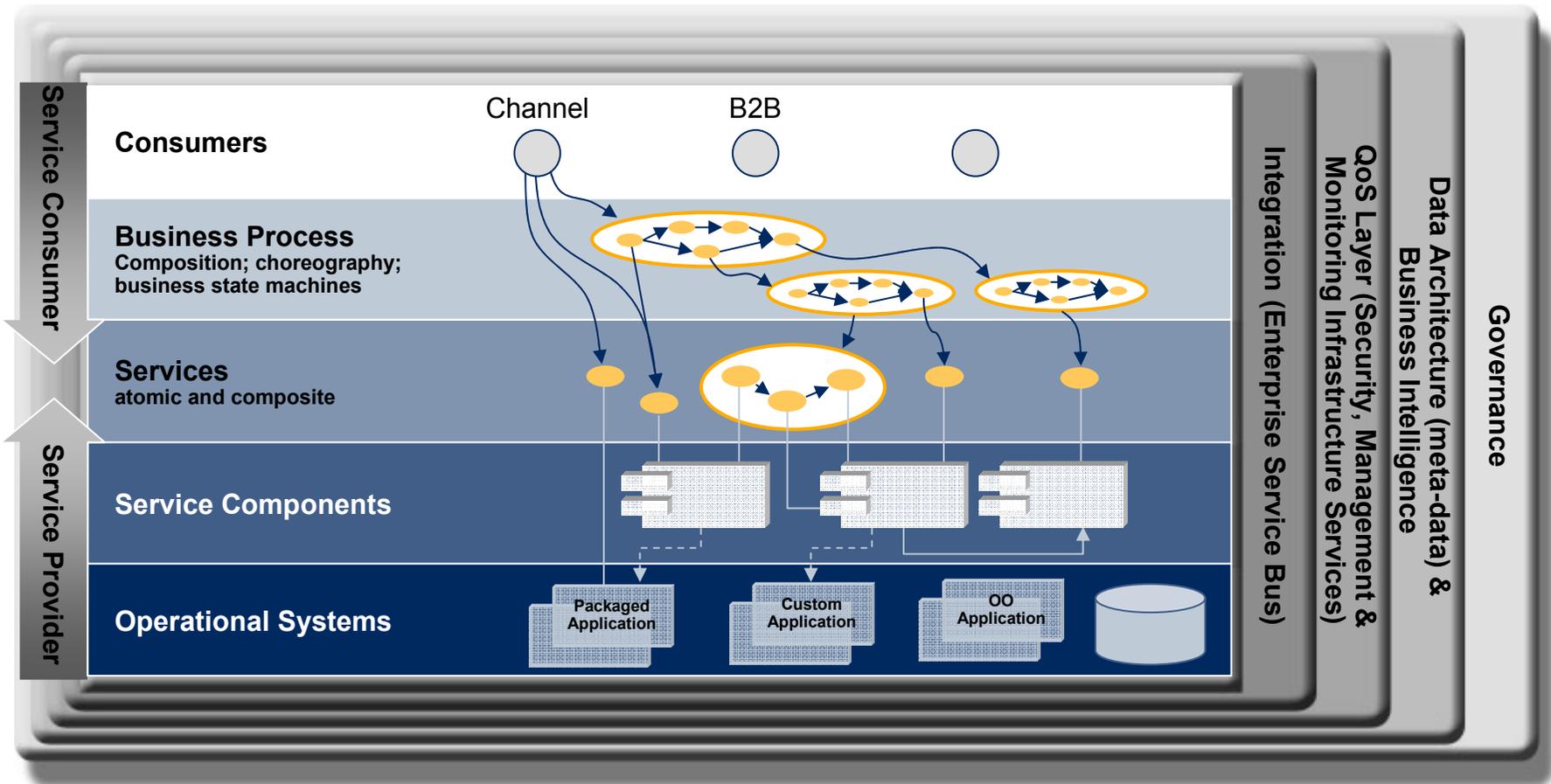
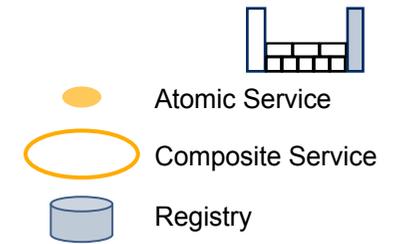


Functional domains	Functional domain services	Relevant protocols and standards
1 Management services	1.1 Business activity and service-level compliance monitoring 1.2 Service deployment support	<ul style="list-style-type: none"> • WSDM • WS-Policy, WS-RF
2 Service process orchestration	2.1 Execution of processes/integration logic across applications 2.2 Metadata-driven configurable service interaction	<ul style="list-style-type: none"> • BPEL, JCA • WS-Coordination
3 Quality of service	3.1 Service liability	
4 Security services	4.1 Authentication and encryption	<ul style="list-style-type: none"> • WS-Policy • WS-Security
5 Service mediation and transformation	5.1 Broking, intelligent routing, and address indirection 5.2 Data translation and interaction model transformation* 5.3 Metadata management on global repository	<ul style="list-style-type: none"> • XSLT • WS-Addressing
6 Service connectivity and transport	6.1 Assured delivery and protocol conversion	<ul style="list-style-type: none"> • SOAP • JMS • WS-Reliable Messaging • WS-Coordination

* For example, transform synchronous SQL query to asynchronous message

Source: CBDI report "Modernizing Application Integration with SOA," 2005; Microsoft on ESB

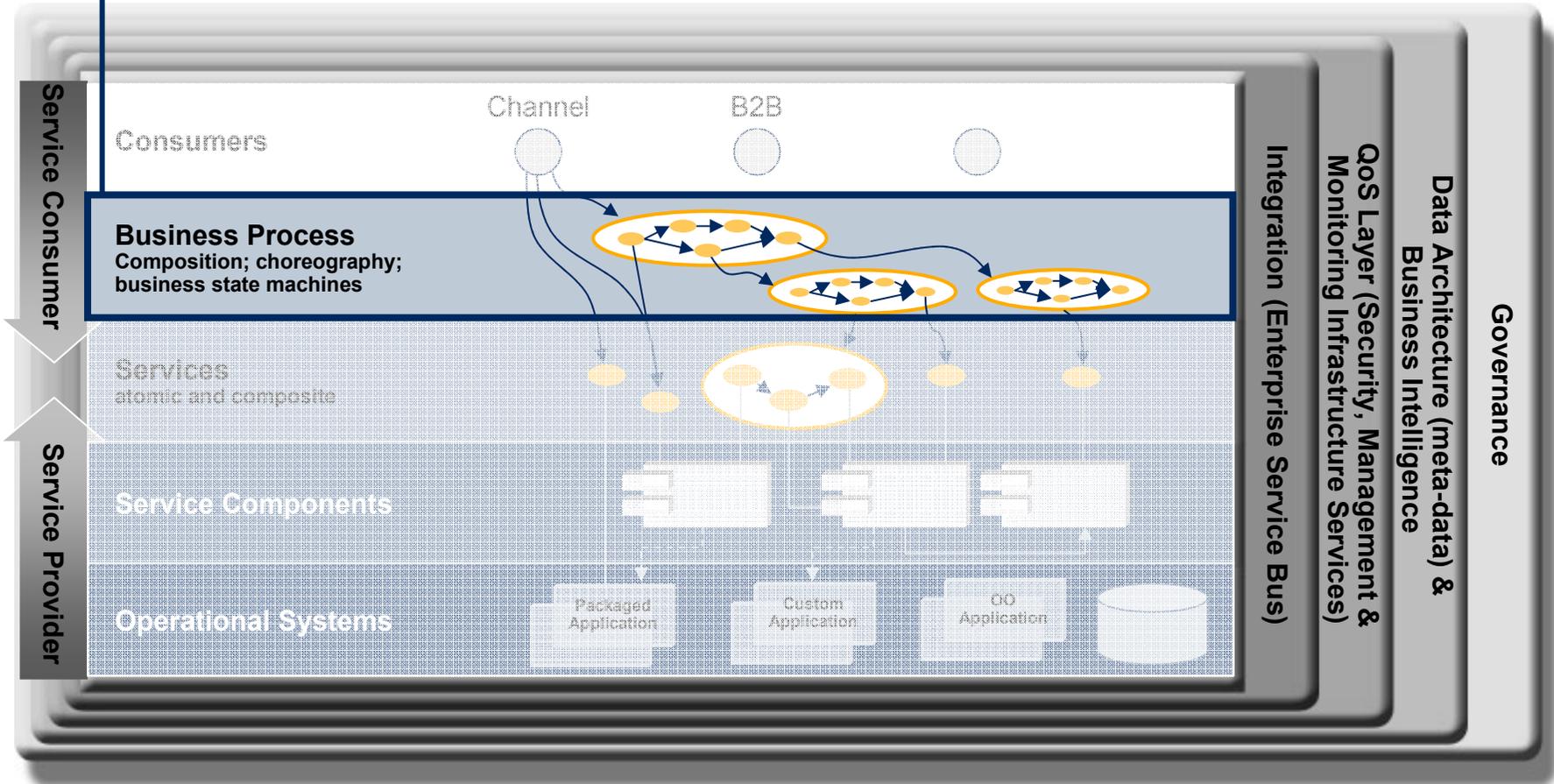
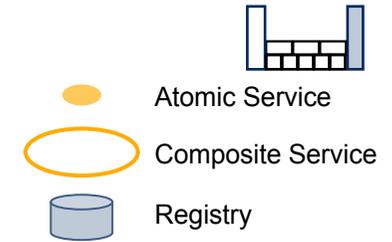
Requirements for SOA Management (1/4)



Quelle: McKinsey

Requirements for SOA Management (2/4)

Business Process Monitoring:
Monitor state of business processes

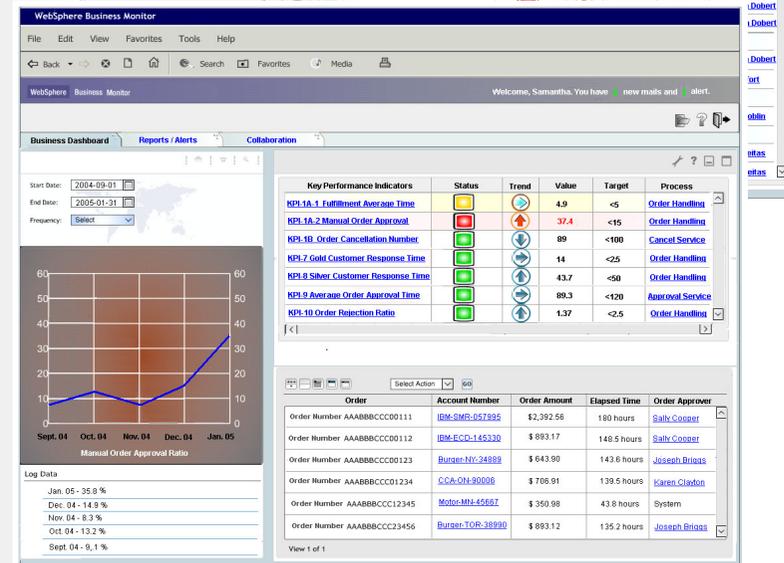


Quelle: McKinsey



Business Process Monitoring

- Report on business performance measured against targets (scorecard)
 - Share growth and new product revenue
- Track business process flow
 - Status of particular insurance claim
 - Bottlenecks due to human tasks
- Monitor business process metrics
 - Duration, cost, branch ratios
- Business Analysis through aggregation and multidimensional reporting
 - Total monthly revenue by customer



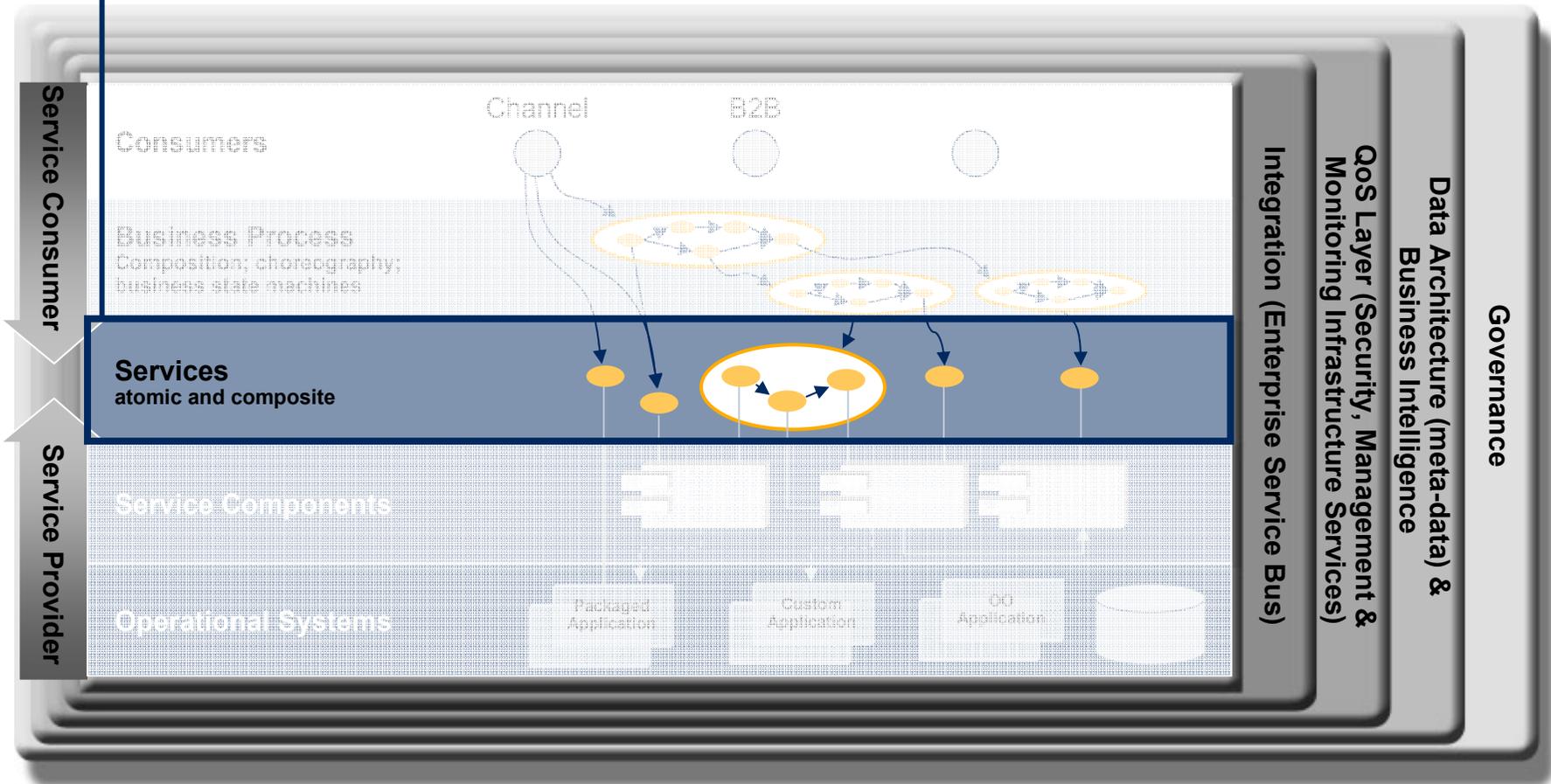
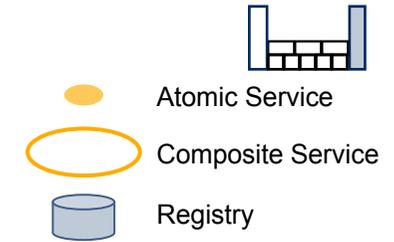
Was ist der Unterschied zu BAM?

Quelle: McKinsey

Requirements for SOA Management (3/4)

Services Management:

Discover, monitor, secure and manage services to meet SLAs



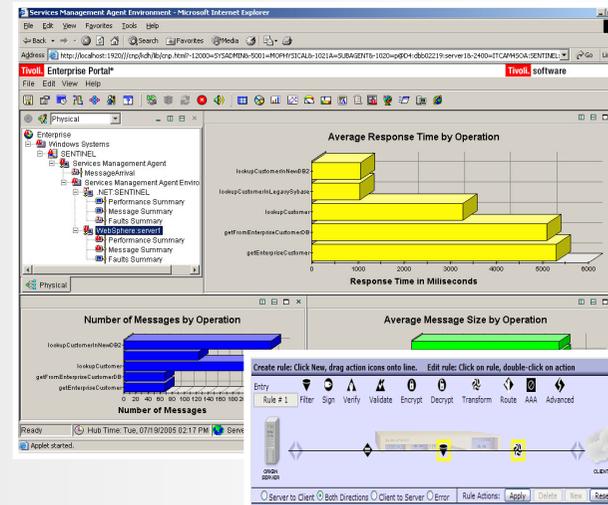
Quelle: McKinsey



Services Management Lifecycle Support for Web Services

To ensure service levels conform to agreed upon specifications, you need:

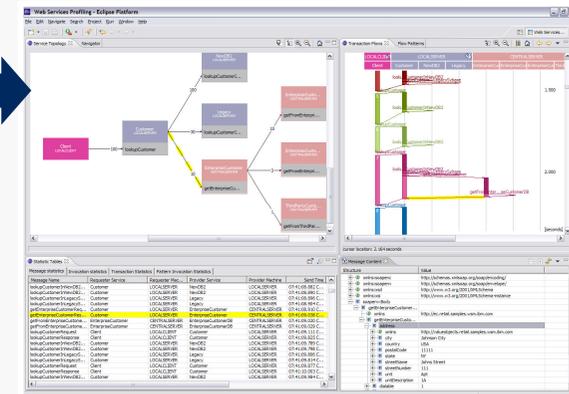
- Views and analysis of Web service interactions for IT Operations to quickly identify source of errors, and take corrective action through situations, workflow and mediations
- Detailed views of operational SOAP/XML message content, flow patterns and topology for Web services experts and support teams
- Highly performing and flexible enforcement points



IT Operations
"Don't give me another console"



Web Services Expert
"Show me the service details!"

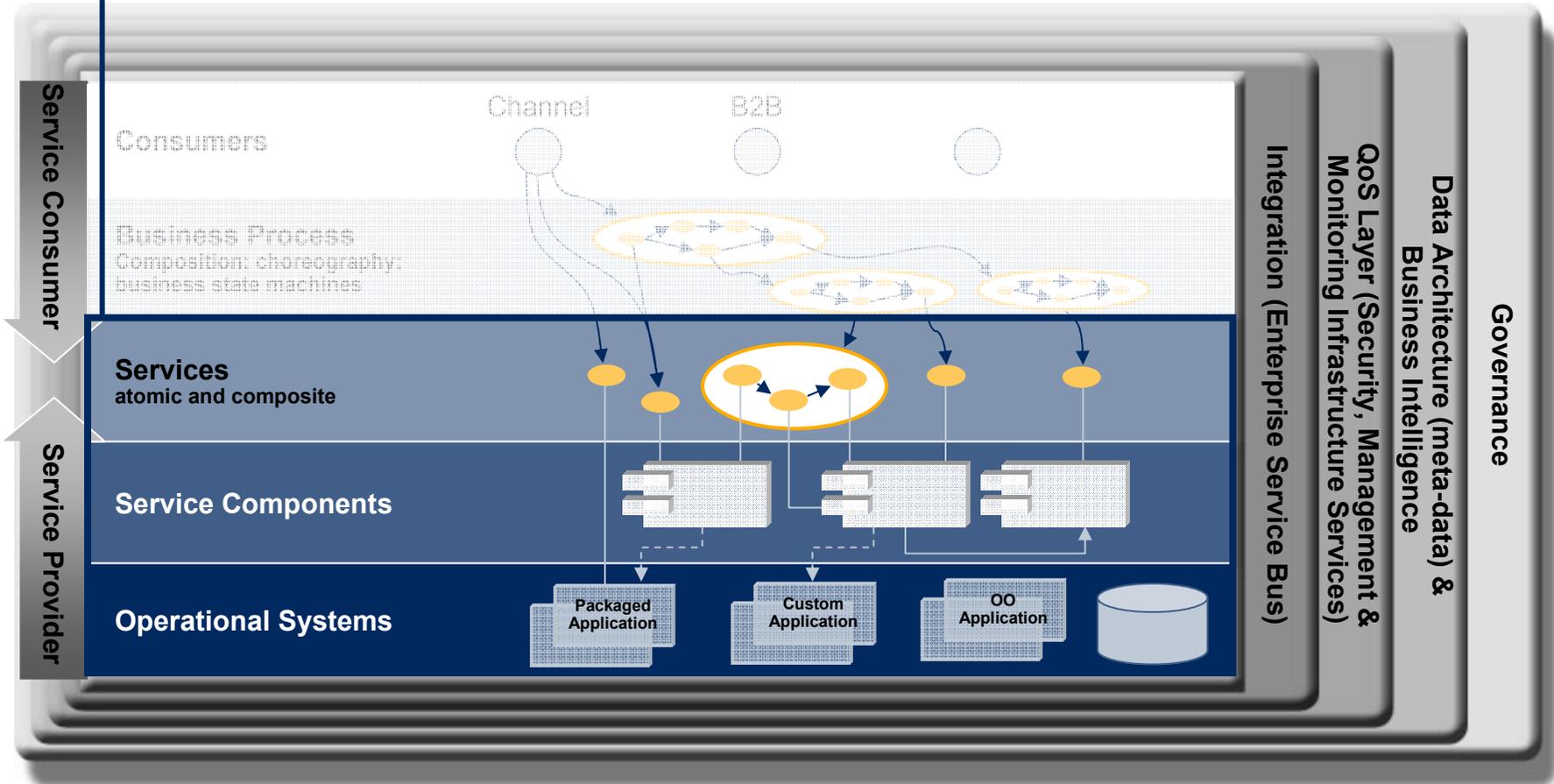
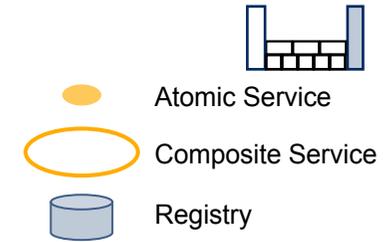


Wie und warum ist Versionierung von Services anders als traditionelle Versionierung?

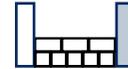
Requirements for SOA Management (4/4)

Manage Transaction Performance:

Measure transaction response times to discover bottlenecks, isolate infrastructure



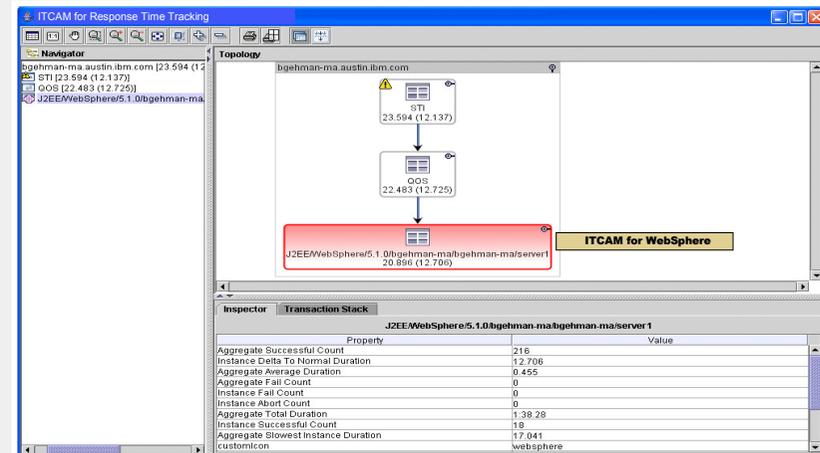
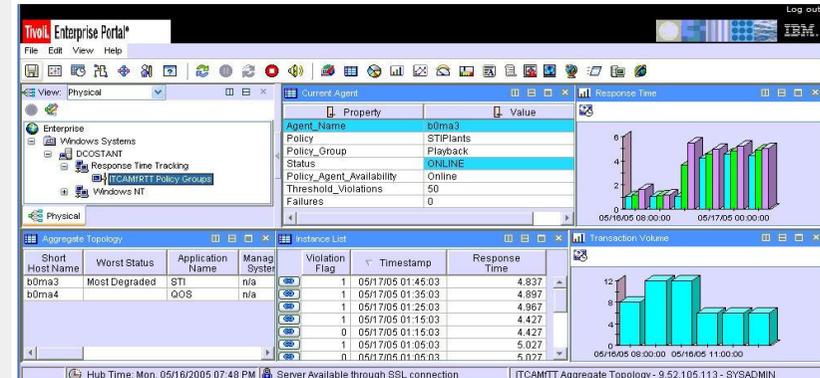
Quelle: McKinsey



Manage Transaction Performance

Provide Key Response Time Metrics Across Platforms

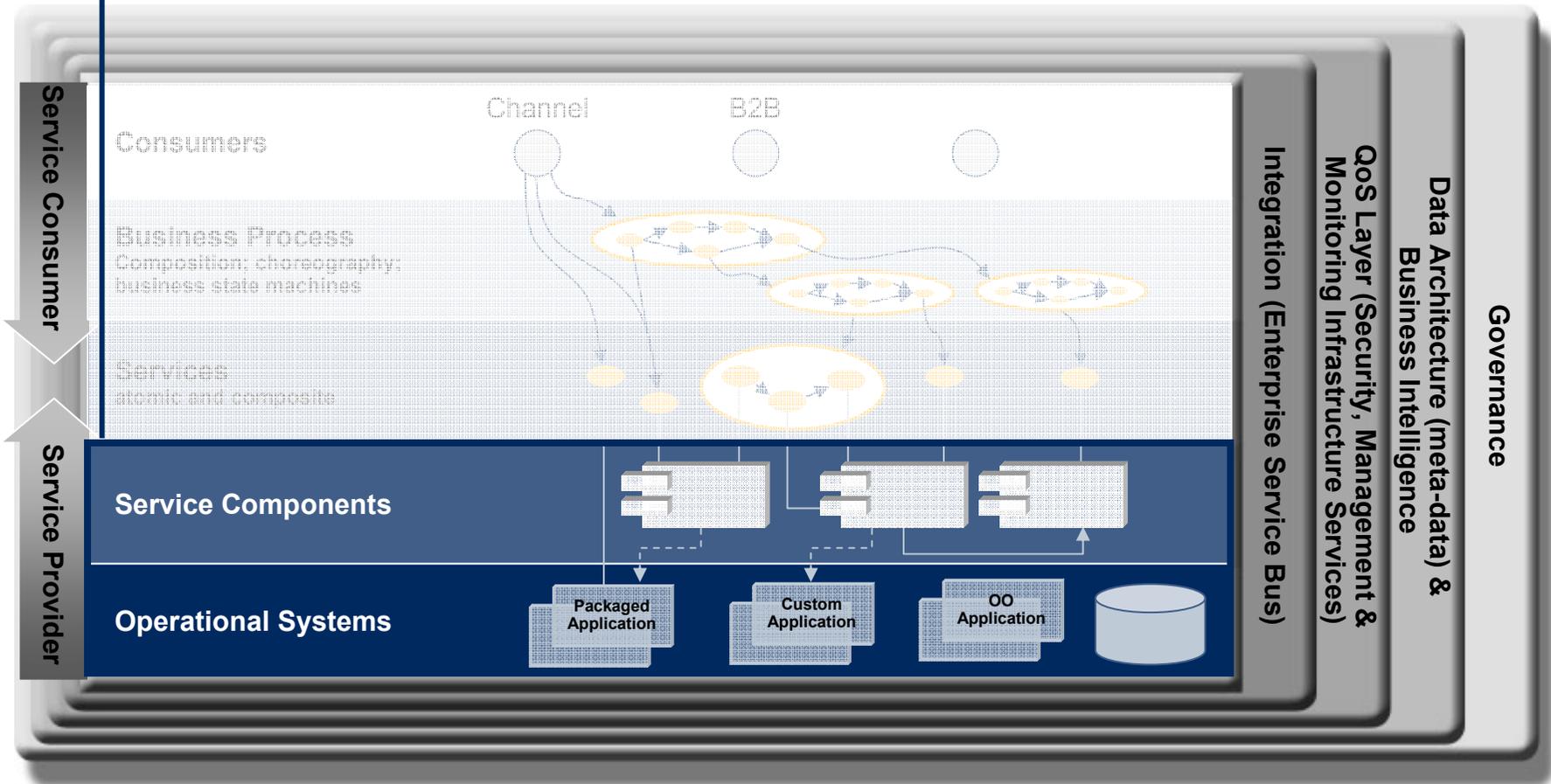
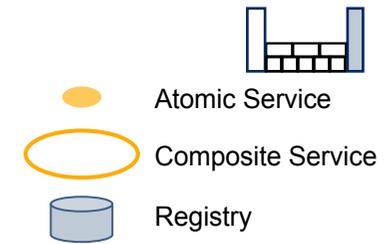
- Customers find it very difficult to identify and isolate performance bottlenecks in composite applications that span technology and platform boundaries
- Need to provide performance instrumentation that is lightweight and can be dynamically configured to identify problems before customers call
- ARM-based instrumentation is the industry standard that can be leveraged to isolate the problem



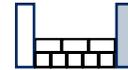
Quelle: McKinsey

Requirements for SOA Management

Manage the Infrastructure:
Deep dives into specific resources



Quelle: McKinsey

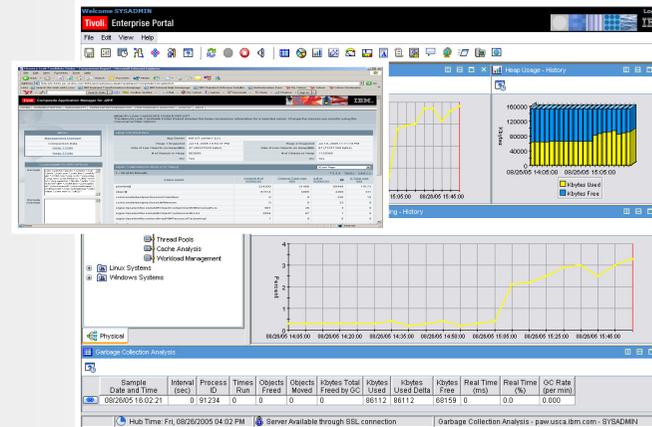


Manage Supporting Middleware

Comprehensive Deep-dive Monitoring

- Identify and quickly correct applications that are down or performing slowly
- Need to provide comprehensive in-flight transaction display that includes the name of the hung class/method
- Introspect messaging and brokering subsystem for real-time metrics and historical data analysis
- This can significantly improve the performance and availability of J2EE applications by reducing problem identification and resolution time

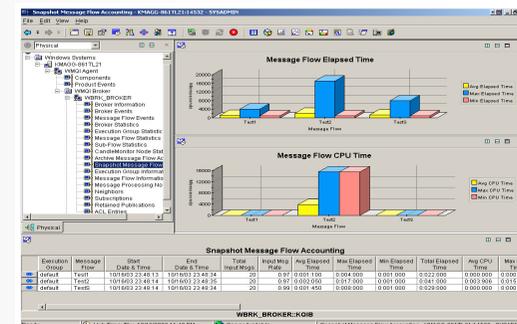
J2EE Applications



IT Operations

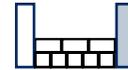


Subject Matter Expert



Messaging Infrastructure

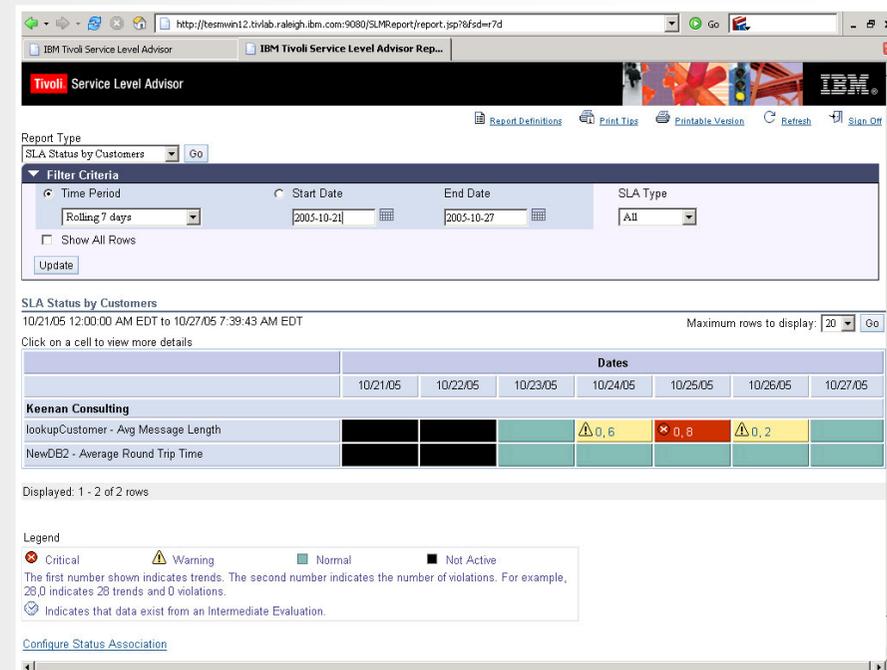
Quelle: McKinsey



Manage Service Levels

Enterprise-wide Reporting on Service Level Compliance

- Need to define reports on a customer basis
- Report on compliance to service levels and service level agreements
- Represent all aspects of service level performance, from business process layer to IT infrastructure
- Perform trending analysis to predict outages

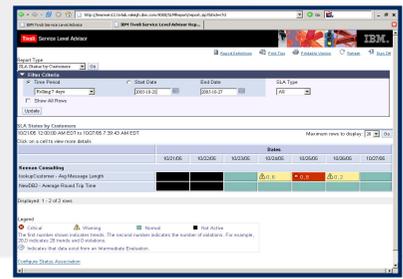
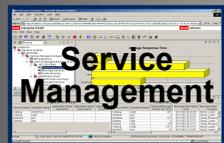
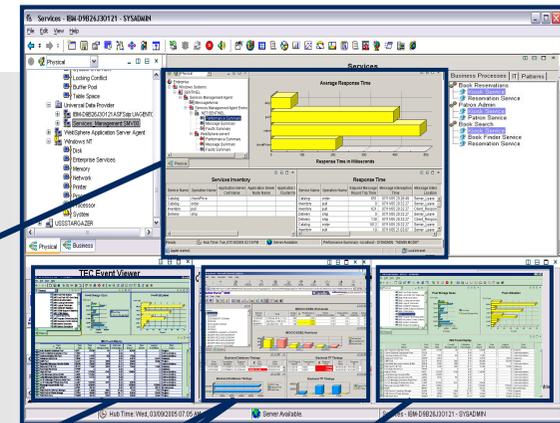


Quelle: McKinsey



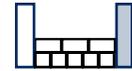
A Comprehensive view of SOA Resources

Integrated Console



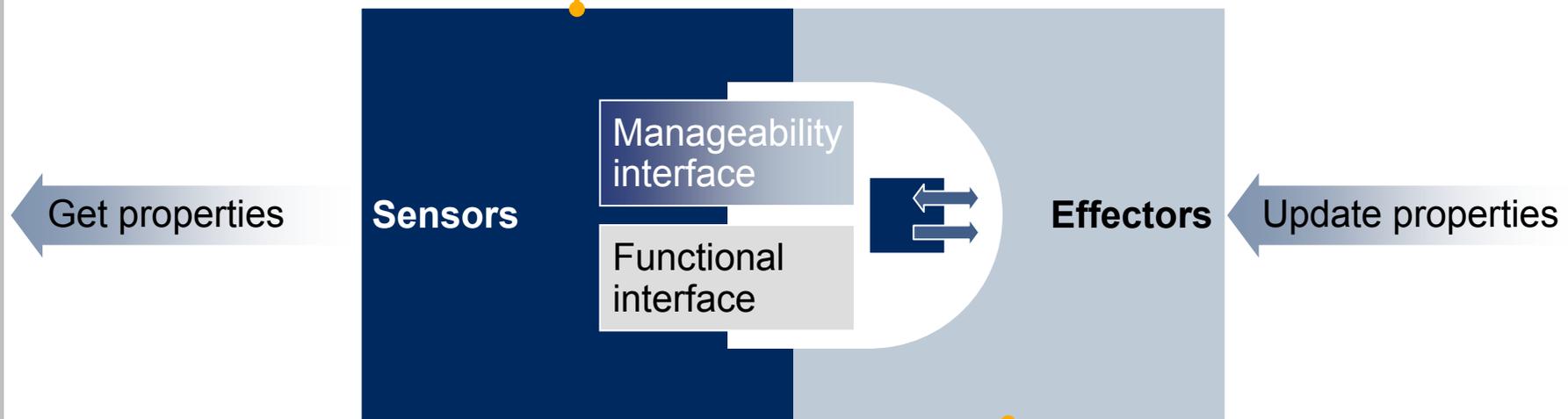
Integrated Reporting

Quelle: McKinsey



Sensors and effectors of services

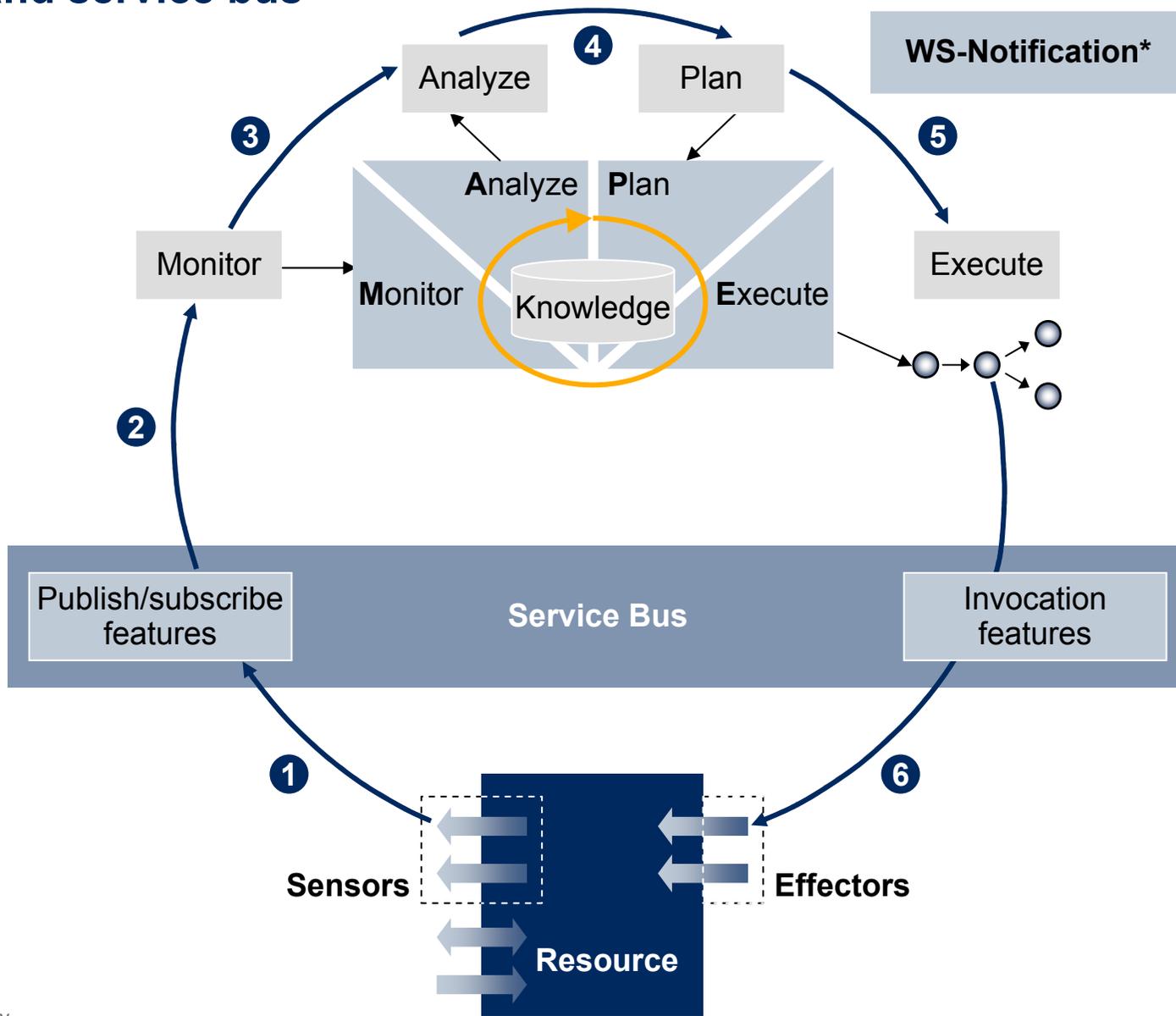
Sensors enable monitoring by providing **queries** ("pull") on properties (actual state, metadata, and data) as well as change **notification** ("push")



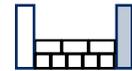
Effectors enable explicit state management by providing lifecycle operations and update operations on state and data



MAPE and service bus



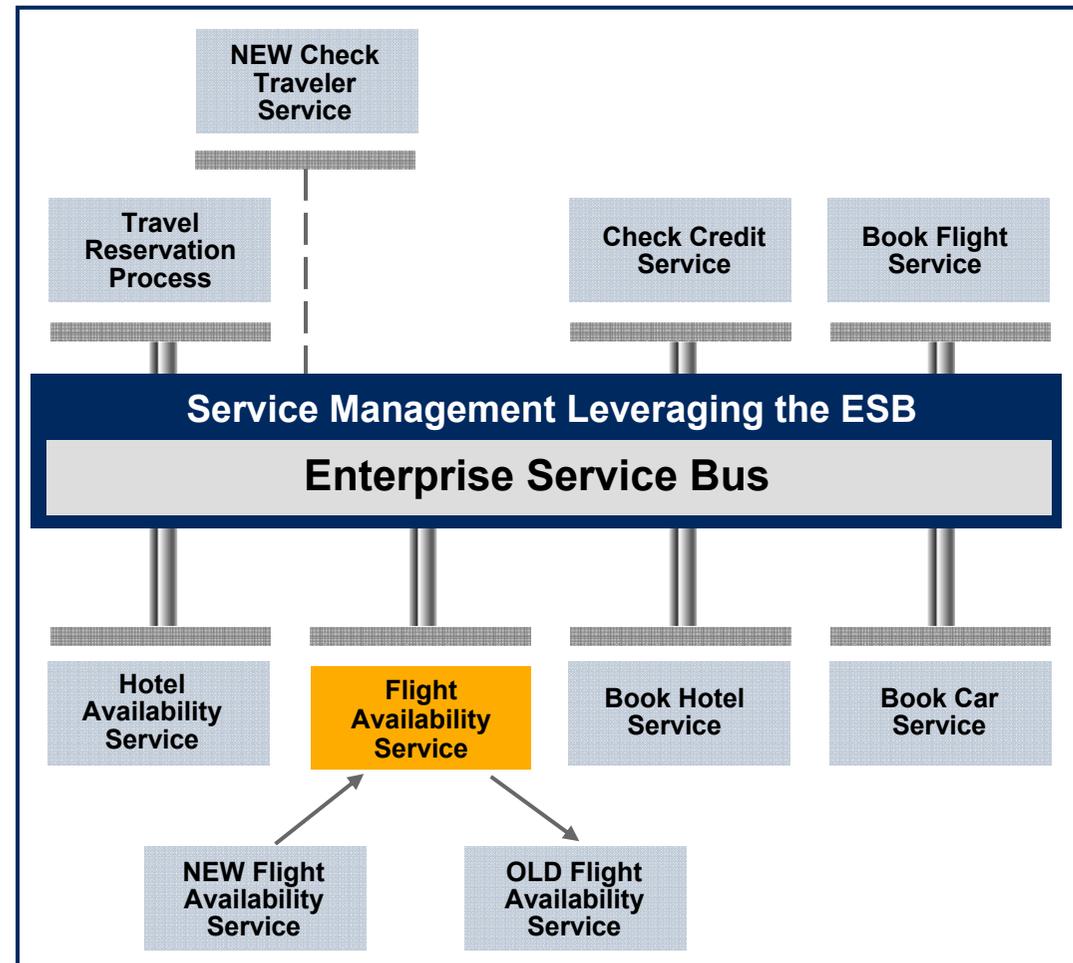
Source: McKinsey



Enterprise Service Bus and SOA Management

Management tools naturally target ESBs as enforcement endpoints

- To perform **Routing** of messages based on system capacity, Quality of Service, and SLAs
- Leverage **Conversion** and **Transformation** capabilities to comply with policy
- Centralize **Handling** of IT events related to Services



Source: McKinsey



SOA Exposes New Management Pains in Application Lifecycle

Model	Assemble	Deploy	Manage
 <p>"I need a service, does it exist?"</p> <p>"How can I debug my production application without reproducing the problem?"</p> <p>"I now have to write a service – how do I make sure it works securely with other services I'm dependent on?"</p>	 <p>"Before I deploy it in production, how can I be sure that the service flow matches the design?"</p> <p>"Does my new SOA application meet its performance goals?"</p>	<p>"Some of our services are used by our partners? How can I be sure they are meeting their SLAs?"</p> <p>"Which part of the SOA infrastructure is causing this service problem? The app server or the messaging connections?"</p>	 <p>"What's the root-cause of this service problem – the BPEL service flow or the application?"</p>

Source: McKinsey

Inhalt



Eine typische Unternehmensstruktur

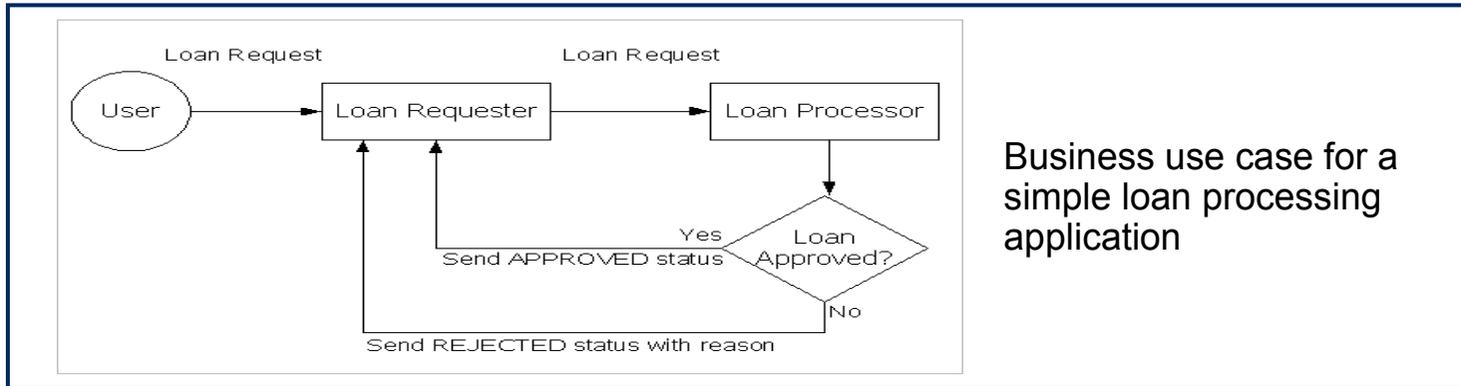
Probleme einer IT-Abteilung

SOA-Grundlagen

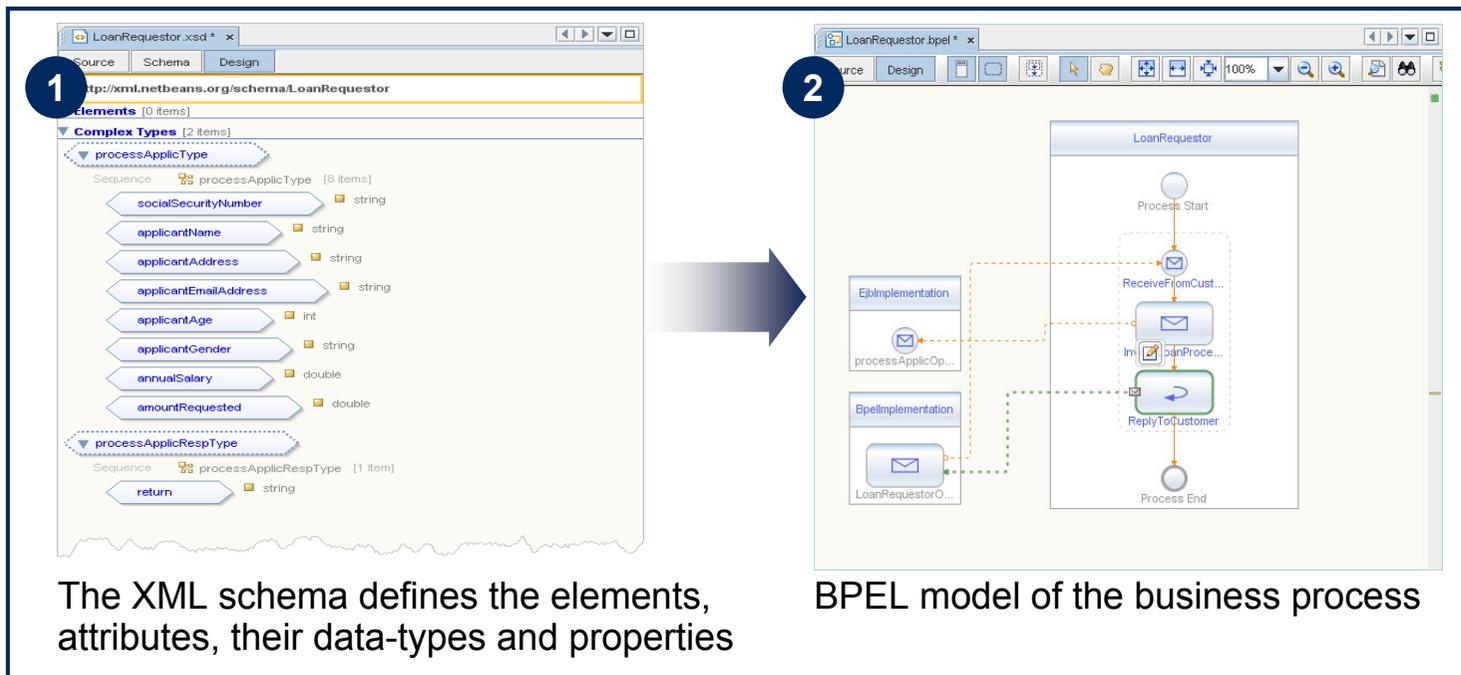
SOA-Laufzeitumgebung

SOA-Beispielanwendung

Example SOA application



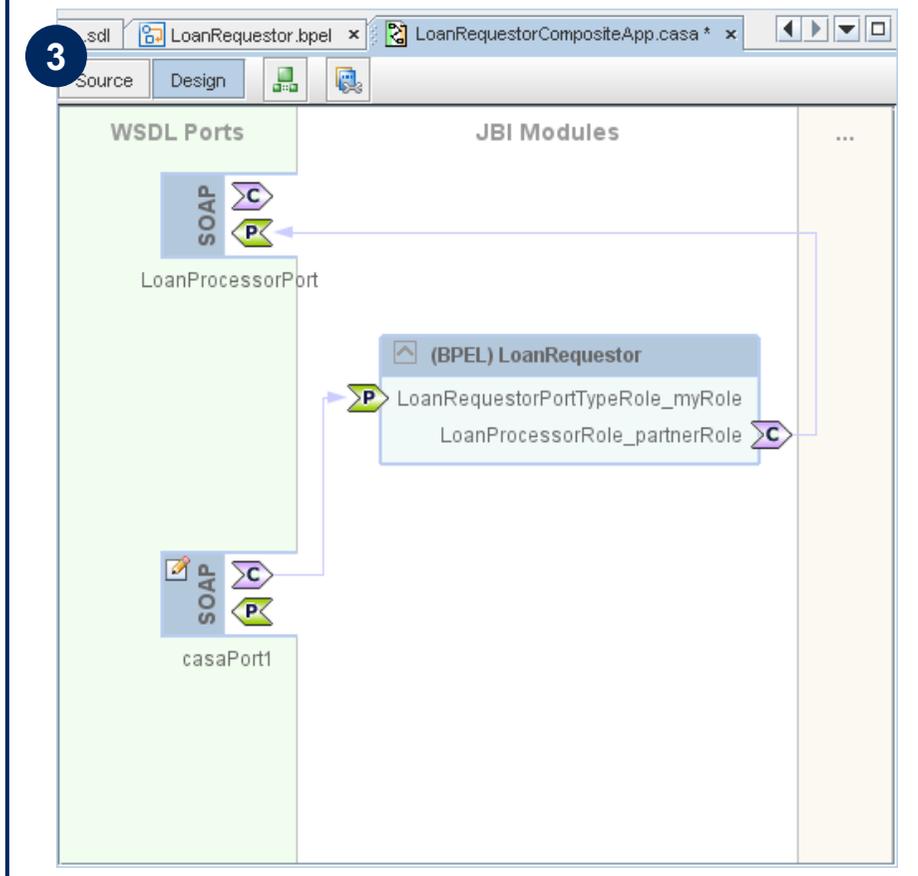
Business use case for a simple loan processing application



Source: netbeans.org

Example SOA application (Contd)

Composite Application Service Assembly (CASA) connects the BPEL process to the WSDL ports and external modules to create a composite web-based application



Source: McKinsey

Encapsulation – The loan processing service is encapsulated from I/O of account information

Loose coupling – Change in user database doesn't affect processing

Service contract - Service descriptions are well defined

Service abstraction - Beyond what is described in the service contract, the process logic is hidden

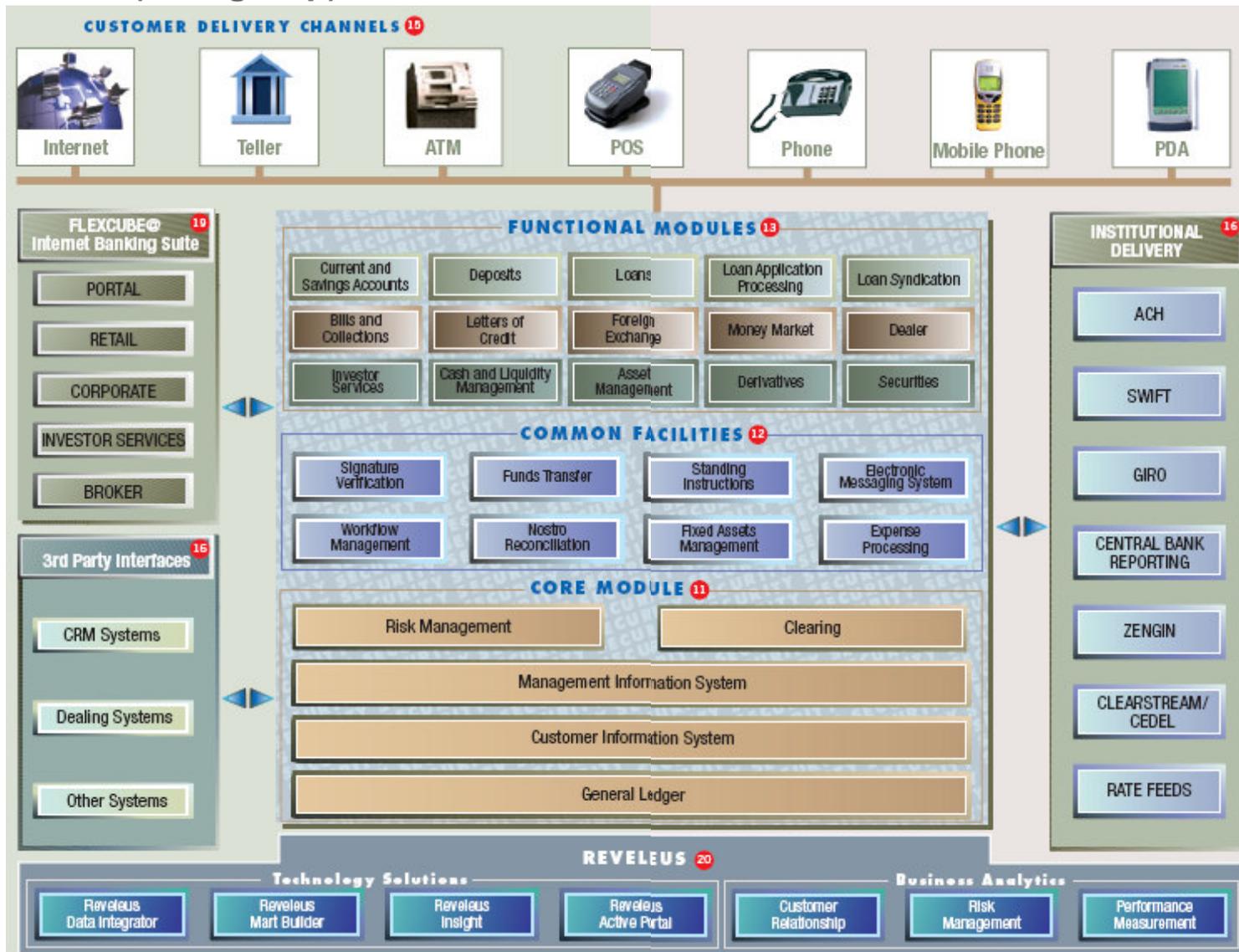
Service reusability - Logic is divided into services with the intention of promoting reuse (maybe launch of a new loan product)

Service autonomy – Services have control over the logic they encapsulate

Service discoverability – Services are designed to be outwardly descriptive so that they can be found and assessed via available discovery mechanisms to create a composite application

SOA enterprise software example

Flexcube, iFlex (Tata group) architecture



Nächste Schritte und offene Fragen

Nächste Schritte

SOA im Gesamtkontext 1/2 (Sam 26. Jan)

- Unternehmenskomplexität
- Bedarf für holistische Landschafts-Architektur
- Architekten und die Architekturabteilung
- Ableitung von der Business Strategie
- Effektivität von Architektur
- Frameworks
- Domänen und Service-Granularität
- Sektorspezifische und generische Architekturen
- Konzepte
- Prinzipien
- Beispiel eines Architekturprojektes

SOA im Gesamtkontext 2/2 (Sam 8. März)

- Effizienz von Architektur
- Strategien zur Umsetzung
- Grundlagen des Veränderungsmanagement
- Organisation & Governance
- Architektur Prozess
- Delivery Modelle
- Beispiel einer SOA Business Applikation
- Zusammenfassung

Offene Fragen

1. Was sind andere Anzeichen für eine Verschiebung des Anwendungsentwicklungsparadigma?
2. Wie interagieren Business Process Management (BPM) und Services? Geht es auch ohne BPM?
3. Was unterscheidet Business Process Monitoring und Business Activity Monitoring (BAM) oder Application monitoring?
4. Wie und warum ist Versionierung von Services anders als traditionelle Versionierung?