



IBM Global Services

The IBM Legacy Transformation Offering

Rainer Gimnich
Senior IT Architect
IBM Business Consulting Services (BCS)
Financial Services, Frankfurt/Main

6. Workshop Software Reengineering

Bad Honnef, 3.-5. Mai 2004

The IBM Legacy Transformation Offering

1 Motivation, goals

2 Legacy Transformation Offering

- Structure
- Alignment with Application Portfolio Management
- Project examples

3 Offering context

- On Demand business
- Service Oriented Architecture (SOA)

Motivation für neue/erweiterte Reengineering-Ansätze

Gründe für heutige Reengineering-Nachfrage ('Pain Points'): u.a.

- ▶ **Mangelnde Flexibilität beim Anpassung heutiger Legacy-Anwendungen an neue Geschäftsanforderungen (neue Gesetze / Produkte / Tarife / ...)**
- ▶ **Hohe Fixkosten im IT-Budget für Legacy-Wartung**
- ▶ **Fehlende langfristige Planung des Legacy-Portfolios (Ungewissheit)**
- ▶ **Inkrementelle, evolutionäre Transformation erwünscht**
- ▶ **Zurückgehende Legacy-Programmierkenntnisse (im Haus, am Markt)**
- ▶ **Fehlende unternehmensweite, durchgängig nutzbare Datenhaltung**
- ▶ **Probleme bei der Integration von Legacy-Anwendungen und bei ihrer Anbindung an Web-Technologien**

Evolution zu On-Demand-Geschäftsmodellen



Identify

Publish

**Integrate
Internally**

**Integrate
Externally**

**Adapt
Dynamically**



Ansteigender Geschäftswert durch Legacy Transformation

Aktuelle Reengineering-Anforderungen

3 Hauptzielrichtungen (auch kombinierbar):

- ▶ **Legacy Transformation Value**
(‘self-contained’ Reengineering)
- ▶ **e-business On Demand, Utility Computing**
- ▶ **Web Services, Service Oriented Architectures**

The IBM Legacy Transformation Offering

1 Motivation, goals

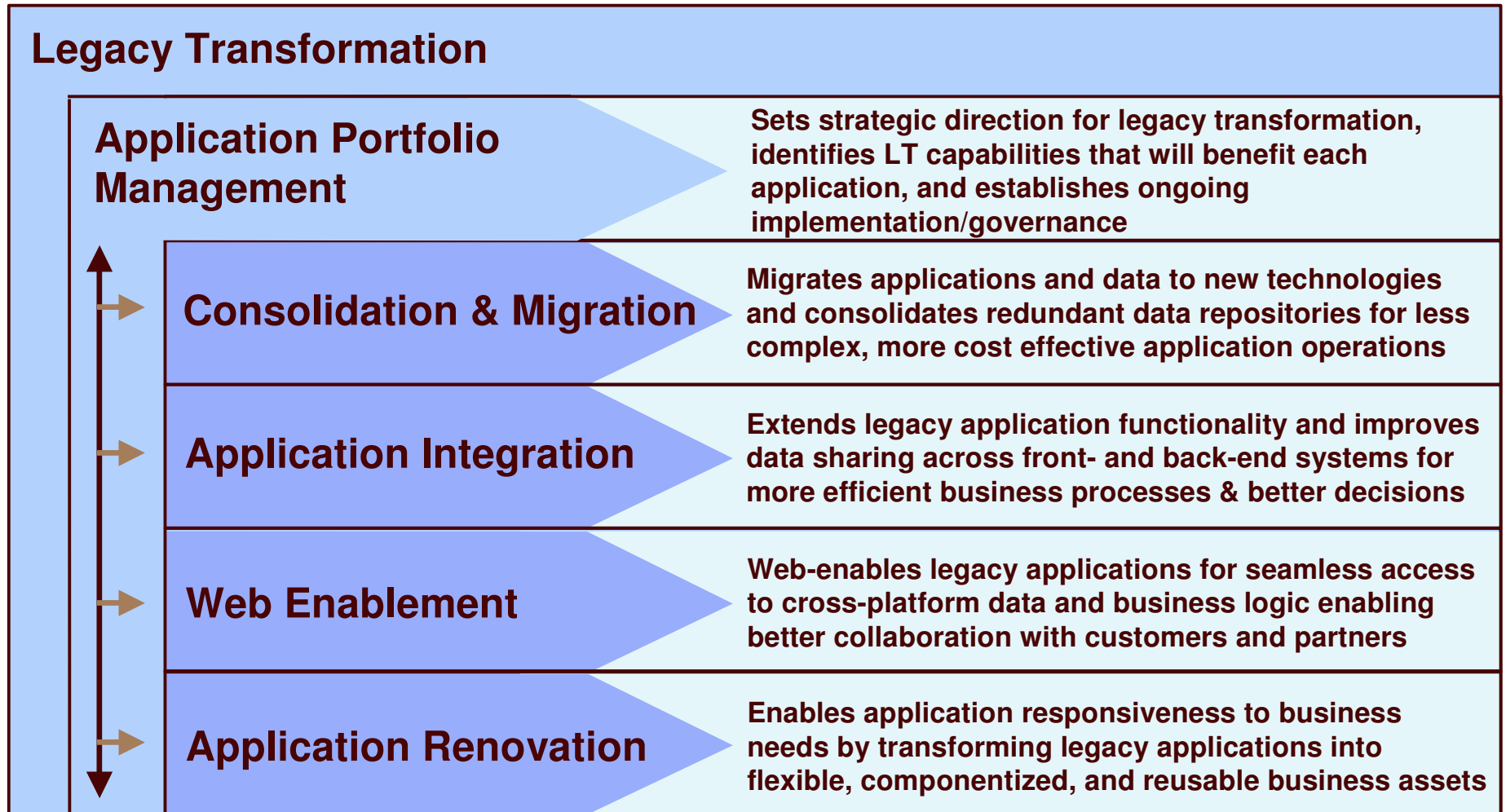
2 Legacy Transformation Offering

- Structure
- Alignment with Application Portfolio Management
- Project examples

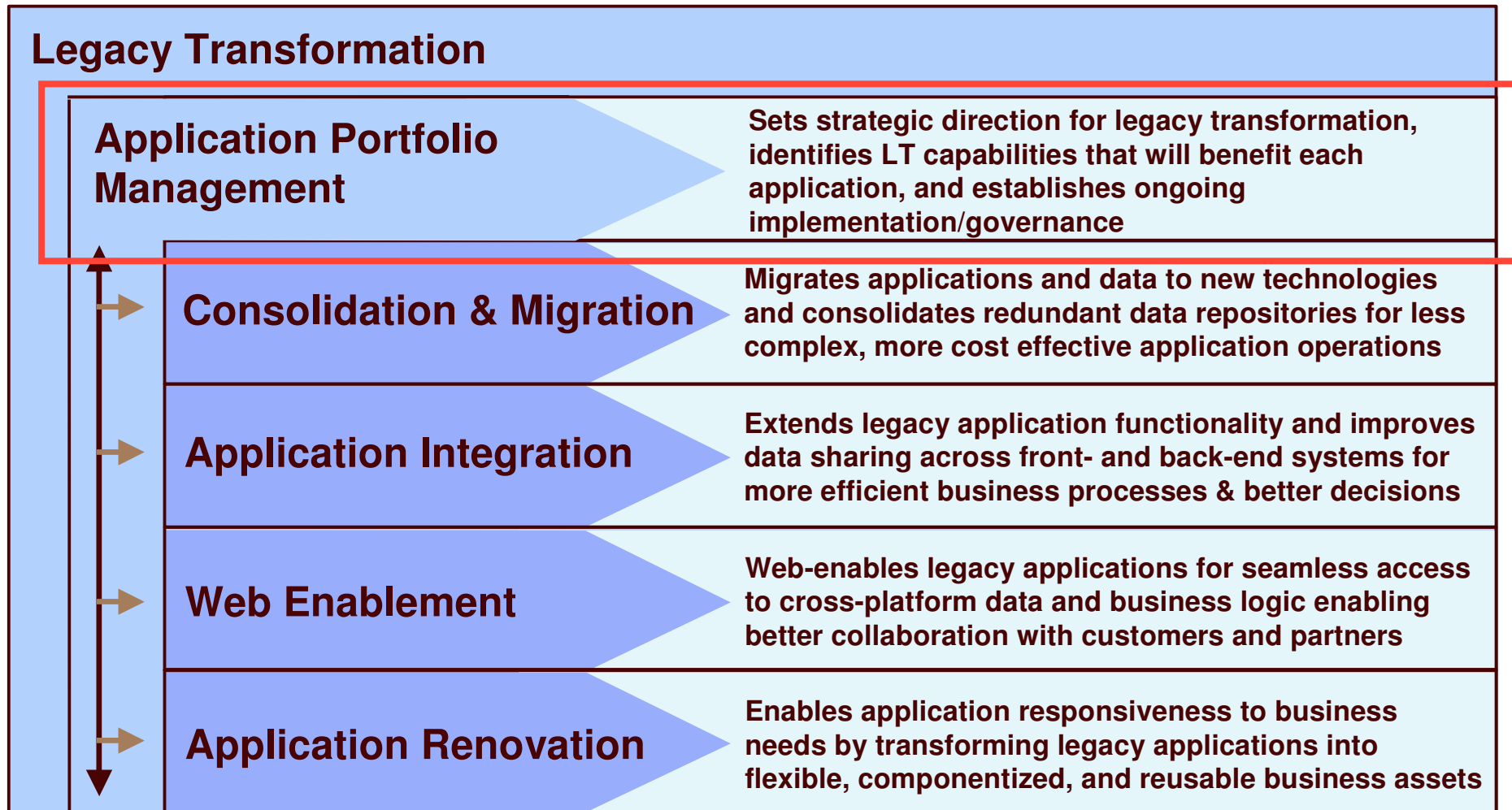
3 Offering context

- On Demand business
- Service Oriented Architecture (SOA)

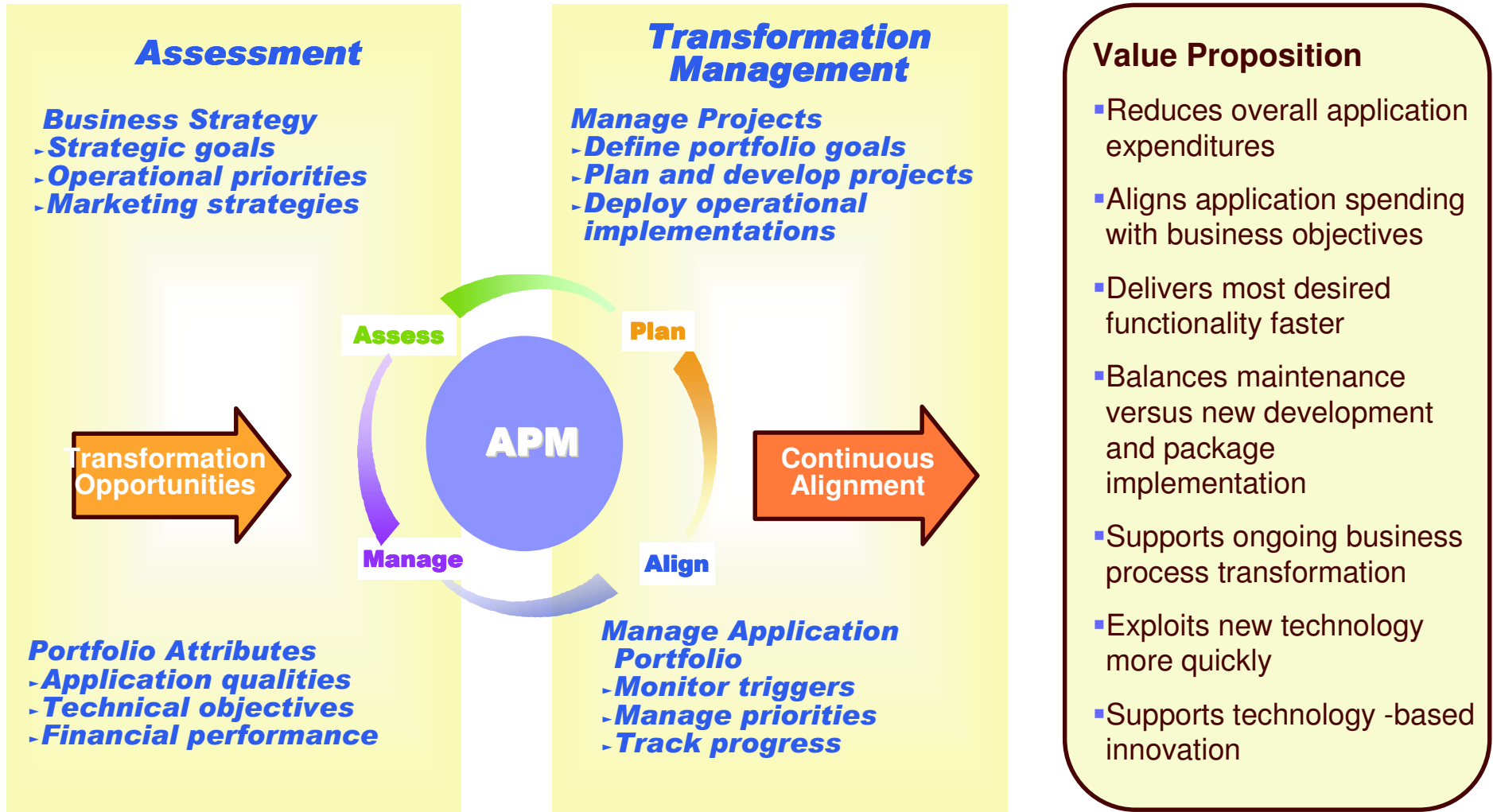
Legacy Transformation Offering Structure



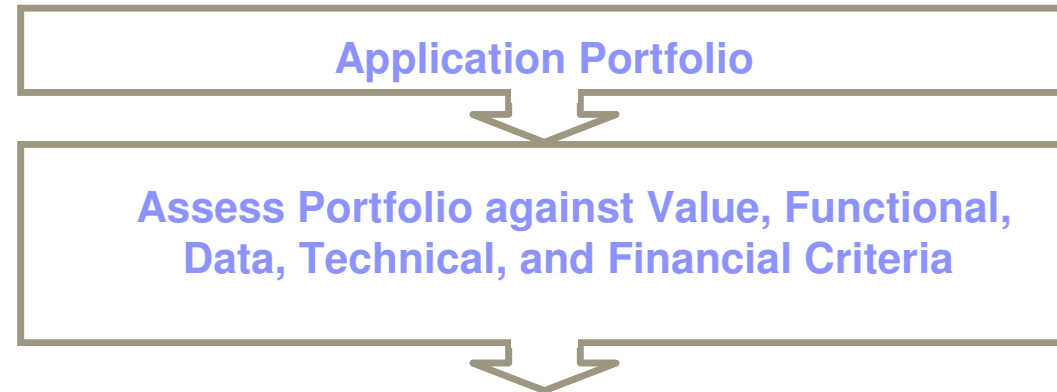
Ganzheitlicher Ansatz: Legacy Transformation



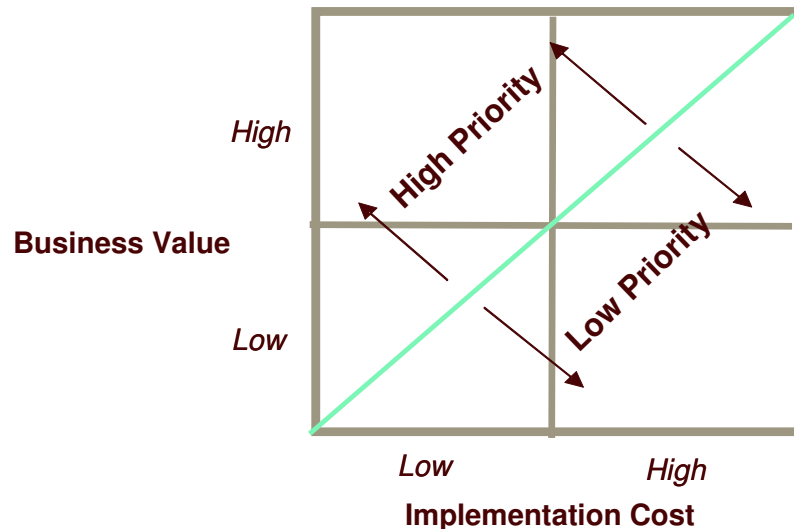
Application Portfolio Management (APM)



Application Portfolio Assessment



Matrix of recommendations



Focus Areas

- Collect business process, strategy, goals and objectives information
- Collect application and architecture information
- Assess the application portfolio
- Identify transformation opportunities
- ROI analysis of the recommendations
- Make final recommendations to retire, replace, restructure, reprioritize or relocate applications

Projektbeispiel: Application Portfolio Management (APM)

Projekt: IBM

1999 Beginn APM (global)

- Ziel: Senkung der (jährlich gestiegenen) Wartungskosten
- Freisetzung von Budget für strategische Entwicklungen
- Stärkere Unterstützung der IBM Geschäftsziele

Anfangs: Mehrere Analysen und Pläne

- Aktuelles Portfolio Assessment
- Definition von Zielen, Einsparungsstrategien, Umsetzungsplänen

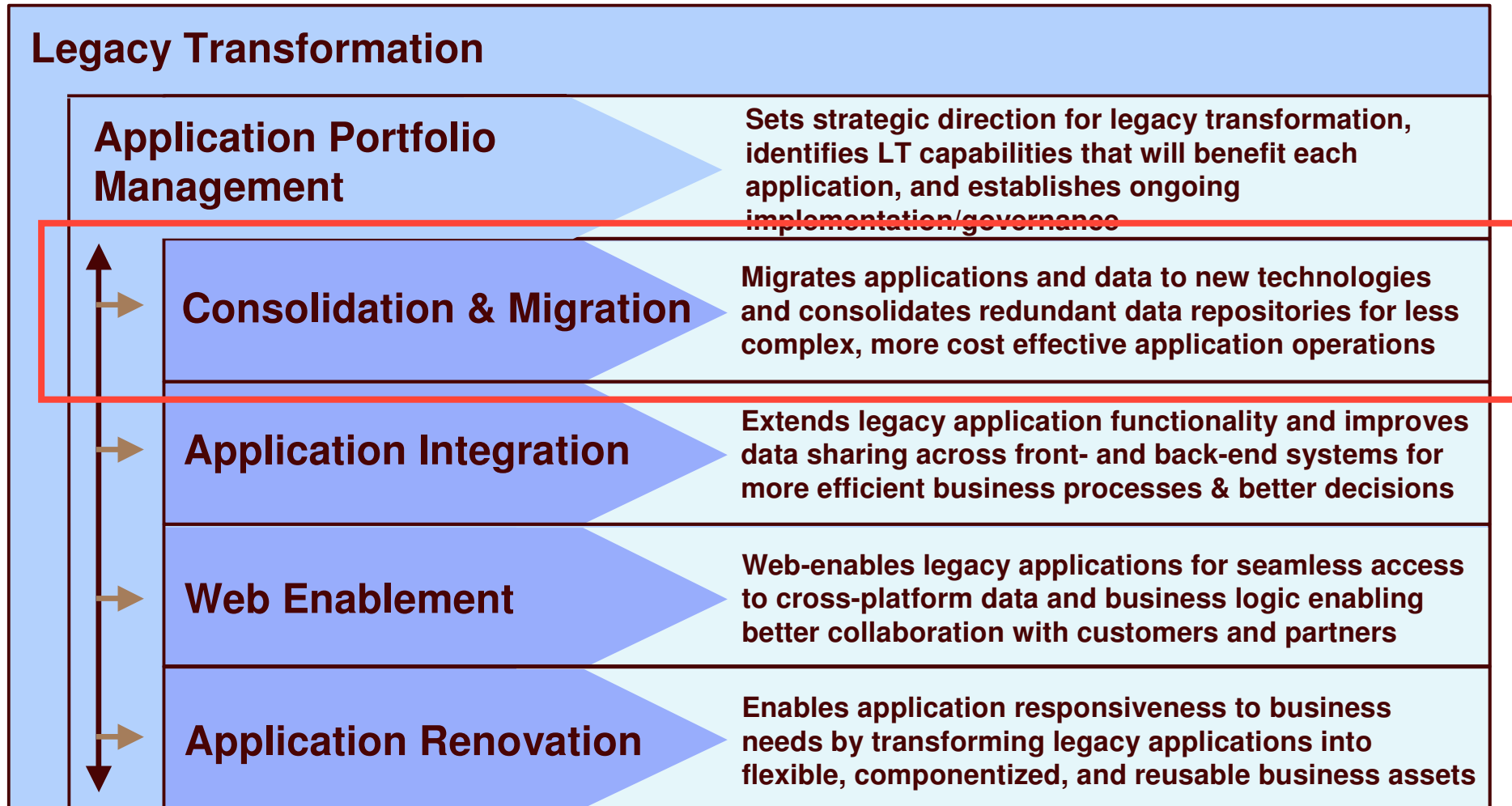
Umfassender APM-Prozess

- Identifikation **redundanter und obsoleter Anwendungen**, Jahresziele (prozentual) für abzulösende Anwendungen
- Klassifikation der Anwendungen nach **strategischem Wert** und Anpassung der Service Level Agreements (SLAs): Fokussierung
- Wesentliches Kriterium: **Benutzerführung** (hohe Wartungskosten bei mangelnder Strukturierung). Bei niedrigen Qualitätswerten: präventive Wartung, Code-Restrukturierung, Modularisierung, Redokumentation.
- Verstärkung der **Offshore-Wartung** aus Kostengründen (Indien, Mexiko, Russland, China)

Ergebnisse der ersten 2 Jahre:

- **Reduktion der Anwendungen:**
16.000 → 6.800
- **Setzen von Prioritäten**
SLAs basieren auf objektiven Geschäftskriterien
- **Reengineering von Anwendungen**
Fehlerrate um 58%, Wartungskosten um 20% gesenkt
- **Wartungsverteilung nach Gesamtkosten:**
20% der Wartung in Ländern mit signifikant niedrigeren Kosten als in den USA

Legacy Transformation (LT)



Consolidation and Migration

Zielsetzung / Vorgehen:

- Konsolidierung von Produktversionen
- Konsolidierung und Migration von Daten in mehrfach nutzbare Repositories
- Code-Redokumentation, Unterstützung des Programmverstehens
- Code-Restrukturierung (Fehlervermeidung, Wartungsverbesserung)
- Anwendungsanpassung, z.B. zur Server-Konsolidierung

Arbeitsbasis:

- Projekt-/Service-Erfahrungen (u.a. IMS/DB → DB2; ASM → COBOL; Schlüsselumstellungen)
- Produkte (u.a. WebSphere Studio Asset Analyzer, Enterprise Developer; Business-Partner-Tools)
- Interne Werkzeuge, PD-Tools



WebSphere®

MICRO
FOCUS

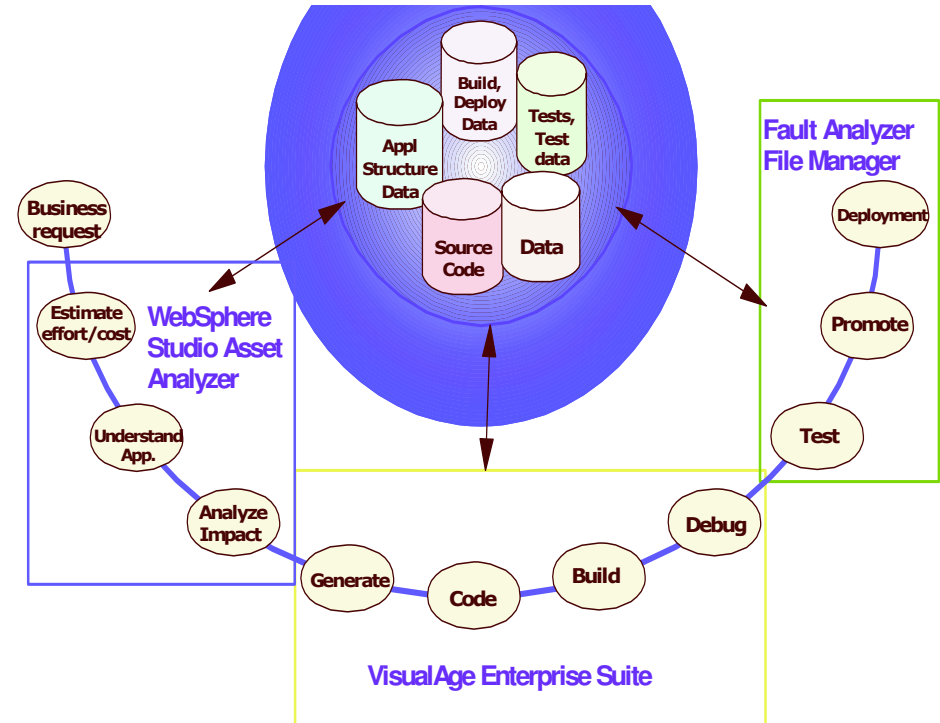


Relativity
TECHNOLOGIES

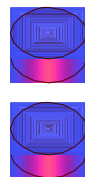
WebSphere Studio Asset Analyzer (1)

Application Asset Scanners

- ▶ Batch JCL
- ▶ Online Transaction / File Definitions
- ▶ Application Program Source
- ▶ Record definitions, call data element use
- ▶ COBOL, PL/I and Assembler
- ▶ File and Database access
- ▶ SCM Adaptors (Serena and SCLM)



Enterprise
Customer
AD artifacts



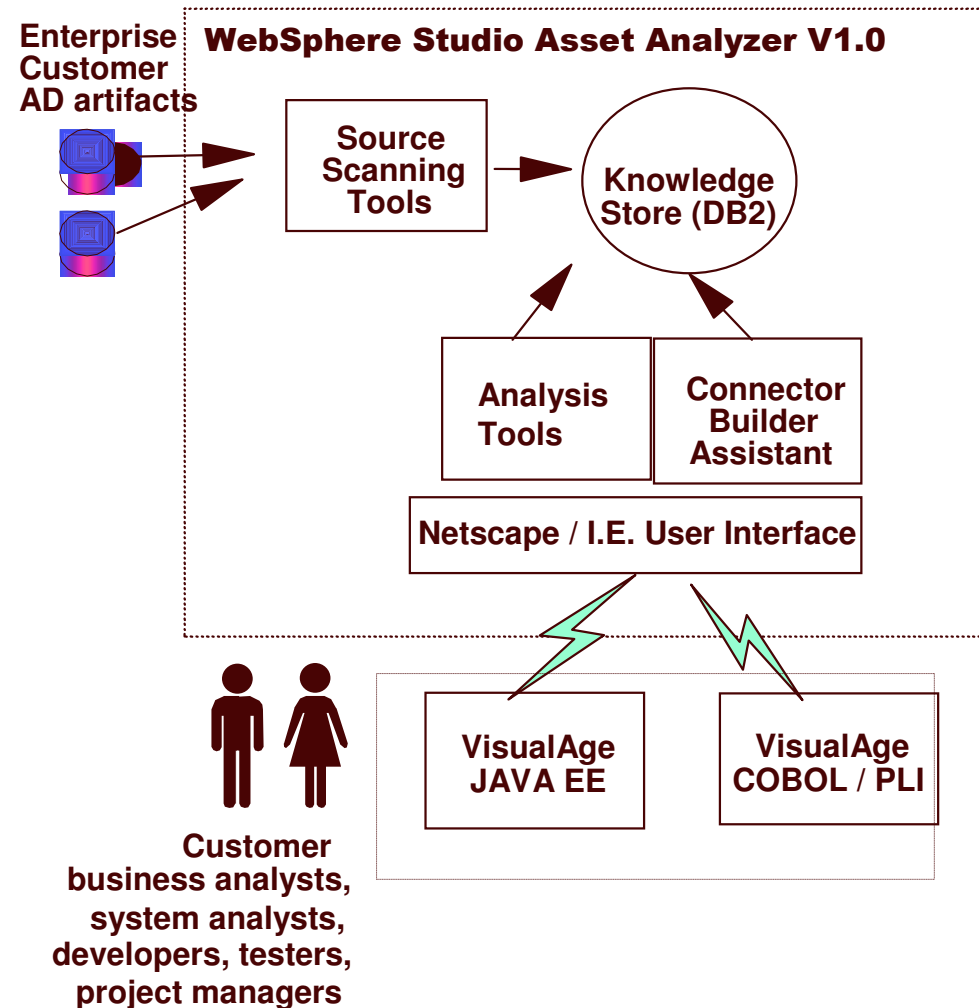
WebSphere Studio Asset Analyzer (2)

- **Application Impact and Understanding Tools**

- ▶ Data item and list scanning
- ▶ Impact identification
- ▶ Worklist generation
- ▶ LOC identification

- **Connector Builder Assistant**

- ▶ CICS, IMS, Batch candidate identification
- ▶ Constructs connector definition consumable by Connector Builders



Projektbeispiel: Konsolidierung (1)

Kunde: Transaktionsbank, Ziel: Physische Mandantentrennung

- Legacy **Abwicklungssystem**, Kernfunktionen ca. **2,8 MLoC**
[“Peripherie” (Ordermanagement, Buchungsschnittstelle, etc.): + ca. 3 MLoC]
- **Datenhaltung**: ca. 700 DB2-Tabellen, 50 VSAM-Dateien;
Verteilung und Replikation unter weitgehender Beibehaltung der Struktur
(über **ENTITY_ID**)
- **Anwendungsprogramme** (COBOL, Assembler, REXX): ca. 2600 Programme,
1000 Copybooks; CICS und Batch; mit ‘Entity-Branching’, weitgehend ohne
Funktionstrennung nach Mandanten
- **Ablaufsteuerung (JCL, OPC)**: ca. 6000 Jobs; Abläufe:
 - generell (bankweit; z.B. WM-Daten-Update (täglich, File Transfer))
 - mandantenübergreifend (z.B. Leistungsabrechnungen)
 - mandantenspezifisch (z.B. Gebührenberechnungen)

Projektbeispiel: Konsolidierung (2)

Configuration Management für separierte Mandanten (auf Basis SCLM)

Layers:

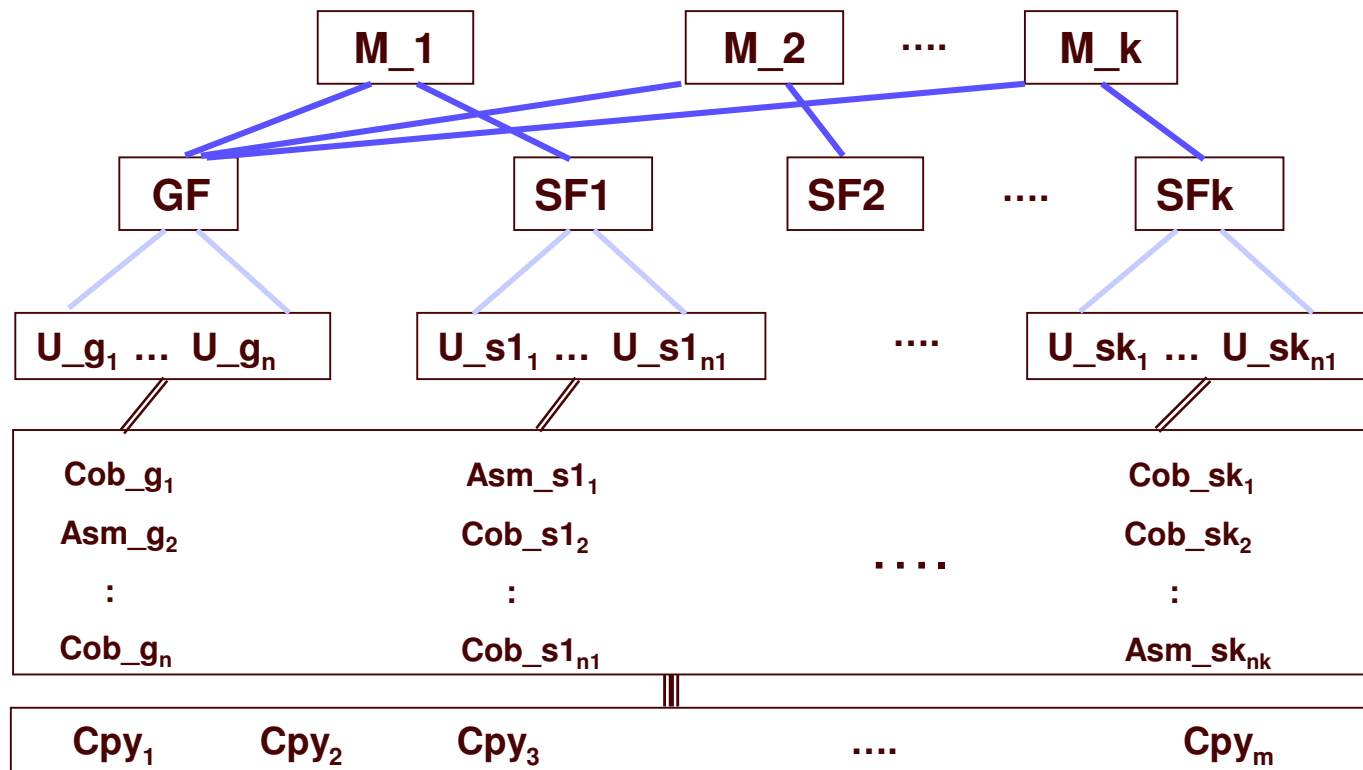
ARC3
(Product)

ARC2
(Sub-product)

ARC1
(Unit)

Programs
(COB,
ASM)

Includes
(CPY)

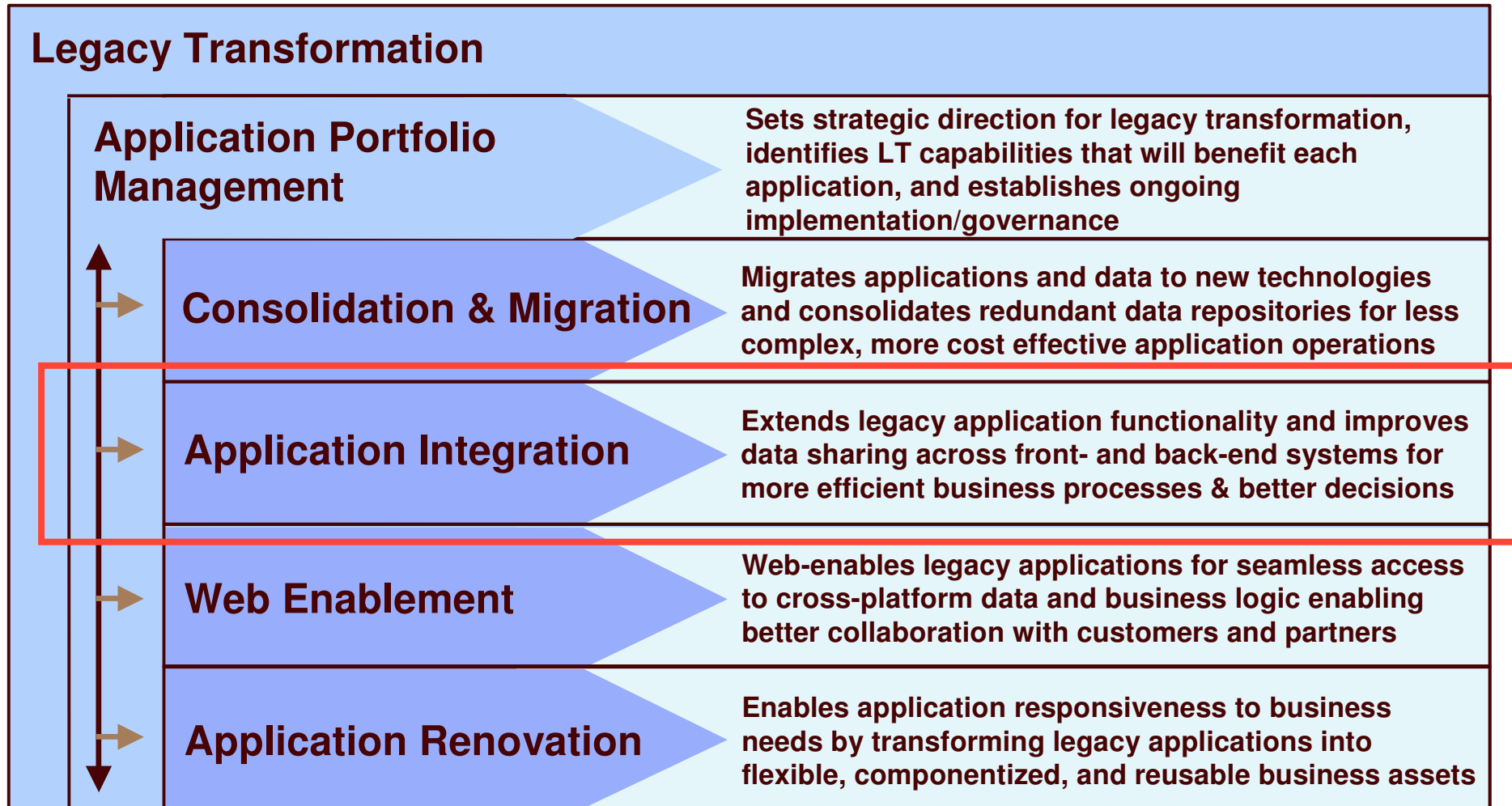


Umfassende Migrationsprojekte

Viele Reengineering-Projekte in dieser Kategorie betreffen problemspezifische Feldtypen im Gesamtsystem, die vollständig zu ermitteln und umzustellen sind:

- Identifizierungsnummern, Kennungen und Nummernkreise:
 - Ordernummer
 - Depotnummer
 - Mandantenummer/-kennung
 - Kundennummer
 - Vertragsnummer
 - Kontonummer, Bankleitzahl (→ IBAN)
 - Wertpapierkennnummer (→ ISIN)
 - Währungskennzeichen
 -
- Berechnungsgrößen:
 - Betragsfelder, insbesondere Summenfelder
 - (Programminterne) Tabellen, u.a. mit Betrags- und Stück-Informationen

Legacy Transformation (LT)



Application Integration

Zielsetzung / Vorgehen:

- Bereinigung inkonsistenter Daten über Anwendungen hinweg
- Integration von Anwendungen über ‚Hub-and-Spoke‘-Architekturen
- Nutzung von Portalen zur Integration auf Benutzerschnittebene
- Management der Datentransformationen über Messages und Queues (Robustheit, Zuverlässigkeit, Wartbarkeit des Anwendungskomplexes)
- Einfügung von ‚Mid-Tier‘-Anwendungen zur Flexibilisierung

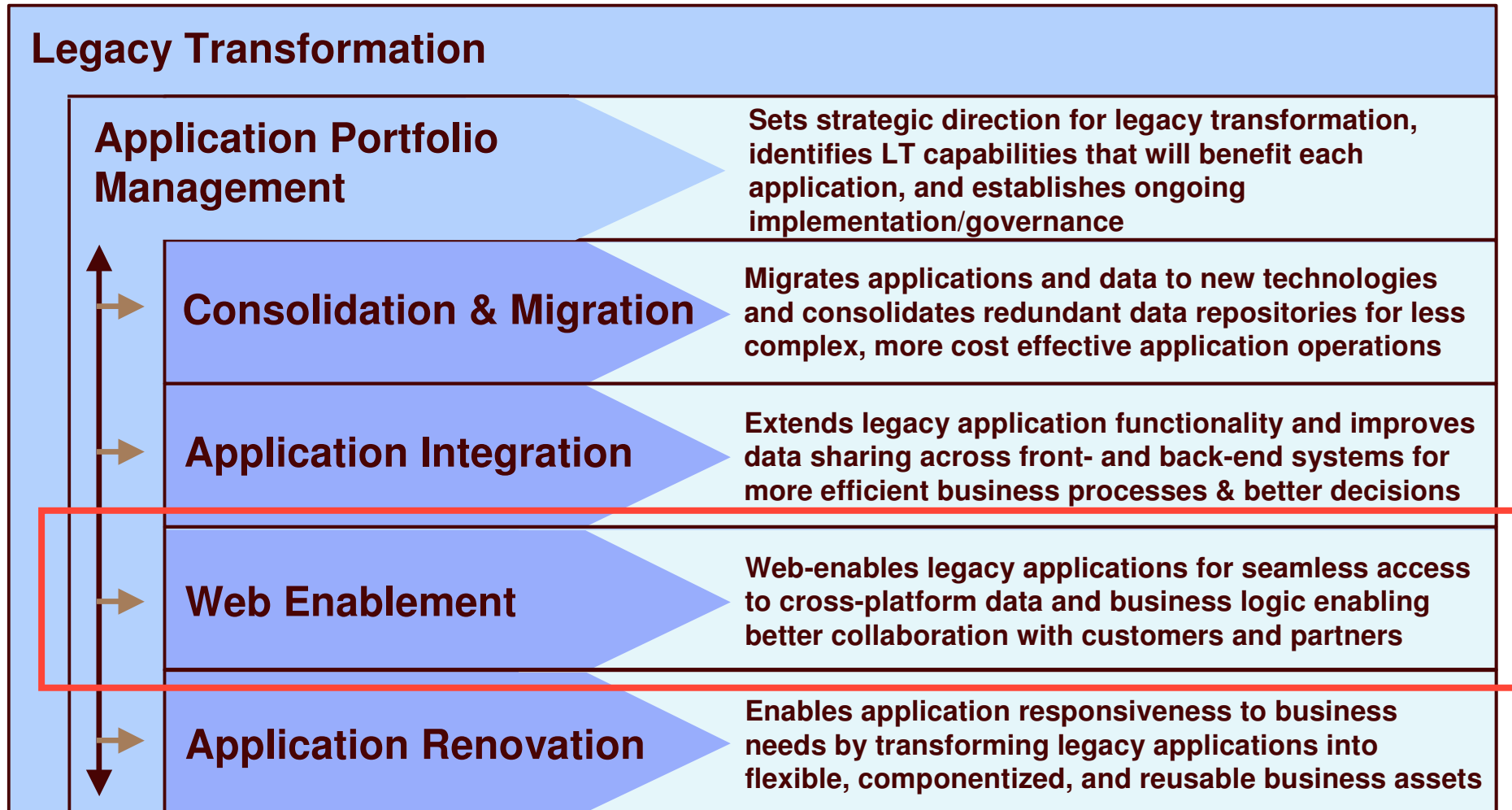
Arbeitsbasis:

- Projekt-/Service-Erfahrungen (viele EAI-Projekte, Fachwissen im Anwendungsbereich)
- Produkte (u.a. WebSphere MQ Familie:
MQ Business Integration Broker,
MQ Workflow;
CCF (Common Connector Framework))
- Interne Werkzeuge, PD-Tools



WebSphere®

Legacy Transformation (LT)



Web Enablement

Zielsetzung / Vorgehen:

- Benutzerschnittstellen-Modernisierung durch Portal-Software
- Flexiblere Nutzung von Legacy-Funktionen durch ‚Mid-Tier‘-Anwendungen, z.B. über XML-Wrapper
- Trennung von Geschäftslogik und Präsentation
- Anbindung weiterer Medien und Web-Technologien an Legacy-Systeme
- Einfacherer Datenaustausch mit Kunden, Partnern, Lieferanten (geringere Interface-Kosten)

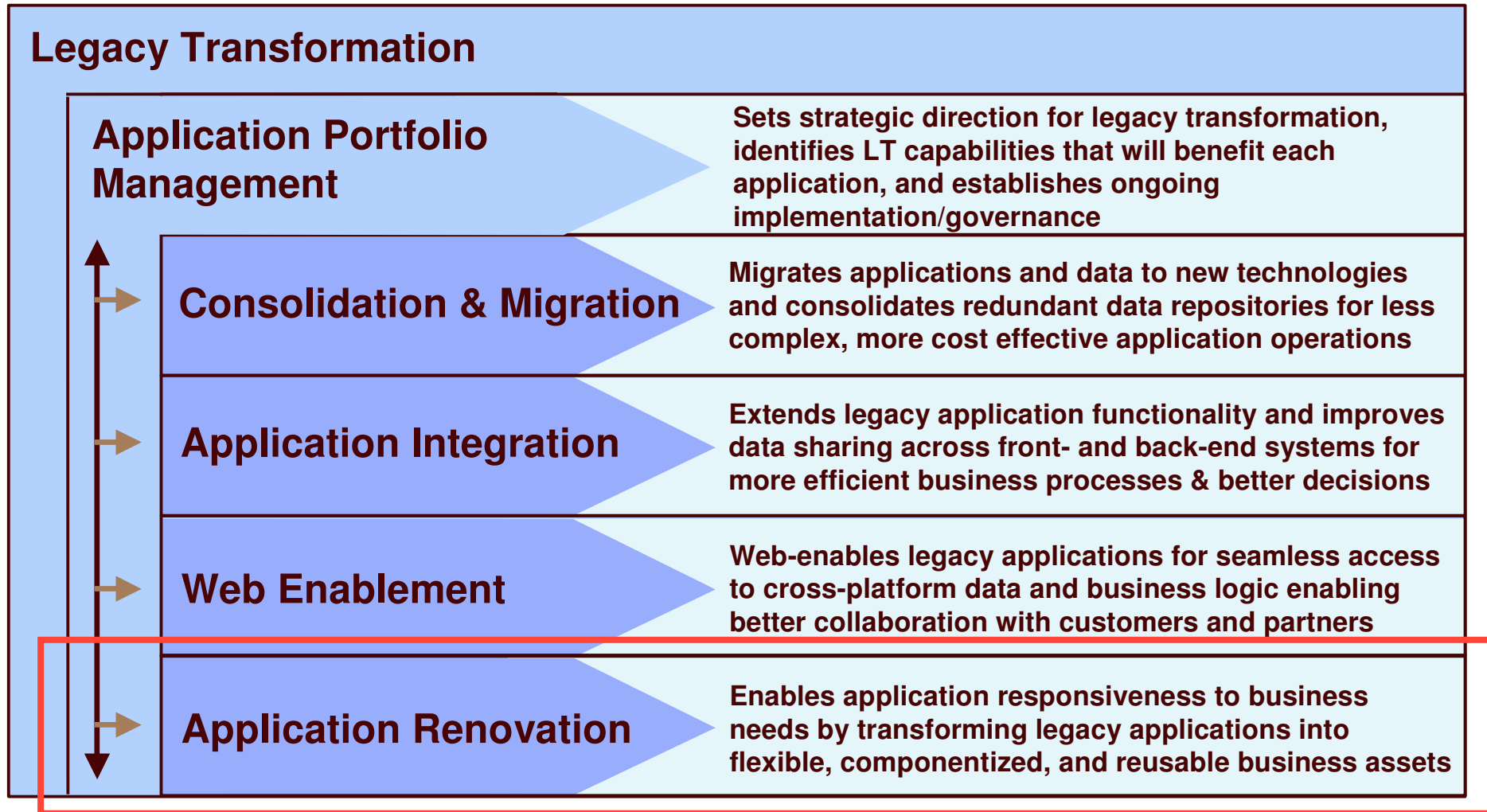
Arbeitsbasis:

- Projekt-/Service-Erfahrungen (viele Portal-Projekte, Fachwissen im Anwendungsbereich, Ergonomie-Fachwissen)
- Produkte (u.a. WebSphere Portal Server, Content Manager, Application Server; WebSphere- und CrossWorlds-Adapter)



WebSphere®

Legacy Transformation (LT)



Application Renovation

Zielsetzung / Vorgehen:

- (Re-)Dokumentation
- Extraktion und Zentralisierung von Geschäftsregeln und –logik
- Modularisierung und Einführung von Software-Wiederverwendungsstrategien
- ‚Komponentisierung‘ und Transformation in neue Architekturen
- Flexibilisierung und Vereinfachung des ‚Change Management‘

Arbeitsbasis:

- Projekt-/Service-Erfahrungen (insbesondere technische Analyse, komponentenbasierte Entwicklung, Web Services)
- Produkte (u.a. WebSphere Asset Analyzer, Relativity für Modularisierung, WebSphere Application Server, WebSphere MQ Workflow, CrossWorlds, Holosofx für Modellierung und Simulation)



WebSphere®



IBM Holosofx Family

The IBM Legacy Transformation Offering

- 1 Motivation, goals
- 2 Legacy Transformation Offering
 - Structure
 - Alignment with Application Portfolio Management
 - Project examples
- 3 Offering context
 - On Demand business
 - Service Oriented Architecture (SOA)

On-Demand-Geschäftsmodelle



Ein Unternehmen, bei dem sämtliche Geschäftsprozesse end-to-end integriert sind, um dynamisch auf Kundenanforderungen, Marktchancen, externe Beeinflussungen und Bedrohungen reagieren zu können.

Reaktionsfähig

Intuitive Fähigkeit, dynamische und unvorhergesehene Veränderungen aufzuspüren und umzusetzen

Variabel

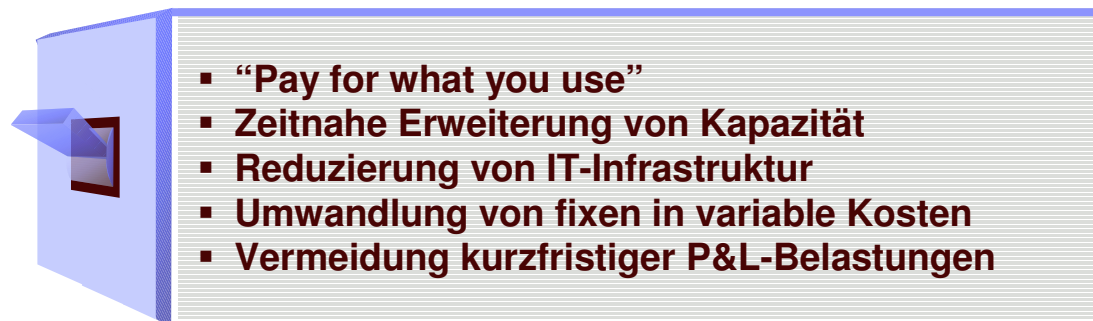
Verwenden von variablen Kostenstrukturen und flexible Anpassung von Prozessen

Fokussiert

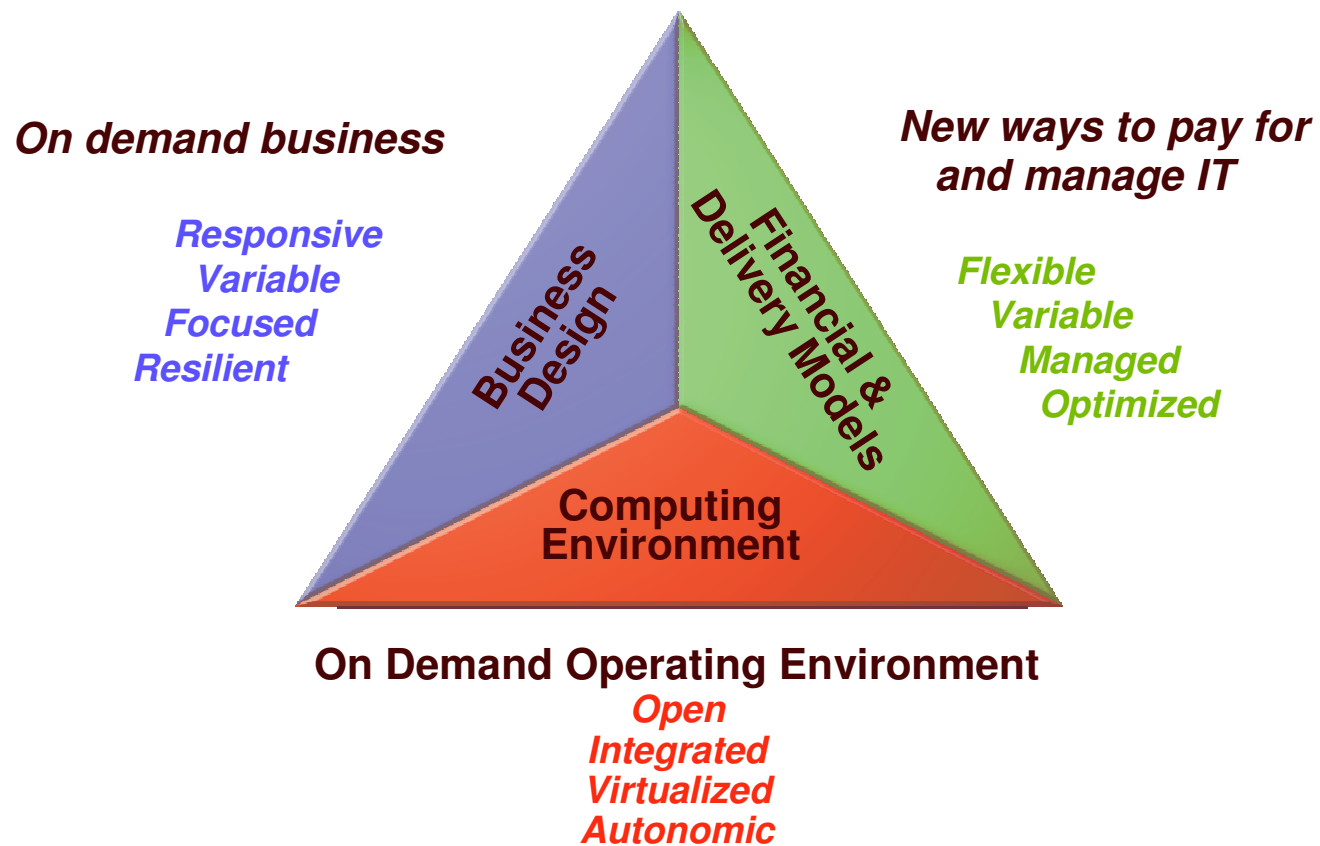
Konzentration auf Kernkompetenzen, während sich voll integrierte strategische Partner um ausgewählte Aufgaben kümmern

Widerstandsfähig

Auf Veränderungen und Bedrohungen mit konsistenter Verfügbarkeit und Sicherheit reagieren können

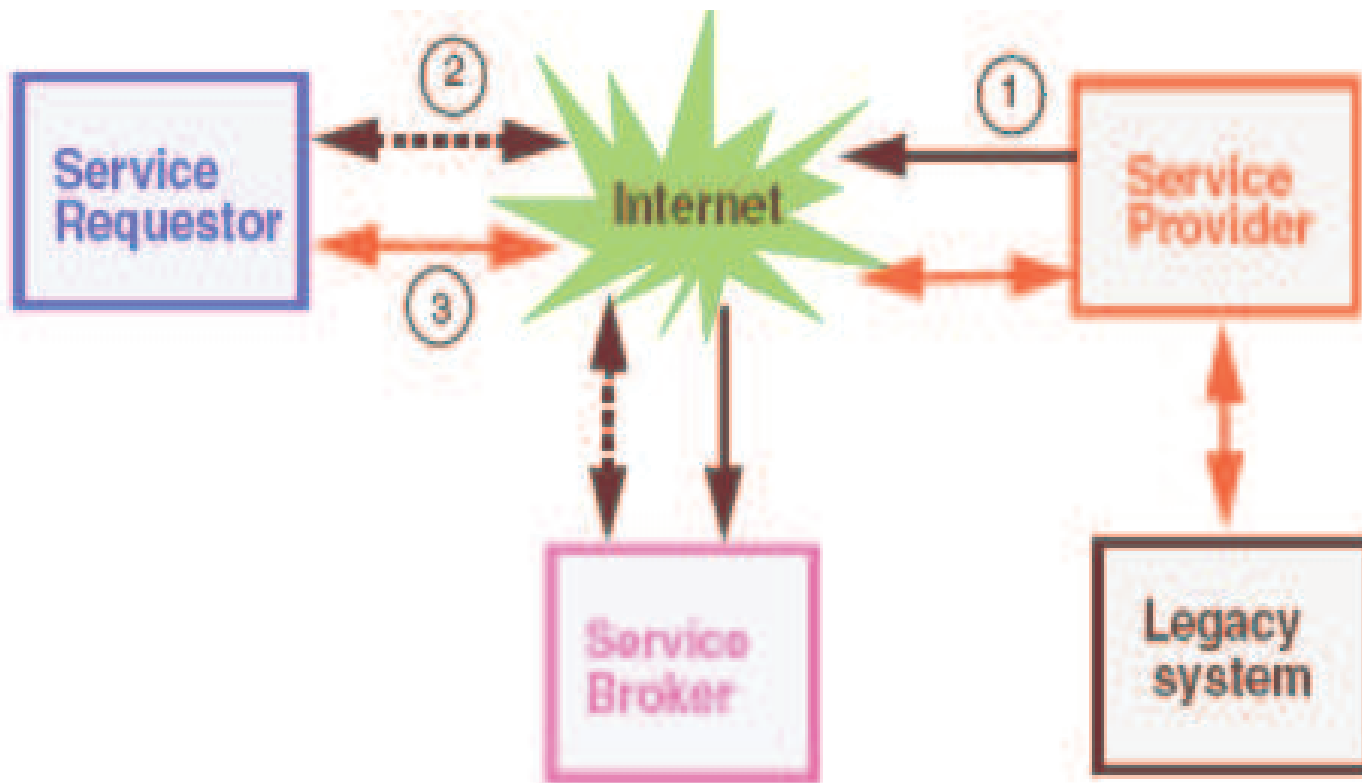


On Demand business / IT / finance



Mögliche Realisierungsbasis für On-Demand-Unternehmen: Service-Orientierte Architekturen (SOA)

SOA-Implementierungen nutzen i.d.R. **Web Services** als internen ‚Standard‘ zur Definition und Nutzung der Komponenten



Konzept der Service-Orientierten Architektur (SOA)

- SOA: Struktur implementierter Geschäftsprozess(teil)e, die wiederverwendbare Funktionen beinhalten und standard-basiert intern und extern aufgerufen und integriert werden können
- Der Übergang von Legacy-Anwendungen zu SOA erfolgt in der Regel **evolutionär** über mehrere Jahre.
- Auch die Zwischenstände auf dem Weg zur SOA haben ihren Nutzen (ROI messbar) und können jahrelang produktiv bleiben.
- Die Implementierung der SOA(-Übergänge) erfordert i.d.R.
 - eine **EAI-Zugriffslogik**, die z.B. über ein EAI-Tool oder eine eigenentwickelte Zugriffsschicht bereitgestellt wird;
 - eine Möglichkeit, die entstehenden Services in ihrem ‚Zusammen- spiel‘ zu definieren (Process **Orchestration**, Workflow Management)

Fazit zu SOA

- **Granularität der Komponenten:**

- Frei wählbar, hat aber deutliche Auswirkungen auf Performance, Akzeptanz, Wiederverwendbarkeit,
- Grobe Granularität: großer Funktionsumfang, einfaches Management, aber geringe Wiederverwendbarkeit/Austauschbarkeit
- Feine Granularität: kleiner Funktionsumfang, hohe Wiederverwendbarkeit/Austauschbarkeit, aber hoher Kommunikationsaufwand, Performance-Nachteile

- **Voraussetzung für ‚gute‘ Granulierung:**

Wissen über die geschäftlichen und technischen Eigenschaften der Anwendungen sowie ihre erwartete Änderungshäufigkeit

- Portfolio-Analyse, Anwendungs- und Programmverstehen
- Restrukturierung, Refaktorisierung