



IBM Software Group

Enterprise SOA Transformation – Experience from the Telco Industry

Dr. Matthias Grützner

*EMEA Lead Architect
SOA Advanced Technologies
IBM Software Group*



ON DEMAND BUSINESS™

March 12, 2007

© IBM Corporation 2007

Agenda

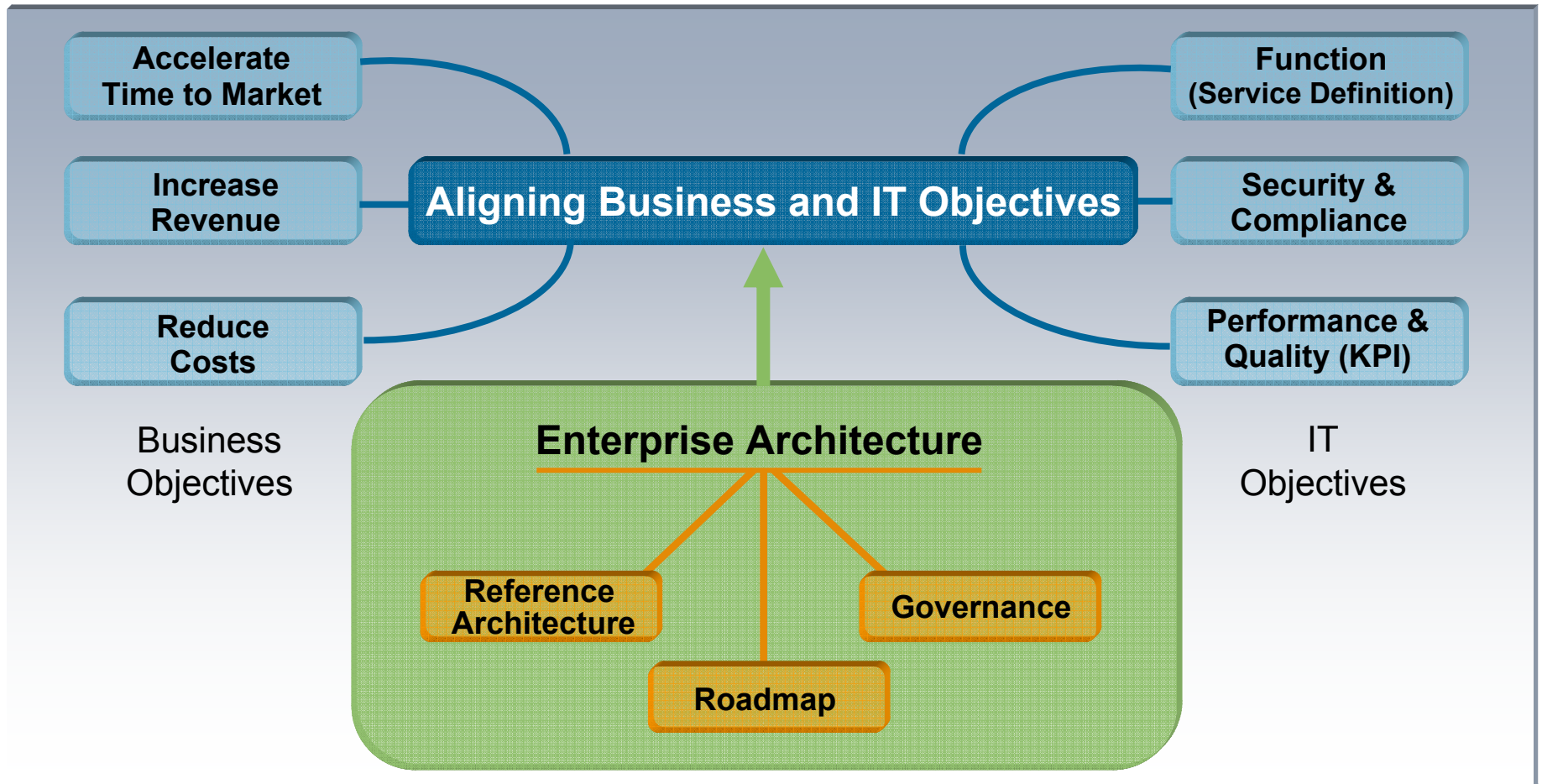
1. What is Enterprise SOA Transformation?
2. Methods to support the Transformation
3. Case Studies from the Field
 - ▶ Study 1: Ad hoc transformation for fast success
 - ▶ Study 2: eTOM level 4/5 mapping onto Packaged applications (COTS)
 - ▶ Study 3: Bottom up SOA, attempt to transform legacy
4. Lessons Learned
5. Conclusion and Outlook



“A holistic approach to the control and co-ordination of IT based business projects”

Ed Kahan, IBM Fellow, 2003

SOA-based Enterprise Architecture



The “A” in SOA

A set of services that a business wants to expose to customers and clients

Business

An architectural style which requires a service provider, requestor and a service description.

A set of architectural principles and patterns which address characteristics such as *modularity, encapsulation, loose coupling, separation of concerns, reuse, composable and single implementation.*

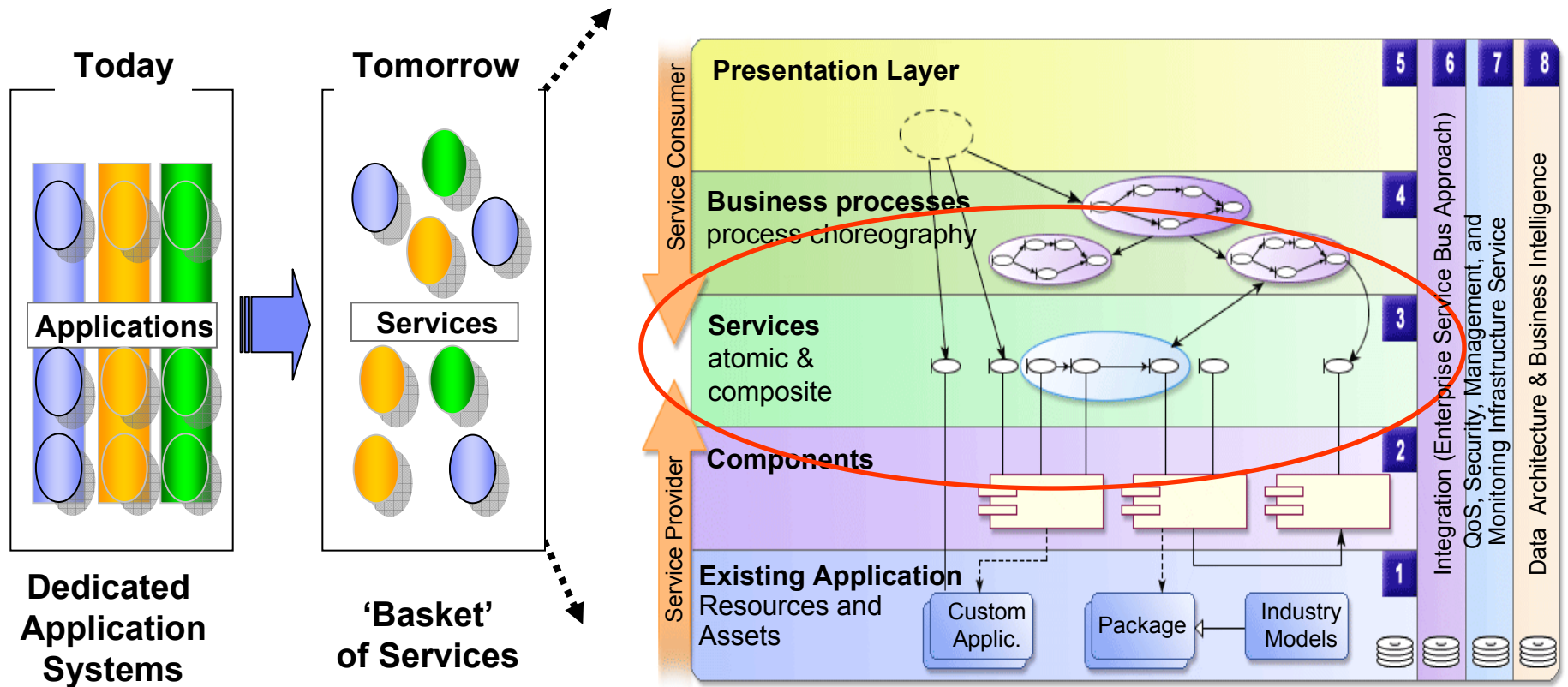
Architecture

A programming model complete with standards, tools, methods and technologies such as web services.

Implementation



SOA – Vision



Including Open Standards (Web Services, WSDL, SOAP, HTTP,



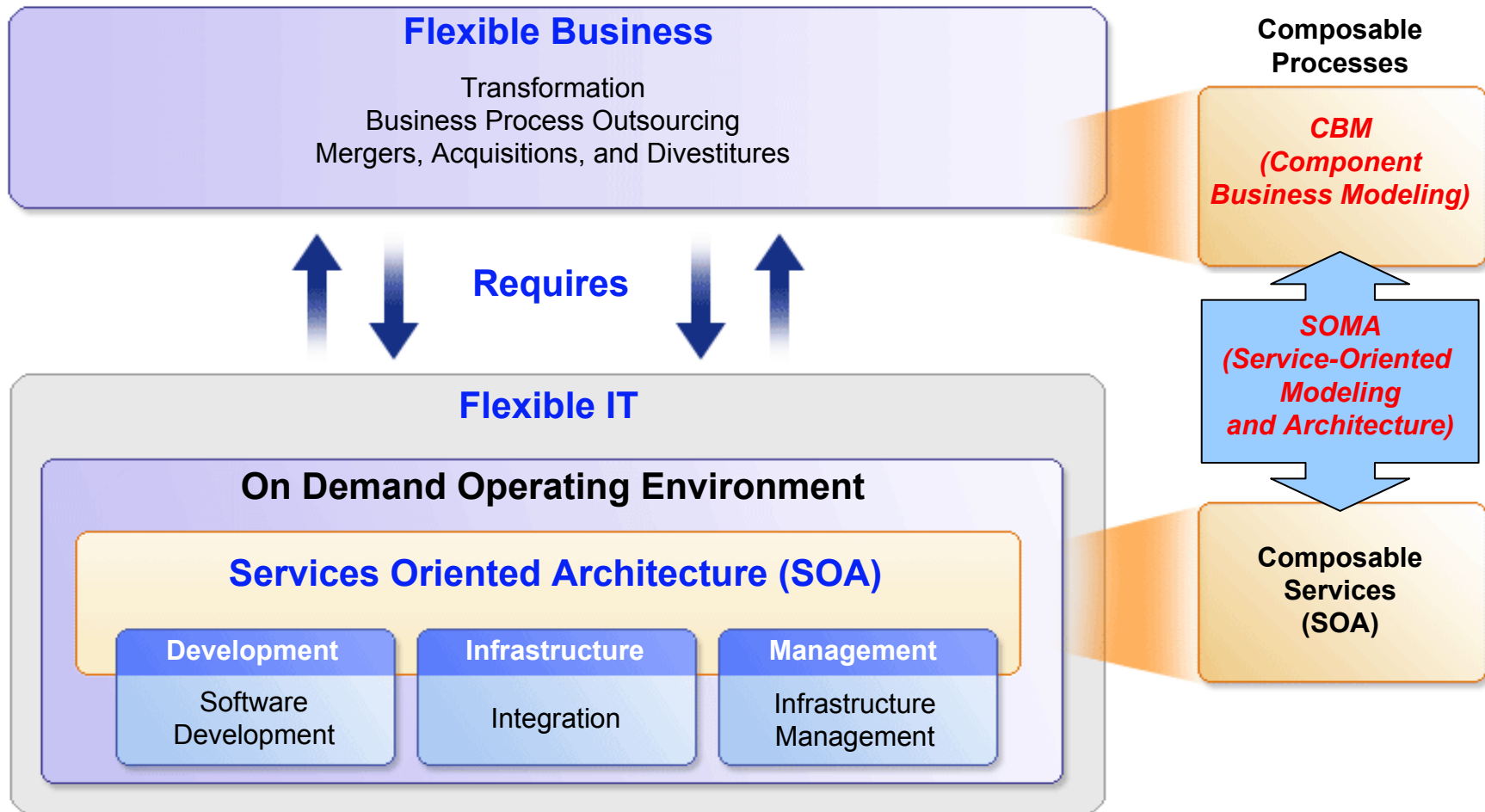
Business Goals & Technical implications in SOA Transformation

- **To optimize profitability there is a requirement on streamlining the operation as well as being able to act more quickly (agility & flexibility) on change in the marketplace**

- **Enterprise benefits to be realized by introducing an SOA:**
 - ▶ Leverage existing *assets*
 - ▶ Provide an well defined service catalog from new and leveraged assets
 - ▶ Reduce Redundant and non-reusable programming (→ application portfolio consolidation, common components, reuse)
 - ▶ Support all required types of integration (user interaction, process integration, application connectivity, information integration; build to integrate)
 - ▶ Allow for incremental implementations and light weight migration of assets
 - ▶



Methods (CBM, SOMA) to support the Transformation



Method: Component Business Modeling (CBM)

Domains

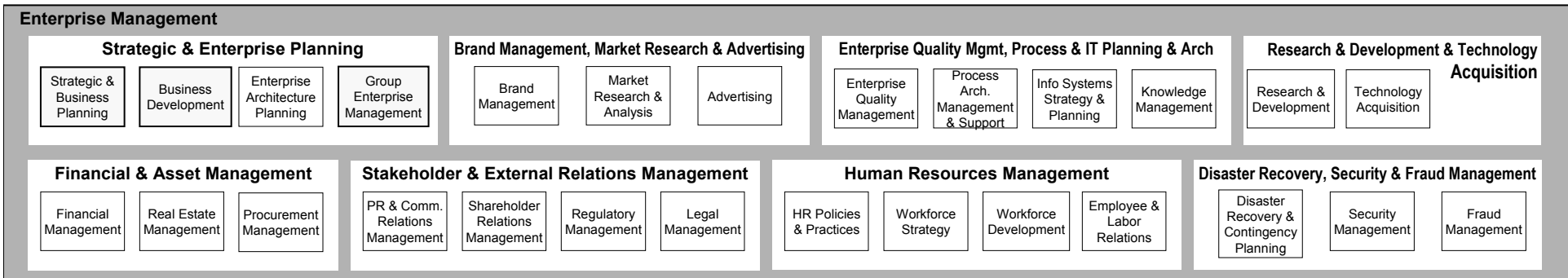
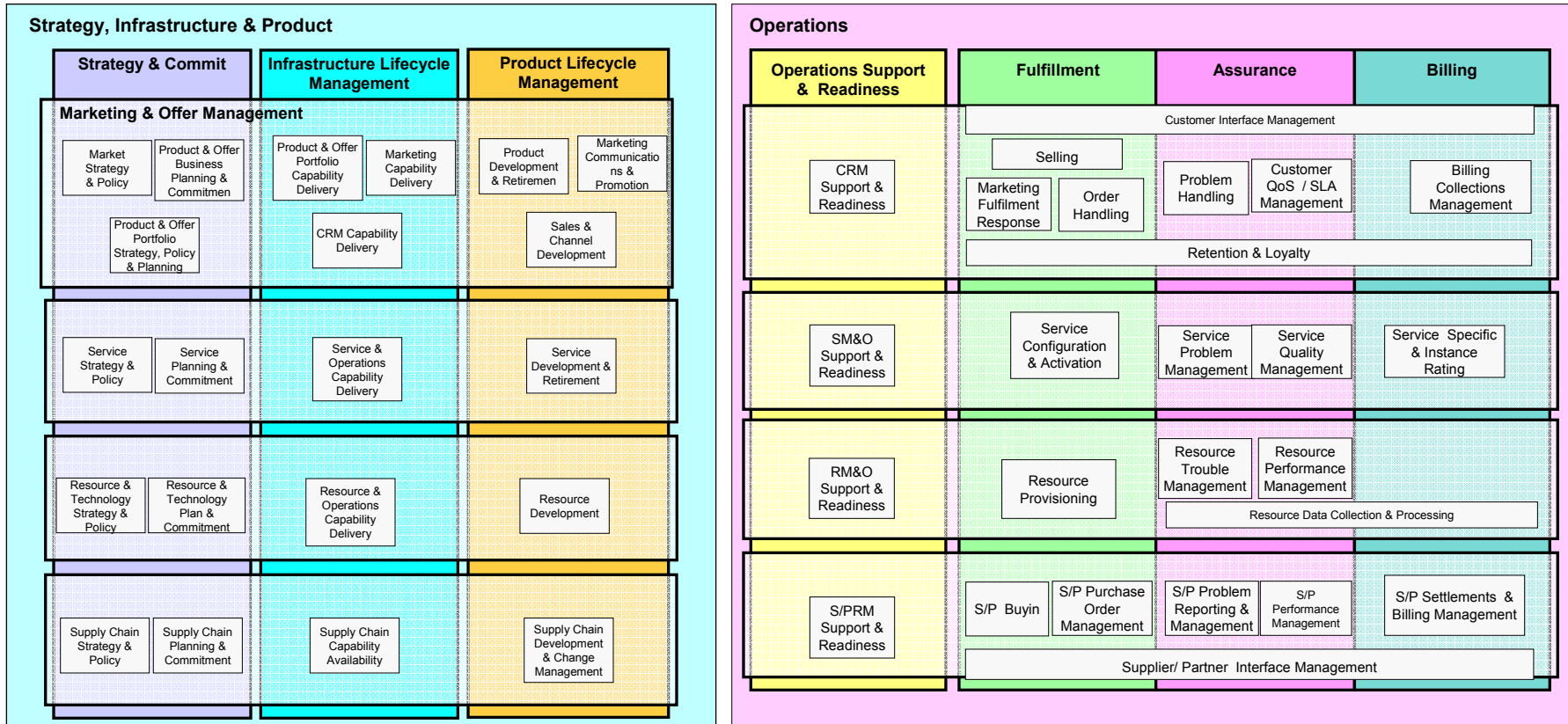
		Customer Relationship	Business & Finance	Compliance, Risk & Quality	Information Management	Solution Development	Solution Value Chain (Supply Chain, Support & Services)		
Responsibilities	Directing	Client Business Development	Business & Finance Strategy	Business Resilience Strategy	Asset & Information Management Strategy	Research & Development Strategy	Value Chain Strategy		
		Market Management	Business Portfolio Management	Regulatory Compliance Strategy	Knowledge Management Strategy	Platform Strategy	Value Chain Planning		
		Channel Strategy	Bus. Enterprise Architecture			Development Capability Management	Value Chain Rules & Policies		
	Controlling	Market Portfolio Management	Manage Standard Business Models	Business Process Compliance	Information Resource Management	Solutions Architecture	Value Chain Management		
		Client profile management	Financial Management	Security, Privacy & Data Protection		Developm. Process Deployment	Demand & Supply Management		
		Opportunity Management	Bus. Process Performance	Integrated Risk Management	Intellectual Property Management	Developm. Program Management	Partner Process Integration		
		Sales Management	IT Management	Regulatory Compliance Mgt.		Release Integration Planning	Partner management		
	Executing	Account & Territory Management	Financial Operations	Implement Compliance Policies	Data & Content Management	Research solution Engineering	Procurement Execution	Services Management	
		Solution Marketing	Bus. Processes & IT Infrastructure	Quality Management	Knowledge Capture & Availability	Solution Component Development	Production Management	Solution Operations & Maintenance	
		Manage Contract Lifecycle	HR Operations	Business Resilience & Disaster recovery	Asset Lifecycle Management	IP Capitalization	Collaborative Design-In	Solution Integration	Warranty Management
			Legal Management & Support			Engineering support	Solution Logistics	Client Inventory Management	

eTom is the Telco "Domain Analysis" reference map

www.tmforum.org

Customer

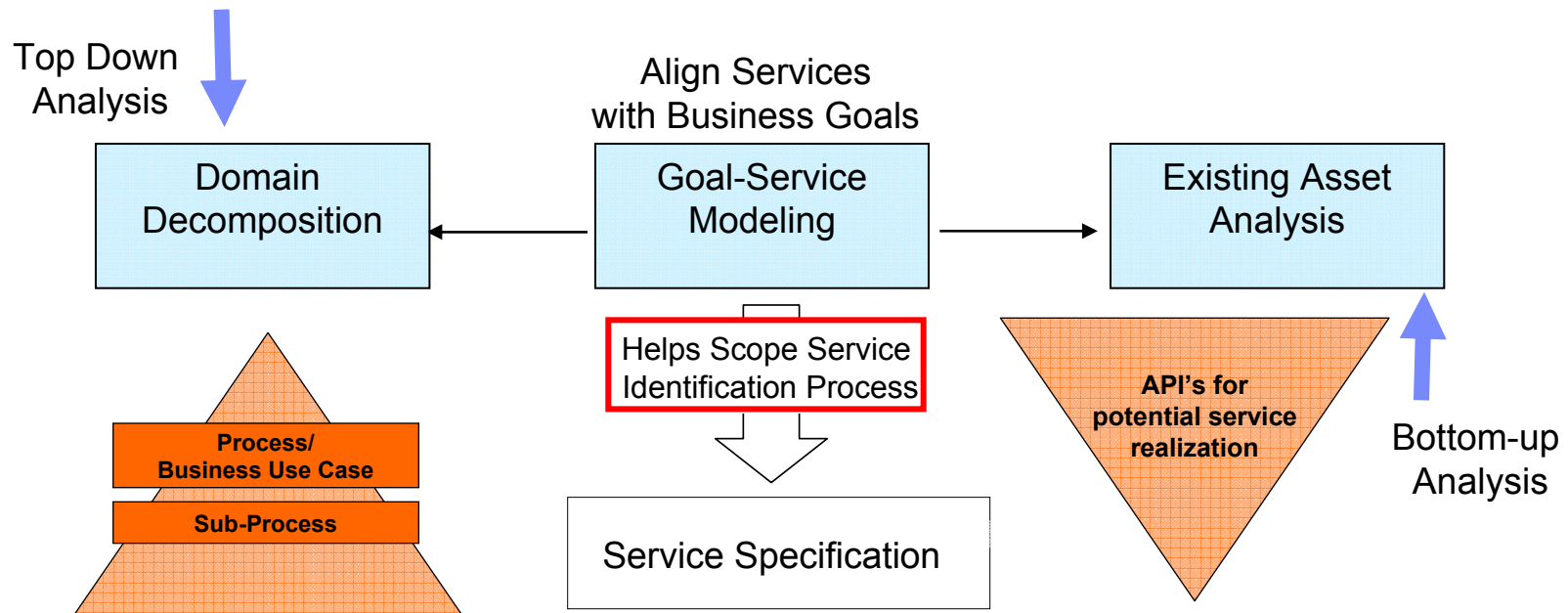
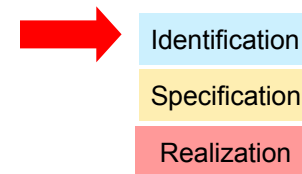
telemanagement forum



Method: SOMA Service Identification

SOMA identifies services through three complementary techniques:

- **Domain Decomposition** (Top-down Analysis)
- **Existing Asset Analysis** (Bottom-up Analysis)
- **Goal-Service Modeling**



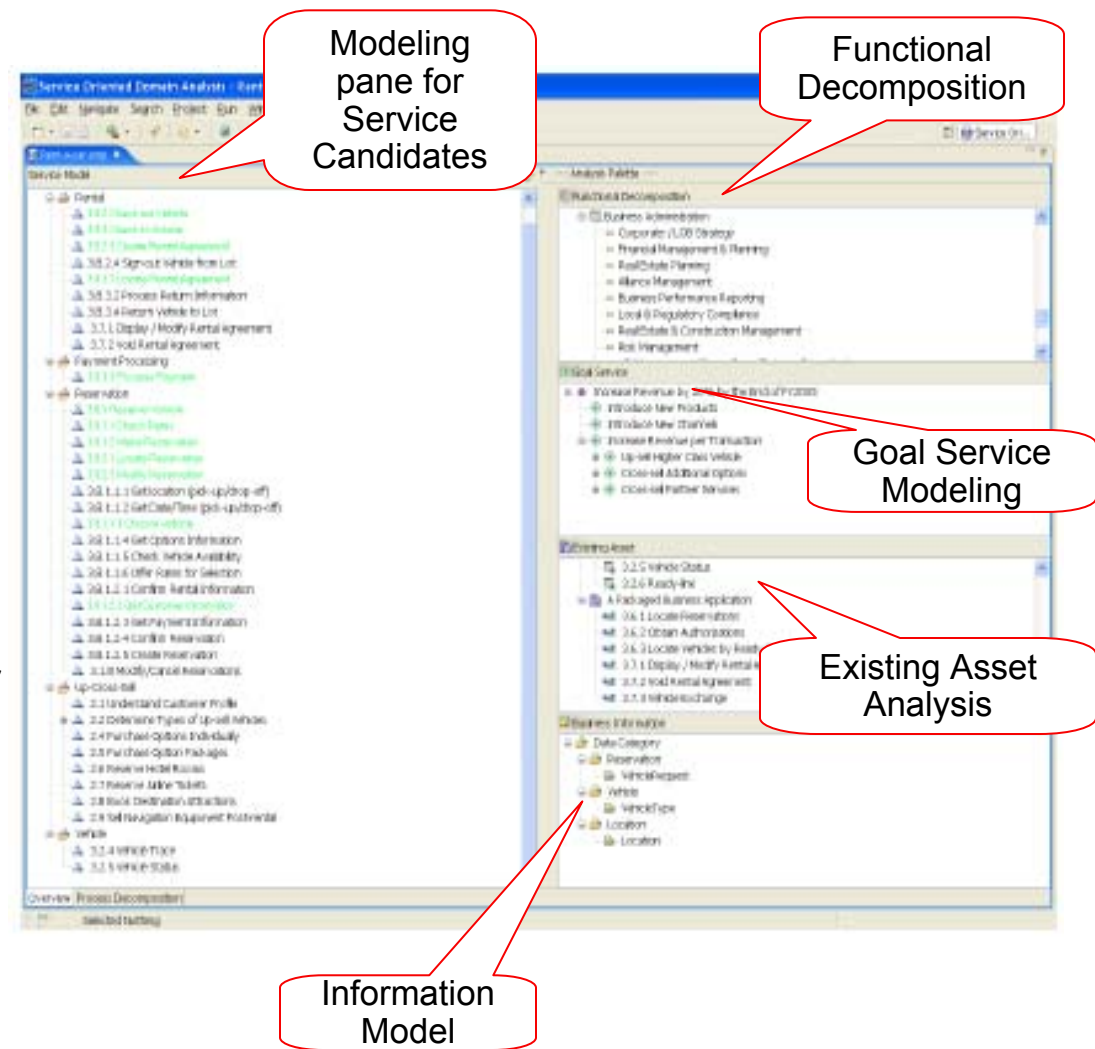
Method support by tooling: Using the SOA Integration Framework (SOA-IF)

SOA-IF Domain Analysis:

- Business Process Analysis
- Goal Analysis
- Existing Asset Analysis
- Model

Capabilities:

- Service-Oriented Modeling and Architecture (SOMA) Service Identification Support
- WebSphere Business Modeler (WBM) Integration
- WebSphere Studio Asset Analyzer (WSAA) support
- Support for Easy Customization of Underlying Meta-Model
- Access to Business Process, Service, and Key Performance Indicator (KPI) Catalogs
- Report Generation
- Export to UML 2.0 (RSA integration)



Case Studies from Telco Industries

- **Study 1:** Ad hoc transformation for fast success
- **Study 2:** eTOM level 4/5 mapping onto Packaged applications (COTS)
- **Study 3:** Bottom up SOA, attempt to transform legacy



Study 1: A path for fast success – Project Goals

- Implement a Pilot to fast path a previously mostly manual business process
- Show immediate business value
- Leverage existing implementations of business functions for services
- Generate baseline information for a potential Enterprise wide role out of SOA Transformation
- Leverage existing infrastructure
- Skill transfer to staff that will go on to transform the organisation

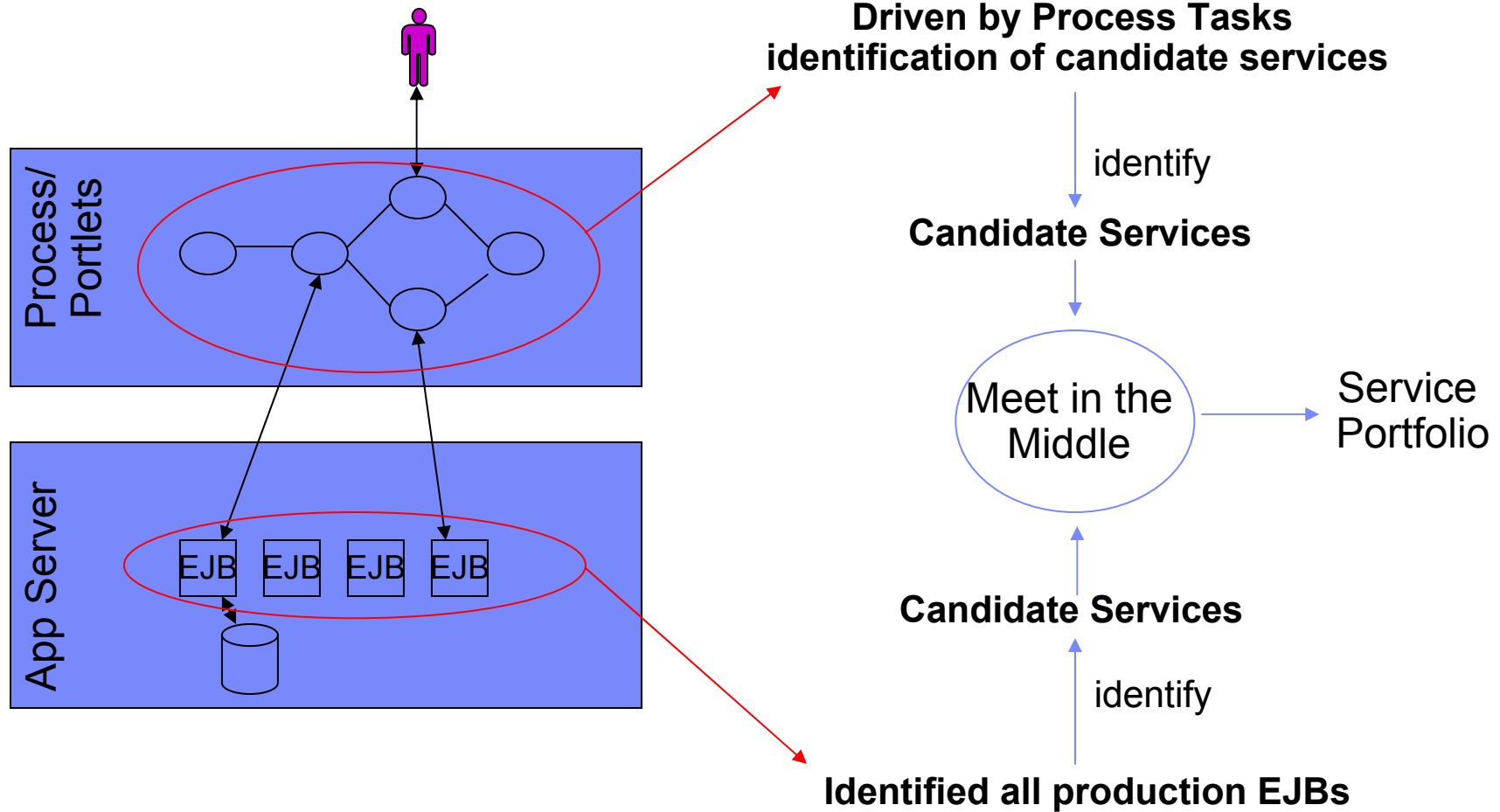


Study 1: A path for fast success - Approach

- Use a Business Process driven approach vs. analysis on Business maps and hot spots
 - ▶ A fragmented manual process was chosen and implemented as human driven business process
 - ▶ A process that has immediate business relevance and has additional value as a service mapping exercise
- Improve the current Process to a TO-BE Process
- Map the process tasks against a set of existing EJBs with the help of Business analysts, i.e develop a “kind of” Service Catalogue
- Show business success by improving process “time to complete” & ease of use (KPI)



Study 1: A path for fast success – Business Process Mapping



Study 1: A path for fast success – The Advantages of the Approach

1. Starting on an existing process accelerates the project right in the execution phase, i.e. very short initiation phase
2. Services are potentially already available requiring only slight modifications
 - relative small development effort,
 - nearly no design phase
3. The application of Method is reduced to an “Acid Test” , i.e. mapping the services to the To Be business processes
4. Skill development can concentrate on the new technology immediately, this keeps the team relatively small and very focused

➤ **It is possible to move quit fast**



Study 1: A path for fast success – The Disadvantages of the Approach

1. Missing alignment with Telco standards if not already reflected in current process
2. Service Catalogue not necessary stable for future since service candidate analysis done on limited scale
3. Application of an end-to-end Methodology missing



Study 1: A path for fast success – Results

- Optimized an existing business process (8 Tasks, 2 sub-processes, 3 Portlets)
- Analyzed and modified existing EJB for Services
- Developed a base line for an potential Enterprise wide role out of an SOA transformation
- Build up skill in particular on the new product stacks (tooling & runtime, monitoring)
- Develop a Business Monitoring (WBI Monitor) model of implemented business process and implemented it
 - ▶ KPIs=13, Aggregate Measures= 23, Metrics=60 defined , Stopwatches=11, Measures=35
- Provides Business value (KPI) by reducing business process duration and throughput
- Went into production in 8 weeks
 - ▶ Products: Application Server, Process Server, Portal, Monitoring, and Tooling



Case Studies from Telco Industries

- **Study 1:** Ad hoc transformation for fast success
- **Study 2:** eTOM level 4/5 mapping onto Packaged applications (COTS)
- **Study 3:** Bottom up SOA, attempt to transform legacy



Study 2: eTOM to COTS – Project Goals

- Provide a standard-based service model for a key business process in the Telco service fulfilment area in order to achieve the following benefits:
 - Abstraction of current COTS packages in order to support plug-and-play them in the Telco infrastructure
 - Provisioning of a business analyst view on standardized business processes implemented over a service model with a shared data model
 - Establishment of an industry standard compliant service catalogue of reusable services



Study 2: eTOM to COTS – Approach

- A model and method driven approach is taken:
 - Modelling of business process starts from eTOM/SID/TAM (with extensions)
 - SOMA is used for service modelling
- Separation of business and technical concerns:
 - Shield standardized business activities (business process and services) from current IT implementation (service components and application packages)
- Service modelling:
 - Performing a top-down approach using eTOM and TAM to categorize and identify services
 - Analyzing the to-be installed COTS packages
 - Perform a bottom-up approach using these COTS packages to realize the services



Study 2: eTOM to COTS –Service portfolio in Domain Analysis tool

The screenshot displays the Service Oriented Domain Analysis tool interface. On the left, a 'Service Model' tree lists various service categories such as 'Market / Sales', 'Product Management', and 'Customer Management'. A large green circle highlights this tree with the text 'Service Portfolio aligned with TMF TAM down to level 5'. On the right, an 'Analysis Palette' shows a 'Functional Decomposition' tree with a green circle around it and the text 'eTOM map decomposition Using the standard GB921D level 3 as a base'. Below this, a 'Goal Service' table is visible, featuring an IBM logo and a grid of colored cells representing different service goals. At the bottom, a 'Business Information' tree lists various service items, with a green circle around it and the text 'Full TMF SID information model for exposing services'.

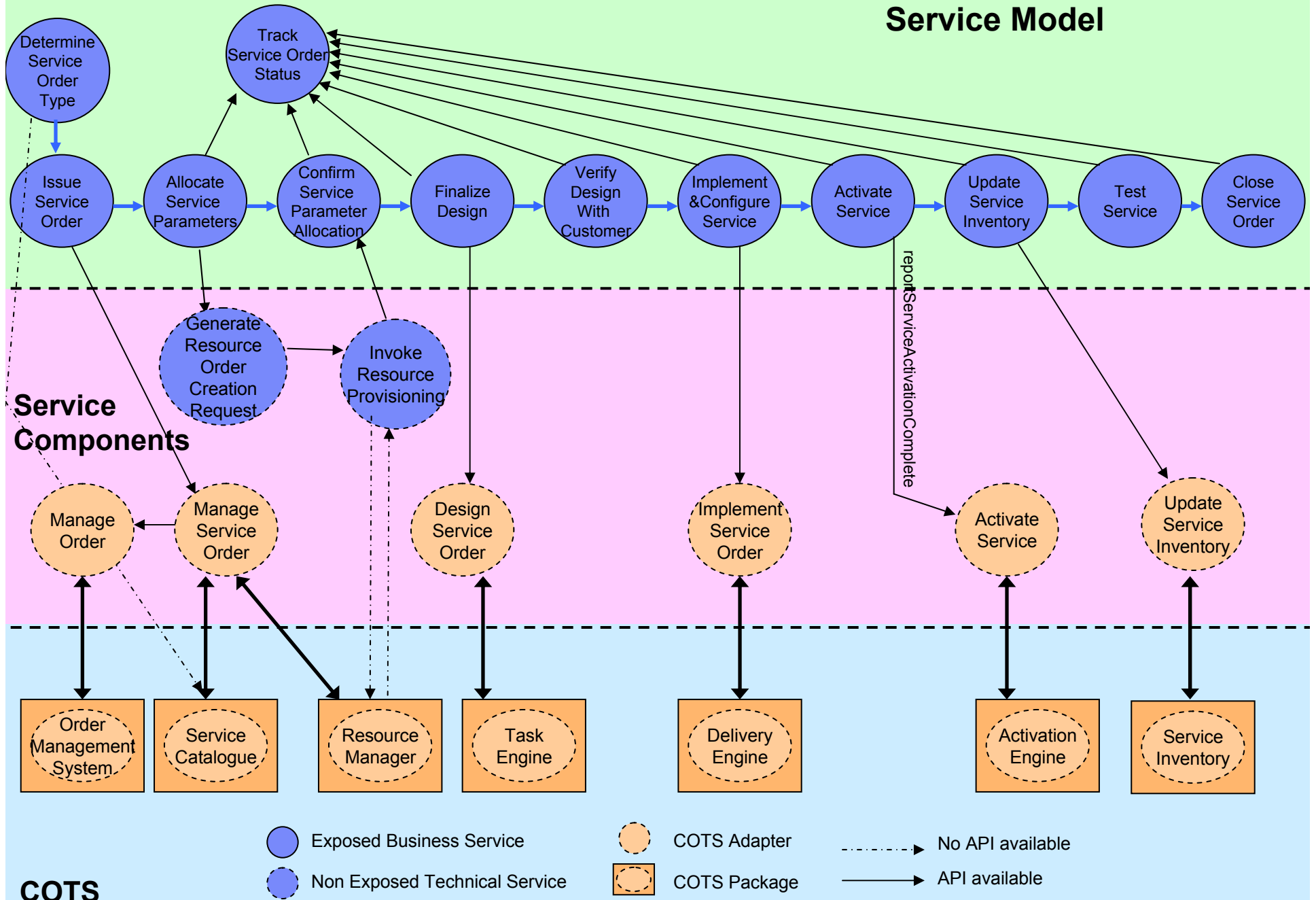
Service Portfolio aligned with TMF TAM down to level 5

eTOM map decomposition Using the standard GB921D level 3 as a base

Full TMF SID information model for exposing services

Services

Service Configuration & Activation Service Model



Study 2: eTOM to COTS – Challenges & Solution Pattern

- Service realization difficult (i.e. mapping services to COTS packages):
 - COTS packages are too coarse-grained
 - COTS packages offer too few APIs
 - Internal packages of COTS packages can't be externally controlled (but offer events to report status progress)
- Solution patterns:
 - Aggregating service (to reflect coarse-grained COTS packages)
 - Wrapper service components using COTS events to control process flow
 - Req: Refactor COTS packages (finer grained COTS components and/or more external APIs)



Study 2: eTOM to COTS – Results

- Service modelling using SOMA:
 - Service identification using top-down and bottom-up approach
 - Refined Service specification using the Service Litmus Test (SLT)
 - Service realization by mapping the identified services to COTS packages
- Layering architected:
 - Standard Model-based business processes (i.e. eTOM & SID with extensions) and Services shield from current implementation
 - Services with standard eTOM service interfaces exposed (i.e. WSDLs with eTOM-based operations and SID-based Business Objects)
 - Delivered standardized Service Catalogue
 - Service components to abstract COTS adapters (i.e. realizing COTS package neutral interfaces for the service layer invocation and performs the mapping to the COTS low level APIs)



Case Studies from Telco Industries

- **Study 1:** Ad hoc transformation for fast success
- **Study 2:** eTOM level 4/5 mapping onto Packaged applications (COTS)
- **Study 3:** Bottom up SOA, attempt to transform legacy



Study 3: Legacy Transform – Project Goals

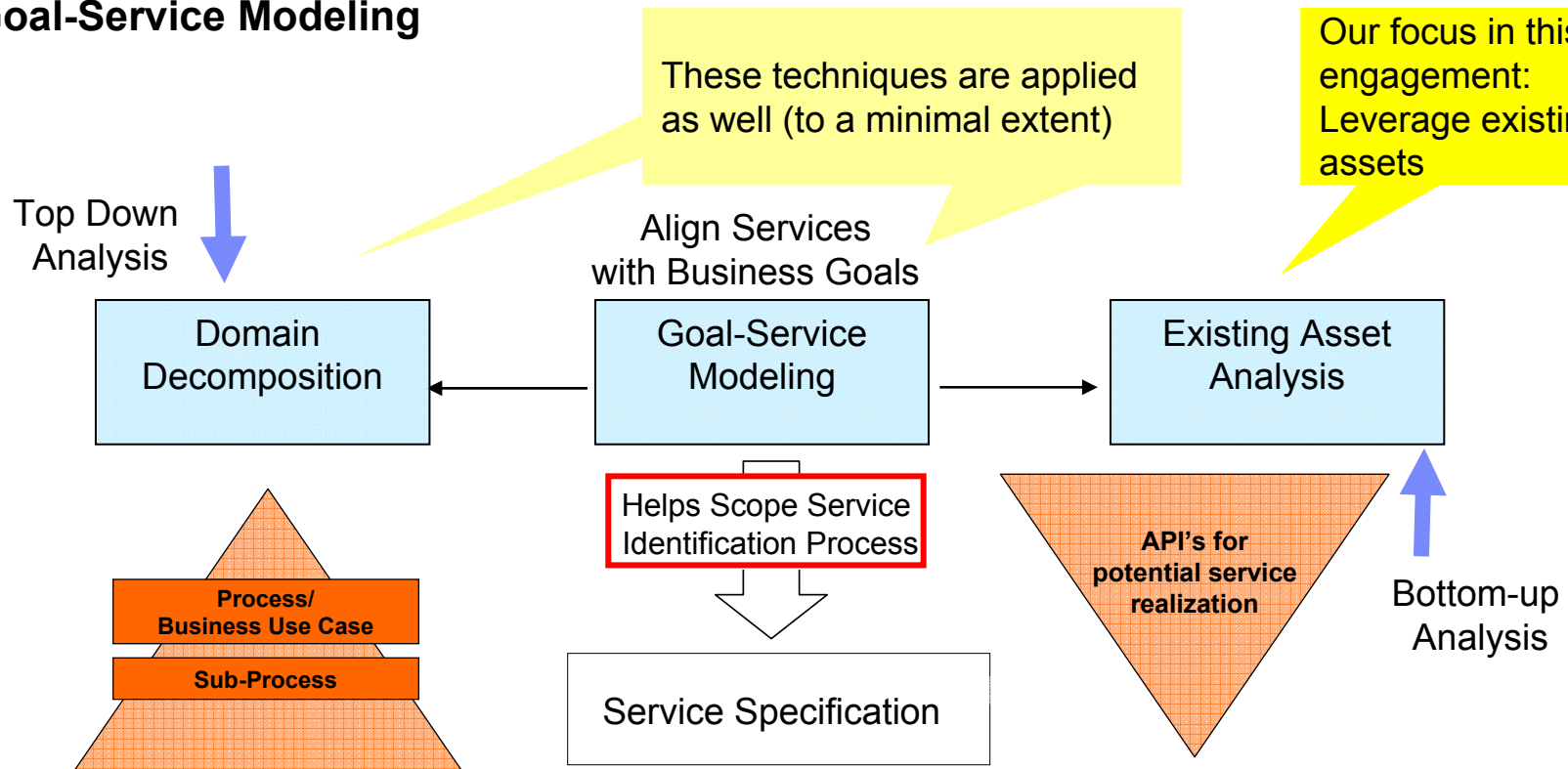
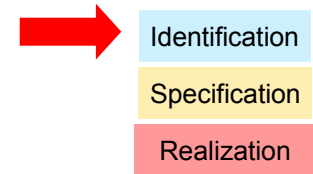
- Develop an SOA concept for **Product Data Management (PDM)** [part of Product Lifecycle Management (PLM)]
- Perform a **bottom-up approach** from the existing IT environment (mostly z/OS, DB2, COBOL/C++)
- Document and assess all PDM-relevant **APIs**
- Provide a list of **recommended PDM services**, based on current IT, APIs, internal usage experience and industry standards
- Propose **naming conventions** for services
- Develop a **service versioning concept** and suitable tool support
- Show the benefit of a PDM SOA and plan next steps



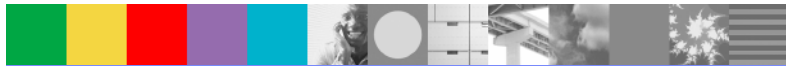
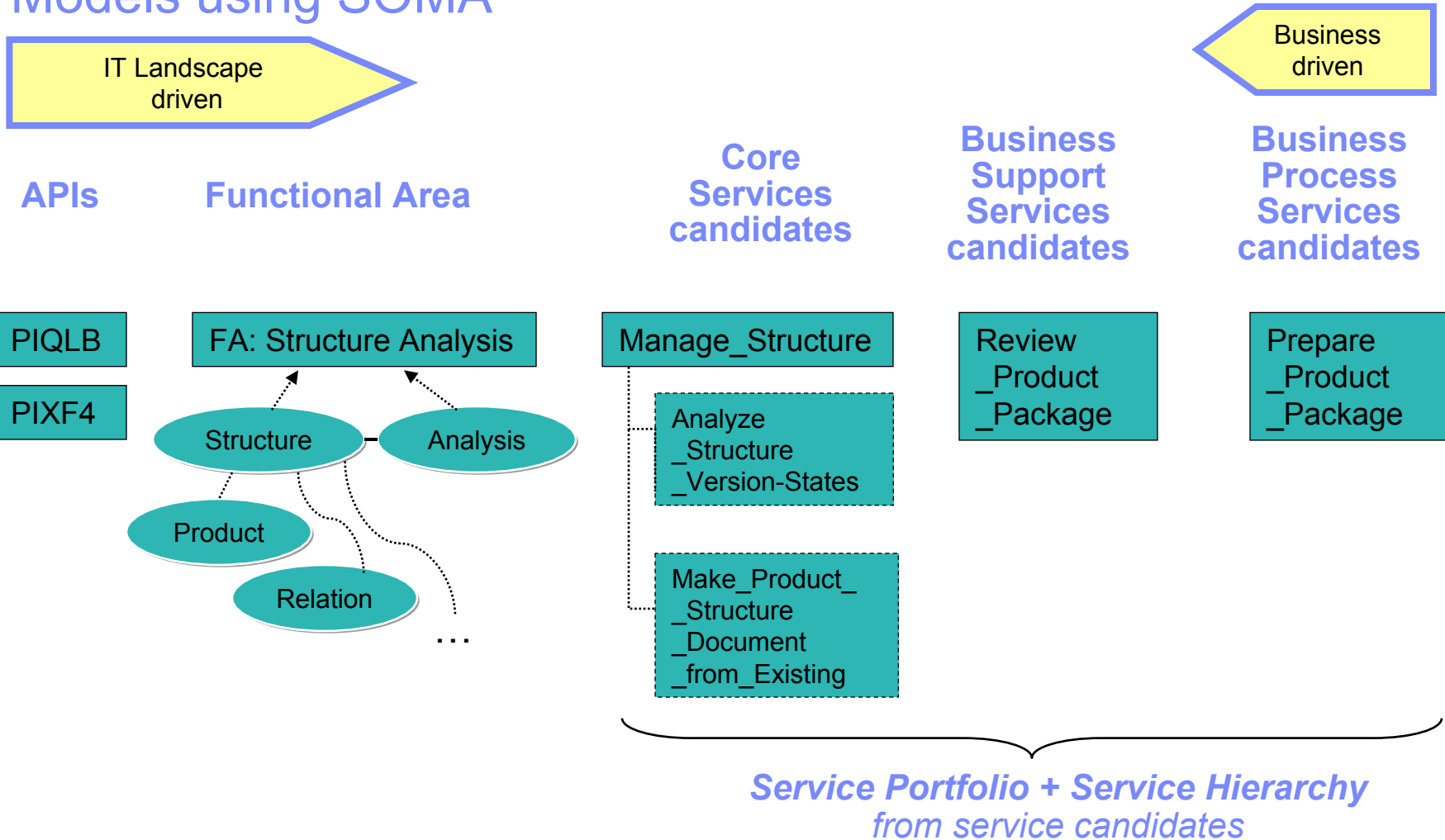
Study 3: Legacy Transform - Service Identification

SOMA identifies services through three complementary techniques:

- **Domain Decomposition** (Top-down Analysis)
- **Existing Asset Analysis** (Bottom-up Analysis)
- **Goal-Service Modeling**



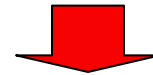
Study 3: Legacy Transform - Solution approach Bottom-up Service Identification extended by Process and Industry Models using SOMA



Study 3: Legacy Transform - Methods / Products / Tools used

Project Goals and Scope:

- Develop an SOA concept for **Product Data Management (PDM)** [part of Product Lifecycle Management (PLM)] — **ISD, SOMA**
- Perform a **bottom-up approach** in the existing IT environment (mostly z/OS, DB2, COBOL/C++) — **SOMA**
- Document and assess all PDM-relevant **APIs** — **APM (Application Portfolio Management)**
- Provide a list of **recommended PDM services**, based on current IT, APIs, internal usage experience and industry standards — **SOMA, SOA-IF, WBM, APM, Industry standards (OAGIS, OASIS, PLCM, ...)**
- Propose **naming conventions** for services — **UML, WSRR**
- Develop a **service versioning concept** and suitable tool support — **UML, WSRR**
- Show the benefit of a PDM SOA and plan next steps



Study 3: Legacy Transform - Building the Service Hierarchy in SOA-IF

The screenshot displays the 'Service Oriented Domain Analysis - PDM_SOA1.smp - Domain Analysis' interface. The left pane shows a 'Service Model' tree with categories like 'PDM Service Hierarchy', 'Authorities Management', 'Information Maintenance', 'Product Maintenance', 'Manage Product Configuration', 'Product Derivatives', 'Publication & Subscriptions', and 'Analysis'. The right pane shows 'Functional Decomposition' with a tree structure including 'Solution management', 'Solution Engineering', and 'Goal Service'. Below this is 'Existing Asset' analysis showing 'PDM relevant APIs' and 'Business Information'.

Business Process Services (points to Authorities Management)

Process support Services (points to Information Maintenance)

Core Services (points to Product Maintenance and Manage Product Configuration)

Functional Decomposition: Starting from CBM competencies down to (reference) process structures (E2E) (points to Solution Engineering)

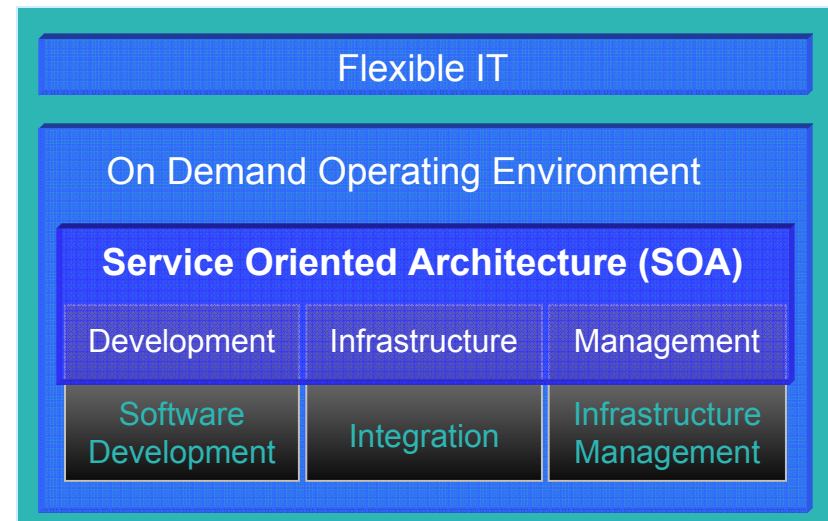
Goal Service Modeling: Business goals, sub-goals and KPIs (points to Business Goals)

Existing Asset Analysis: Captured ~300 APIs in various systems, as potential basis for services and to ensure modeling completeness (points to Existing Asset)

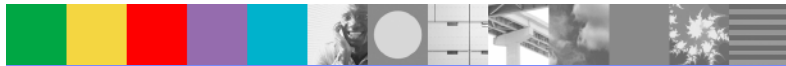
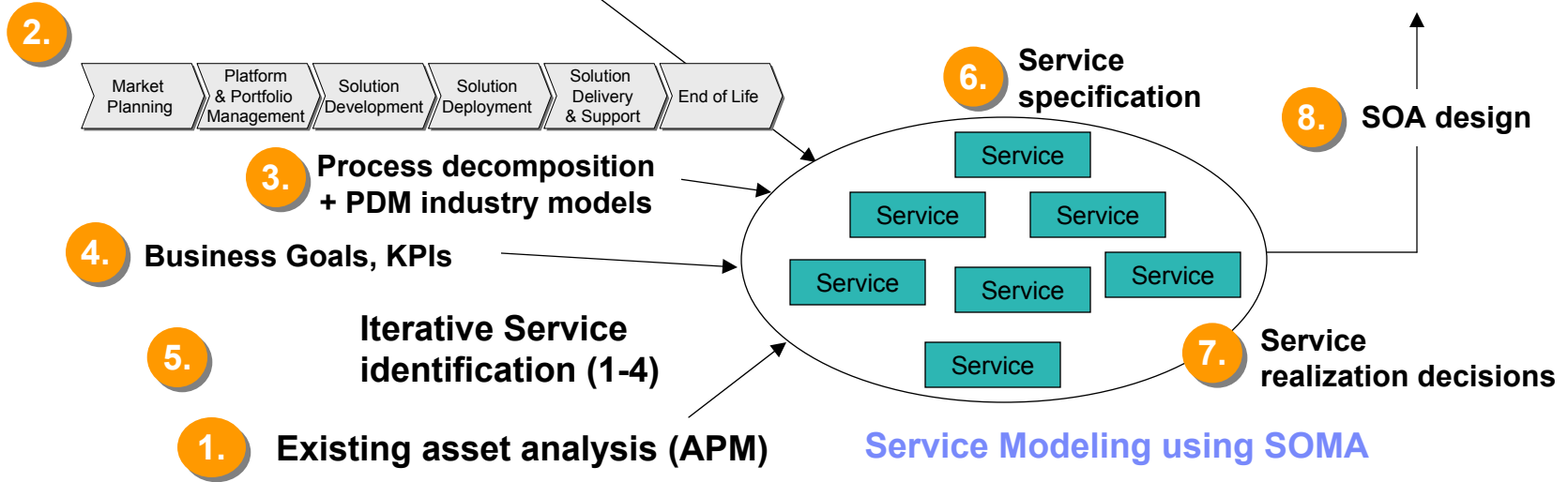
PDM Information Model (points to Business Information)

Study 3: Legacy Transform - Overall approach

	Customer Relationship	Business & Finance	Compliance, Risk & Quality	Information Management	Solution Development	Solution Value Chain (Supply Chain, Support & Services)
Directing	Client Business Development	Business & Finance Strategy	Business Resilience Strategy	Asset & Information Management Strategy	Research & Development Strategy	Value Chain Strategy
	Market Management	Business Portfolio Management	Regulatory Compliance Strategy	Knowledge Management Strategy	Platform Strategy	Value Chain Planning
	Channel Strategy	Bus. Enterprise Architecture	Enterprise Resource Strategy	Information Resource Management	Solutions Architecture	Value Chain Rules & Policies
Controlling	Market Portfolio Management	Manage Standard Business Models	Business Process Compliance	Security, Privacy & Data Protection	Develop. Process Deployment	Value Chain Management
	Client profile management	Financial Management	Integrated Risk Management	IT Management	Develop. Program Management	Demand & Supply Management
	Opportunity Management	Bus. Process Performance	Regulatory Compliance Mgt.	Intellectual Property Management	Release Integration Planning	Partner Process Integration
	Sales Management	Human Resources Management	Implement Compliance Policies	Data & Content Management	Research	Partner Management
Executing	Account & Territory Management	Financial Operations	Quality Management	Knowledge Capture & Availability	Solution Engineering	Services Management
	Solution Marketing	Bus. Processes & IT Infrastructure	Business Resilience & Disaster recovery	IP Capitalization	Solution Component Development	Production Management
		HR Operations	Asset Lifecycle Management	Asset Lifecycle Management	Collaborative Design-In	Solution Integration
				Engineering Support	Solution Logistics	Client Inventory Management



CBM import



Study 3: Legacy Transform - Results

- **Existing PDM APIs** (~ 300) documented with APM method
- **CBM Model for Electronics** imported into SOA-IF
- Domain decomposition based on the CBM competencies
- **Industry models** (from IBM ISD) contributed
- Domain decomposition refined with **models of key processes** (in WBM), e.g. Engineering Change, Commercial Product Packaging.

- Elicited business goals, sub-goals and KPIs

- Business information pane used for Product Data Model

- **PDM service hierarchy** created: 3 levels
 - Guidance to expose higher-level business services
 - Clearly shows how existing APIs can be used in service realizations (unique / redundant / overlapping / missing APIs)

- Issues solved:
 - Initial reluctance of sponsor to share business information (e.g. business goals)
 - “Cultural move” from fine-grained APIs to coarse-grained business services



Lessons Learned from the Studies (the narrow view)

- Could show fast transformation success by a straight forward mapping of Services to EJBs for a human driven process in production, serving as a base line for full Enterprise role out
- Could show results to encapsulate COTS packages and expose their Services in an SOA way by offering standard services and standard process artifacts, avoiding typical silo issues
- Could show that legacy transformation to Service Catalogs and an alignment to Business needs is feasible



Conclusion (the broader view)

- ▶ Studies show that **powerful** methods (CBM, SOMA, APM, ...), tools, and products for SOA are the key to an Enterprise reengineering success
 - **Method adoption** provides flexibility: focus on business/process-driven, top-down aspects or technically-driven, bottom-up aspects or other approaches
 - **Products and tools** speed up the method usage considerably, e.g. SOA-IF, RSA, RMC (Rational Method Composer) plug-ins, i.e. RUP-SOMA
Also Source Analyzer tooling to help decompose existing SW packages

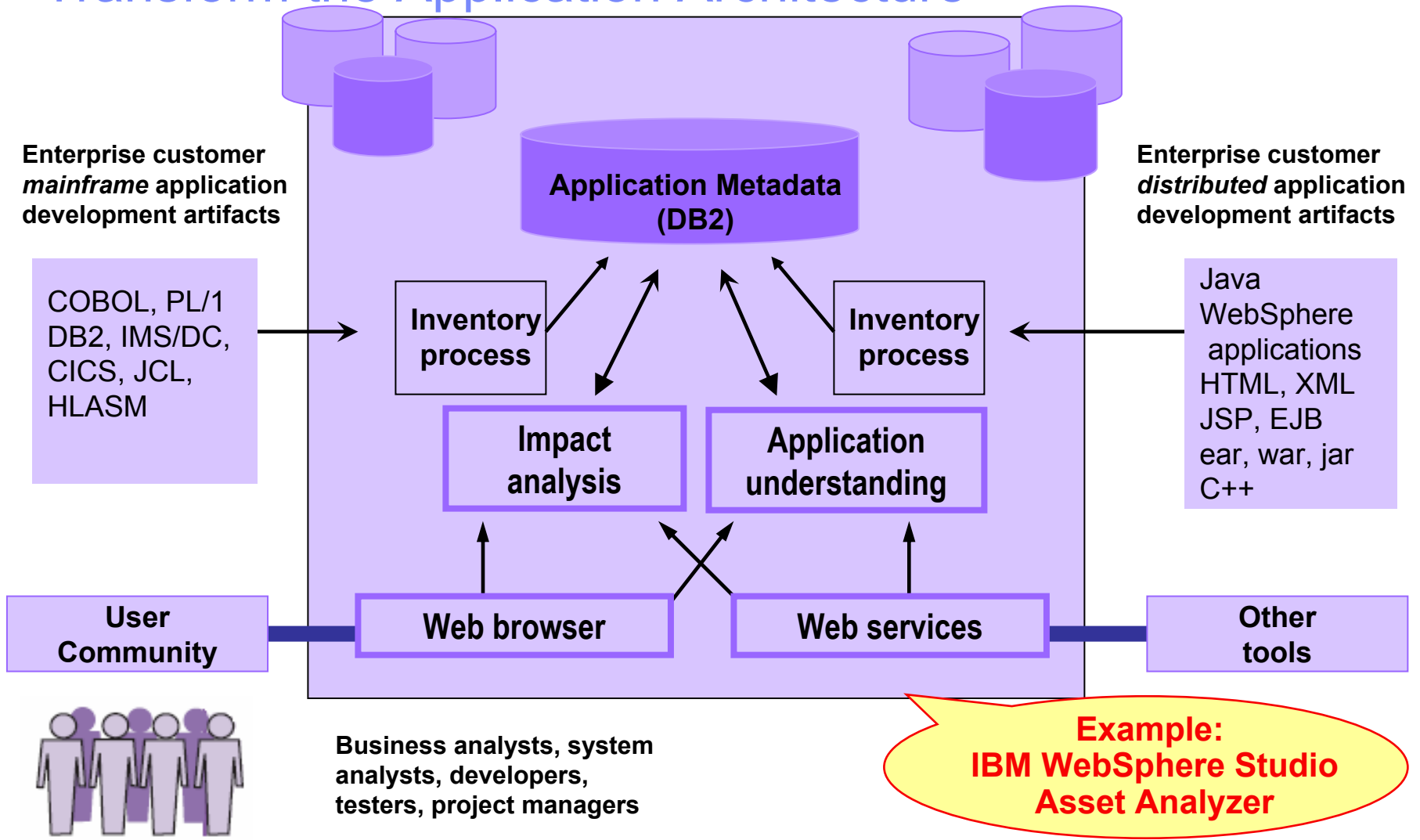
AND not to forget....

- **SOA Governance** issues have been raised from each SOA project
 - ▶ Follow-on work usually includes guidelines, roles and responsibilities for SOA design, implementation and management decisions.
 - ▶ Software support by Registry & Repository functionality



Back-up

Powerful Source Analyzers help Understand and Transform the Application Architecture



More information on SOMA ...

- [Arsanjani 2004]
A. Arsanjani: Service-Oriented Modeling and Architecture.
<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>
November 2004.
- [IBM 2006]
IBM RUP for Service-Oriented Modeling and Architecture V2.4
(Rational Method Composer plug-in)
http://www-128.ibm.com/developerworks/rational/downloads/06/rmc_soma/
November 2006.



General Lessons: Avoiding Pitfalls (1)

1. **Beware of vendor proprietary service offerings.** Do not get locked into SOA vendor offerings that are proprietary in nature; you could lose the interoperability and flexibility benefits of a true SOA.
2. **Seek stability in the use of open standards.** The latest open standard specification in the industry is not always the most stable; as a result, it may not be mature enough for adoption.
3. **Carefully assess your legacy modernization.** Take a holistic view of the enterprise when choosing particular legacy systems for modernization. Silo approaches for SOA transition may create redundancy.
4. **Avoid "waterfall" development and lack of service versioning.** SOA transition should be iterative in nature. A service life-cycle management should possess the capability to maintain multiple versions of a service.
5. **Know the technical constraints of your legacy system.** Consider all the technology limitations of a legacy system before jumping ahead into a legacy modernization effort.
6. **Don't equate SOA with Web services.** Acknowledge the difference between SOA (an architectural style) and Web services (a set of standards for SOA implementation).
7. **Avoid the silo approach to service creation and ownership.** Understand the paradigm shift between traditional application development and an SOA-based development.



General Lessons: Avoiding Pitfalls (2)

8. **Steer away from the use of fine-grained services.** A service is a higher-level abstraction than fine-grained application program interfaces (APIs). Services should be coarse-grained and business aligned.
9. **Avoid point-to-point invocation.** Make an SOA ecosystem manageable and loosely coupled. Bring in a mediation layer that handles service discovery, invocation, and neutralizes underlying technical differences between different SOA implementations.
10. **Avoid lack of adherence to standards.** Adopt stable and proven industry-specific standards. This approach will bring in interoperability benefits for your SOA.
11. **Use redundant data stores.** Concentrate on a data consolidation strategy. Mask the data redundancy by creating a virtualized data service.
12. **Stay away from using a "Big Bang" approach.** For complex SOA transitions, forget a Big Bang approach to the finish line. Acknowledge and respect that a smooth SOA transition is best achieved by adopting an iterative approach.
13. **Allocate service ownership.** Do not orphan a service. Give it a home and make a line of business its owner. This ownership allows someone to be responsible to maintain the nonfunctional qualities of your services.
14. **Institute SOA governance.** Empower a governance body to manage the entire service life cycle.

→ *Tilak Mitra in DeveloperWorks:*
<http://www-128.ibm.com/developerworks/library/ar-soapit/>

