IBM HRL

# Model-Driven Development for Service-Oriented Applications

Vladimir Shcherbina

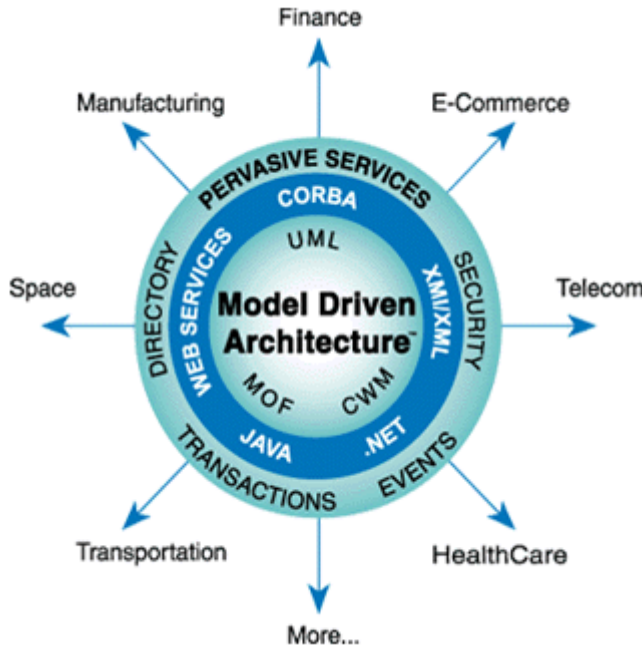
28/05/2007

# Agenda

- **MDA**
- **SOA and SOMA**
- **SOMA Evolution**
- **SOMA Latest Development**
  - SOMA-ME
  - CBS/CBA
  - Integration and Transformation
  - RS4C/PANDOORA
  - IBM's end-to-end SOA Method
- **Conclusions**

# Model Driven Architecture is…

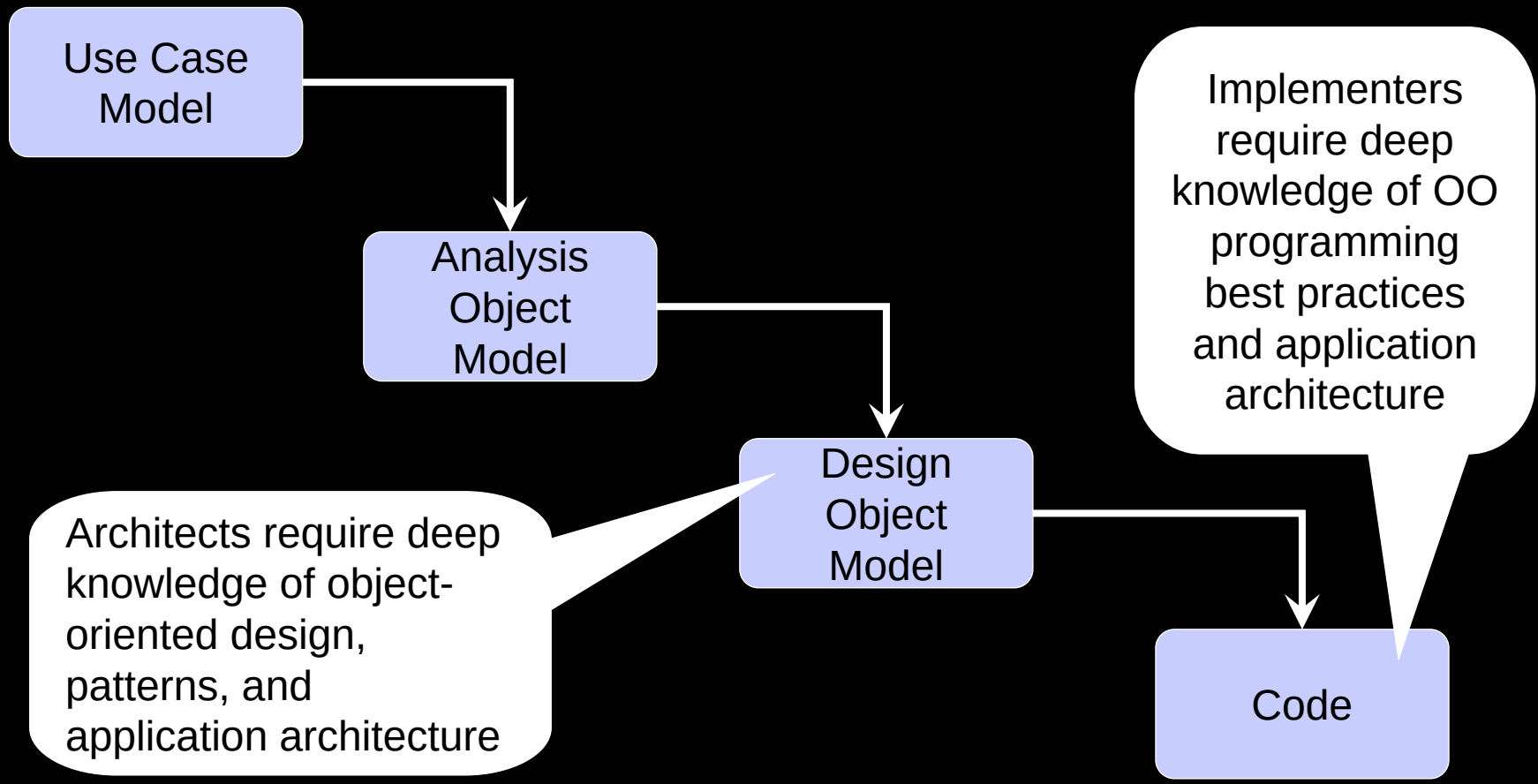
- A standard specification defined by a consortium of companies, managed by the OMG
- A set of technologies that enables you to automate the application of standards throughout the development lifecycle
  - Automate the creation of a Design Model from an Analysis Model
  - Automate the creation of Code from a Design Model, etc.
- A technical solution to keep the Domain Model of a project independent of the hardware and software technologies upon which it is built, so that the Domain Model can be rapidly redeployed to emerging technologies



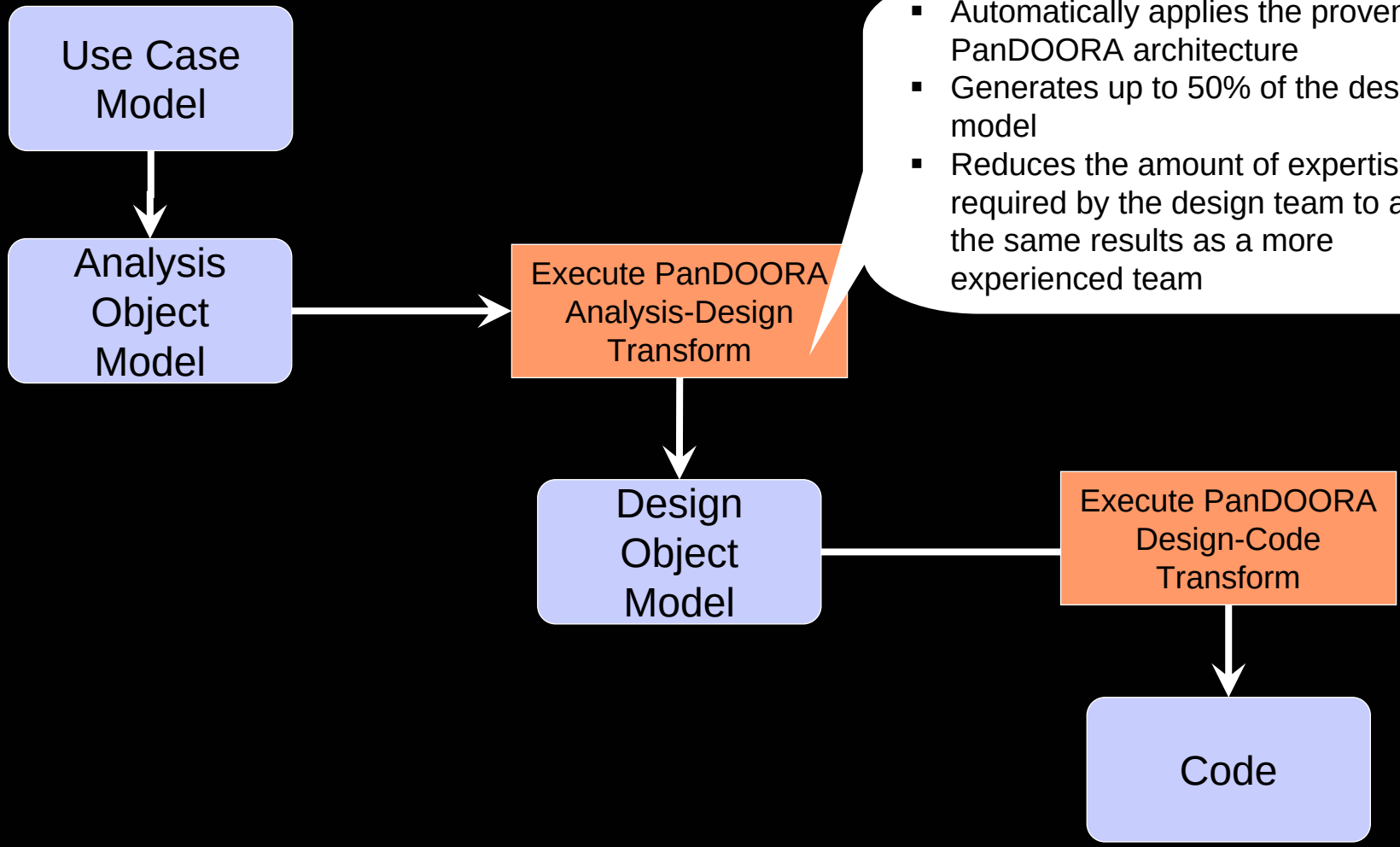
# Primary Concepts of MDA

- **CIM (Computation Independent Model)**
  - The traditional domain model, independent of architectural component structures
- **PIM (Platform Independent Model)**
  - The architectural models (micro design) to support design concepts/best practices independent of the implementation technology links
- **PSM (Platform Specific Model)**
  - The micro design typed for the specific implementation technologies
  - Some also view this as the actual forward engineered code
- **Model Transforms (or Transforms for short)**
  - Process (automated executable) for converting one model to another model for the same system
  - For example, a transform to generate the Design Model (PIM) from the Analysis/Domain Model (CIM)

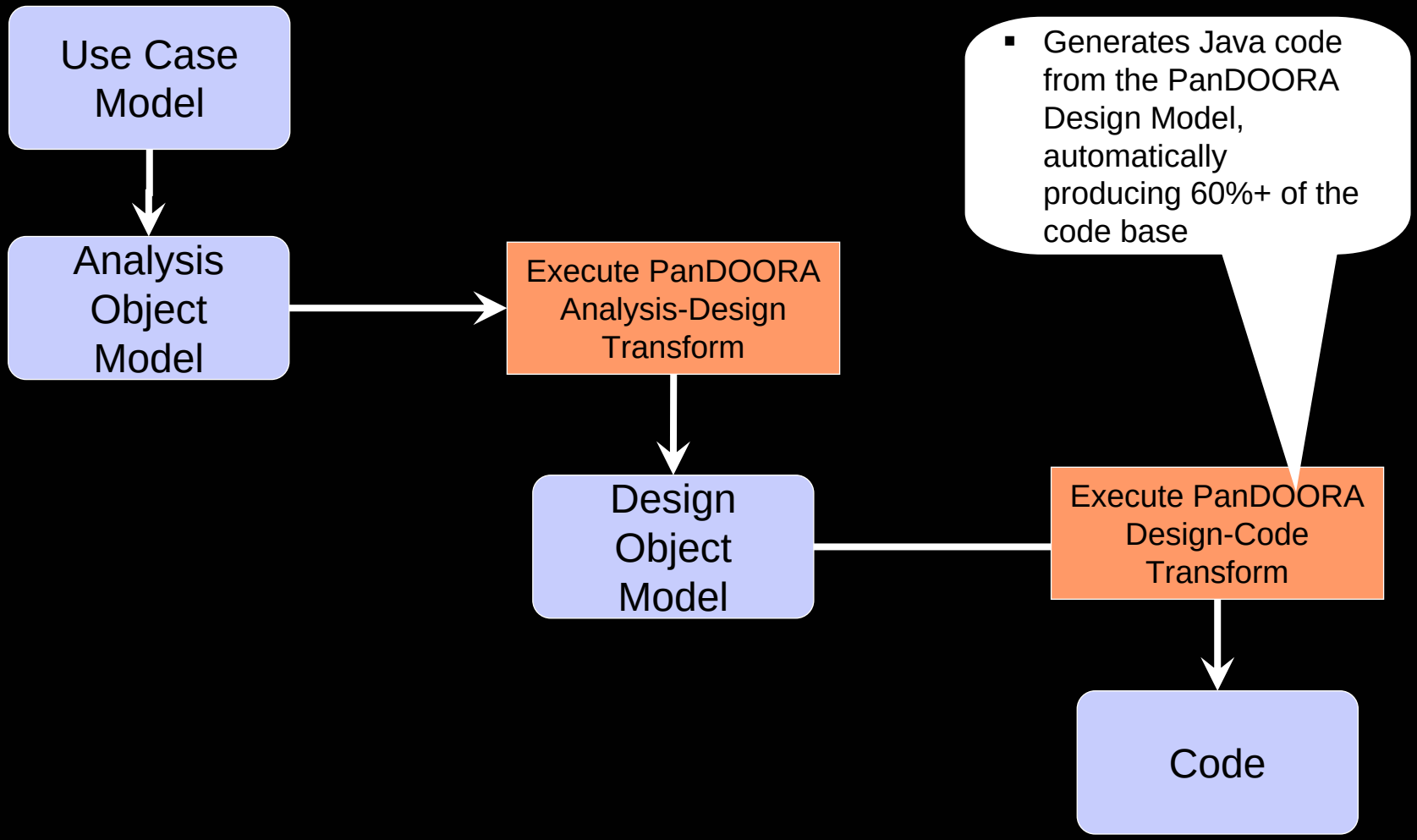
# Without MDA work products are manually produced



# MDA automates the application of architecture standards in Design

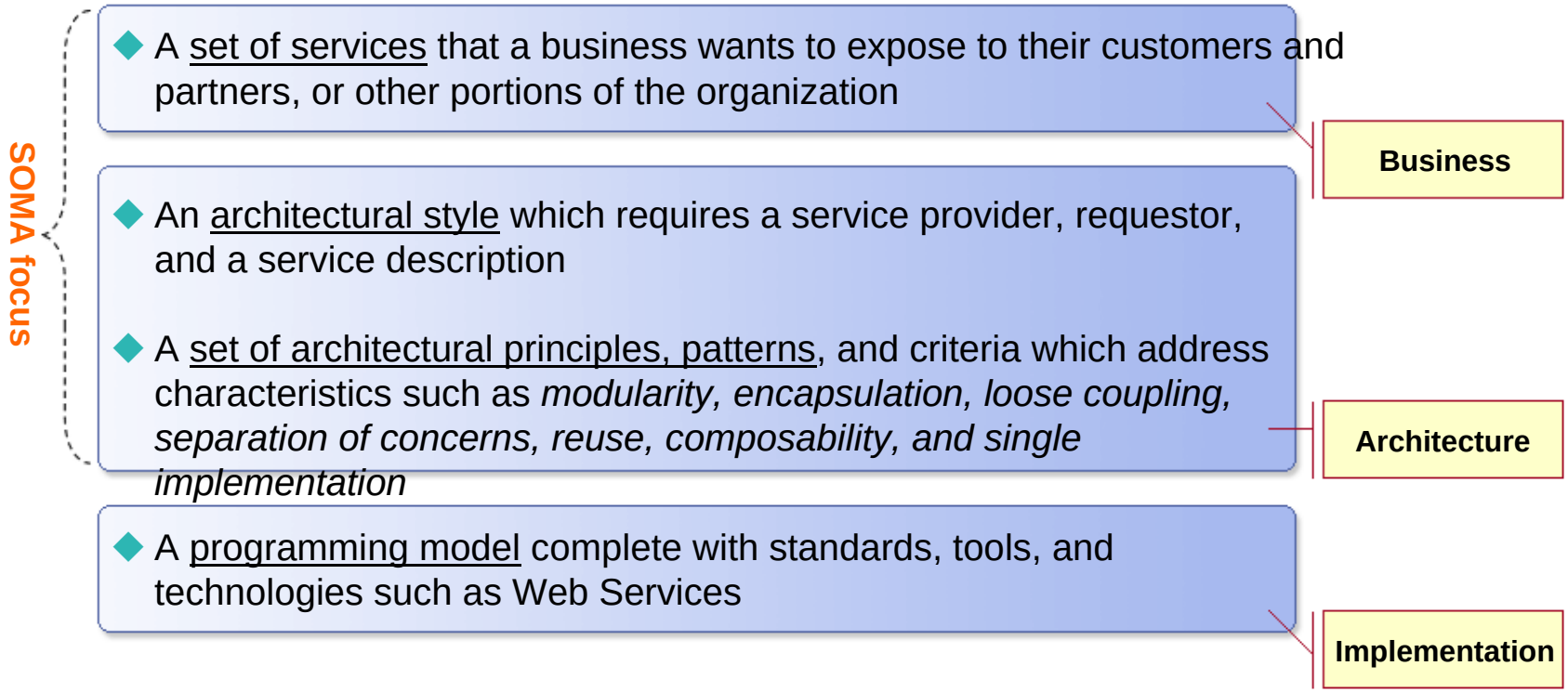


# MDA enables more code generation than XDE/Rose provided in the past

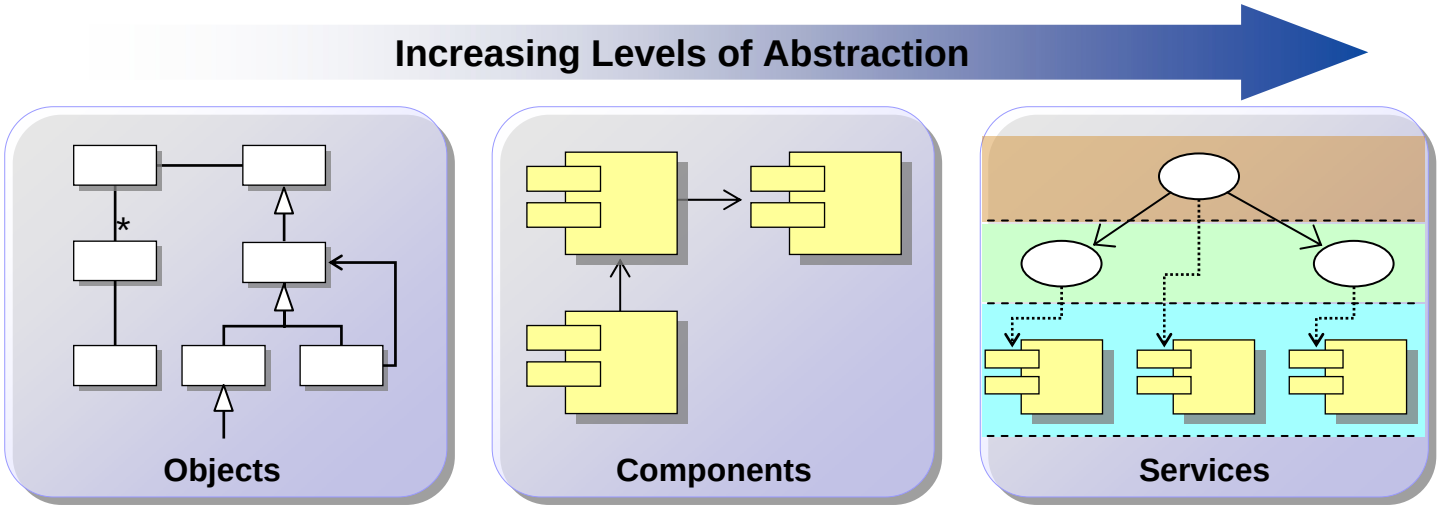


# SOA Various Perspectives Focus on Different Attributes

*"SOA in context …"*



# Service-Oriented Modeling Objectives




Just as OOAD is necessary to define object-oriented systems and component-based development is used to define component-based architectures, **service-oriented modeling is necessary to define a service-oriented architecture**.

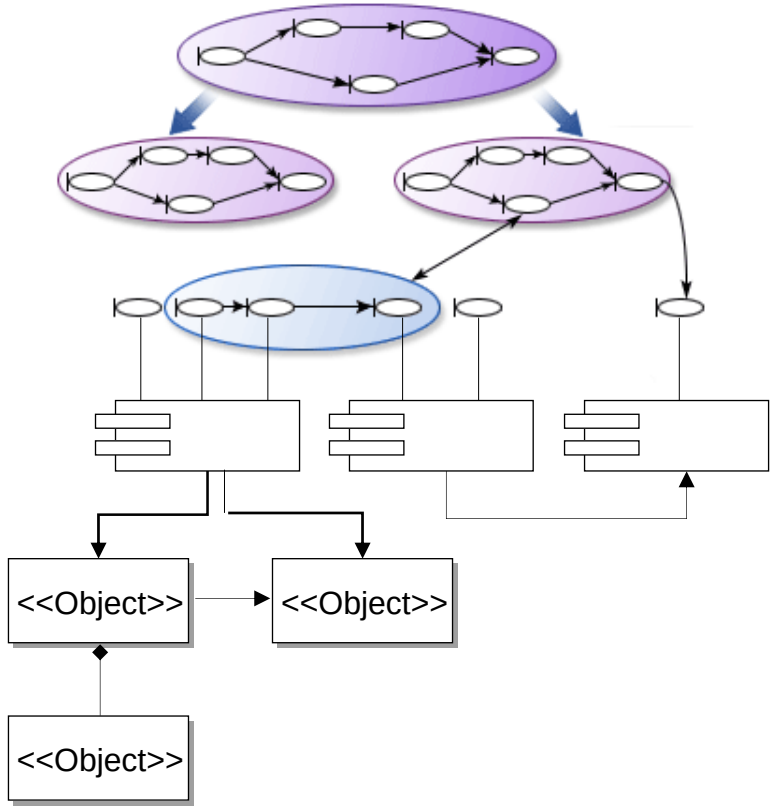
While SOA **builds on** well-established software architecture principles (for example, information hiding, modularization and separation of concerns), it also **adds** additional aspects; thus, service-oriented modeling needs additional techniques for these new aspects.

# SOA Modeling Constructs

**Business Flows**
**(Processes, Orchestration, Choreography)**

**Services**
**Atomic and Composite**

**Service Components**



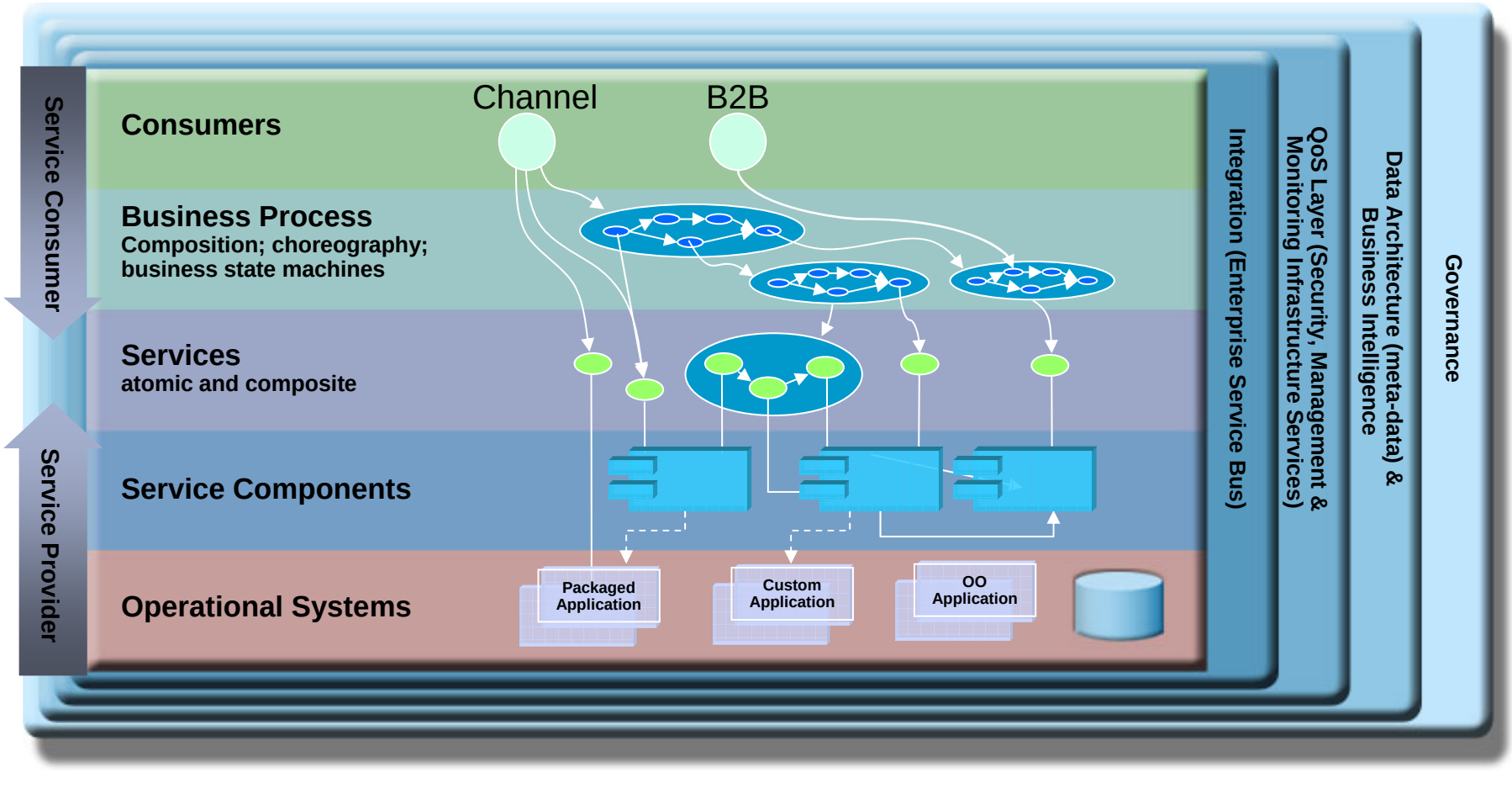

SOMA was created to specifically address modeling (analysis, identification, and specification) of all three constructs.

# Limitations of Traditional Approach

Traditional methods do not address **SOA key elements**: services**,** flows, and components realizing services, which means:

- Explicitly address the identification, specification and realization of services, their flows and composition
- Appreciate the distinct requirements of 2 key roles in a SOA: the service provider and service consumer
- Applications assumed to be built for one enterprise must be exposed to business partners who might compose, combine, and encapsulate them into new applications

# SOA Reference Model
## Solution View



# SOMA Activities Are Grouped into Three Major Steps

**At the heart of SOMA is the identification and specification of services, components, and flows.**

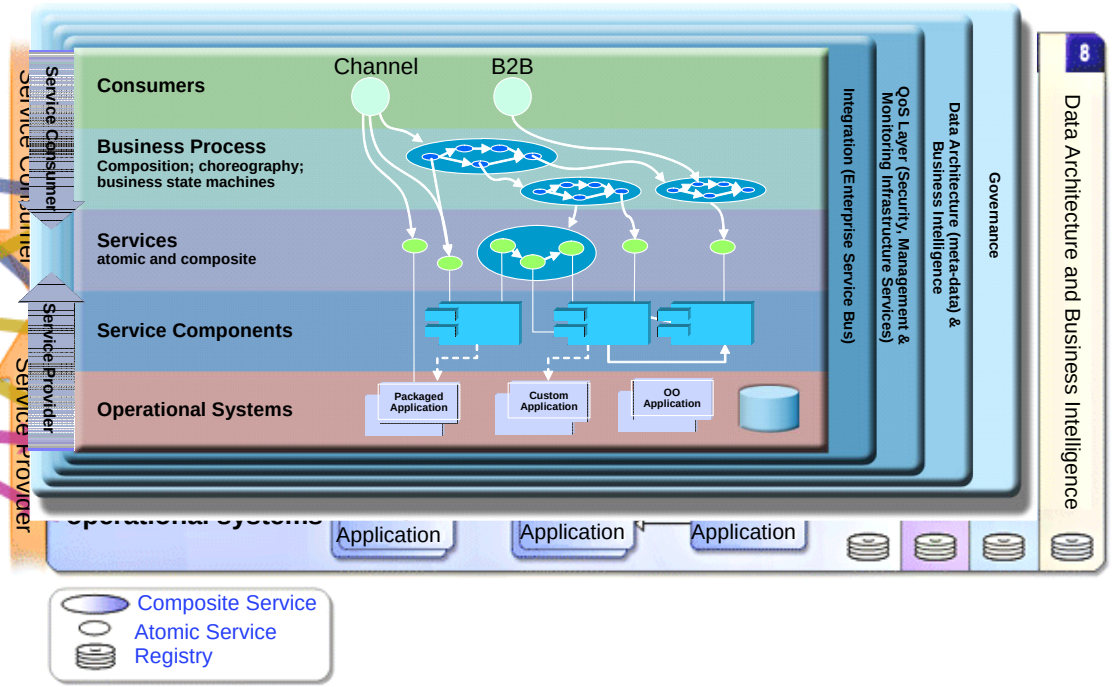
<< Input from: Business Componentization/Analysis >>

**Identification**
of Candidate Services and Flows

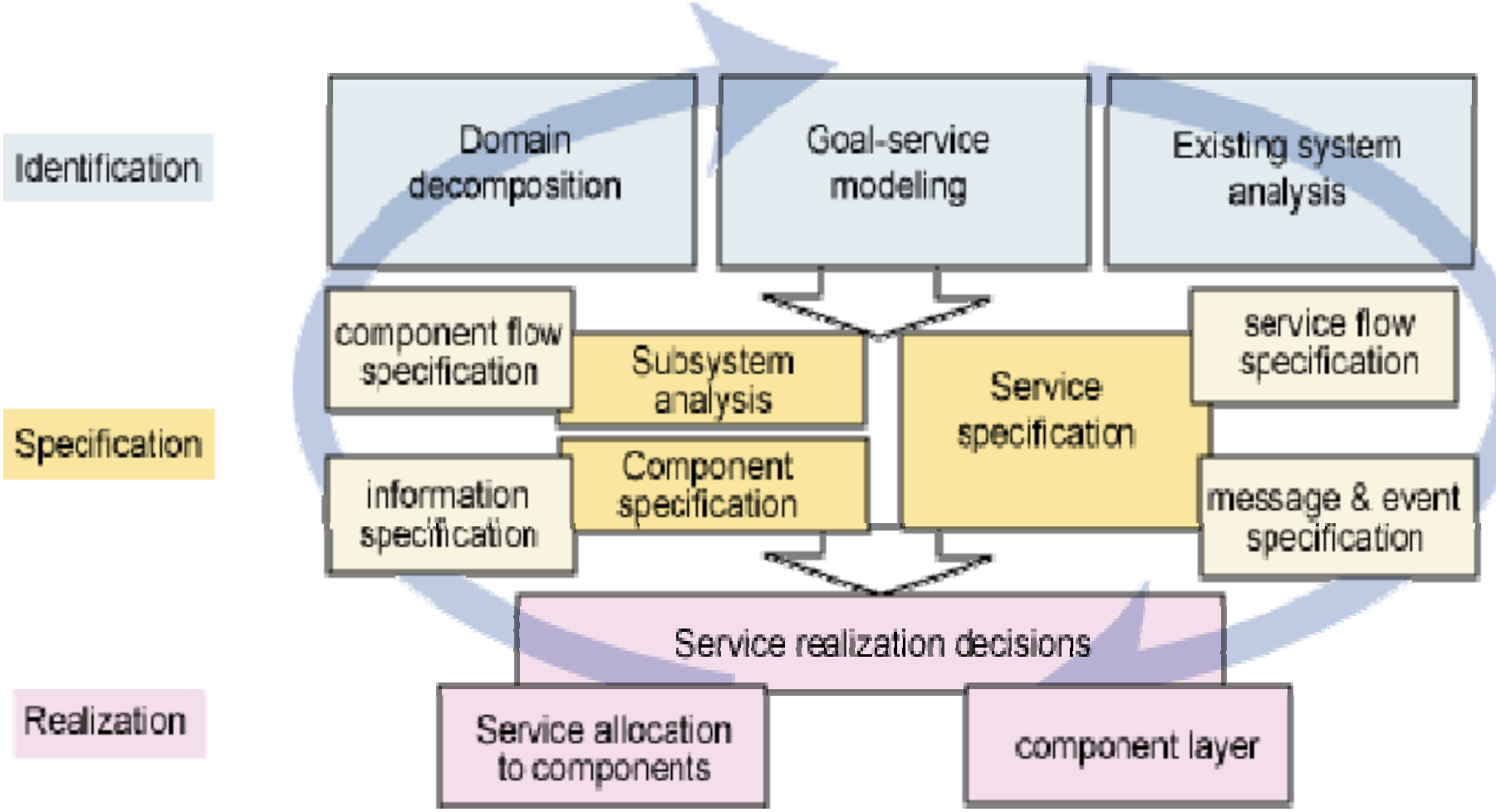
**Specification**
of Services, Components, and Flows

**Realization**
Decisions

<< Output to: SOA Implementation >>



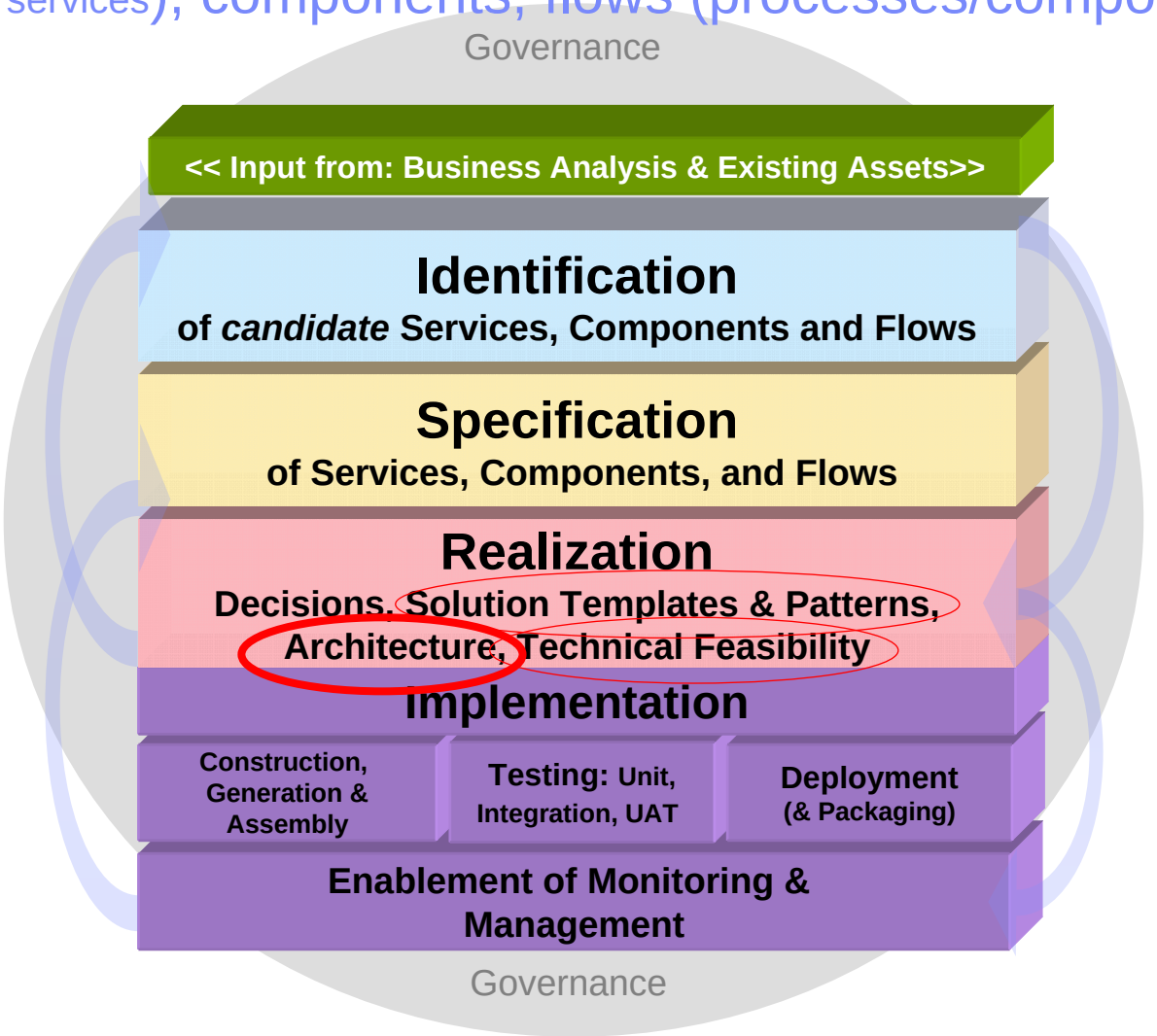
# SOMA. Initial Approach.



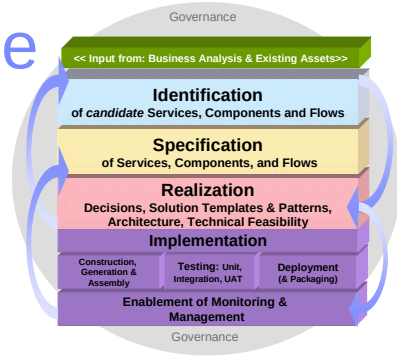
# SOMA. Initial Approach.

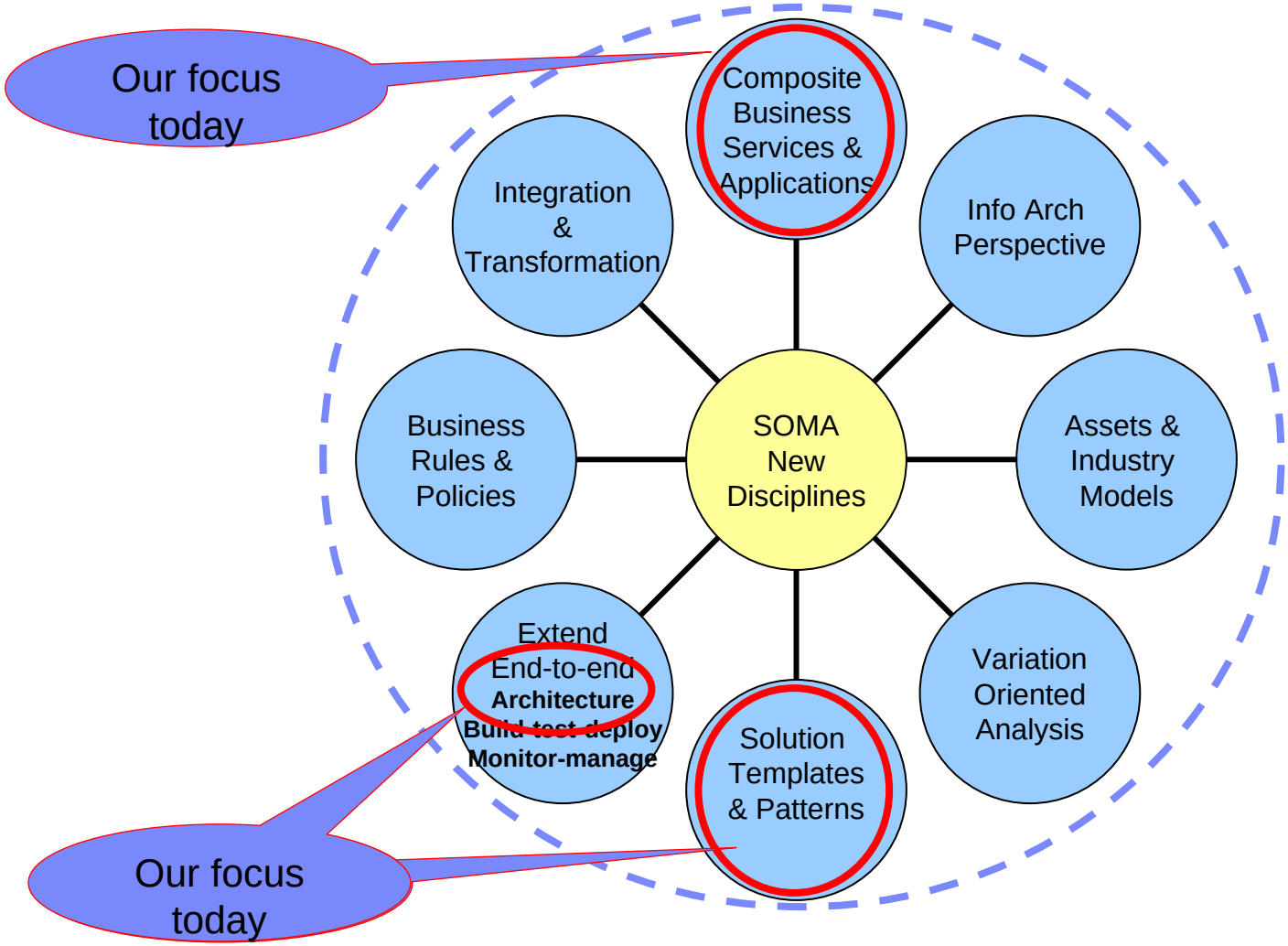
- Domain decomposition that exploits Component Business Modeling (CBM) maps
- Successful in domains, where industry assets were well developed:
  - Financial – IFW
  - Insurance - IAA
- Top-down process of manual selection and customization of appropriate models and processes in asset catalogs
  - Results: high level description of identified business services and corresponding components in the enterprise.
- Significant gap between created SOMA service definitions and required IT artifacts

# SOMA is an end-to-end SOA Method for the identification, specification, realization and implementation of services (including information services), components, flows (processes/composition)



# SOMA New Disciplines Based on Field Experience

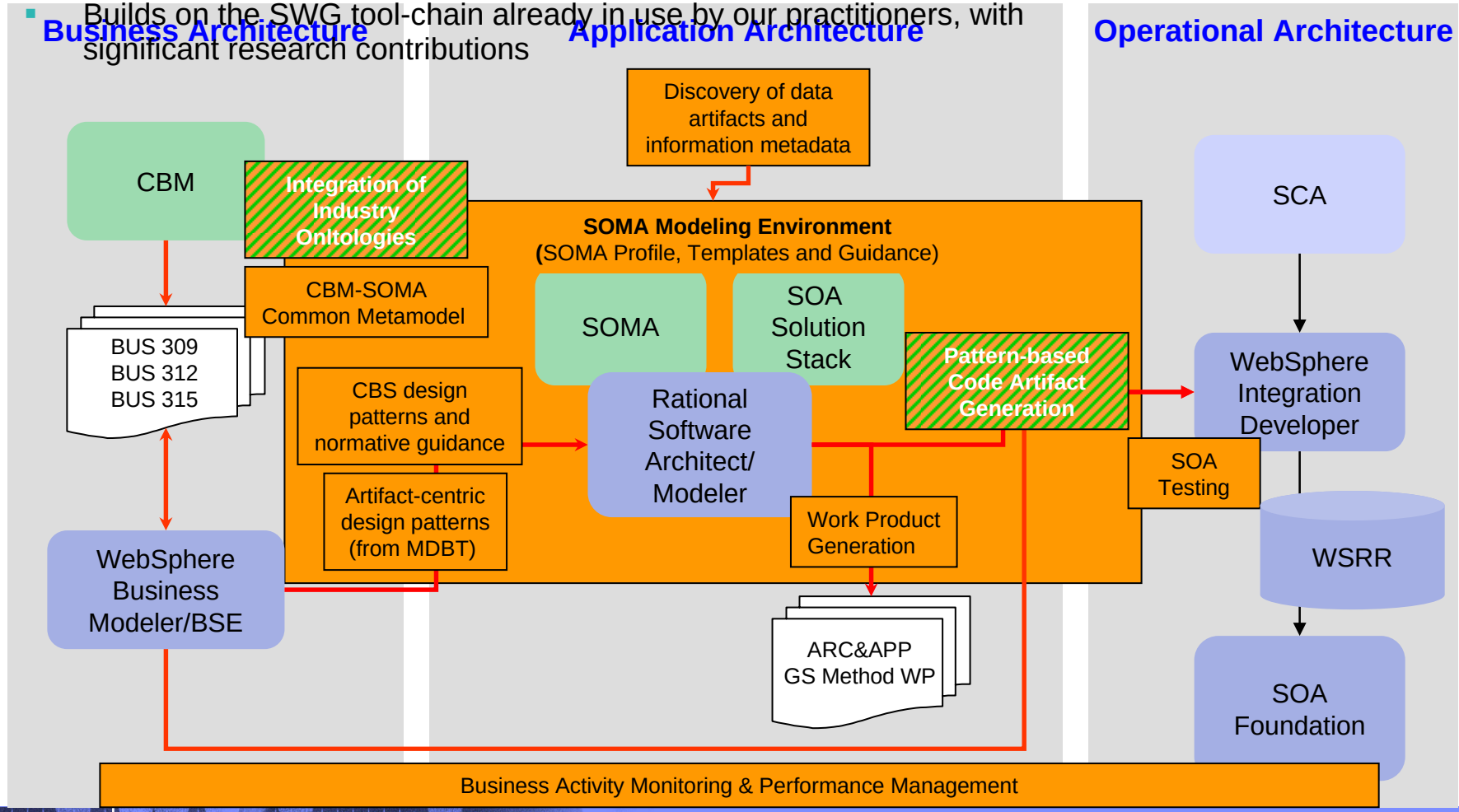


# SOA Solution Method & Tooling: SOMA & SOMA-ME

- Provides a cohesive linkage of models, methods, tools and content to support asset-based development of SOA solutions (Engagements, AS Factory, CBS…)
- Builds on the SWG tool-chain already in use by our practitioners, with significant research contributions

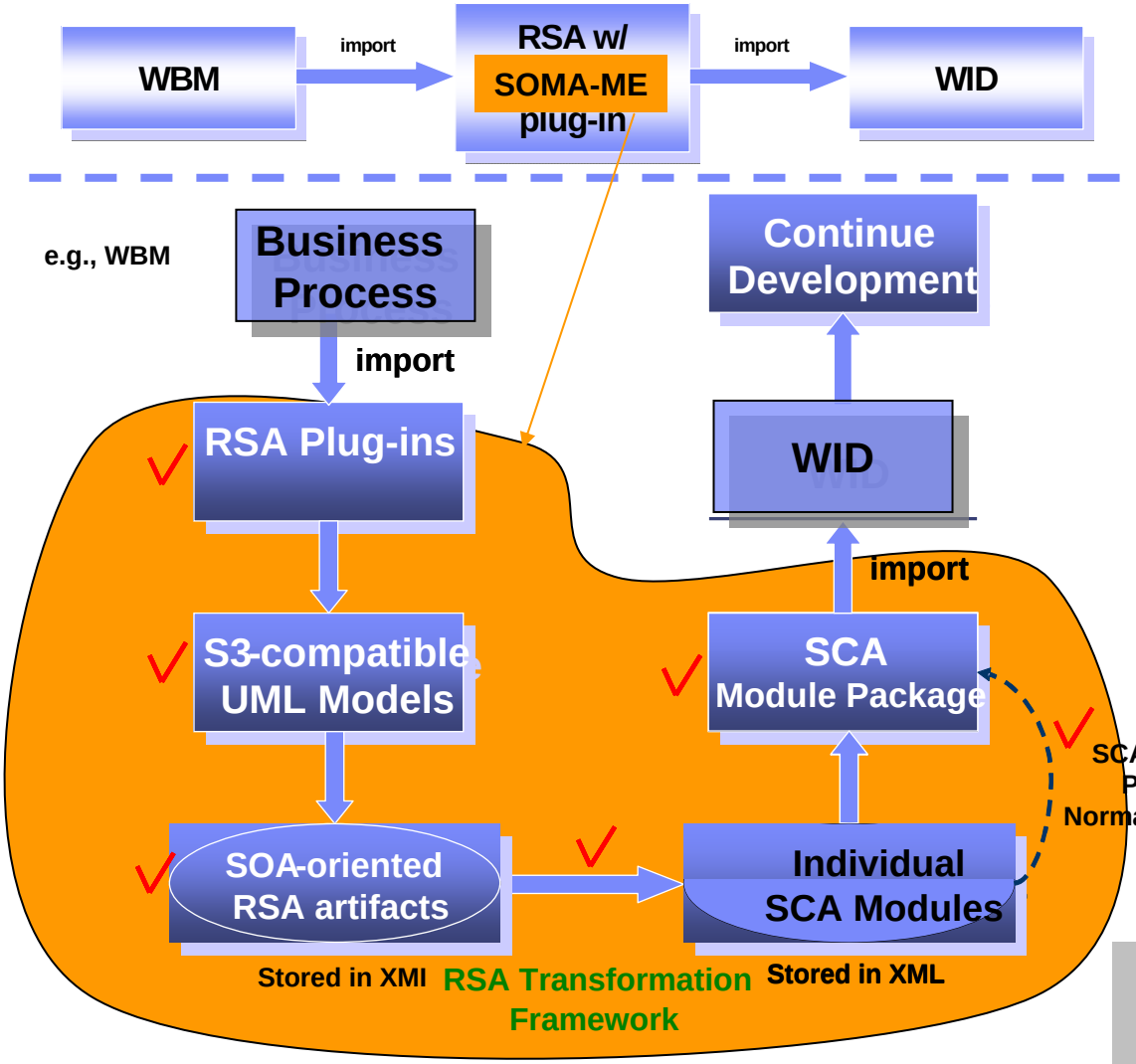
GBS Intellectual Capital | Research Contributions | SWG Tool-chain & Runtime | Joint Efforts



# SOMA Modeling Environment - An Overview

- Adds SOMA support to **Rational Software Architect** and **Rational Software Modeler**.
- Based on **SOMA** and **SOA Solution Stack**
  - Extends UML meta-model for SOMA methodology - **UML 2.0 Profiles for SOMA**.
  - Provides **Model Template** suitable for SOMA phases and activities.
  - Automates repeatedly performed tasks with MDD enhancements for SOMA - **Transformations and Patterns**.
  - Reduces effort and duration drastically with SOMA **Work Product Generation** capabilities.
  - Ease modeling with **User Interfaces** to create SOA/SOMA model elements.
  - **Model Validation** for completeness and consistency.

# Capability: SOA Solution Stack Pattern-based SCA Module Generation in RSA




- **Leverages Rational Transformation Framework (RSA) & IGS Pandoora (WIP)**
- **SCA realization of SOA Solution Stack (S3) artifacts developed using SOMA**
- **New research algorithms for**
  - Optimized service specification
  - Flexible bindings/realization

✓ **SOMA-ME Action Item**

**SCA transformation connects abstract business process definitions to deployed code**

# Composite Business Services and Applications. WBSF.

- WBSF enables following steps:
  - Assemble Composite Business Services – defining business policy and subscribers models in OWL
  - Deploy CBS into client environment – defining provision service model
  - Manage lifecycle of CBS -- enabling service monitoring and configuration
- WBSF development process is asset-based and highly dependable on existing metadata. There is a goal to enable an automatic transformation from Business Services models to WBSF metadata

# WBSF. Proposed Model Transformations in RSA

1. CBM UML profile model -> SOMA ME/or any other SOMA UML model ->SCA Modules or SCA components test configurations
2. Service Entitlement Model[1]->UML-OWL Profile model->OWL assertions/channels/roles for specific services (WSDLs)
3. Industry models (IFW, IAA, etc.)->WBSF Industry Content Packs
4. Service Ts&Cs/SLA ->UML Service Entitlement Model[1]->Tivoli TLM/TUAM/TCAM artifacts

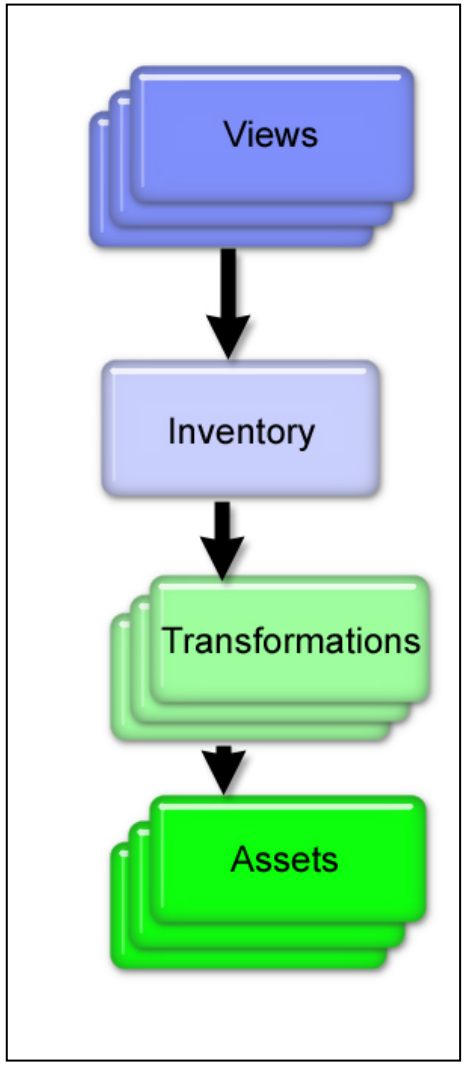
# Integration and Transformation. The Model Driven Integration Framework

- **Built On Model Driven Architecture (MDA)**
  - Defined by the Object Management Group (OMG)
  - Open industry standard, widely accepted
  - Based on UML, the de-facto modelling standard
  - Mature, proven tools available (Rational Software Architect, etc.)
- **Works on different levels of abstraction**
  - Enables mapping between different model views
  - Integrates well with SOA/SOMA models
  - Enables code generation to application environments (Application Servers, BPMS, SOA etc)
- **Built on the Eclipse Platform**

# MDIF is based on a Layered Architecture with a Single Representation (Inventory) Of The Problem Space

## Description

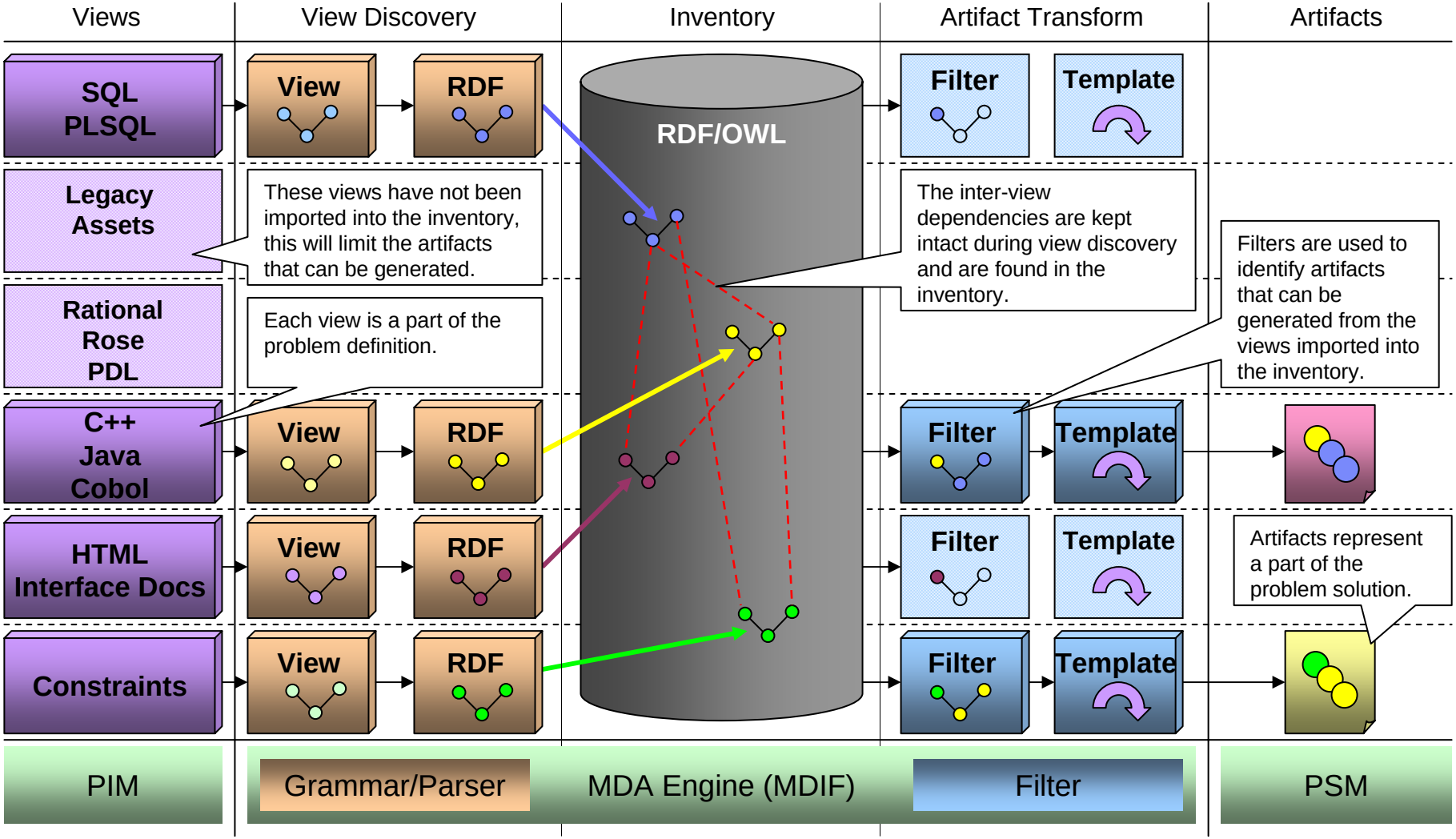
- The **Views** provide the means to create, view and update controlled subsets of the contents of the Inventory
- The **Inventory** stores the metadata. It supports versioning, access control and assumes ownership of the contents is distributed
- A **Transformation** reads metadata from the Inventory and combines it with a template to generate an Artefact
- An **Asset** is a "document" that is directly useful to the project. It could be a report or runtime code/XML or test program. It is considered read-only




## Implementation

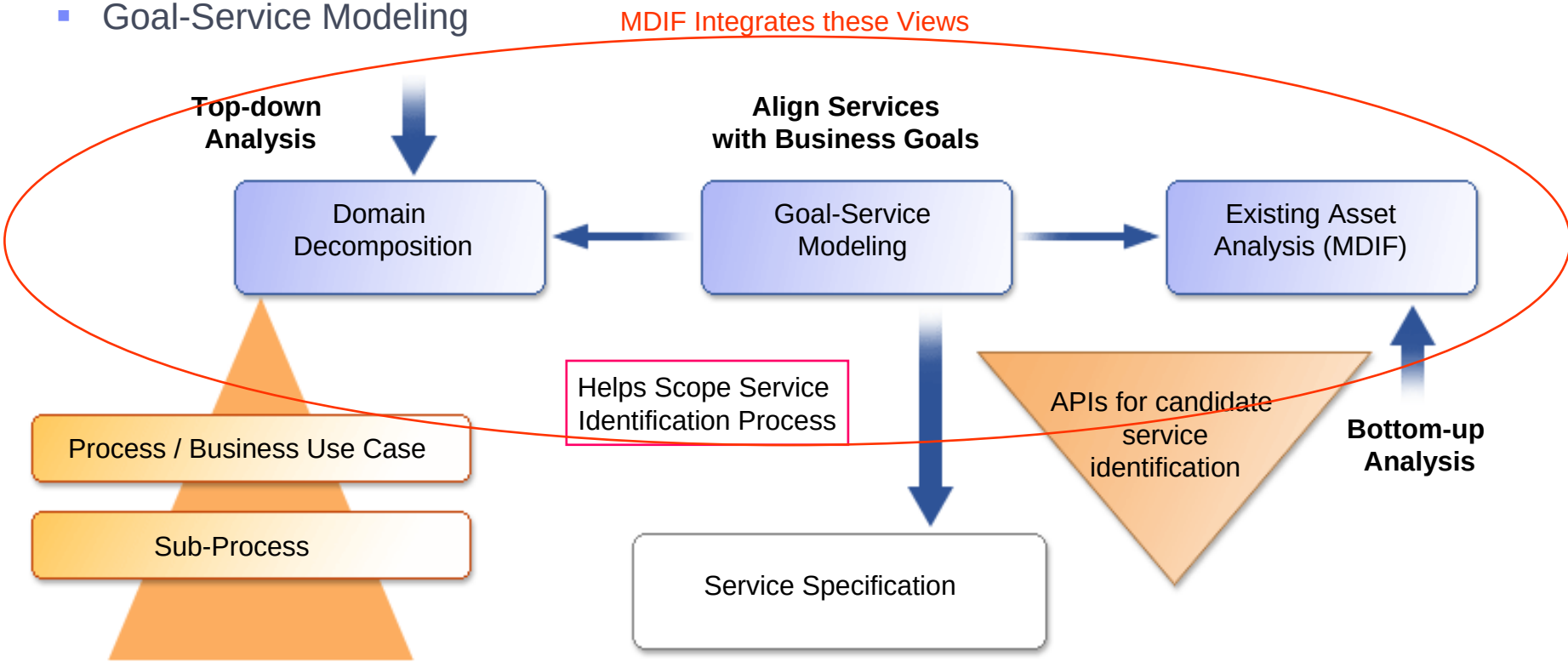
- **Context, Service or Component Views, Sequence Views, Architectural Patterns, Structural Metadata, WSDL extracts, DDL, SQL, Cobol Copybooks, IDL, other Defs, Non-Functional Requirements, Constraints, AST**
- **Implemented in RDF/OWL uses inference for impact analysis. Traceable back to source view**
- **Build in XSLT or Java uses the inventory as the only input**
- **WSDL, XML Schema, Java/C++ Code, scripts, html, Eclipse RCP UI, DOU, UML, Test Harness, Deployment environment**

# Using MDIF in Legacy Systems Transformation



# Combine with Techniques such as SOMA enabling reuse of existing functionality

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



# R4SC – Recipe for Service Creation a.k.a. PANDOORA

Part of the IBM EAD4J framework. It provides:

- Customizable model for design/architecture to client specific domain
- MDA transform for automating the development of design models/code from analysis model details.

Pilot MDA transforms in RSA format cover the following:

- Analysis-Design Transform automates development of class diagrams within the Design Model
- Design-Code Transform currently supports design model to Java transform
- PANDOORA for SOMA MDA transforms are under development.

# What is PanDOORA: A set of project accelerators comprised of software engineering best practices, reference architecture, project planning aides, and code components for building e-business Applications

Application Reference Architecture for e-business and related common component designs (accelerates project start-up and design by 1000 hours)

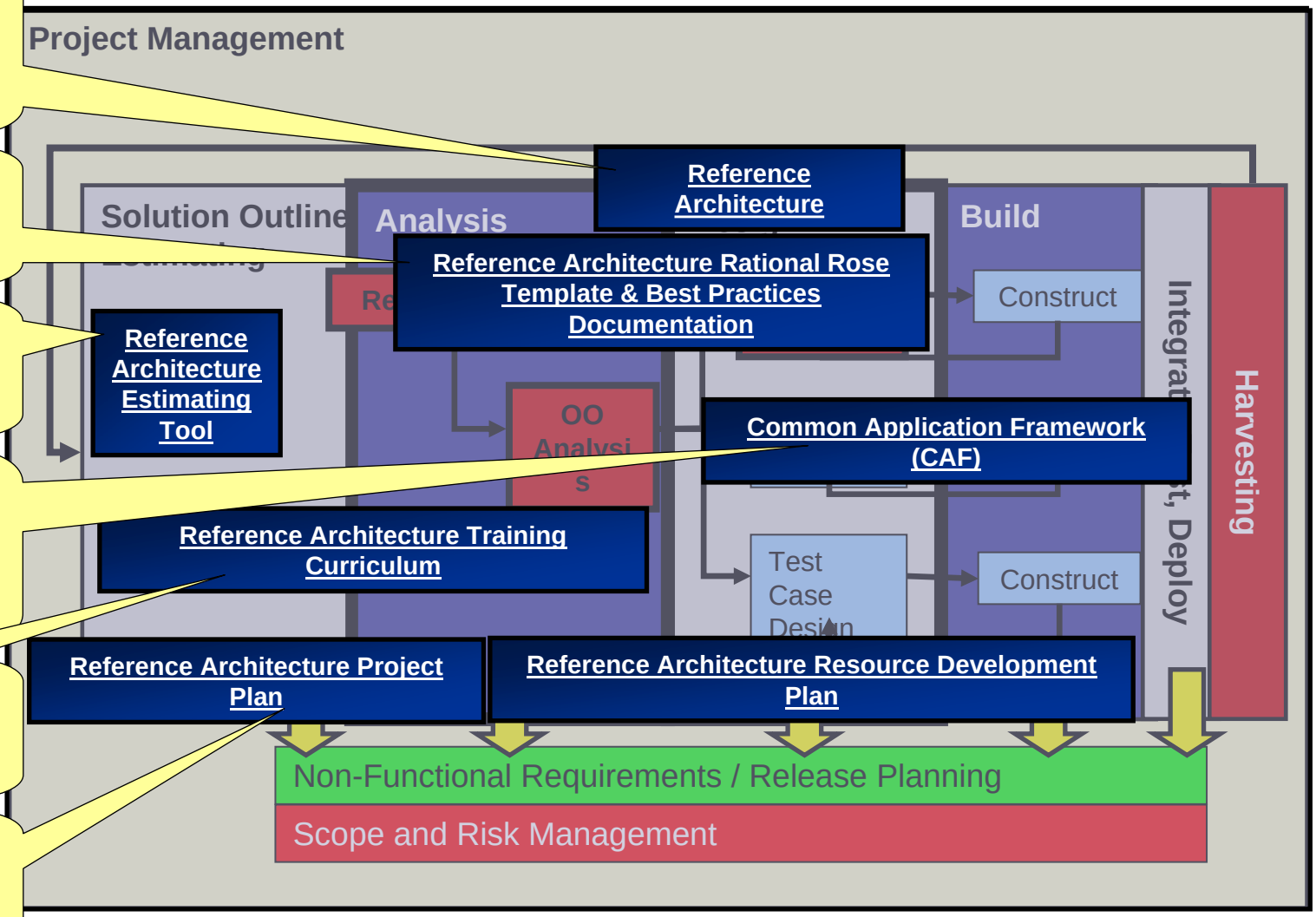
Rational Rose best practices and templates based on the application architecture

Estimating models based on project metrics from past PanDOORA solutions

Logging, exception handling, caching, session management, security, EAI, etc. code components (accelerates project design/implementation by as much as 1500 hours)

Training curriculum for applying PanDOORA throughout the development process.

Project plan templates for iterative object-oriented methods applying PanDOORA



# IBM's end-to-end SOA Method

- Extension of SOMA into implementation, testing, deployment, monitoring and management
- It concentrates on integration with IBM GS Method and RUP and integration with other activities such **MDIF** and **PANDOORA**
- MDD/MDA model-to-text transforms can help here in order to synchronize SOMA models with GSM Work Products

# MDD Opportunities

- MDD based tooling in order to automate end-to-end SOMA development in WBSF environment
  - SOMA-ME Integration with WBSF
  - OWL-UML transformation pack in SOMA area
  - Tivoli TLM/TUAM/TCAM artifacts generation in order to enable a monitoring of “Return on Investment for SOA”
  - MDD support for Industry Models
- Other
  - Legacy SOA Realization - MDIF
  - Information Architecture - support for WIS (not covered here)

धन्यवाद

Hindi

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

Thank You

English

شكرا

Arabic

Obrigado

Brazilian Portuguese

多谢

Simplified Chinese

Danke

German

Grazie

Italian

Merci

French

நன்றி

Tamil

Tamil

ありがとうございました

Japanese

감사합니다

Korean