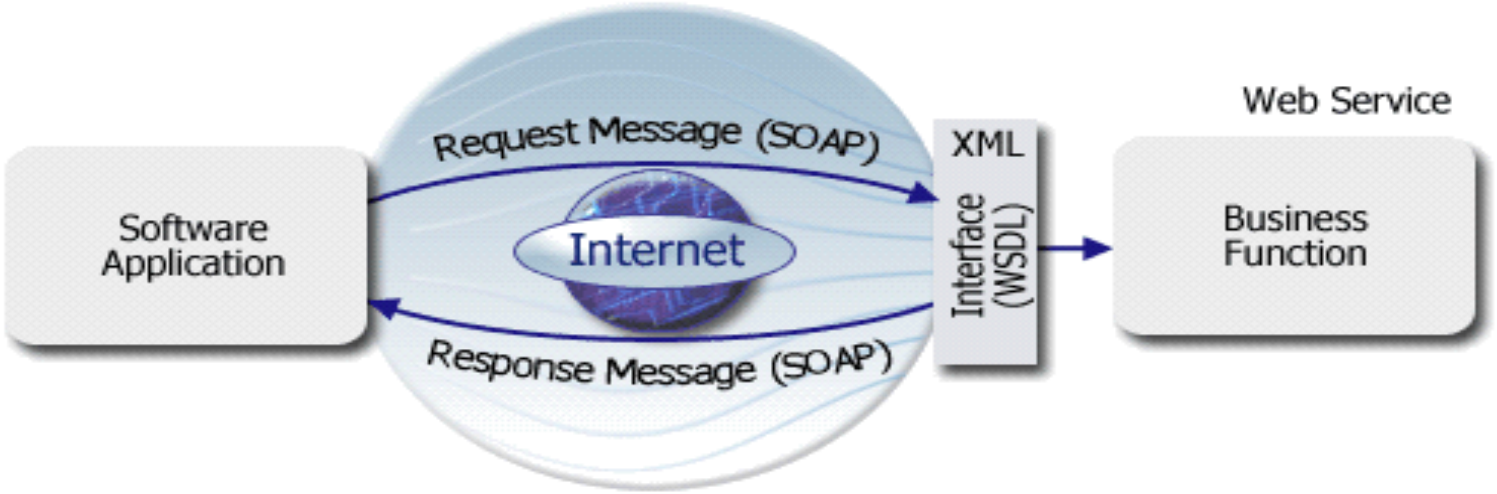
IBM Service-oriented Architecture

Smart SOA in Action

SOA Service Creation

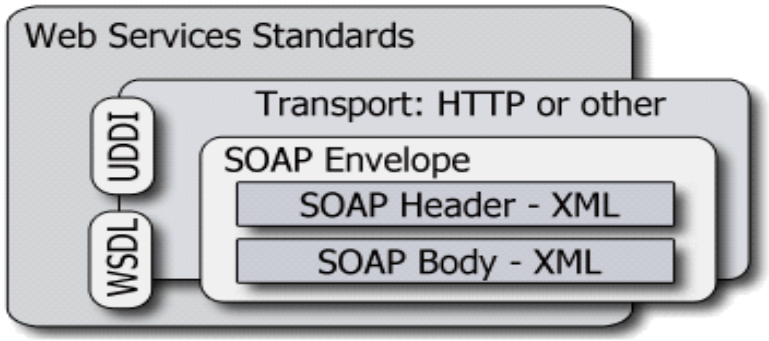
Simon Chan
SOA Architect
hychan@hk1.ibm.com

What is Web Service



Key Technologies for Web Services

- **SOAP** over HyperText Transfer Protocol (HTTP) is the common protocol for Web services.
- **WSDL** is the interface contract language between Web service providers and requestors
- Web services can be discovered using Universal Description, Discovery, and Integration (**UDDI**).
- **WS-I** (Web Services Interoperability) has developed specifications that have become the foundation for even the most basic Web services. WS-I is an industry consortium chartered to promote standards for interoperability of Web services.



How is Web Service Defined

WSDL <portType>

```
<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
       <input message="tns:GetLastTradePriceInput"/>
       <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>
```

WSDL portType

A portType element defines the abstract operations used by the Web service. .

WSDL <message>

```
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>
```

WSDL message

A message element defines each message used by the Web service and associates it with a particular name and data type.

WSDL <types>

```
<types>
   <schema targetNamespace="http://example.com/stockquote.xsd"
         xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
         <complexType>
            <all>
               <element name="tickerSymbol" type="string"/>
            </all>
         </complexType>
      </element>
      <element name="TradePrice">
         <complexType>
            <all>
               <element name="price" type="float"/>
            </all>
         </complexType>
      </element>
   </schema>
</types>
```

WSDL type

A types element defines the data types, typically expressed using XML Schema, describing the messages exchanged with the Web service.

How is Web Service Defined

WSDL <service>

```
<service name="StockQuoteService">
    <port name="StockQuotePort" binding="tns:StockQuoteSoapBinding">
       <soap:address location="http://example.com/stockquote"/>
    </port>
</service>
```

WSDL service

A service element groups a set of related ports together. Each port is associated with a particular binding definition. The actual location of the Web service is defined within the port definition of a service.

WSDL binding

A binding element defines the message format and protocol details for operations and messages defined by a particular portType. Each operation defined under portType is associated with an access protocol, such as SOAP, and various formatting options.

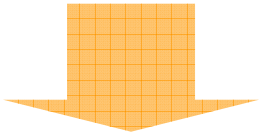
WSDL <binding>

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
 <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http"/>
 <operation name="GetLastTradePrice">
    <soap:operation soapAction=
     "http://example.com/GetLastTradePrice"/>
    <input>
       <soap:body use="literal"/>
    </input>
    <output>
       <soap:body use="literal"/>
    </output>
 </operation>
</binding>
```

How to access Web Services using SOAP

SOAP request message

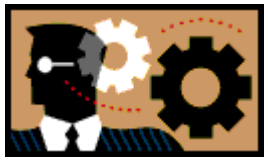
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <m:GetLastTradePriceInput xmlns:m="Some-URI">
            <m:tickerSymbol>IBM</m:tickerSymbol>
        </m:GetLastTradePrice>
    </soapenv:Body>
</soapenv:Envelope>
```



SOAP response message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <m:GetLastTradePriceOutput xmlns:m="Some-URI">
            <m:price>75.5</m:price>
        </m:GetLastTradePriceResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Question



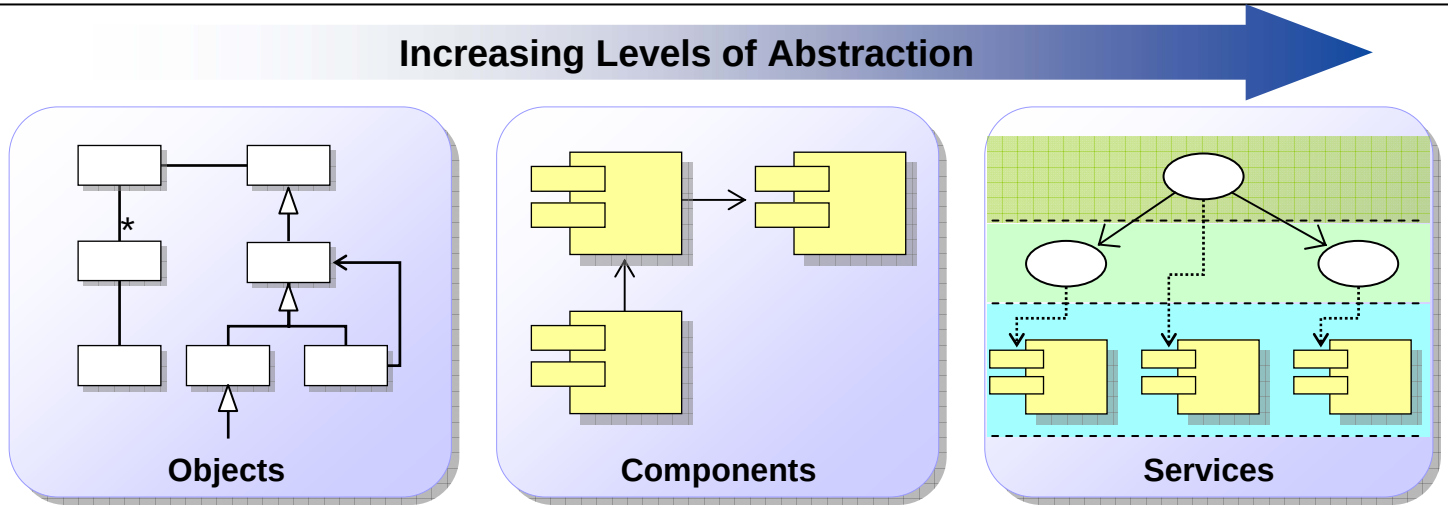
- Do we need to turn all application functions into services
- Why or Why not?

Question

- How should service granularity be defined?

vs

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**.

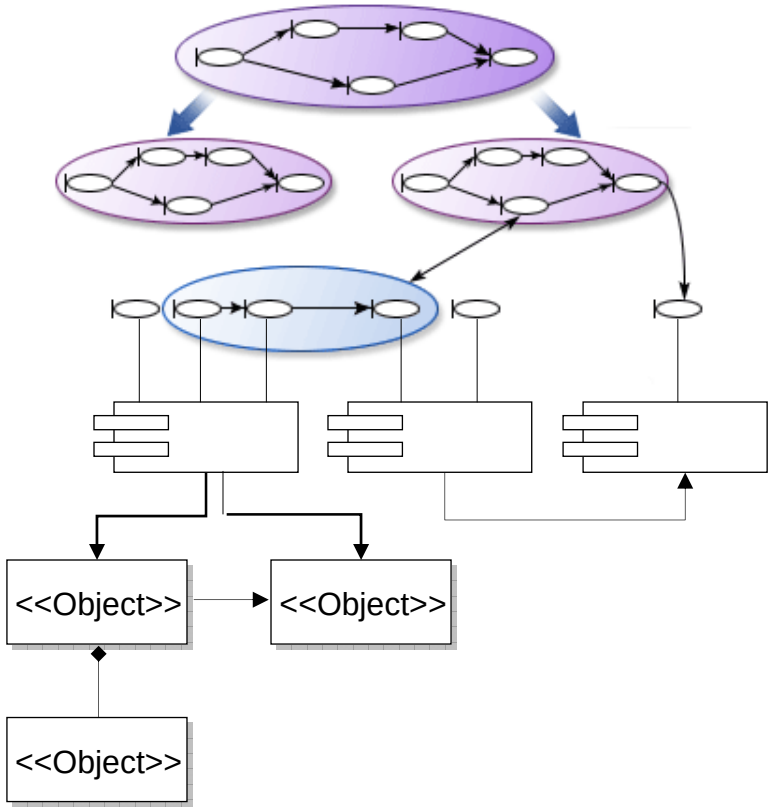
While SOA **builds on** well-established software architecture principles (for example, information hiding, modularization and separation of concerns), it also **adds** additional aspects; thus, service-oriented modeling needs additional techniques for these new aspects.

SOA Modeling Constructs

Business Processes
(Flows)

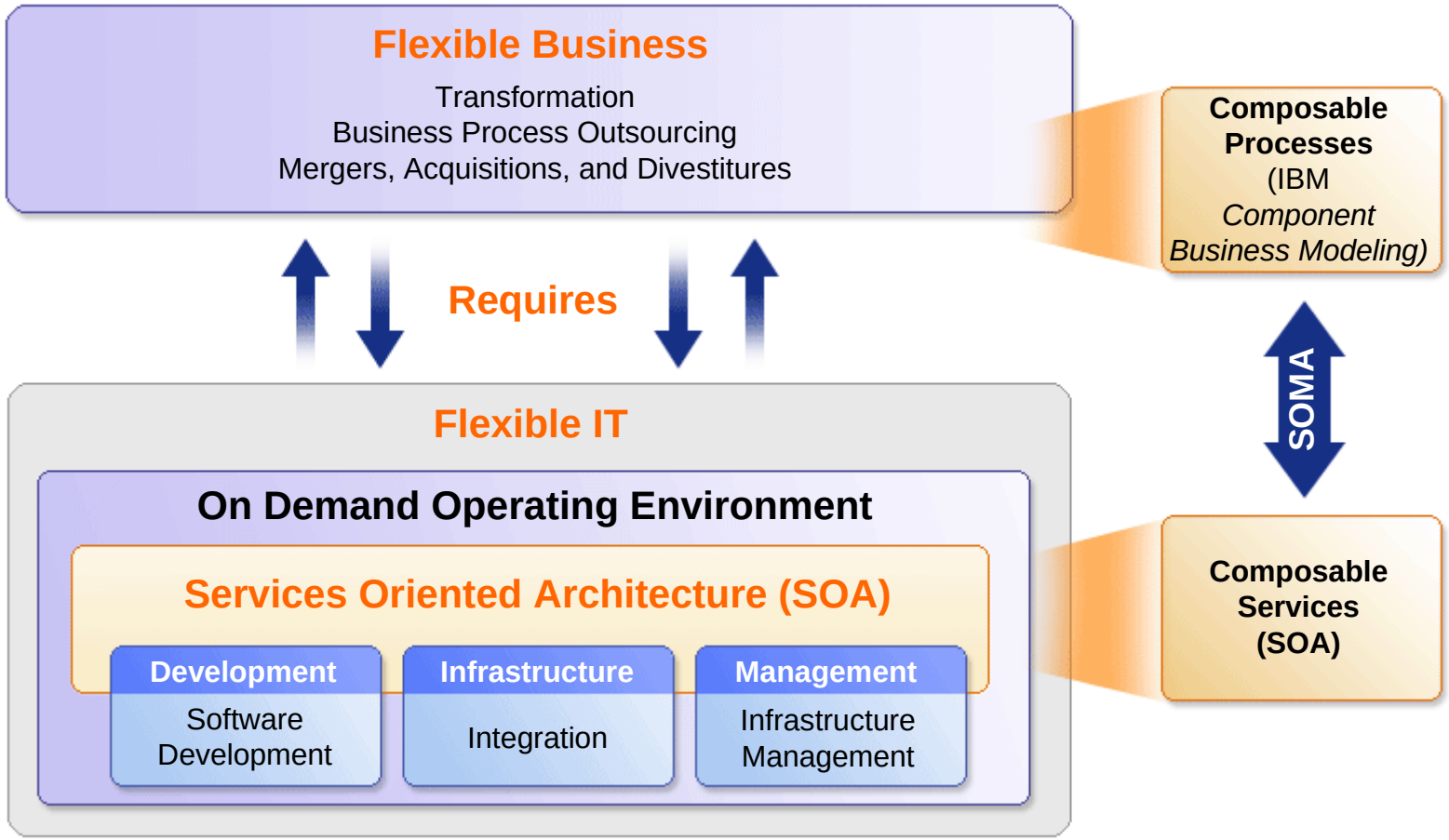
Services
Atomic and Composite

Service Components



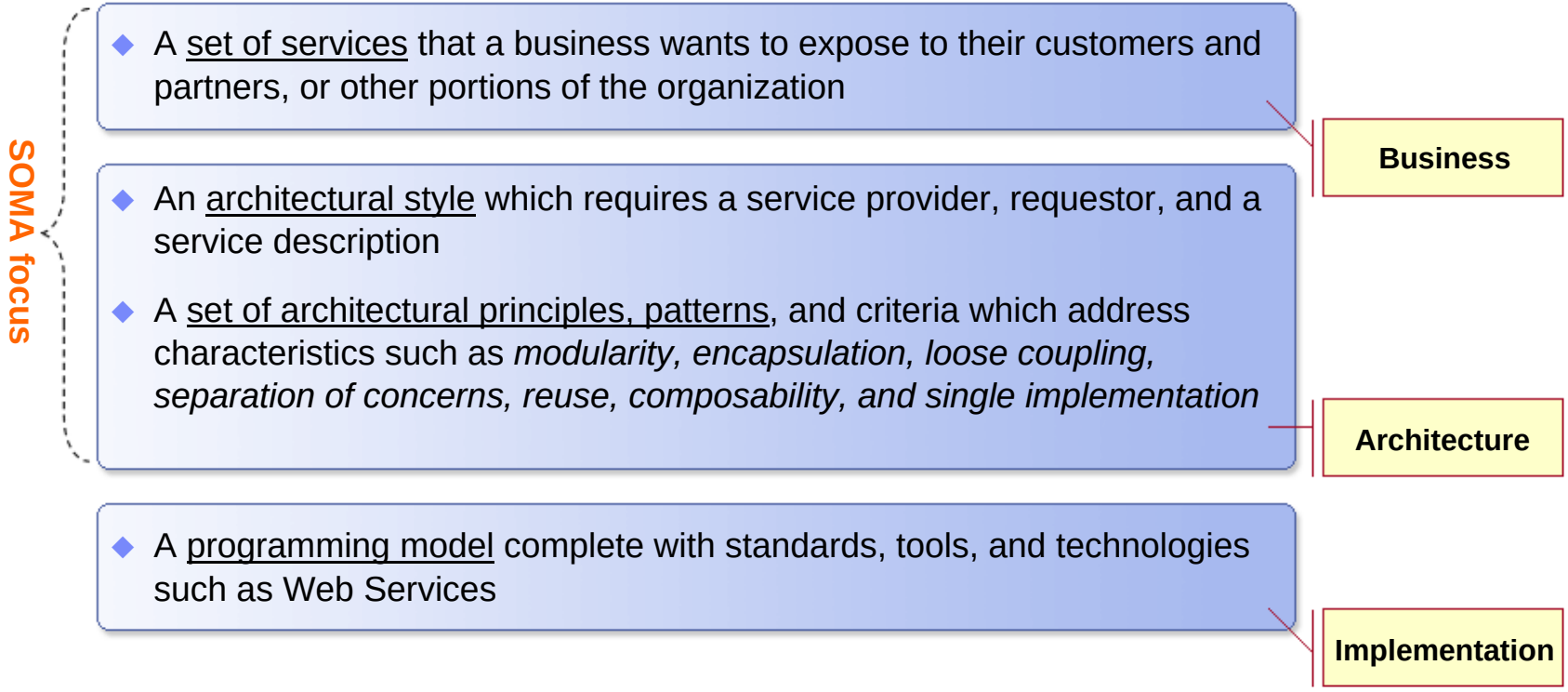
SOMA was created to specifically address modeling (analysis, identification, and specification) of all three constructs.

Flexible Business Requires Flexible IT

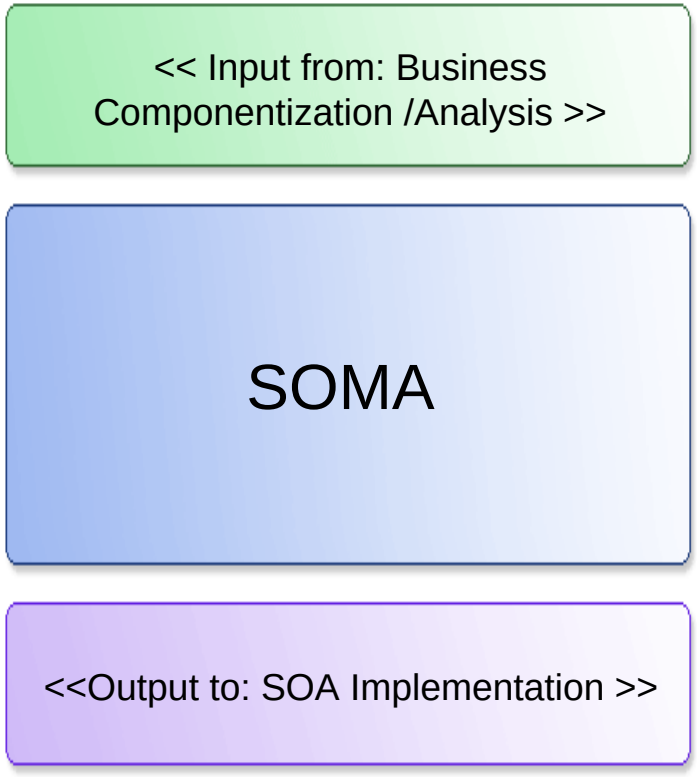


SOA Various Perspectives Focus on Different Attributes

"SOA in context …"

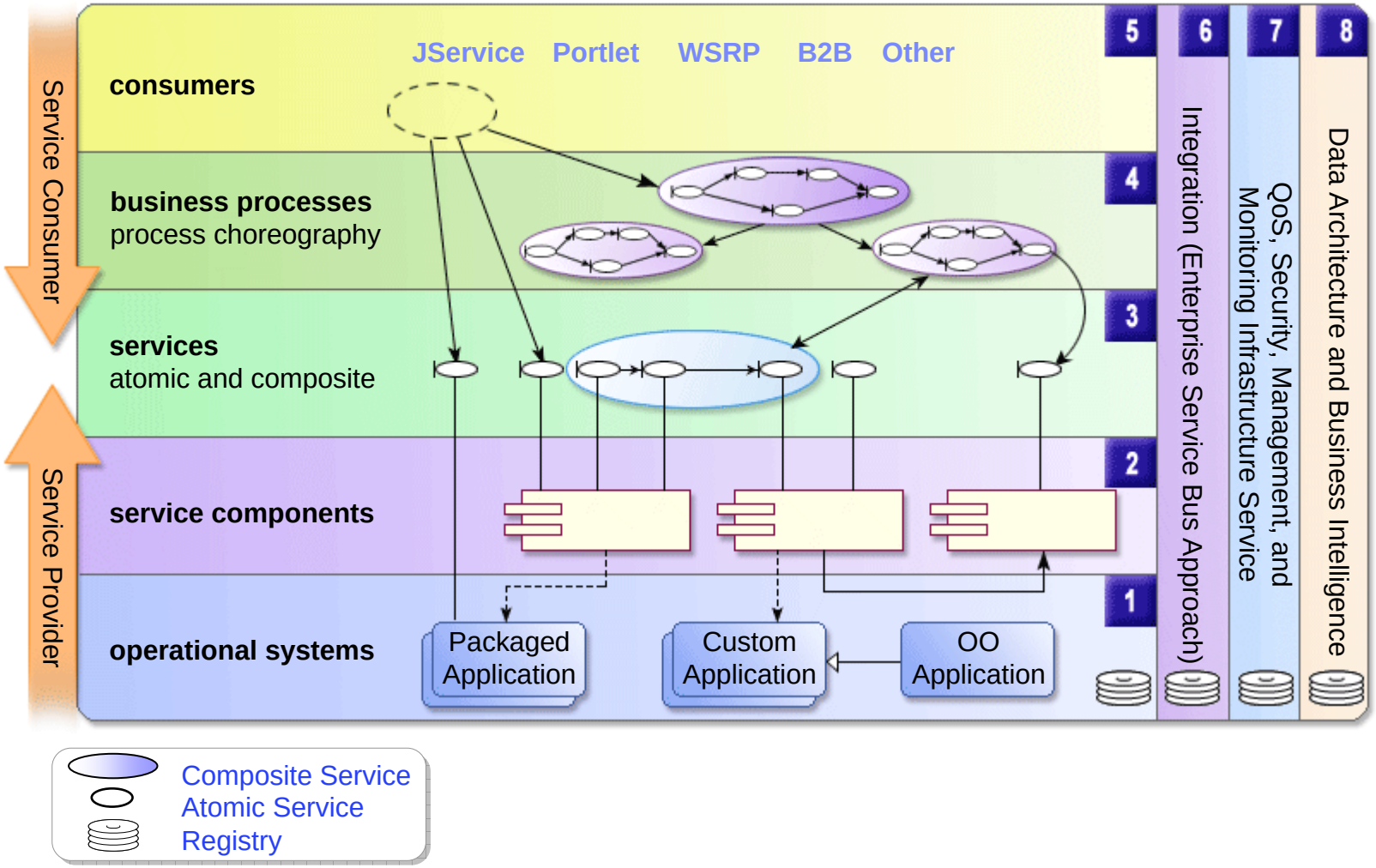


SOMA Links Business Intent and IT Implementation



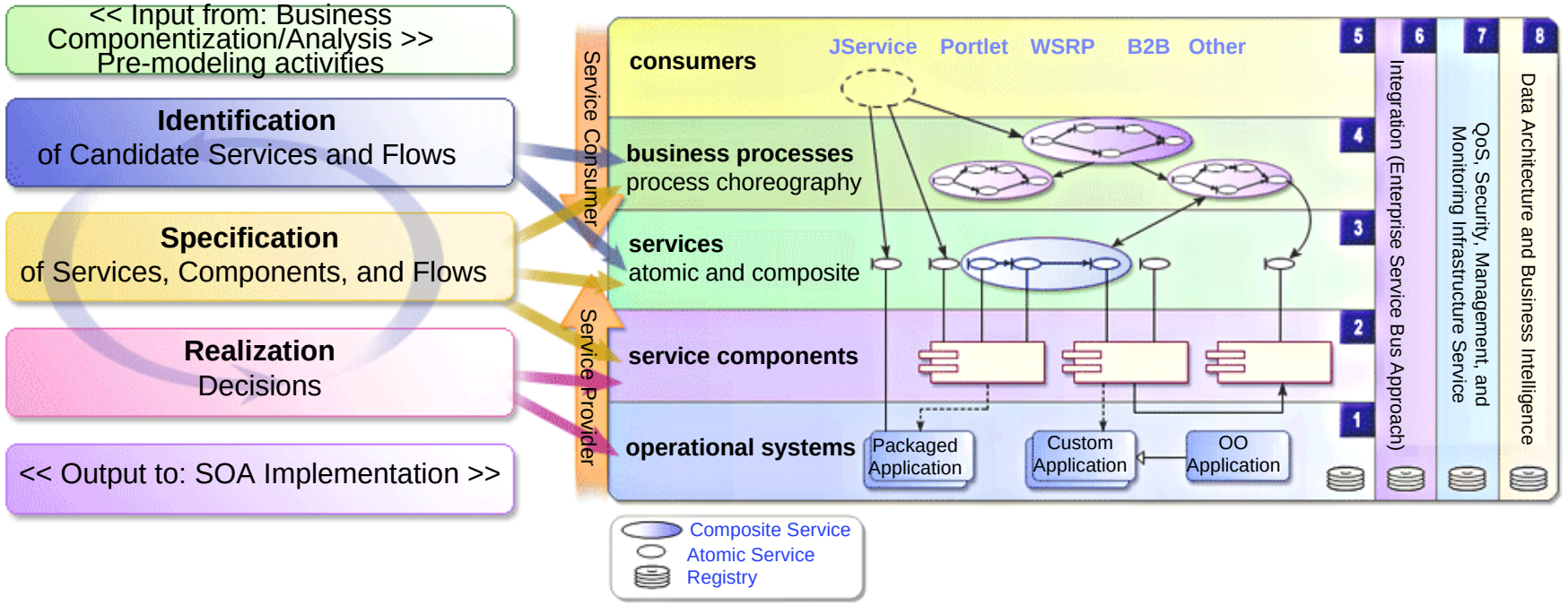
- SOMA gets inputs from business componentization and analysis activities, and produces outputs necessary for SOA implementation.
- The analysis and modeling performed during SOMA is **technology and product agnostic**, but establishes a context for making technology and product specific **decisions** in later phases of the lifecycle.

The Ultimate Goal of SOMA Is to Build an SOA-Based Solution



SOMA Activities Are Grouped into Three Major Steps

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



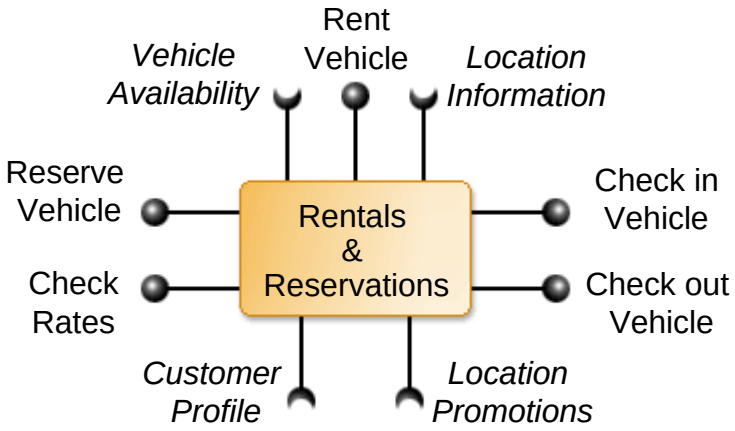
Rent-A-Car: Case Study Introduction

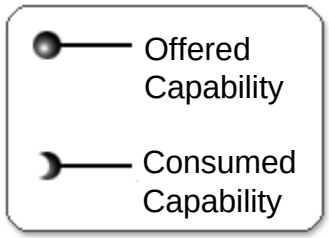
- The example is from the **car rental segment** of the Travel & Transportation Industry.
- The scope of this example covers **three business processes**:
  - Reservation
  - Vehicle Check-out
  - Vehicle Check-in
- Input considerations:
  - Although SOMA does NOT require CBM as the source of inputs, the componentized and service-oriented view of CBM makes it a natural fit with SOMA. For this example, we will be using outputs from an illustrative CBM engagement as inputs to SOMA.

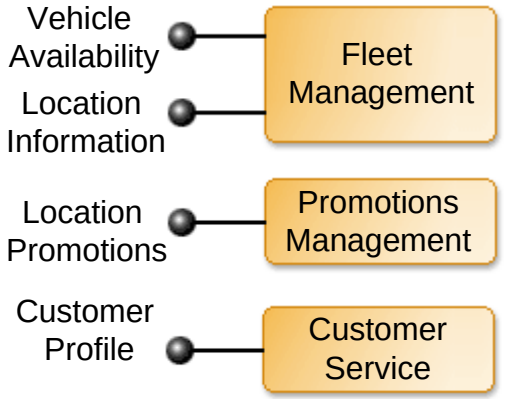
CBM Heat Map of Rent-A-Car "Hot" Components

	Marketing & Customer Mgt.	Products	Rentals Management	Rental Fleet Logistics	Business Administration
Direct	Customer Segmentation Customer Relationship Strategy Marketing Strategy & Planning	Rental Product Strategy Product Development / Design	Location & Channel Strategy Location Design & Layout Channel Design & Layout	Fleet Strategy Fleet Planning OEM Relationship Planning	Corporate / LOB Strategy Financial Management & Planning Real Estate Planning
Control	Customer Behavior Modeling Market & Competitor Research Segmentation Management Call Center Campaign Management	**Promotions Management** Pricing Management	Channel & Location Profitability Location Operations Management Reservations Management Workforce Management	OEM Performance Management In-bound Logistics	Alliance Management Business Performance Reporting Legal & Regulatory Compliance Real Estate & Construction Management Risk Management Stock Ledger HR Management (Career Dev., Training, Recruiting)
Execute	**Customer Service** Preferred Member Mgmt Customer Communications Mass Marketing & Advertising Target Marketing	Purchasing / Sourcing Demand Forecasting	**Rentals & Reservations** Time & Attendance	Location Operations Fleet Servicing **Fleet Management**	HR Administration / Payroll Corporate Audit Corporate Accounting (GL, AP, A/R, Treasury, etc.) Indirect Procurement PR & Investor Relations IT Systems & Operations

Rent-A-Car Rentals and Reservations "Hot" Component



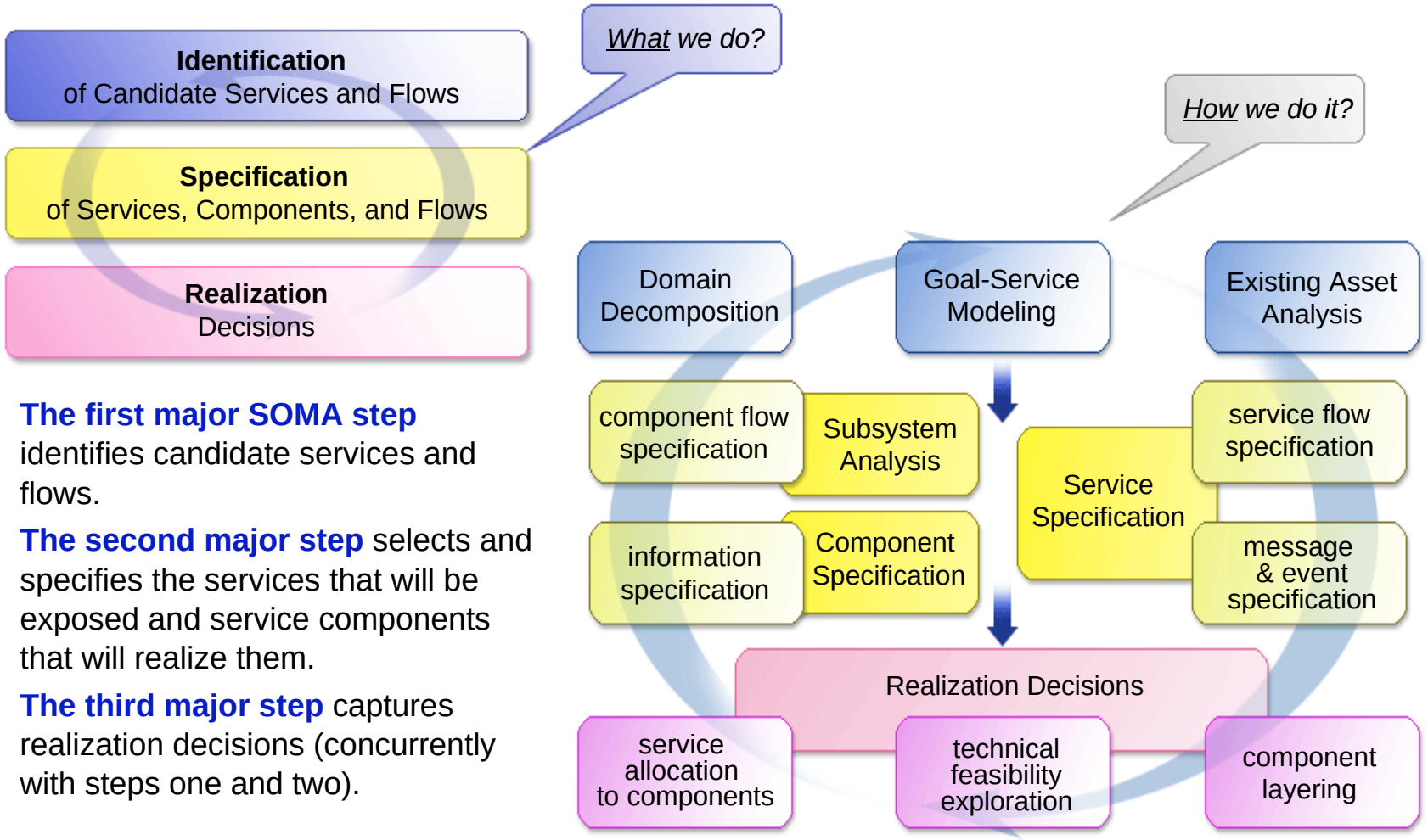




Rentals & Reservations	*Execute Level Business Component*. Provides business functionality related to vehicle rental reservations. Provides support for both the reservations and rentals business processes.
Customer Service	*Execute Level Business Component*. Responsible for servicing the customer. Offers, among other things, capabilities relating to maintenance of customer profile.
Promotions Management	*Control Level Business Component*. Responsible for the management of promotions across the board.
Fleet Management	*Execute Level Business Component*. Responsible for providing functionality around fleet management, fleet availability, and so on.

Rent Vehicle
Business Process

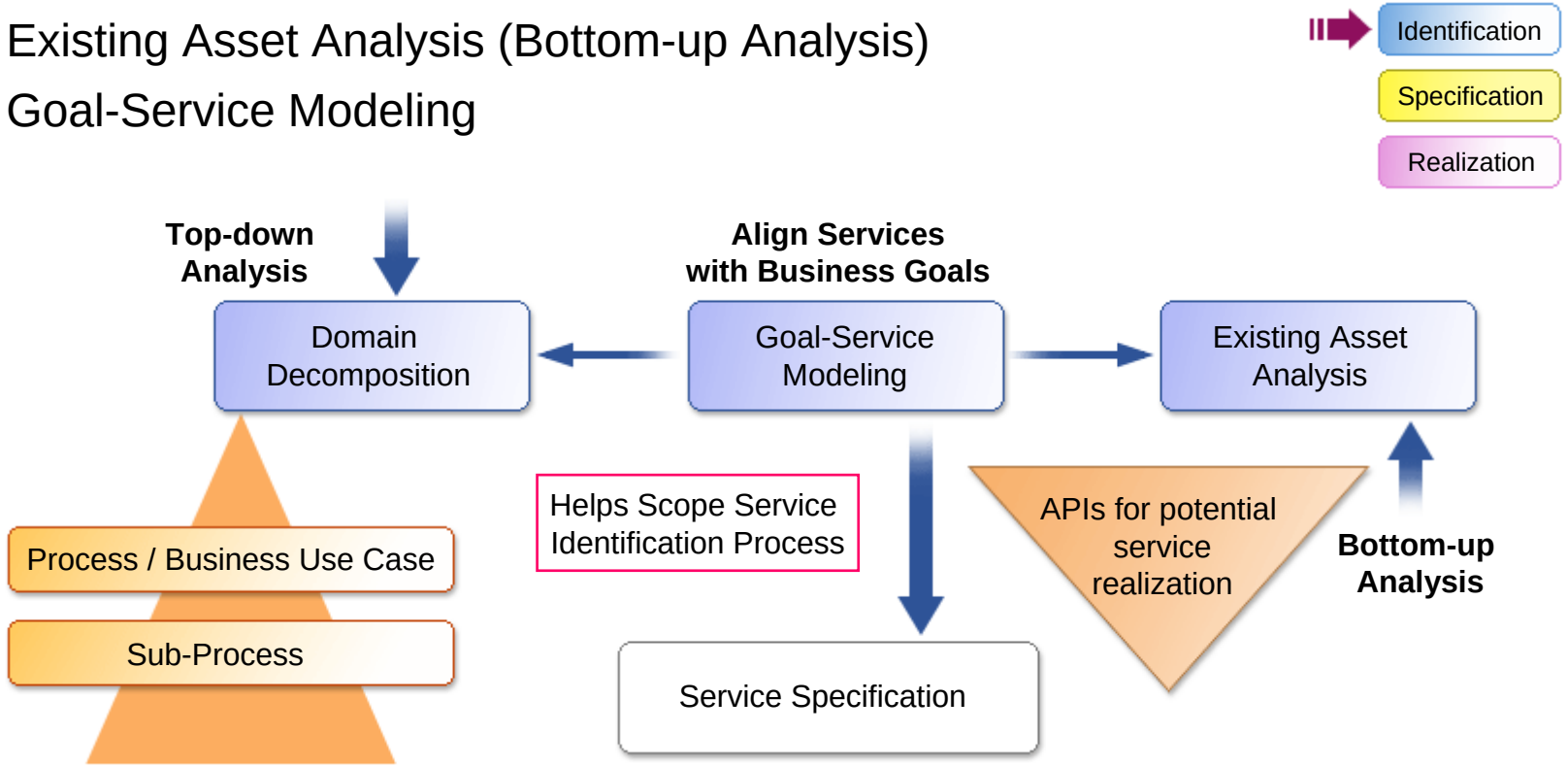
Each SOMA Step Is Carried Out by Complementary Techniques



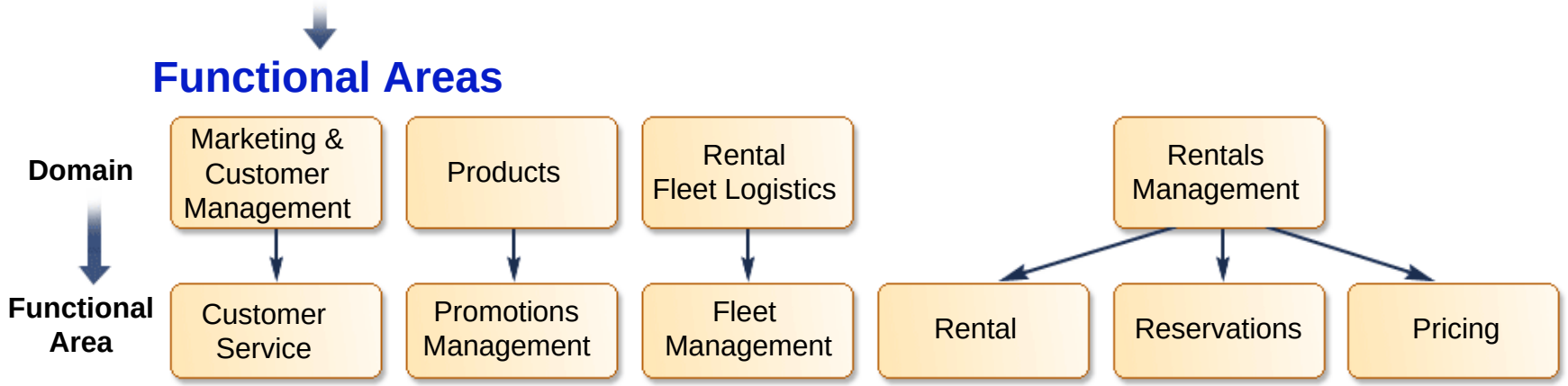
- **The first major SOMA step** identifies candidate services and flows.
- **The second major step** selects and specifies the services that will be exposed and service components that will realize them.
- **The third major step** captures realization decisions (concurrently with steps one and two).

SOMA Identifies Services Through Three Complimentary Techniques

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



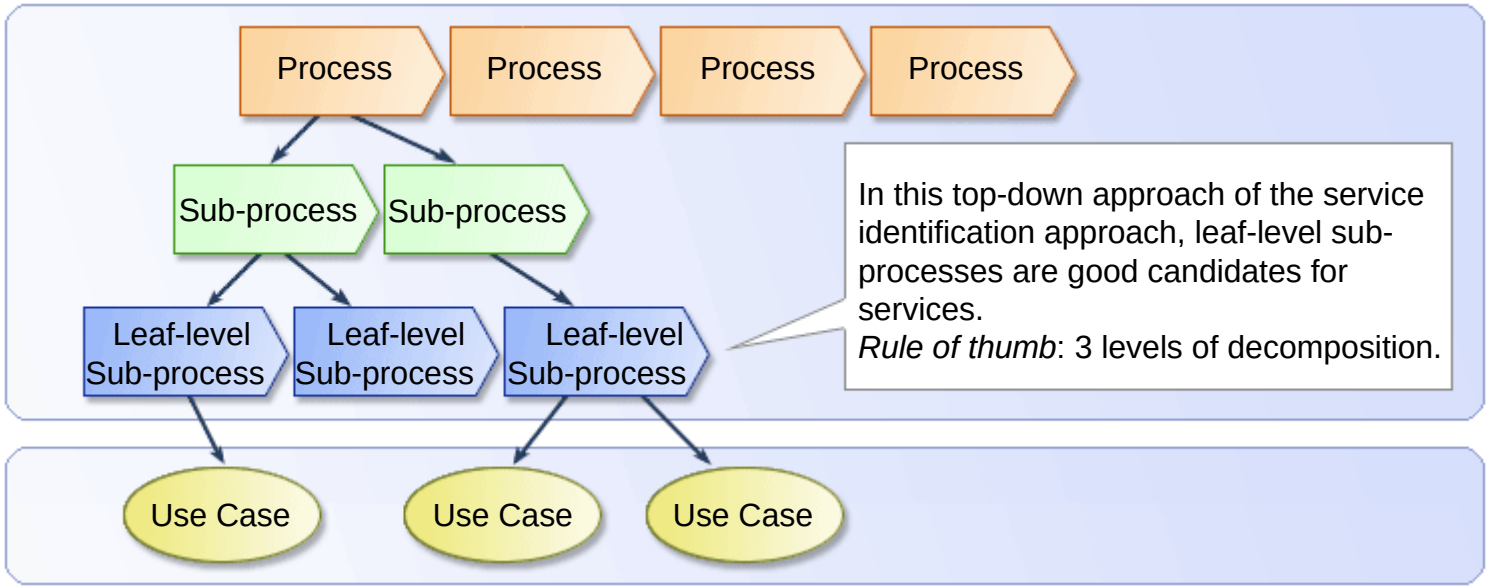
Rent-A-Car Domain Decomposition – Functional Area Analysis



Functional Area	Description	Functions
Rental	Manage the rental process for clients	**Check out** car as defined by revised client business process
		Check in car as defined by revised client business process
Fleet Management	Manage the fleet of rental cars	**Find car** for customer based on style, make/model, availability, and location of the car
Reservations	Create and manage reservations for clients	**Create the reservation** for the customer based on the customer reservation business process
		Locate reservation for an existing customer with a reservation
		Cancel a customer's existing reservation

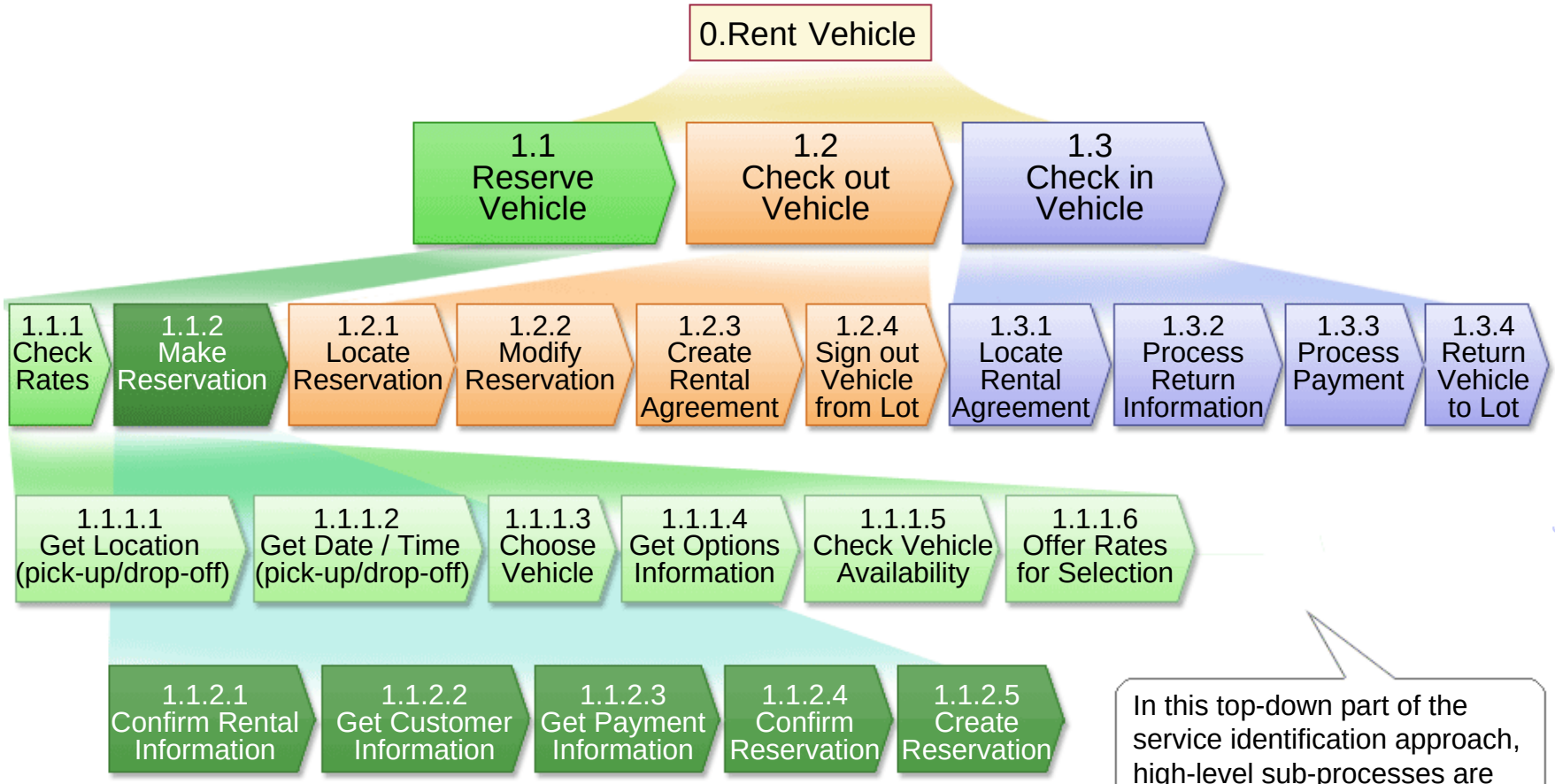
EXAMPLE
For illustration only

Process Decomposition Helps Identifying Candidate Services



- A sub-process is a convenient construct used to denote further levels of refinement to a process into its constituent parts (sub-processes), recursively.
- Sub-processes are used to identify candidate services.
- The list of use cases provides the initial scope for system design (“business as usual”).

Rent-A-Car Process Decomposition



EXAMPLE
For illustration only

Question: Which should be exposed as services?

In this top-down part of the service identification approach, high-level sub-processes are good candidates for business services. *Rule of thumb*: 3 levels of decomposition.

All candidate services are captured in the Service Model's "Service Portfolio" part

- In the case study example, every "box" in the process decomposition is initially considered a "Candidate Service."
- Service Portfolio
  - 0 Rent Vehicle
  - 1.1 Reserve Vehicle
  - 1.2 Check out Vehicle
  - 1.3 Check in Vehicle
  - 1.1.1 Check Rates
  - 1.1.2 Make Reservation
  - 1.2.1 Locate Reservation
  - 1.2.2 Modify Reservation
  - 1.2.3 Create Rental Agreement
  - 1.2.4 Sign out Vehicle from Lot
  - 1.3.1 Locate Rental Agreement
  - 1.3.2 Process Return Information
  - 1.3.3 Process Payment
  - 1.3.4 Return Vehicle to Lot
  - 1.1.1.1 Get Location (pick-up/drop-off)
  - 1.1.1.2 Get Date/Time (pick-up/drop-off)
  - 1.1.1.3 Choose Vehicle
  - 1.1.1.4 Get Options Information
  - 1.1.1.5 Check Vehicle Availability
  - 1.1.1.6 Offer Rates for Selection
  - 1.1.2.1 Confirm Rental Information
  - 1.1.2.2 Get Customer Information
  - 1.1.2.3 Get Payment Information
  - 1.1.2.4 Confirm Reservation
  - 1.1.2.5 Create Reservation

EXAMPLE
For illustration only

The services in the Service Model is classified and organized into "Service Hierarchy" – Functional areas provide a useful business context for developing a classification scheme

Functional Areas Rental, Reservation, and Payment Processing are also good candidates for subsystems.

Rental

- 1.2 Check out Vehicle
- 1.3 Check in Vehicle
- 1.2.3 Create Rental Agreement
- 1.2.4 Sign out Vehicle from Lot
- 1.3.1 Locate Rental Agreement
- 1.3.2 Process Return Information
- 1.3.4 Return Vehicle to Lot

Payment Processing

(Note: This is a new Functional Area that was identified by evaluating the services in the Service Portfolio.)

- 1.3.3 Process Payment

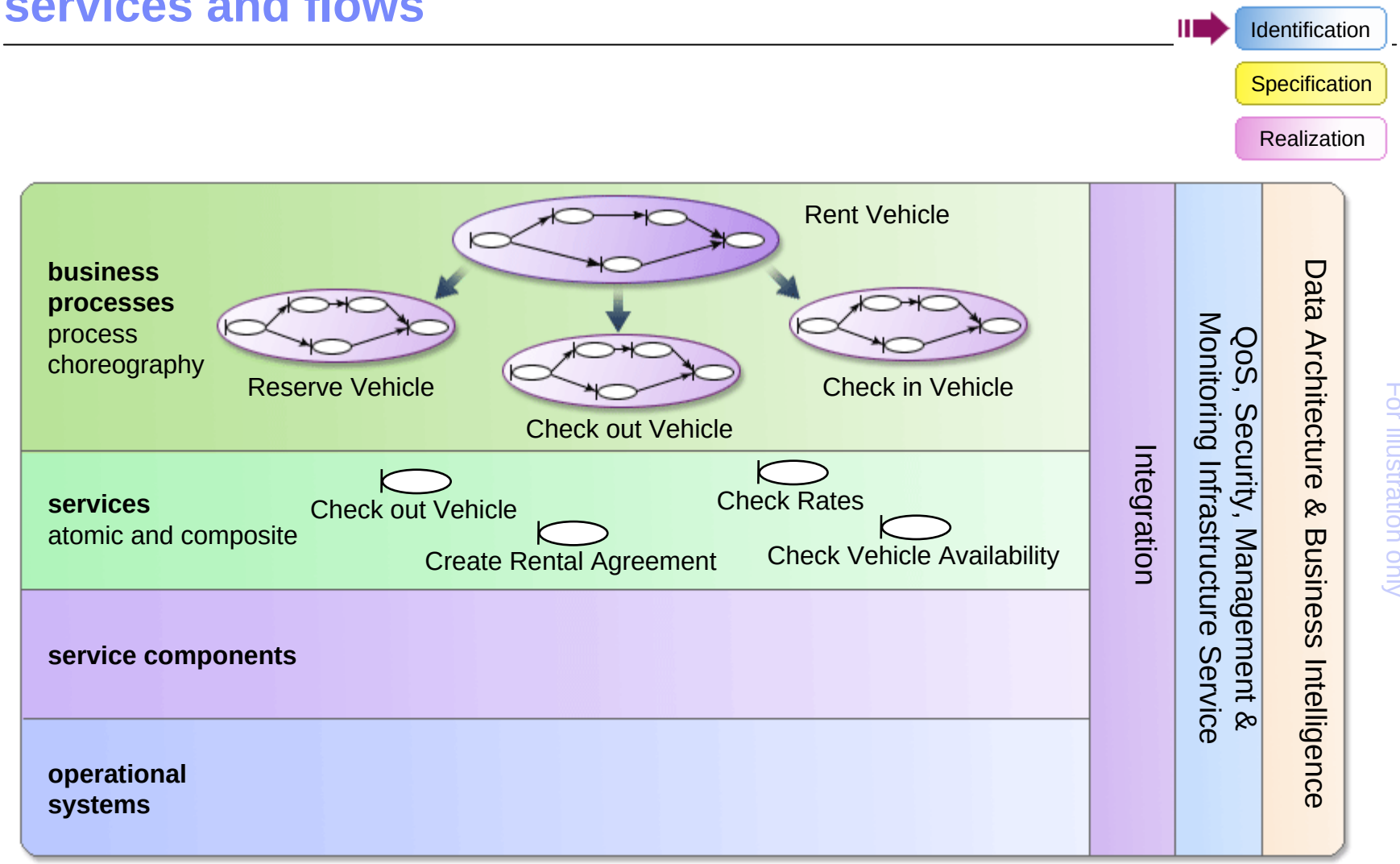
Reservation

- 1.1 Reserve Vehicle
- 1.1.1 Check Rates
- 1.1.2 Make Reservation
- 1.2.1 Locate Reservation
- 1.2.2 Modify Reservation
- 1.1.1.1 Get Location (pick-up/drop-off)
- 1.1.1.2 Get Date/Time (pick-up/drop-off)
- 1.1.1.3 Choose Vehicle
- 1.1.1.4 Get Options Information
- 1.1.1.5 Check Vehicle Availability
- 1.1.1.6 Offer Rates for Selection
- 1.1.2.1 Confirm Rental Information
- 1.1.2.2 Get Customer Information
- 1.1.2.3 Get Payment Information
- 1.1.2.4 Confirm Reservation
- 1.1.2.5 Create Reservation

Green Bold: Functional Area / Service Hierarchy Category

EXAMPLE
For illustration only

Rent-A-Car Top-Down Service Identification identifies services and flows



Goal-Service Model may identify new services

Goal-Service Model

- Increase Revenue by 20% by the End of FY2005
  - Introduce New Products
  - Introduce New Channels
  - Increase Revenue per Transaction
    - Up-sell Higher Class Vehicle
      - *Understand Customer Profile*
      - *Determine Types of Up-sell Vehicles*
      - **Check Vehicle Availability**
    - Cross-sell Additional Options
      - *Purchase Options Individually*
      - *Purchase Option Packages*
    - Cross-sell Partner Services
      - *Reserve Hotel Rooms*
      - *Reserve Airline Tickets*
      - *Book Destination Attractions*
      - *Sell Navigation Equipment Post-rental*

Service Model::Service Hierarchy

Up-cross-sell

- 2.1 Understand Customer Profile
- 2.2 Determine Types of Up-sell Vehicles
- 2.4 Purchase Options Individually
- 2.5 Purchase Option Packages
- 2.6 Reserve Hotel Rooms
- 2.7 Reserve Airline Tickets
- 2.8 Book Destination Attractions
- 2.9 Sell Navigation Equipment Post-rental

Reservation

- 1.1.1.5 Check Vehicle Availability

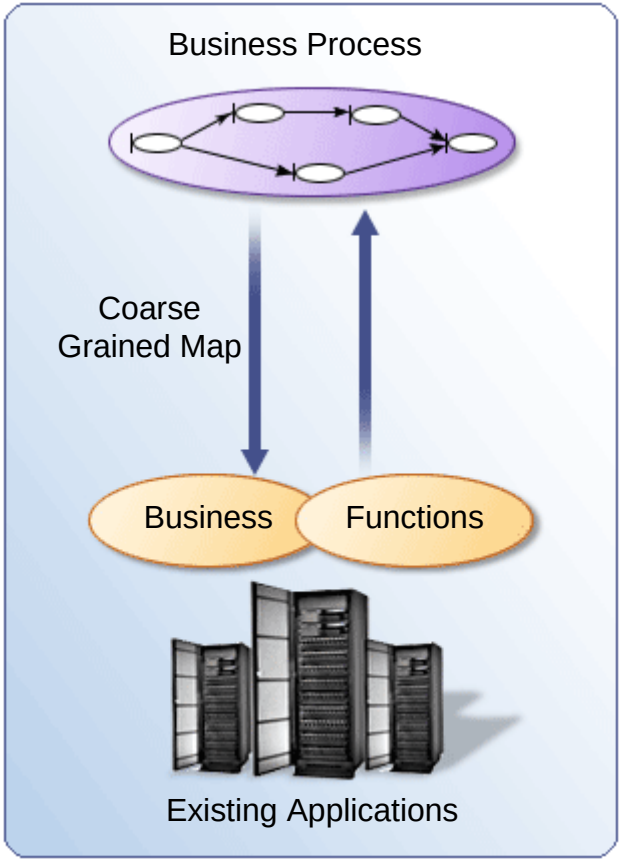
(Partial List)

EXAMPLE
For illustration only

Red Italics: Identified candidate service to be added to Service Model::Service Hierarchy
Bold Blue: Matches an existing service – no need to add

Existing Asset Analysis: Coarse-Grained Mapping of Candidate Services to Existing Applications

- Understand the business functions supported by each application
- Record attributes of existing applications in terms of technologies used, architectural styles, and so on
- Identify applications that perform common services



Rent-A-Car Existing Asset Analysis

- 3.1 Reservation
  - 3.1.1 Display Station Status
  - 3.1.2 Rate Shop
  - 3.1.3 Create/Display Reservation
  - 3.1.4 Display Rate Information
  - **3.1.5 Display Vehicle Availability**
  - 3.1.6 Display Station Information
  - 3.1.7 Create Account Number
  - ***3.1.8 Modify/Cancel Reservations***
  - 3.1.9 Verify Flight/Arrival Time
  - 3.1.10 Display Make Model Information
  - 3.1.11 Display Credit Cards Accepted at Location
  - 3.1.12 Display Manifest
- 3.2 Vehicle
  - 3.2.1 Vehicle Display
  - **3.2.2 Vehicle Availability**
  - 3.2.3 Daily Status Report
  - ***3.2.4 Vehicle Trace***
  - ***3.2.5 Vehicle Status***
  - 3.2.6 Ready-line
- 3.3 Customer
  - 3.3.1 Display Customer Profile
- 3.4 Additional Pre-Payment
- 3.5 Rapid Movement
- 3.6. Check-out
  - **3.6.1 Locate Reservations**
  - 3.6.2 Obtain Authorizations
  - 3.6.3 Locate Vehicles by Ready-line
  - **3.6.4 Up-selling**
  - 3.6.5 Delay Check-out Rentals
  - 3.6.6 Driver's License Verification (if Applicable)
  - 3.6.7 Risk Capabilities
  - 3.6.8 Coupon Processing
  - 3.6.9 Miscellaneous/One Way Fee Processing
- 3.7. Check-in
  - ***3.7.1 Display / Modify Rental Agreement***
  - ***3.7.2 Void Rental Agreement***
  - 3.7.3 Vehicle Exchange
  - 3.7.4 Obtain Authorizations
  - 3.7.5 Real-time Check-in
  - 3.7.6 Delay Check-in
  - 3.7.7 Miscellaneous/One Way Fee Processing
  - 3.7.8 Cash Calculation
  - 3.7.9 Coupon Processing
  - 3.7.10 Customer Survey
  - 3.7.11 Record Vehicle Damage
  - 3.7.12 Rates Processing

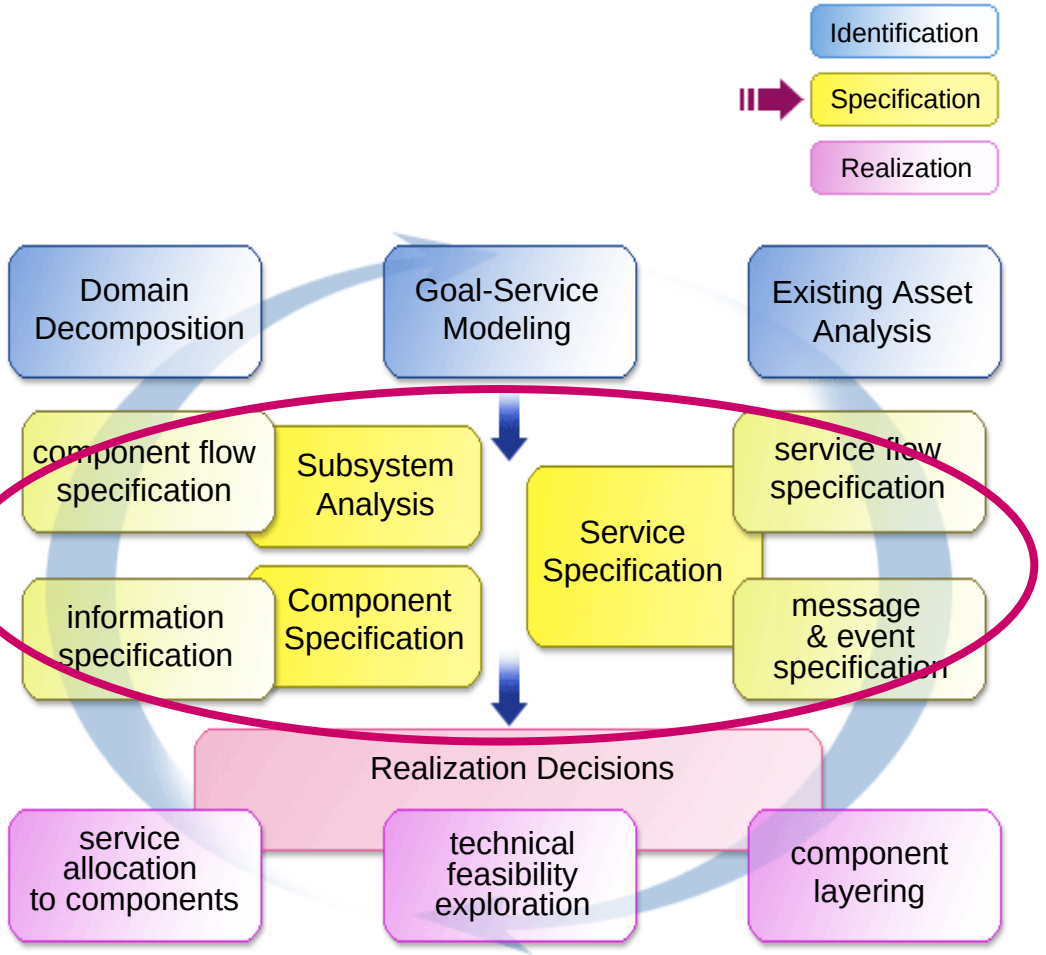
Red Bold Italics: New candidate service to be added to service portfolio.

Blue Bold: Already identified as a candidate service - likely to be of value during service realization.

Gray: Not relevant for this initiative. Identify these as quickly as possible to maximize focus on those that are relevant.

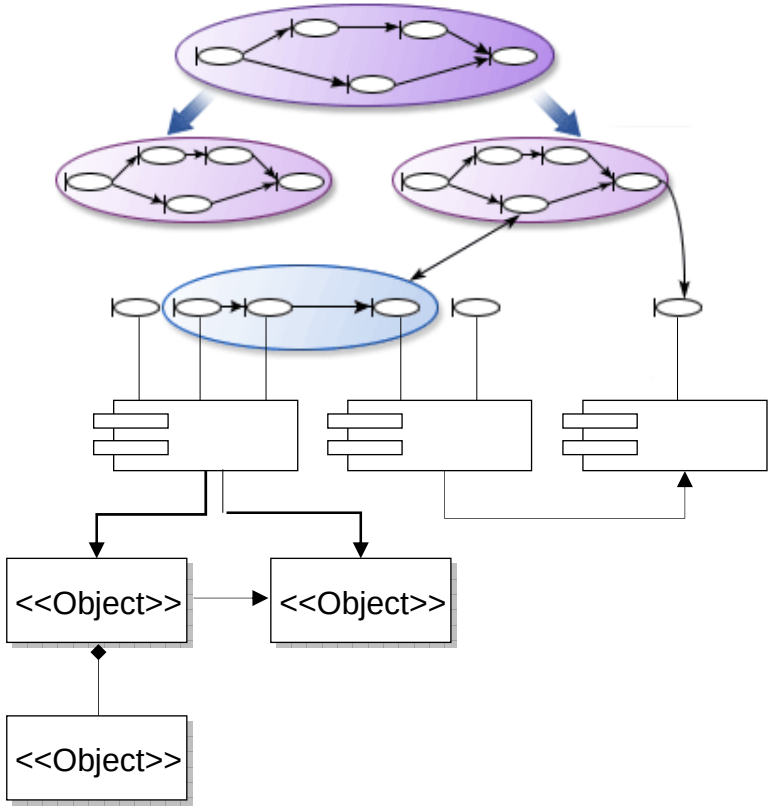
SOMA Specification Specifies Services, Service Components, and Flows

- Service Specification
  - Elaborates the *Service Model*, for example, service dependencies, composition and flow, non-functional requirements, service message specifications, design decisions, state management
  - Includes **Service Litmus Test** that "gate" service exposure decisions
- Subsystem Analysis
  - Partitioning into service components that will be responsible for service realization
- Component Specification
  - Detailed component modeling, flow, information architecture, and messages

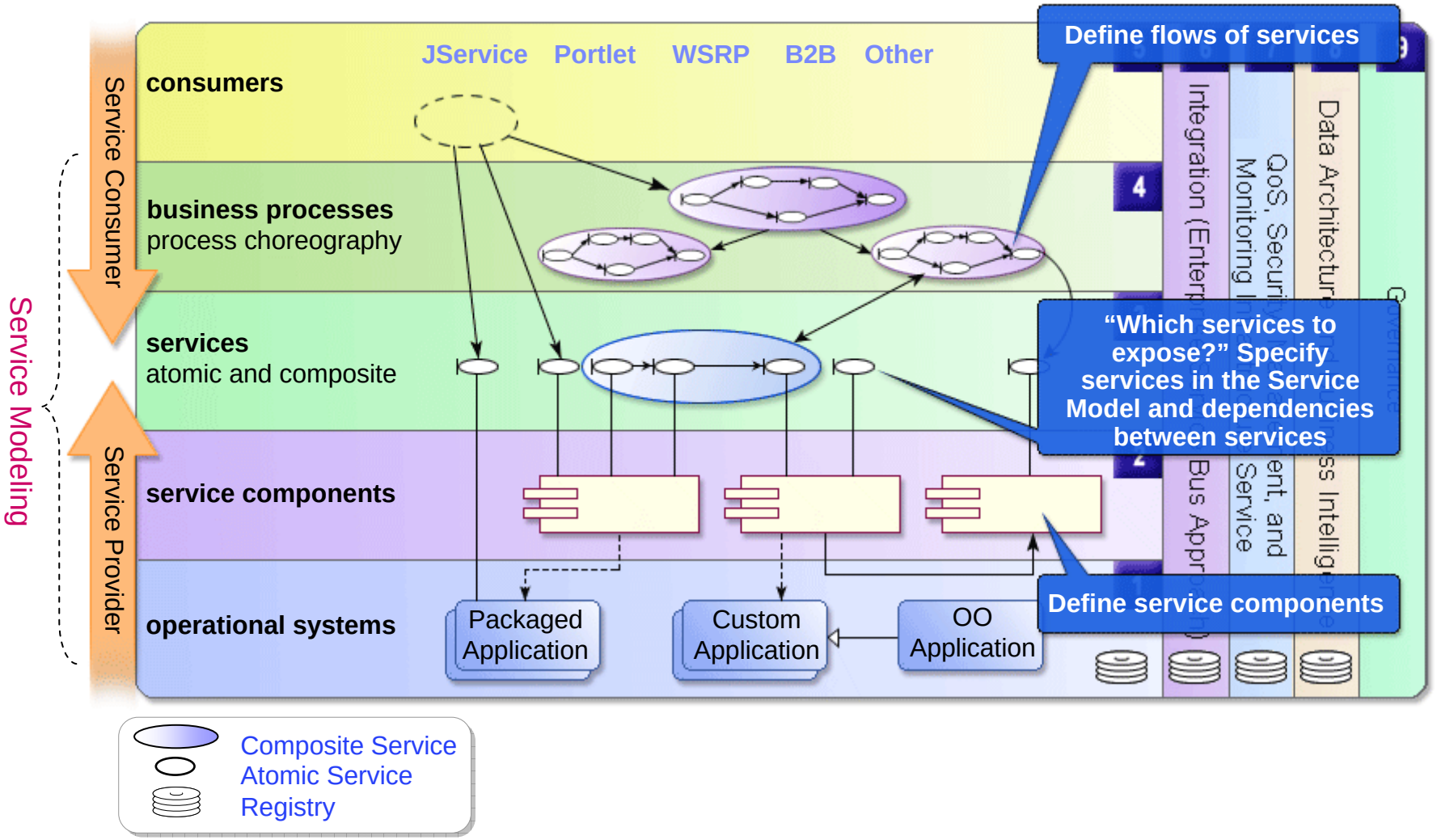


SOMA Specification Helps Design the Details of the Three First-class Constructs of SOA

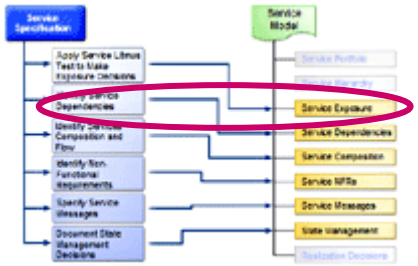
- **Services** – Use service litmus tests to decide which services to expose, and to detail their specifications
- **Service Components** – What large-grained components will realize and maintain QoS for the exposed services
- **Flows (Processes)** – Specify messages, choreographies, and information architecture required



Specification in Context

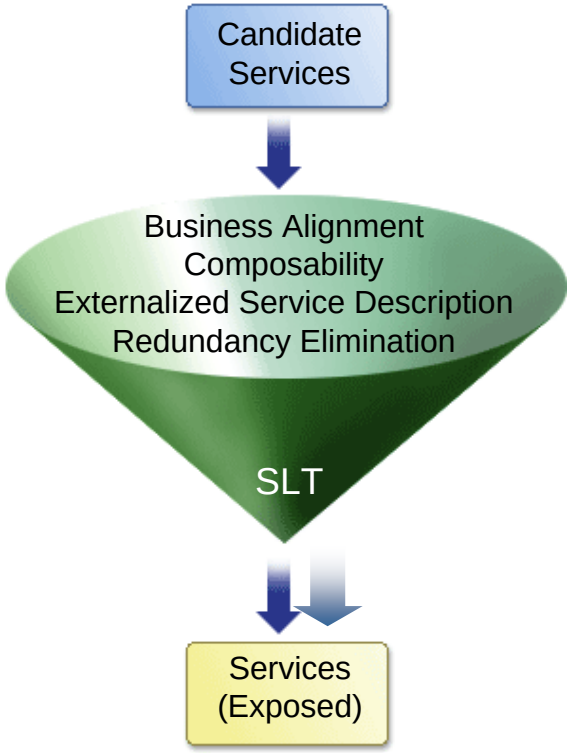


Service Litmus Test



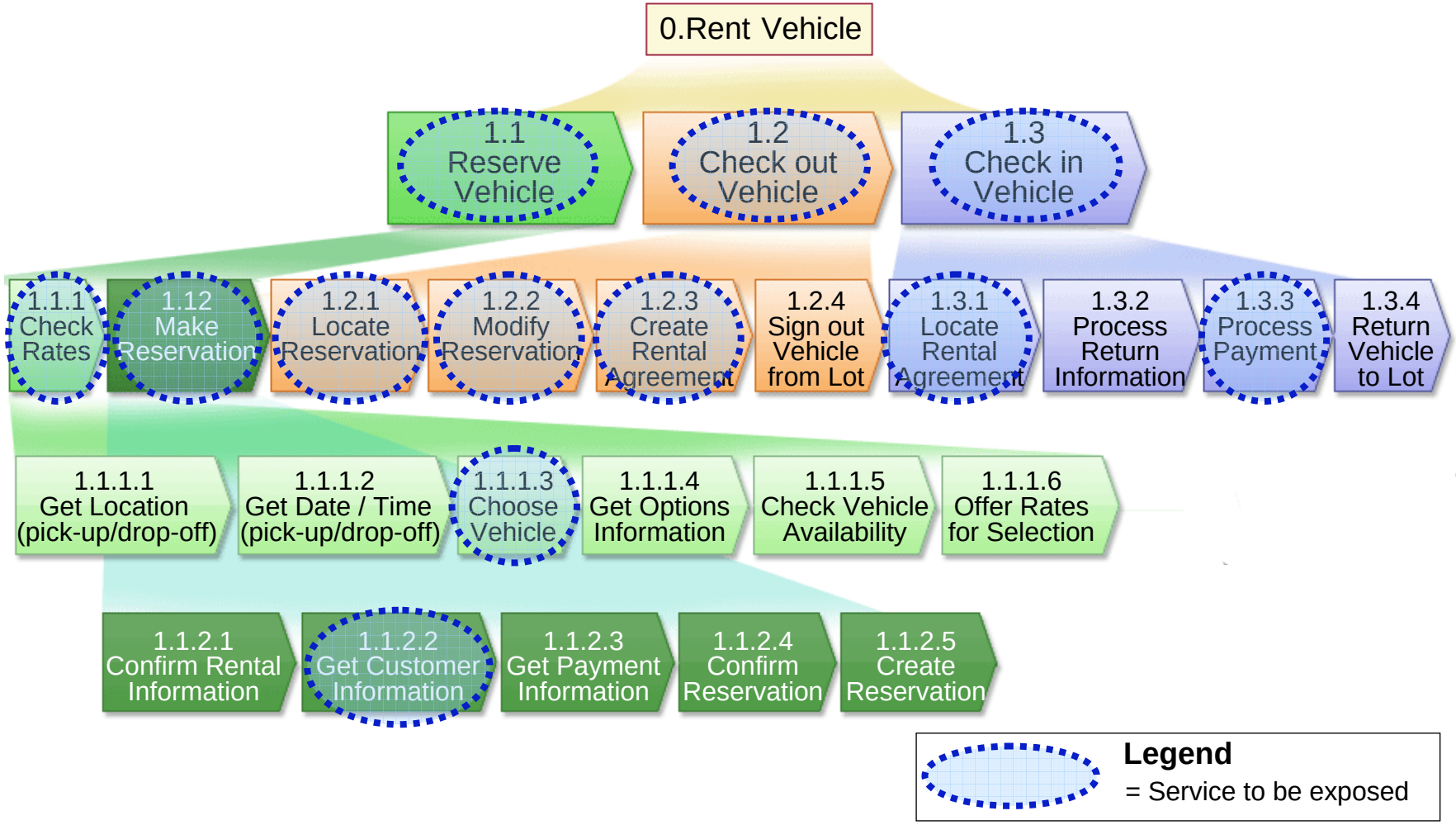
During the Service Specification, we make **service exposure decisions**: “From all the candidate services, which ones should we expose?”

- Not all candidate services should be exposed.
- Every implemented service has costs and risks.
- SOMA “**Service Litmus Test**” helps make exposure decisions.

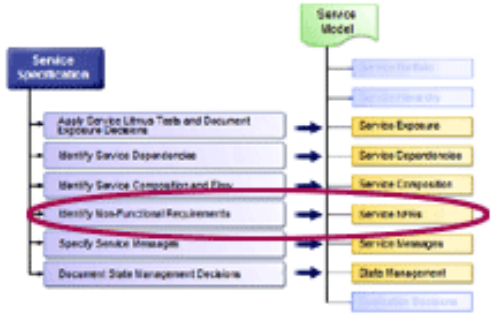


SLT 1-1 : Does the service provide a required unit of business functionality that supports business processes and goals?

Rent-A-Car Service Exposure Decisions reflected in the process



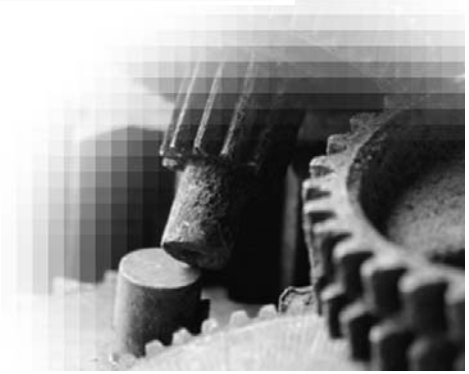
Non-Functional Requirements / QoS



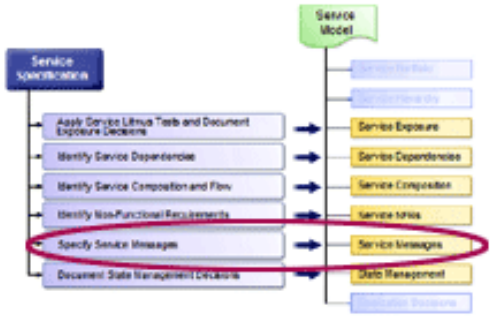
- SOA provides the opportunity to choose a service provider based not only on the functionality they provide, but also on the Quality of Service (QoS) they guarantee.
- Quality of Service is what has been achieved based on the non-functional requirements specified.

Service	NFR Type	NFR
Rent Vehicle	Availability	99.999%
Check Rates	Performance	Rates returned within 5 seconds

EXAMPLE
For illustration only



Message Specification - input, output and the internal message format

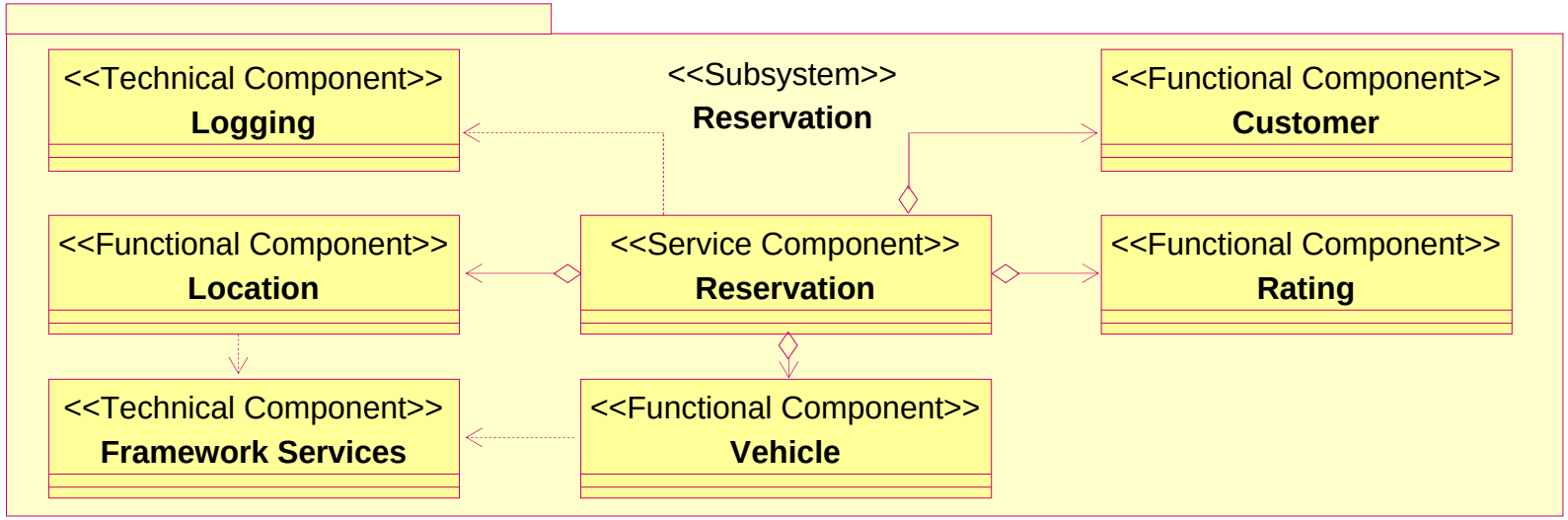


- The Service Model captures a high-level description of the messages associated with a service.
- Later engagement activities create a detailed message specification the form of a fixed-format, e.g. XML message. The message specification defined should be transport protocol (HTTP, MQ, JMS, FTP, SMTP, etc.) agnostic.
- Perspectives to take into consideration
  - Message standards defined at an industry, enterprise or application level
  - Appropriate meta or data model that are part of a data architecture
  - Message transformation standards that are part of an integration architecture.

EXAMPLE
For illustration only

Message Segment	Value
Service	ReservationServices
Topic	CreateReservation
Input Message	VehicleRequest
Output Message	Boolean flag showing success or failure

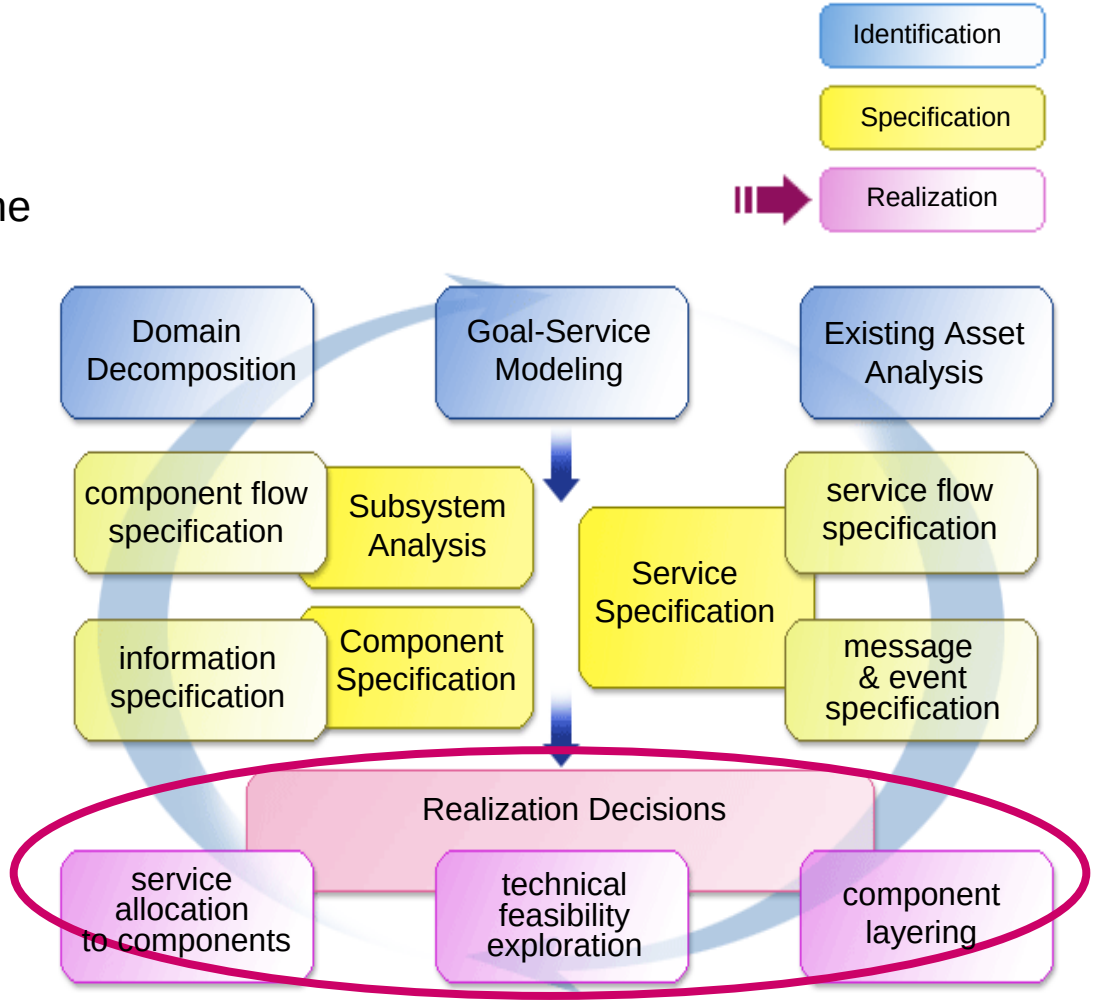
Service, Functional and Technical Components



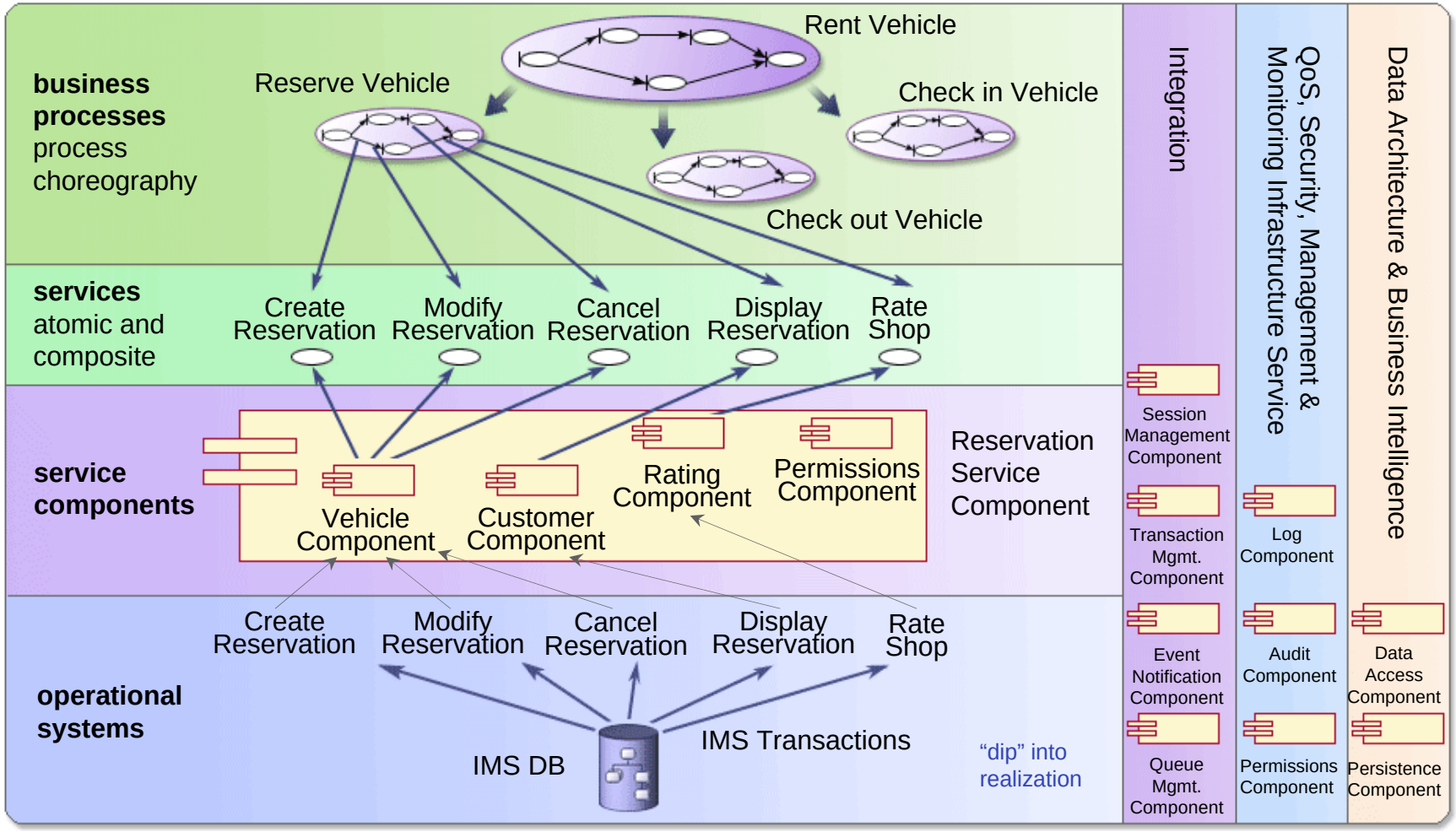
- A (coarser-grained) service component realizes the subsystem and has the responsibility of implementing the functionality of the services and delivering the QoS of the services it will realize. The services become the interfaces on the façade of the associated service component.

Realization Decisions Help Map Design to Actual Technologies

- Component Layering
  - Allocation of component to the application architecture layers
- Allocation of services to service components
- Technical Feasibility Exploration
- Realization Decisions with Justifications

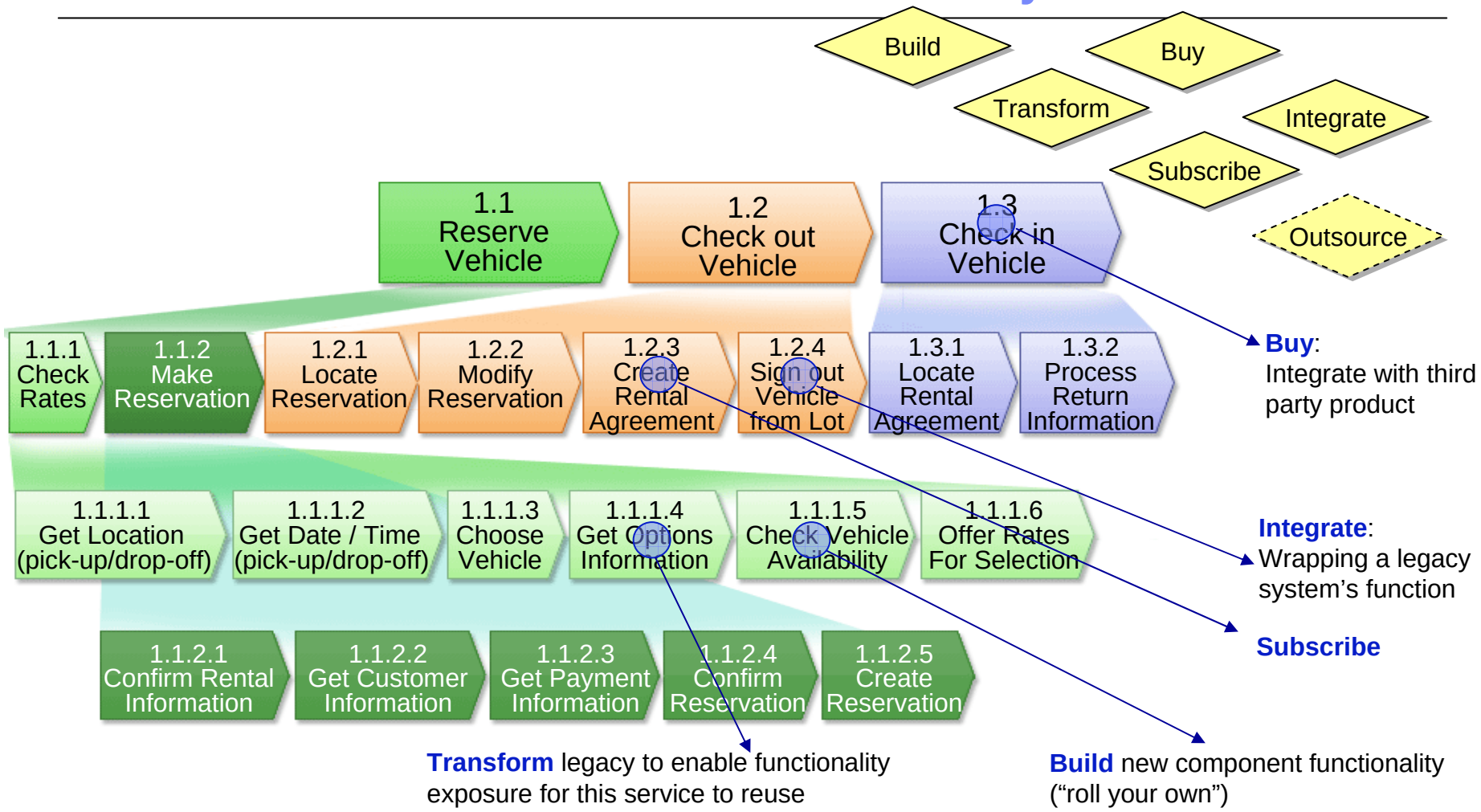


Allocation of Components to SOA Solution Stack Layers

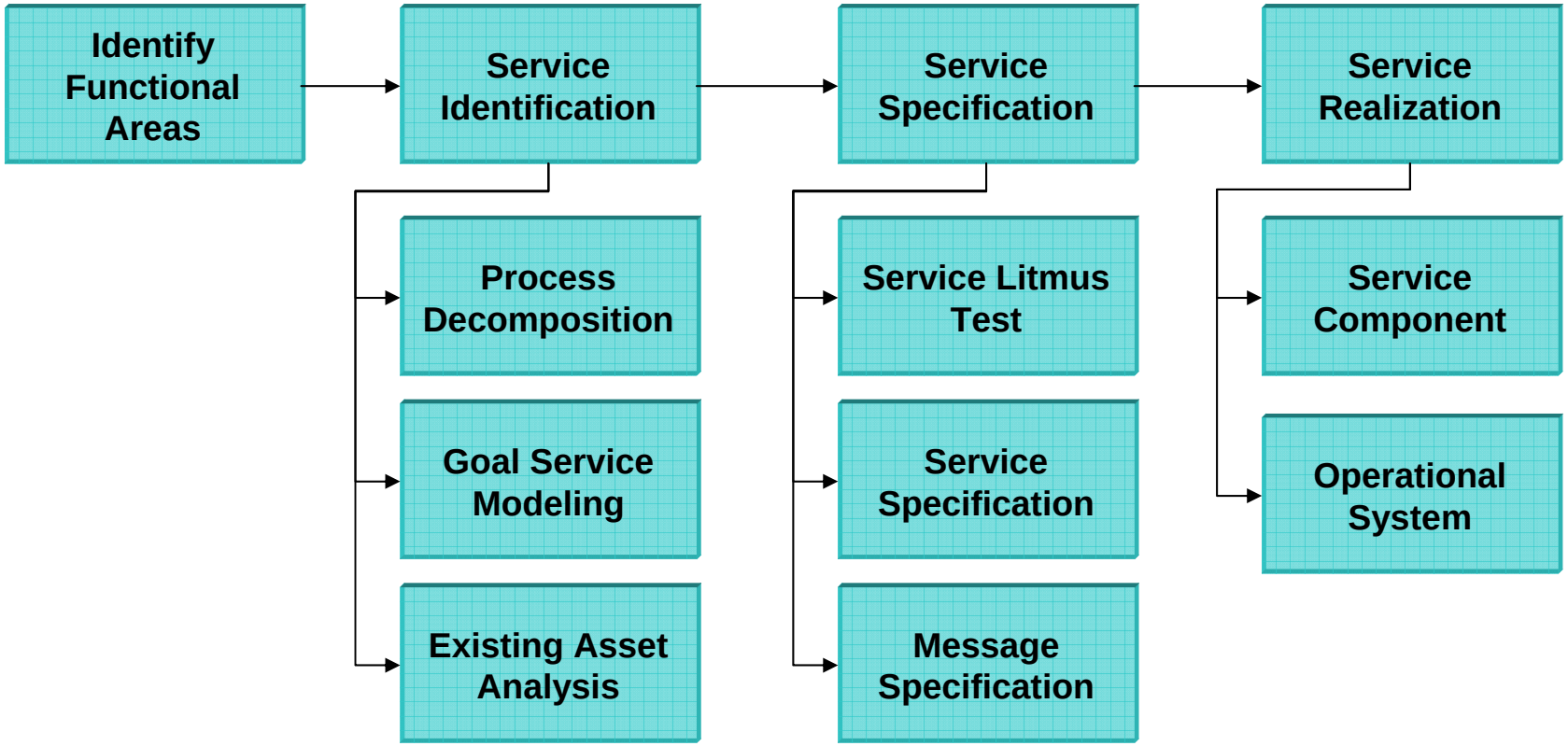


EXAMPLE
For illustration only

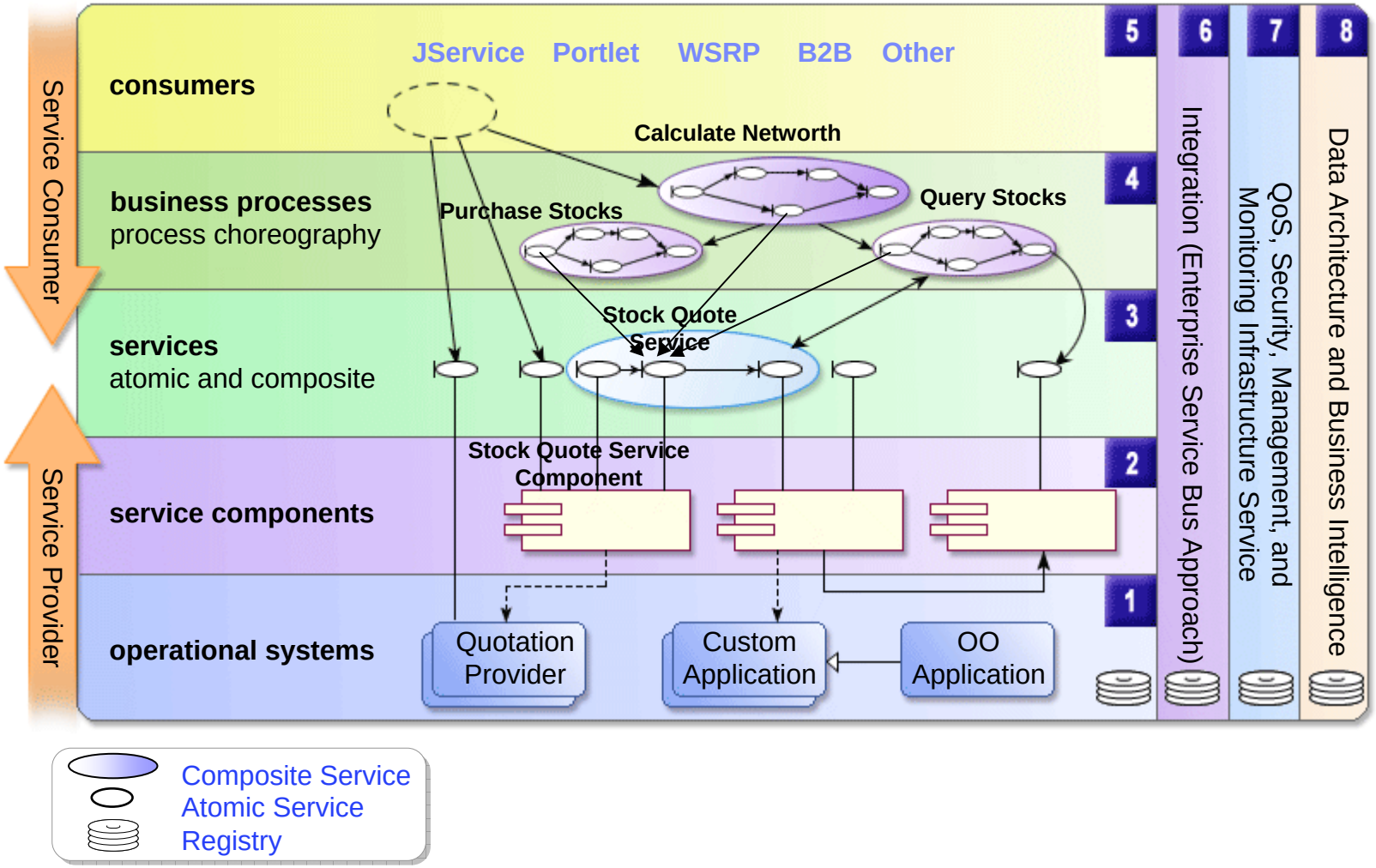
Realization can be achieved in different ways



Service Creation Summary

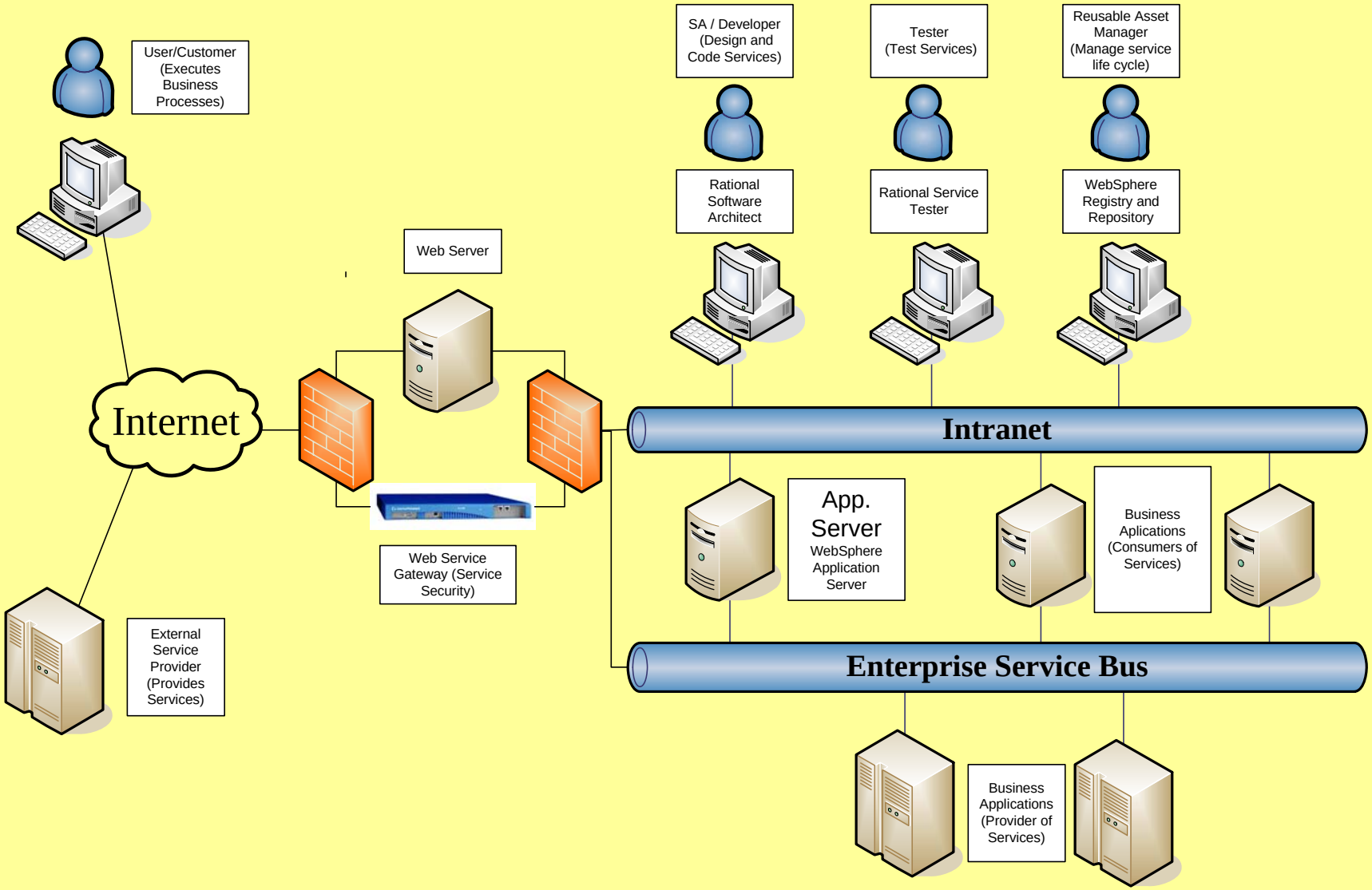


Demonstration: Development of a Stock Quote Service



Demonstration: Development of a Stock Quote Service

Architecture for Service Creation and Testing



Service Creation and Testing