

# ***Service Oriented Realization of The HTNG Reference Architecture***

***Version 0.6***

## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
June 24, 2008	0.1	First Draft with Structure	Chris Laffoon (IBM)
August 20, 2008	0.2	Updated structure and some initial content	Chris Laffoon Travis Grigsby (IBM)
November 10, 2008	0.3	Added content in chapters 3 and 4	Chris Laffoon Viswanath Srikanth (IBM) Gee Chia (IBM)
November 25, 2008	0.4	Split Ch. 4 and combined with chs. 3 & 5 Completed first draft of Ch.4 and 5 content	Chris Laffoon Viswanath Srikanth Gee Chia
December 2, 2008	0.5	Reorganized Ch.1 and cleaned up entire document.	Chris Laffoon Viswanath Srikanth Gee Chia
January 7, 2009	0.6	Minor changes made from reference architecture team recommendations	Chris Laffoon

# Contents

- 1 Service Orientation in Hospitality Industry ..... 3**
- 2 Increasing Service Orientation using HTNG Reference Architecture..... 4**
  - 2.1 Brief Overview of HTNG Reference Architecture ..... 4
  - 2.2 Taking a SOA approach to the HTNG Reference Architecture ..... 4
- 3 Jumpstarting SOA through Business Analysis ..... 5**
  - 3.1 Laying out the Business Domain ..... 5
  - 3.2 How to Identify High ROI Business Processes..... 6
  - 3.3 Modeling Business Processes ..... 7
  - 3.4 Realizing Business Processes through Service Choreography..... 8
- 4 Business Application Services for an Agile Business ..... 10**
  - 4.1 Service Characteristics ..... 10
  - 4.2 A Word on Service Granularity ..... 10
  - 4.3 Identifying Business Application Services ..... 11
  - 4.4 Specifying Services..... 14
  - 4.5 Considerations on Realizing and Deploying Services ..... 16
  - 4.6 Guide to Exposing Services from Packaged Applications..... 16
- 5 Ensuring Success through SOA Governance ..... 19**

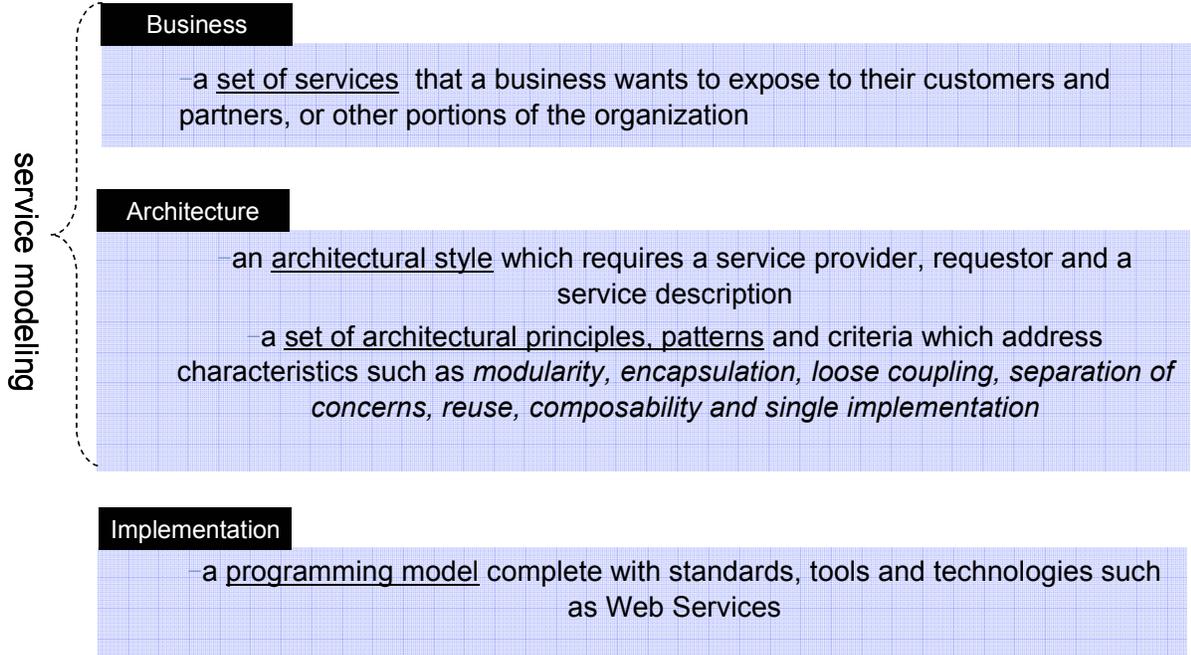
# 1 Service Orientation in Hospitality Industry

Service Oriented Architecture or SOA is an architectural principle that offers the prospect of better alignment of business and Information Technology in the Hospitality Industry. Change is a constant in the business, and being able to respond to change without investing considerable time and effort in modifying IT systems is a huge benefit of SOA. Furthermore, Service Oriented Architecture enables the Hotelier to not only transform internal systems to be more service oriented, but also permits collaboration amongst business partners as well as outsourcing of useful business functions in a seamless, non-disruptive fashion.

Before describing how a Hotelier can increase the service orientation of their IT systems, it is beneficial to look at the different viewpoints into Service Orientation. Broadly, there is the:

1. Business View of Service Orientation
2. Architectural View of Service Orientation
3. Implementation View of Service Orientation

as shown in Figure 1 below.



**Figure 1 – Different Viewpoints of Service Orientation**

This document will speak to both the Business View through Business Domain Analysis and Business Process Analysis, and the Architectural View through sections on creating Business Application Services and Governance. The actual implementation of Services and Business Processes is briefly touched upon, but not dealt with in detail in this document.

## **2 Increasing Service Orientation using HTNG Reference Architecture**

### **2.1 Brief Overview of HTNG Reference Architecture**

The HTNG Reference Architecture, which this document serves as an addendum to, has been developed in the Architecture Workgroup by numerous hoteliers and vendors. This document will reference much of the content found in the Reference Architecture, making it valuable to be familiar with the Reference Architecture.

The HTNG Reference Architecture is split into three different parts: Business Architecture, Application Architecture, and Data Architecture. Content from all three, such as business processes, data definitions, and functional capabilities, will be used throughout this document.

### **2.2 Taking a SOA approach to the HTNG Reference Architecture**

One of the pillars of Service Orientation is alignment between the architecture and the business. In order to achieve this alignment, it is critical that both Business and Information Technology are speaking the same language. HTNG's Reference Architecture for Hospitality helps with this critical piece by defining industry specific vocabulary, including common business processes, data definitions, and functional capabilities.

This asset alone brings great value to the industry by helping vendors and hoteliers to speak the same language and share a common understanding of the industry. The remainder of this document builds on this foundational reference architecture, providing a starting point for hoteliers building service oriented architectures and for vendors building applications that will plug into these architectures in a more efficient and predictable way.

After analyzing the Business Domain and revisiting the common business processes identified in the HTNG Reference Architecture, this document will focus on decomposing these business processes to identify business application services, which will be valuable for the integration of vendor applications with hoteliers' service oriented architectures.

In addition to a business domain analysis and an overview of business application services, this document will conclude by looking at how to ensure success during transformation through life cycle management and SOA governance.

### 3 Jumpstarting SOA through Business Analysis

A first important step to developing increased service-oriented enterprise architecture is analyzing the business. This includes describing the business domains and core business processes. After looking at a reference business domain analysis, taken from the HTNG Reference Architecture, this chapter will focus on how to get started with Service Oriented Architecture, including how to identify the higher ROI business processes to optimize.

#### 3.1 Laying out the Business Domain

The following diagram (Figure 2) lays out a typical hotel business by describing core domains (Guest Services, Facilities, etc.) and the key business processes within those domains. This breakdown comes directly from the Business Architecture of the HTNG Reference Architecture V1.0, which is initially focused on a limited service, independent, US-based hotel. In that document, descriptions for each of these domains and business processes can be found.



Figure 2 – Core Business Domains and Processes

The Business Architecture, represented by Figure 2, helps with increasing Service Orientation in an Enterprise Architecture by helping to identify who owns each set of services. Traditionally, Business Processes have been owned and managed by one organization or division within the business. However, in an effort to realize the benefits of service orientation, services will be

developed that will be used by more than one business process and from more than one business domain. With a focus on reuse and agility, service ownership will become as important as business process ownership. For services, ownership includes defining service contracts, managing costs and fees, and thinking about scalability and the use of the services.

In addition to helping identify owners and developing proper governance procedures, this Business Architecture can serve as an important tool for those in the line of business to communicate how they think about the business. This will help create a more formal contract on what Information Technology organizations must support and align with.

## 3.2 How to Identify High ROI Business Processes

The first challenge an enterprise faces in driving a business towards increased service orientation adoption is how to get started. A good way to look at commencing the effort is to prioritize the problem areas for investment, demonstrate some incremental success in adopting service orientation in some aspects of business, and receive a return on that investment.

Service Oriented Architecture entry points help businesses pursue SOA the right way based on the business value and priority. These entry points are specific approaches for getting started and are driven by both business and IT needs.

The three primary business-centric entry points, applying directly to the tasks businesses perform to produce value for customers, are as follows:

- **People:** The people entry point focuses on enabling people to interact with application and services that support business processes to drive greater productivity and collaboration.
- **Process:** The process entry point focuses on streamlining business processes to align business and IT goals.
- **Information:** The information entry point leverages information in a consistent and visible way to allow timely information available for making decisions that are critical to success of a business.

Another common SOA entry point is the Re-use entry point. This entry point focuses on deriving value from previous business-centric services investments. By building from existing resources one can streamline business processes, ensure consistency throughout the company, reduce duplicate functionality, and cut development time.

For this document, the Process entry point will be the focus; this is a convenient way to start SOA adoption by transforming one business process at a time.

### 3.2.1 Process Entry Point

A business is comprised of many processes. Investing in optimizing and streamlining those processes offers a promising ROI with a clear focus on meeting business goals. With SOA, business processes can be realized as choreography of distinct, reusable services, which can be re-combined to address the changing business priorities and support new business processes.

To identify high ROI business processes, first look for areas of problems as well as opportunities to improve and optimize business processes that could lead to a positive return. The candidate business processes identified may be prioritized into different phases for transformation:

**Phase I:** Looking for business processes where pain-points can be or have been identified. These are the challenges that are potentially attributed to inefficient processes. Examples of obvious red-flag are:

- Loss of employee productivity/time
- Loss of new/existing business opportunity
- Failure to produce enough to satisfy the demand
- Failure to keep pace with competitors

**Phase II:** Looking for business processes that could benefit from improvement and optimization:

- Is there a method to monitor business activity for making informed business decisions on the success of a process, as well as recognizing problem areas and responding to them quickly?
- Can one change processes quickly to remain competitive?
- Can one rapidly develop automated processes?
- Can one monitor results and performance to optimize processes?
- Is stored data being utilized for business benefit?

**Phase III:** Identifying business processes that can benefit from increasing business partner relationships. This includes increasing collaboration among current strategic partners and finding new ways to strategically outsource non-core operations to new or existing partners.

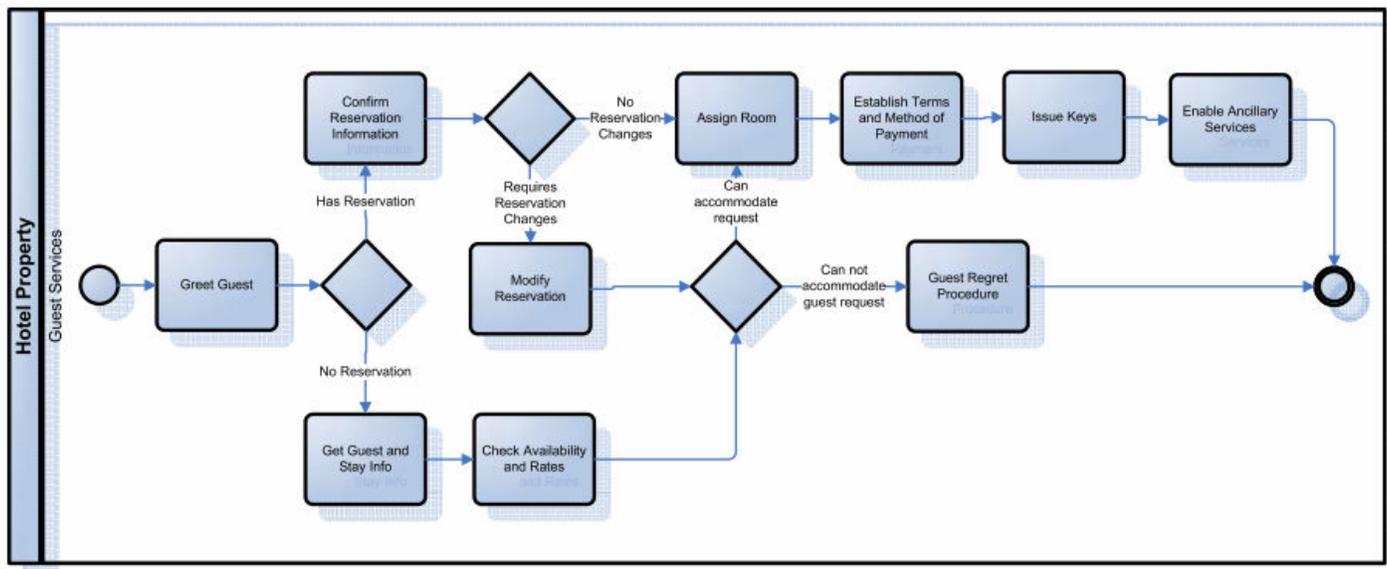
The process entry point focuses on building services that needs better business processes automation or monitoring. During the design of new, improved business processes, redundant processes should be eliminated where possible and similar processes should be merged into singularly integrated processes. The services developed for a business process can be reused and possibly combined to develop other business processes to fulfill business goals.

### 3.3 Modeling Business Processes

Modeling Business Processes is critical for optimization. First, modeling the existing processes will help determine areas for improvement. Second, modeling the new, optimized business processes is critical to developing a service oriented realization of this process, as will be described in the next section. While modeling these new optimized business processes, the HTNG Reference Architecture Business Processes, modeled for each process in the Business Architecture, can serve as an excellent starting point and guide.

There are many tools, patterns, and standards to assist in modeling business processes. One of the most useful standards is the Business Process Modeling Notation (BPMN). This graphical notation standard describes how business processes should be represented in models. As an example, Figure 3 below is a BPMN representation of the HTNG Reference Architecture standardized Check-In Process.

## HTNG Check In Process



**Figure 3 – HTNG Reference Architecture Check-In Process in BPMN**

BPMN 1.1 spec is managed by the Object Management Group (OMG) and tutorials can be found at <http://www.bpmn.org/>.

### 3.4 Realizing Business Processes through Service Choreography

In a mature Service Oriented style of Architecture, business processes are realized through business application and infrastructure services. Often, a business process, modeled with BPMN for example, is transformed into an execution language, like Business Process Execution Language (BPEL), where each task or activity is mapped to different services that provide the necessary capabilities.

The following diagram (Figure 4) attempts to show how business process choreography and supporting services are deployed. The top layer contains the typical touch points found at a hotel property; these are often connected through interaction services. The business processes themselves are then composed and choreographed in the second layer. As described above, the processes are then realized by composing together services found in the services layer. These services can be exposed by new service components or components from existing operational systems. The wires on the diagram give an example of how Kiosk Check-in functionality would be realized through a service oriented architecture of this nature.

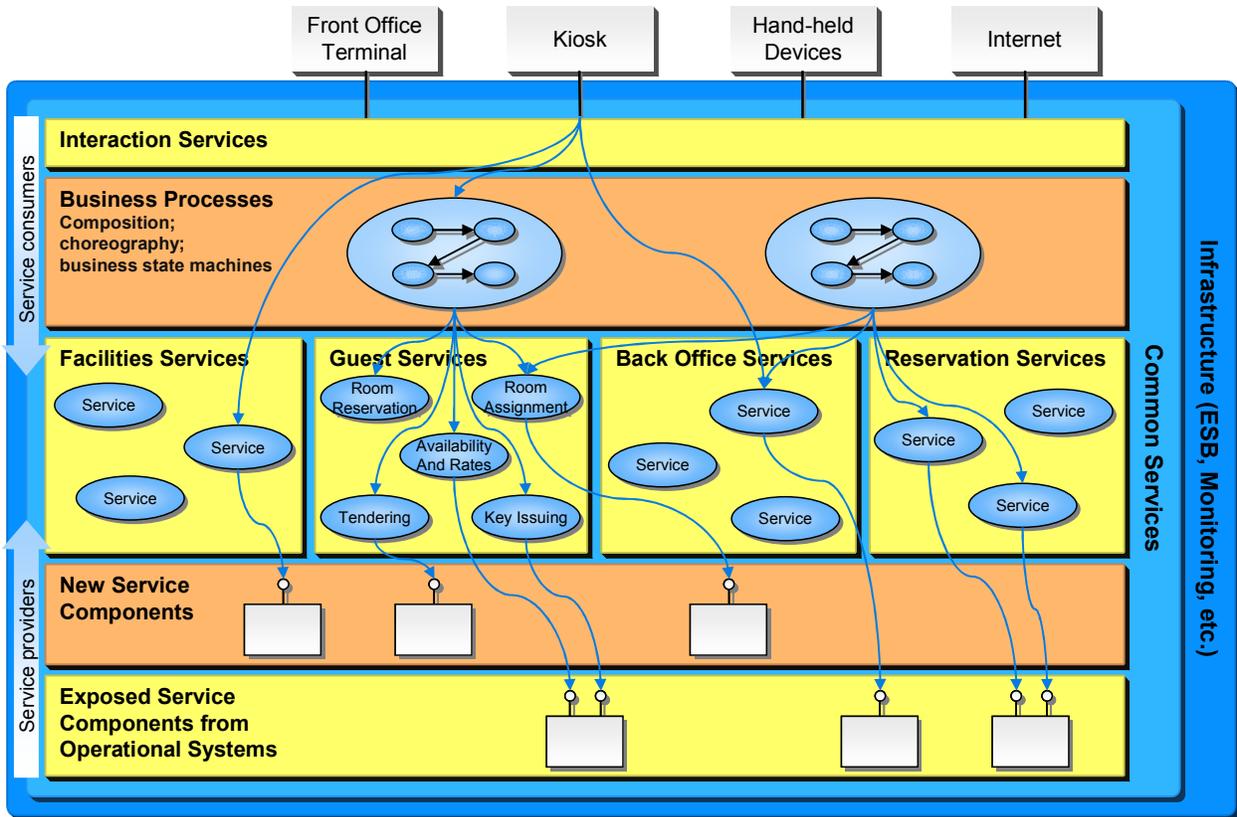


Figure 4 – Processes and Services View

## 4 Business Application Services for an Agile Business

A Business Application Service is the most fundamental unit of a service-oriented solution that is packaged as a reusable component for use in a business process.

### 4.1 Service Characteristics

There are a set of design characteristics that are important in a “well-defined” service. One of the more complete sets of Service characteristics has been captured by Thomas Erl in his book *SOA Principles of Service Design* and is listed below for reference:

- **Standardized service contract** - Services within the same service inventory are in compliance with the same contract design standards.
- **Service loose coupling** - Service contract impose low consumer coupling requirements and are themselves decoupled from their surrounding environment.
- **Service abstraction** - Service contract only contains essential information and information about services is limited to what is published in service contract.
- **Service reusability** - Services contain and express agnostic logic and can be positioned as reusable enterprise resources.
- **Service autonomy** - Services exercise a high level of control over the underlying runtime execution environment.
- **Service statelessness** - Services minimize resource consumption by deferring the management of state information when necessary.
- **Service discoverability** - Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.
- **Service composability** - Services are effectively composition participants, regardless of the size and complexity of the composition.

By minimizing service dependencies and freeing service consumers from having to couple to underlying service implementation, service providers can accommodate changes more efficiently with minimal impact to the rest of the enterprise, as long as the defined contract stays the same.

The fundamental benefits of “well-defined” services are to facilitate interoperability of service interaction and to enable flexibility and reuse of business functionality within an enterprise and with partners. This establishes an environment wherein services produced by different projects at different times can be repeatedly assembled together to help automate a range of business tasks for addressing new business opportunities or changing business priorities.

### 4.2 A Word on Service Granularity

Service granularity is defined by the functional context of the whole service. Services that are intended to cover a larger functional context are considered to have a coarse granularity. On the contrary, fine-grained services expose a narrow specialized functionality.

Looking from the perspective of the type of Service under consideration, you could have:

- **Utility Services (Infrastructure Services)**, usually defined by grouping together infrastructural capabilities with a common purpose, for example Logging Service, and thus granularity is directly defined by the utility functions supported.
- **Entity services** which have their functional context scoped by the entities that they manage. For example, granularity of the Guest Service is defined by the Guest entity. In this case Guest Service would include all the capabilities that are necessary to maintain Guest entity.
- **Task services**, which typically contain a group of capabilities related to the same business task, for example Tax Calculation Service. Task services tend to be fairly granular.
- **Process services**, on the other hand, usually deal with a larger functional context defined by the encapsulated business process. For example, Guest Itinerary Management Service would include all the capabilities necessary to manage a guest's itinerary.

### 4.3 Identifying Business Application Services

Business application services are those services that perform a business function and are specific for a particular industry or business. From the granularities discussion in section 4.2, business application services would be primarily entity, task, and process service types. Utility services are more common and cross-industry.

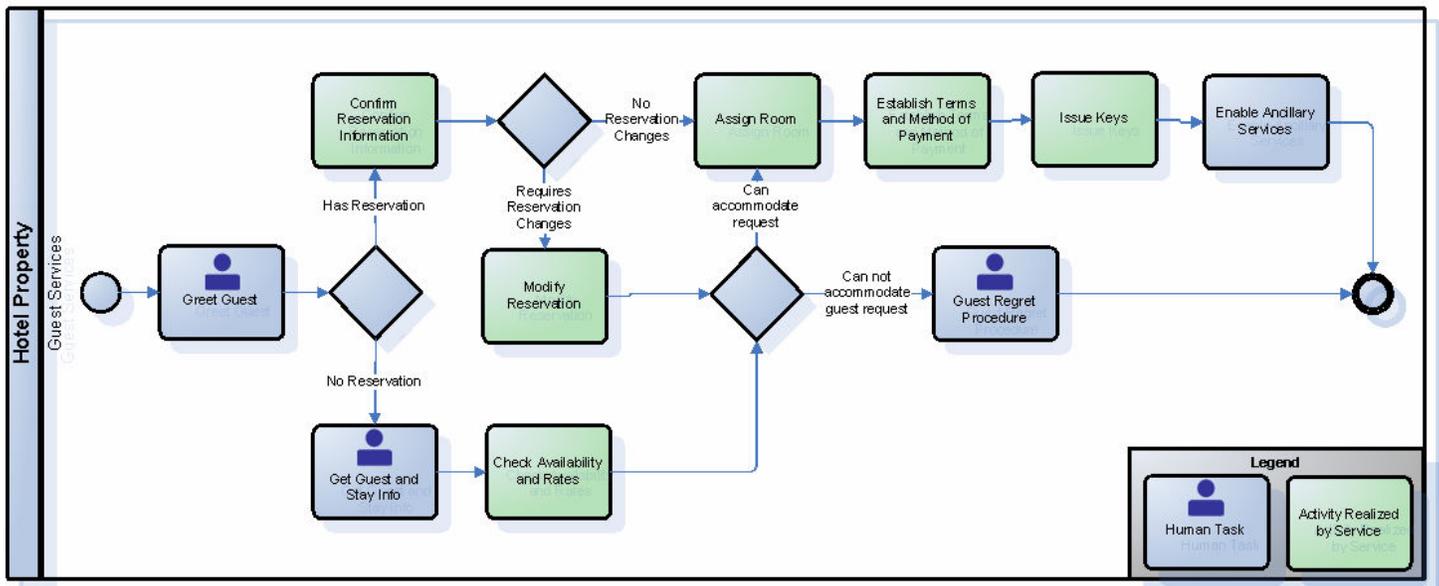
Although business application services can be identified through various entry points, the most common approach is through business process decomposition.

#### 4.3.1 Identifying Candidate Services through Business Process Decomposition

This section will describe how to take the business processes identified in Ch.3 as well as the service characteristics described above in section 4.1 and identify candidate business application services. Although this section will not outline all of the candidate business application services for each process defined in the Business Architecture, the example process chosen could be a reference on how this process and output can be extended to other processes.

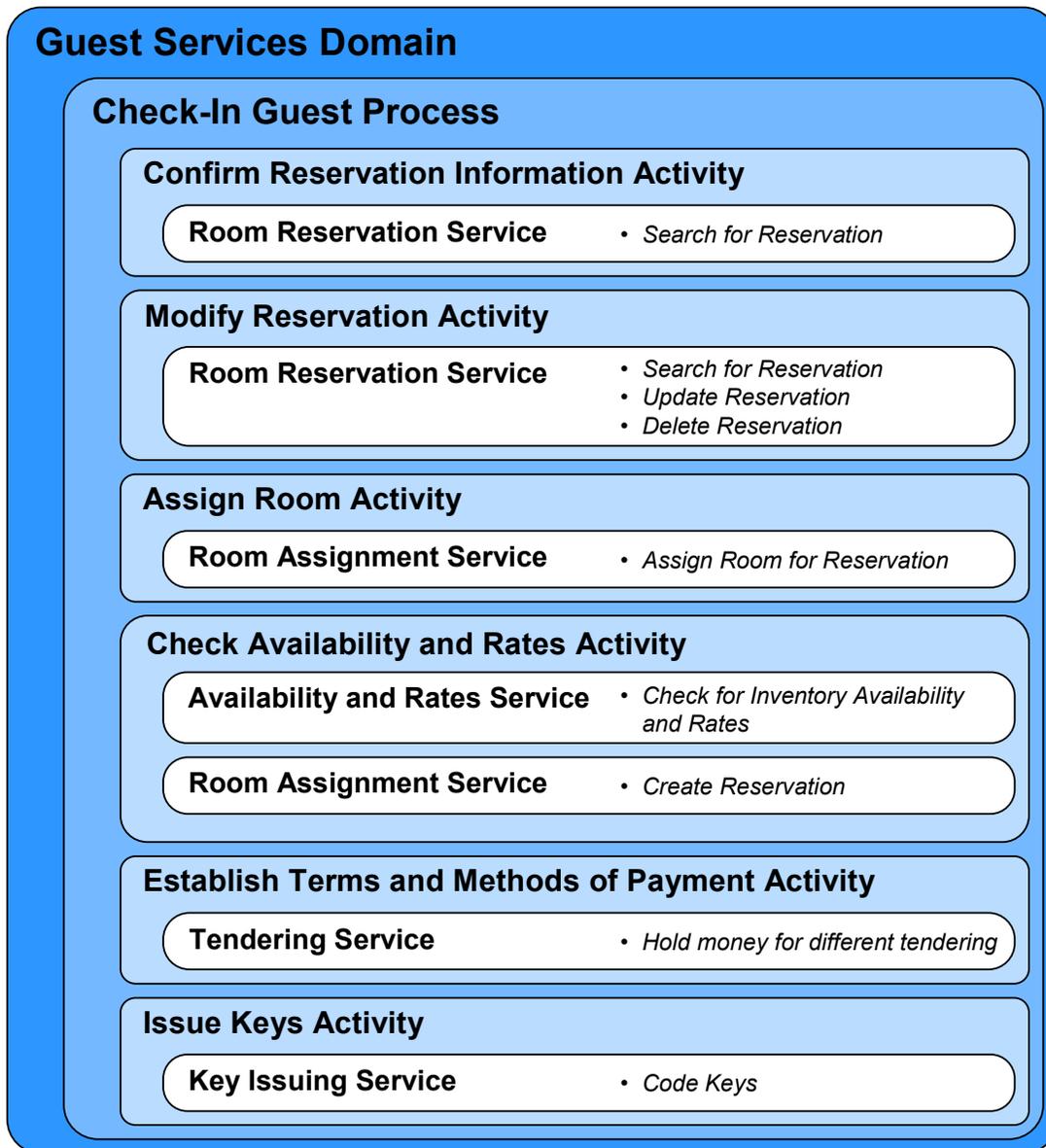
For this exercise, the HTNG Check-In Business Process was chosen because of its core nature to hotels and its overlap with the Reservation Business Domain. The first step after modeling the process in BPMN (as described in section 3.3) is identifying the tasks and activities. Some of the tasks will need to be supported with business applications services, while others do not. These differences are depicted in the updated business process model below (Figure 5).

## HTNG Check In Process



**Figure 5 – Updated HTNG Check-In Process BPMN Diagram**

For the various activities in the process diagram in Figure 5 that are supported by one or more business application services, the next step is to identify candidate services and service capabilities. In this step, the HTNG Application Architecture is referenced to learn more about the different groupings of capabilities needed to support a particular process. These capabilities, in addition to the logical functionality needed for each activity, lead to a first list of candidate services for the Check-In Guest Process. The decomposition chart in Figure 6 describes the different services identified for the Check-In process, including the capabilities needed, for each activity from the process model.



**Figure 6- Check in Guest Process Decomposition**

### 4.3.2 Prioritizing Candidate Services

Following completion of business process to business application services decomposition for all relevant business processes identified earlier as being considered for optimization, the output is a list of candidate services and service capabilities needed to support these business processes. It is foreseeable that creating all these services is not a very realistic goal to achieve timely and significant ROI. Therefore, it is important to be able to step back and prioritize which services to implement based on some important strategic criteria. Below are some key criteria recommended to consider when attempting to prioritize which candidate business application services to implement:

- **Multiple Consumers** – Would this candidate service have more than one consumer?

- **Spanning Domains** – Would this candidate service span multiple domains?
- **Business timely information** – Would this candidate service need to deliver timely business information?
- **Business alignment / cost-effective** – Would this candidate service be cost effective to maintain and govern given its relative alignment with the business?

As a means of demonstrating how to use these criteria, qualitative analysis for a candidate service like “Room Reservation Service” can be performed. Such an analysis might end up with a results table that illustrates how the service met each of the criteria, like the example below in Figure 7 for “Room Reservation Service”.

### Room Reservation Service

Multiple Consumers	Excellent	Different consumers would definitely need this service: front desk, web portal, self service, etc.
Spanning Domains	Excellent	This service would span different domains; specifically, the Guest Services and Reservation domains
Business Timely Information	High	This service would need to provide timely and accurate information that is relevant to the business
Business Alignment / Cost-effective	Medium	This service might be a little costly, but would definitely align with the business priorities

**Figure 7 – Room Reservation Service Prioritization Criteria**

The result of this analysis for this particular candidate service indicates that this service is likely to be chosen for development. The next section will discuss specifying services and will be based on this particular “Room Reservation Service”.

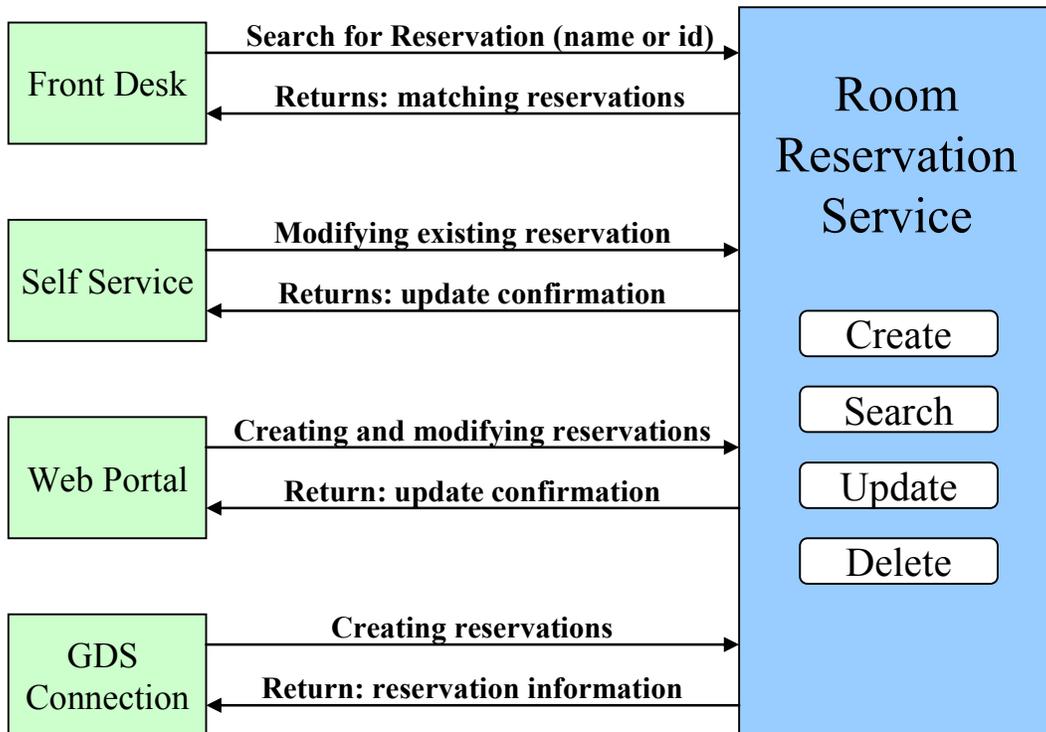
## 4.4 Specifying Services

After identifying services to be developed as described in section 4.3, the next step is start specifying those services. The first step, like in any good software engineering methodology, is to analyze the service that needs to be developed. This analysis primarily consists of defining various types of requirements:

- **Functional Requirements** – defined through Use Case descriptions
- **Non Functional Requirements** – some examples of these are...
  - Logging

- Security
- Availability
- Composability

In addition to the traditional requirements, it is important to think about the consumers for these services, given the importance of reuse in a Service Oriented Architecture. Figure 8 below is an example of how one could describe the different consumers of a particular service. This type of diagram helps significantly when designing, realizing, and ultimately deploying these services.



**Figure 8 – Context Diagram for Room Reservation Service**

While this analysis is being performed, an obvious question will arise whether a service should be implemented from scratch or should expose existing capability from packaged applications, perhaps already deployed. The answer to this question will be important and will significantly influence how the service is implemented. An entire section (Section 4.6) is included below as a guide to exposing services from packaged applications.

Following this decision, a high level design of how a service that will be implemented and realized is created. This high level design includes at least two important pieces:

- Detailed Interface and Contract
- Design of the Messages

When creating this high level design, it is critical to think about what messaging and interface standards could be leveraged. These messages could be ones being developed inside of HTNG or potentially within OpenTravel. For example, the Room Reservation Service first introduced in section 4.3 could potentially leverage Reservation Messaging standards from OpenTravel as well as ones from the Property Distribution workgroup within HTNG. This enables not only the

core business process to be aligned with Open Standards, but the messaging and service interfaces as well; this is significant to improving interoperability at various levels within hotel enterprise architectures.

## **4.5 Considerations on Realizing and Deploying Services**

Most of the decisions and work that needs to be done during service realization is planning how the non functional requirements for the service will be executed. This includes whether or not there is redundancy for the service in order to execute on the stated availability requirements and then creating an appropriate deployment model.

In addition to determining how the nonfunctional requirements will be delivered during this part, how the services will be accessed and integrated is of high importance. In order to be found by other systems, applications, and other services, often services need to be registered in a service repository; then, these services can be discovered by other sources by accessing the service repository.

An important consideration with regards to how the services will be consumed and integrated is whether or not there should be some form of a service bus in the deployment. A service bus is a simple hub and spoke architecture that can perform routing, transformations, logging, etc. Used in the context of a hotel property, this is referred to as a property service bus; this type of bus creates more flexibility at the hotel property by letting the service consumers be agnostic to the location of the service providers, as well enabling easier integration with non-service based applications.

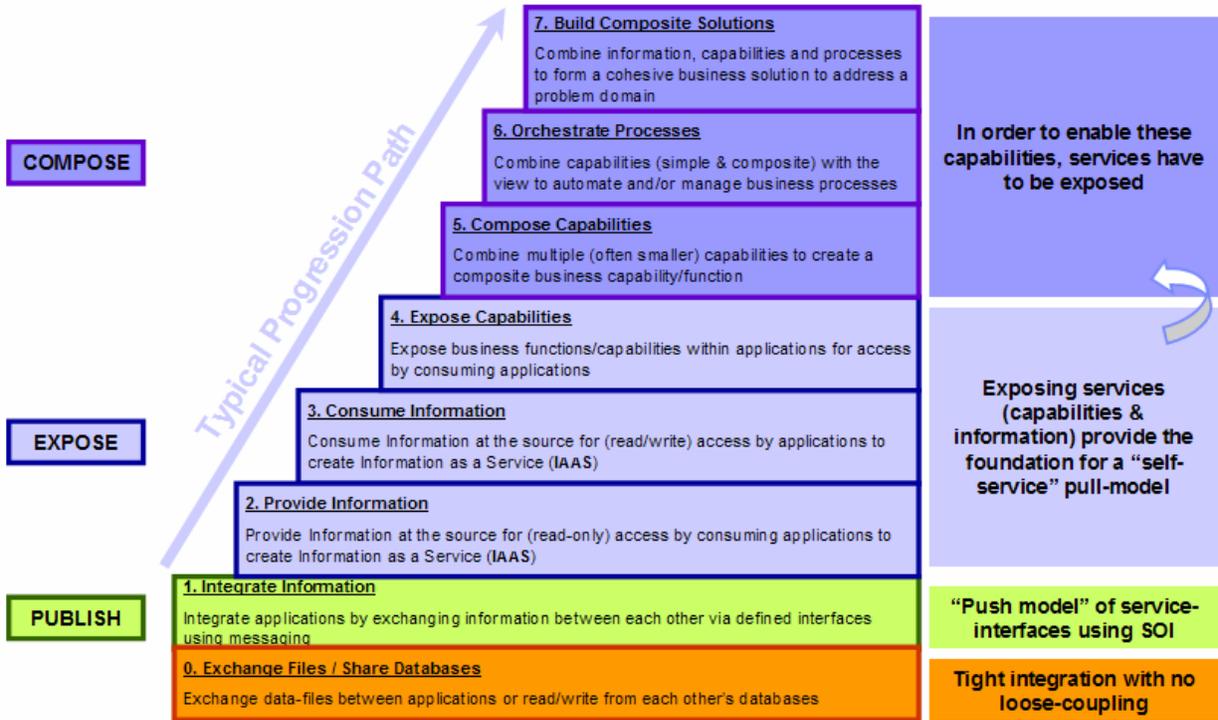
## **4.6 Guide to Exposing Services from Packaged Applications**

In an ideal SOA environment, Hoteliers will have access to an inventory of SOA services that can be used, reused and composed in any fashion that suits the changing business requirements. In other words, in an ideal environment, services would be in the business of providing a service, and business processes would be external to the services themselves.

This type of an environment gives maximum flexibility to a business and makes its IT agile and responsive to change.

There are, however, several packaged applications today providing critical functionality already established in the marketplace and that constitute a significant investment and ongoing cost. Moreover, these applications have been developed with significant cost to the vendor companies as well, and the cost of ripping and replacing them with a basket of services is neither feasible nor advisable.

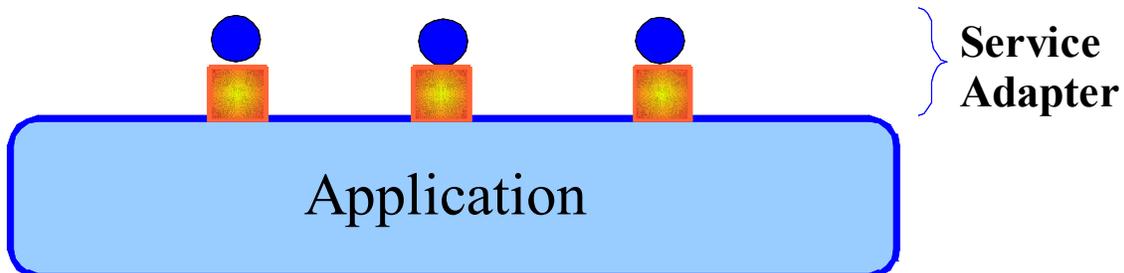
Given that many of these applications, whether they are Property Management Applications or Financial and Accounting Applications, provide useful functions and manipulate crucial data, the best method for these applications to participate in a SOA world is to slowly work their way up the maturity ladder of service oriented integration as shown in Figure 9.



**Figure 9 – Service Oriented Integration Ladder**

While most of the integration between packaged applications is achieved at the Tier 0/1 level through either sharing of database or file exchange, it is easy for most application providers to begin providing and consuming information as a Service. That is, the data/information from the applications can be provisioned as a Service, allowing multiple consumers to get to it in a non-proprietary fashion, and consumers who were previously unable to get information can also now access the same.

Providing information as a Service is the first step towards participating in a Service Oriented world and can be achieved by either having the application vendor directly making the information available as a Service or by getting a solution integrator to create a front-end Service with suitable adapters to the proprietary API exposed by the application, like in Figure 10 below.

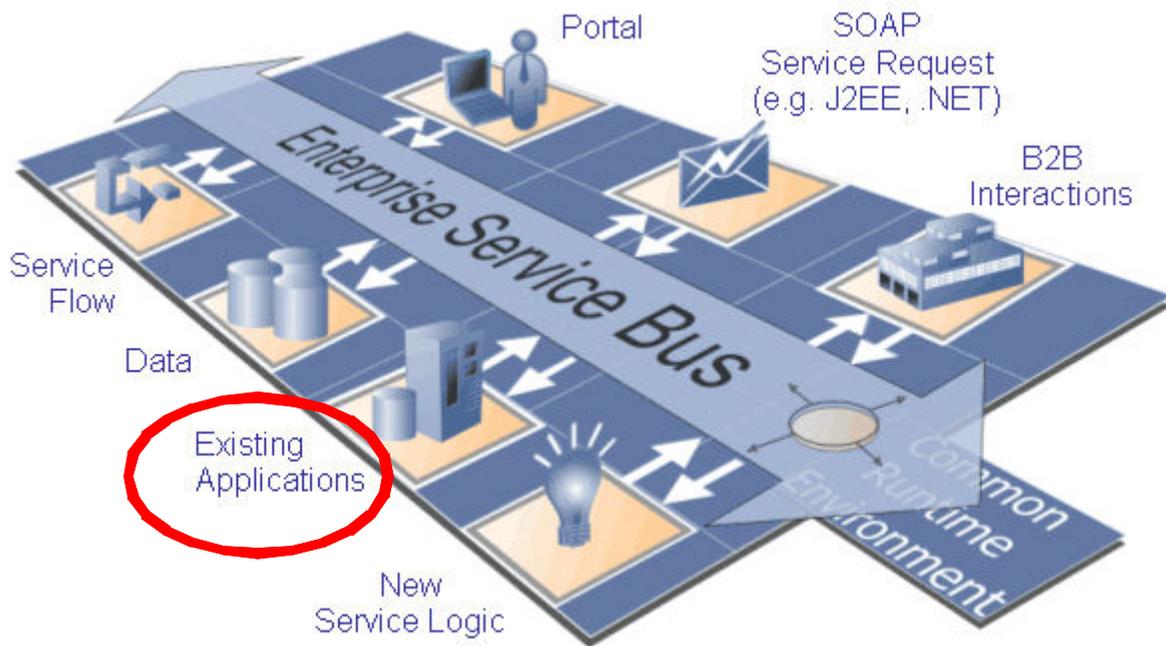


**Figure 10 – Exposing Applications with Service Adapter**

The next step in improving the integration maturity of a packaged application is to begin exposing its business function capabilities as a set of Service capabilities. The fundamental methodology is the same as for exposing Information as a Service; however, the ability for a given application to do this depends on how well defined and modularized is the internal

architecture of the application itself. Nevertheless, key functional capabilities are usually well-defined and can be exposed through an API for external consumption. Adding suitable protocol and message adapters can help in exposing a set of Service capabilities from the application and make them available for participation in business processes.

Once existing application capabilities and information are exposed as a Service, the application can fully participate in a SOA environment by plugging into a bus such as the Enterprise Service Bus, allowing the Hotelier to take full advantage of all the features offered by the application in new, innovative, and collaborative ways.



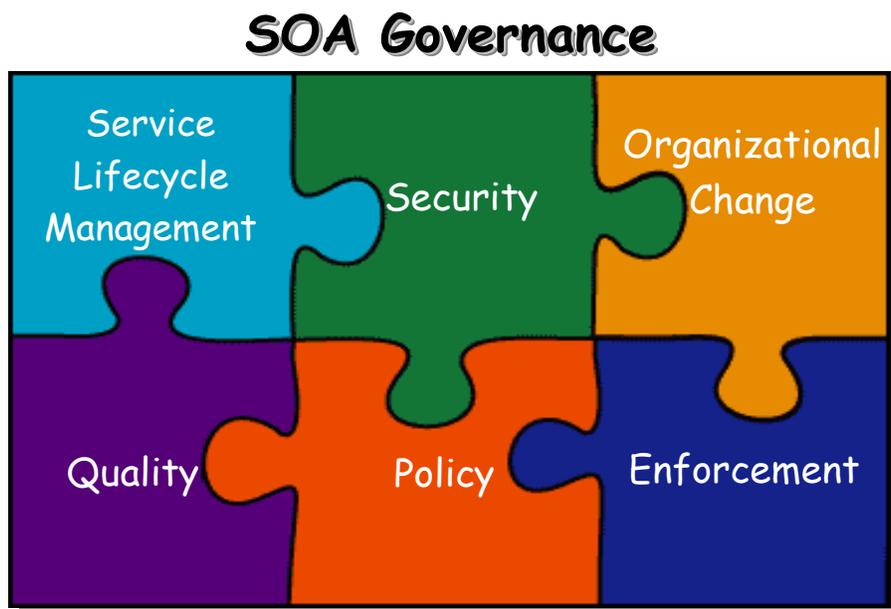
**Figure 11 – Existing Applications connecting to Enterprise Service Bus**

## 5 Ensuring Success through SOA Governance

SOA governance extends IT governance as enterprises increase their level of service-orientation. Governance becomes more critical in any SOA-based enterprise solution, which typically has a dynamic and heterogeneous IT environment. For example, in SOA:

- Service consumers and service providers could run in different processes on different platforms, be developed and managed by different departments, and require coordination to work together successfully.
- Reusable and common services are designed for sharing across applications within an enterprise or across enterprises.

These new challenges call for new governance requirements such as service ownership and funding policies for service providers, service design principles for guiding design and development of common and reusable services, as well as service change management processes.



**Figure 12 – SOA Governance Key Components**

SOA Governance, as pictured in Figure 12, can be broken down into six major components. These components together describe the different elements to always consider for SOA Governance. Below are brief descriptions of each of these components:

### 1. Service Lifecycle Management

The first component of SOA governance, Service Lifecycle Management, is based on modeling, assembling, deploying, and managing a service. The effective management of this lifecycle is a key goal of SOA governance.

Several governance processes are defined to improve the efficiency in Service Lifecycle Management:

- Service definition (the scope, interface, and boundaries of a service)

- Service modeling (detailed design and specification of services based upon design techniques, patterns, and standards)
- Service versioning (including compatibility)
- Service registries (service discovery and access)
- Service ownership (Corporate organization and funding prioritization)
- Service deployment (Service registration and configuration for discovery, usage documentation)
- Security of service (Including range of acceptance protection)
- Service management and monitoring (Problem determination, policy enforcement, service audition, rogue services management, service metrics and measures)

## **2. Security**

The Security component of SOA Governance deals with who is allowed to own, access, and update services.

## **3. Organizational Change**

Organization Change involves creating the culture and organization to support moving to SOA. This includes educating and communicating SOA governance across the organization and setting up environment and tools to allow easy access and use of the governance information.

## **4. Quality**

The Quality component of SOA Governance includes creating an end-to-end process for developing and deploying high quality services. This process would include activities such as monitoring of associated metrics for measuring and reporting performance and effectiveness of implemented SOA components/services.

## **5. Policy**

Policy involves establishing and managing the rules associated with services. Although policies are often expressed as business rules, for example protecting private information, they have important IT components as well.

## **6. Enforcement**

Enforcement is a very important component of SOA Governance that makes sure the service policies are being followed. This includes developing control mechanisms to ensure compliance and conformance to these policies.

As hoteliers use SOA to better align IT with the business, proper SOA governance, including all six of these major components, is necessary to realize the benefits of SOA. For SOA to be successful SOA governance is not optional, it is required.