# Oracle® Reference Architecture

and Service Orientation

Release 3.0

**E15827-03**

September 2010

ORACLE®

ORA and Service Orientation, Release 3.0

E15827-03



Primary Author: Bob Hensle

Contributing Author: Cliff Booth, Dave Chappelle, Jeff McDaniels, Mark Wilkins, Steve Bennett

# Contents

## List of Figures

# Send Us Your Comments

**ORA and Service Orientation, Release 3.0**

**E15827-03**

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this document?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us at its_feedback_ww@oracle.com.

# Preface

This document introduces the concepts of service orientation and describes the relationship between the Oracle Reference Architecture (ORA) and Service-Oriented Architecture (SOA). The document describes why service-orientation is fundamental concept within ORA, provides the unambiguous definition of a SOA Service in the context of ORA, and presents the service-oriented architecture principle underpinning ORA. This document also describes how various Enterprise Technology Strategies (e.g. BPM, MDM, BI) can be combined using a service-oriented approach. This document does not cover any Oracle products nor does it discuss any specific industry standards.

## Audience

This document is intended for Enterprise Architects and Solution Architects who want to understand the foundation upon which ORA is based.

## How to Use This Document

This document is intended to be read from start to finish. However, each section is relatively self contained and could be read independently from the other sections. This document should be read to understand the underpinnings for the entire ORA document set.

## Document Structure

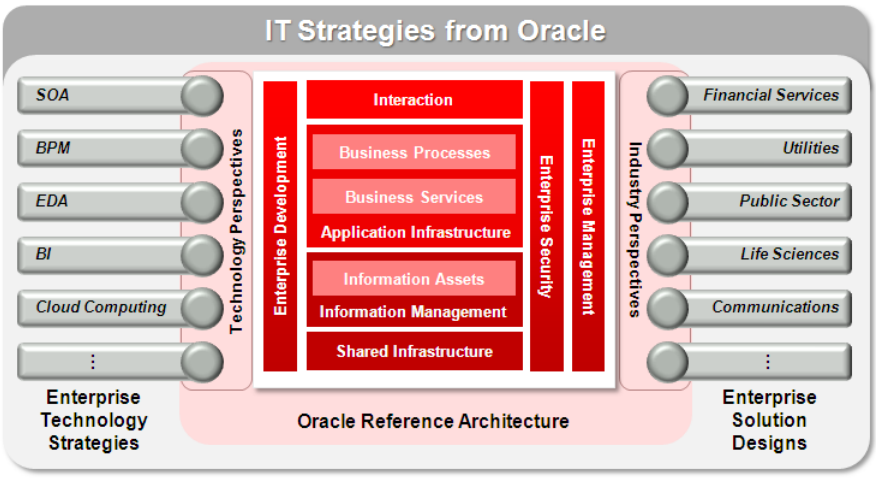
This document is organized in sections that first introduce the primary concepts underpinning ORA and then provides concrete definitions for the terms used. The architecture principles infused into ORA are also identified. Specifically,

- Chapter 1 introduces the fundamental concepts for ORA.
- Chapter 2 provides an unambiguous definition of SOA Service used as a building block with a SOA.
- Chapter 3 describes how different Enterprise Technology Strategies can be combined by following the foundational concepts of ORA.
- Chapter 4 is a summary of the document.

## Related Documents

**IT Strategies from Oracle (ITSO)** is a series of documentation and supporting collateral designed to enable organizations to develop an architecture-centric approach

to enterprise-class IT initiatives. ITSO presents successful technology strategies and solution designs by defining universally adopted architecture concepts, principles, guidelines, standards, and patterns.




ITSO is made up of three primary elements:

- **Oracle Reference Architecture (ORA)** defines a detailed and consistent architecture for developing and integrating solutions based on Oracle technologies. The reference architecture offers architecture principles and guidance based on recommendations from technical experts across Oracle. It covers a broad spectrum of concerns pertaining to technology architecture, including middleware, database, hardware, processes, and services.
- **Enterprise Technology Strategies (ETS)** offer valuable guidance on the adoption of horizontal technologies for the enterprise. They explain how to successfully execute on a strategy by addressing concerns pertaining to architecture, technology, engineering, strategy, and governance. An organization can use this material to measure their maturity, develop their strategy, and achieve greater levels of adoption and success. In addition, each ETS extends the Oracle Reference Architecture by adding the unique capabilities and components provided by that particular technology. It offers a horizontal technology-based perspective of ORA.
- **Enterprise Solution Designs (ESD)** are industry specific solution perspectives based on ORA. They define the high level business processes and functions, and the software capabilities in an underlying technology infrastructure that are required to build enterprise-wide industry solutions. ESDs also map the relevant application and technology products against solutions to illustrate how capabilities in Oracle's complete integrated stack can best meet the business, technical, and quality of service requirements within a particular industry.

This document is one of the series of documents that comprise Oracle Reference Architecture. ***ORAand Service Orientation*** describes the relationship between ORA and service orientation and how service orientation is a fundamental concept used within ORA. It describes how service-oriented principles are used to combine different Enterprise Technology Strategies thereby bringing together disparate technologies into a unified architecture.

Please consult the ITSO web site for a complete listing of ORA documents as well as other materials in the ITSO series.

# Conventions

The following typeface conventions are used in this document:

| Convention | Meaning |
|---|---|
| **boldface text** | Boldface type in text indicates a term defined in the text, the ***ORA Master Glossary***, or in both locations. |
| *italic text* | Italics type in text indicates the name of a document or external reference. |
| underline text | Underline text indicates a hypertext link. |

***"SOA Service"*** - In order to distinguish the "service" of Service-Oriented Architecture from the wide variety of "services" within the industry, the term "SOA Service" (although somewhat redundant) will be used throughout this document to make an explicit distinction for services that were created as part of an SOA initiative; thus distinguishing SOA Services from other types of services such as Web Services, Java Messaging Service, telephone service, etc.

# 1 Introduction

The modern IT environment is generally an eclectic mix of products and technologies that have accumulated over years or even decades. Adding to the complexity is a continuous stream of new products and technologies each with its own promise of value that it can deliver to IT and/or business. CIOs and Enterprise Architects are given the unenviable task of bringing order to the chaos and are expected to deliver more business value while also reducing costs.

Bringing order to the chaos requires creating an architectural vision (i.e. Enterprise Architecture) that functions as the blueprint for the enterprise IT environment. This blueprint must incorporate the existing products and technologies as well as provide guidance for incorporating new products and technologies.

The ***IT Strategies from Oracle*** (ITSO) series of documents provide information and guidance to address this exact problem faced by architects. These documents are written by Oracle architects for architects. This particular document details the fundamental concepts that are incorporated into the reference architecture provided by ITSO - the Oracle Reference Architecture. (Please see the ***ITSO Overview*** document for a description of the scope, structure, and contents of ITSO.)

## 1.1 Oracle Reference Architecture

The Oracle Reference Architecture (ORA) is designed to help companies create their own architectural vision. ORA provides a reference architecture for building and integrating solutions based on Oracle technology. Thus, ORA is narrower in scope than the entire IT environment, but for companies that have invested (or plan to invest) in Oracle technology, ORA brings valuable information, principles, and guidance focused specifically on the problem of building a sound architectural vision.

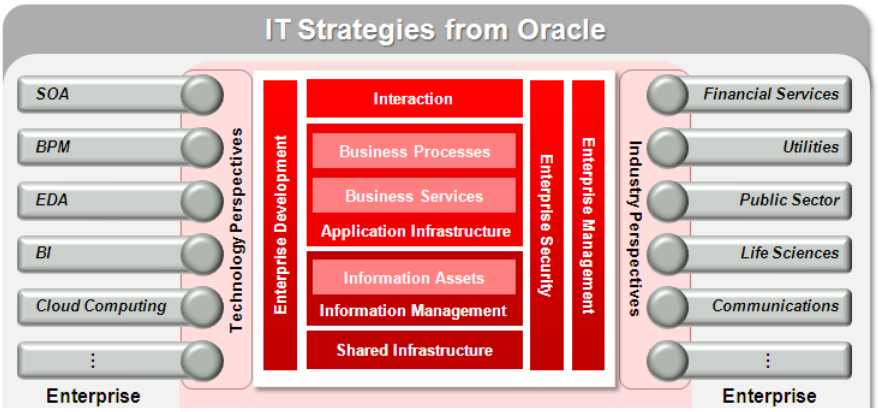
ORA provides a clear architecture blueprint that includes recommendations that promote best practices, and reduce or eliminate ambiguity and inconsistencies. It describes many principles and technology-based strategies to help architects build better solutions and help companies maximize their information technology investments.

ORA offers architecture principles and guidance based on recommendations from Oracle Product Development architects and field experts. It covers a broad spectrum of technology from virtualization to user interaction, security, and management. Information provided by ORA gives architects an understanding of how to design solutions for the Oracle environment and best leverage its capabilities.

### 1.1.1 ORA Perspectives

ORA is focused specifically on architecture and provides a single, consistent reference architecture. This reference architecture can be viewed from multiple different "perspectives" that focus on particular technologies and products (e.g. SOA, BPM, EDA, BI). An ORA perspective is NOT a separate reference architecture; rather it is focused view into the totality of ORA. This relationship is illustrated in Figure 1–1.

***Figure 1–1 ORA Perspectives***




There are both horizontal, technical perspectives as well as vertical, industry perspectives of ORA. As more perspectives are added to ORA, the totality grows, but as each new perspective is created consistency across ORA is maintained. This allows users of ORA to select and combine perspectives to create a customized reference architecture without introducing ambiguities or inconsistencies.

## 1.2 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is a strategy for constructing business-focused, software systems from loosely coupled, interoperable building blocks (called SOA Services) that can be combined and reused quickly, within and between enterprises, to meet business needs.

SOA has garnered widespread attention and adoption due to its promise to deliver agility and cost savings to IT. SOA is an architecture discipline designed to handle the heterogeneity of enterprise IT. SOA facilitates the inter-operability of diverse applications and computing platforms by packaging or wrapping functionality as interoperable SOA Services.

SOA provides high-level architectural principles, but it does not prescribe specific products or technologies. There are multiple products and technologies, frequently with overlapping capabilities, that could be included in an SOA. Therefore, architects must "fill in the details" by creating a reference architecture before any SOA initiative has a chance to succeed within an organization.

Additionally, achieving the benefits of SOA requires more than just the definition of a sound reference architecture. There are many other areas that must be addressed to successfully change the solution delivery process into a service-oriented approach including changes to software engineering, organizational structure, roles and responsibilities, governance, etc.

Thus, SOA requires both architecture and a broader strategy to be successfully implemented. The distinction between these two aspects is generally not clearly

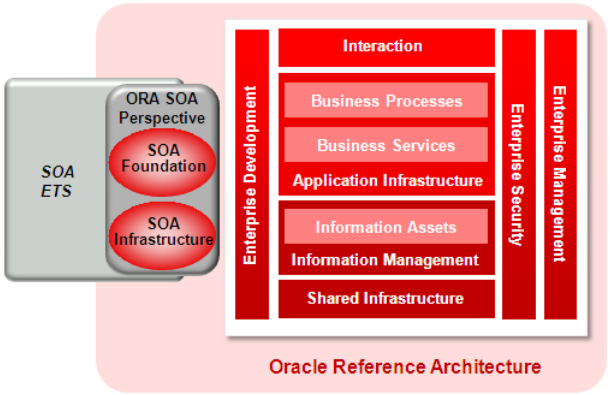
delineated in writings about SOA and frequently leads to confusion and ambiguity about what is SOA and what is not.

## 1.3 ORA and SOA

ORA is strictly focused on architecture. The SOA Perspective is a view of ORA focusing on the architectural components specific to SOA. Other non-architecture aspects of SOA are addressed by the SOA Enterprise Technology Strategy (ETS). The SOA ETS includes practical guidance on how to succeed at SOA by providing a variety of material including guidance on service identification, service engineering, governance for SOA, etc.

In essence, the SOA Perspective is a connection between ORA and the strategy of SOA. This relationship is illustrated in Figure 1–2.

***Figure 1–2 SOA ETS and ORA***




While the illustration is for the SOA ETS, a similar illustration could be shown for any other ETS such as the BPM ETS, BI ETS, etc. This illustrates the composability inherent in the structure of ORA and the ETSs.

However, ORA does have a special relationship with SOA. ORA embraces service-orientation as a core tenet to improve agility, rationalize functions and data, and promote reuse in an effective manner. The entire strategy of SOA is not core to ORA, but the concept of exposing data and functionality as interoperable SOA Services is core to ORA.

ORA must provide interoperability across all Oracle products and must also effectively deal with the heterogeneity that exists in IT environments. SOA Services provide a clean, consistent approach to deal with both of these complexities. This is the reason that ORA includes service orientation as a core tenet. Stated as an architecture principle, this becomes:

- **The architecture embraces services as the primary mechanism for interoperability and integration.**

  Service orientation offers tremendous value in the form of agility, interoperability, rationalization, standardization, manageability, and reuse. By incorporating service-oriented principles at the core, ORA can infuse these benefits into all layers, aspects, and perspectives of the computing environment. Service orientation is a key enabler for emerging strategies such as **Software as a Service (SaaS)** and **Cloud Computing**.

## 1.4 Products Labeled “SOA”

The term “SOA” has become a marketing term and a wide variety of products have the term SOA included in their marketing verbiage. Some of these products are rightfully described as SOA infrastructure products (e.g. ESB, Registry, Web Service Management). Other products that are label with SOA have only a very ancillary association with SOA. Today any product that can either consume or expose a Web Service has been labeled as “100% SOA”, “SOA ready”, or something similar.

This widespread use of the term SOA demonstrates that service orientation has become the primary architectural approach for the IT industry. Unfortunately, the overuse of the term SOA has diluted the true meaning and has led to confusion and is, in some cases, creating a backlash. Nonetheless, the concept of using SOA Services as the modus for interoperability is well established, and it is this concept that ORA embraces. It should be noted that ORA has a unambiguous definition of what constitutes a SOA Service (see Chapter 2).

## 1.5 ORA without SOA

Although widespread, the adoption of SOA is far from universal. For some companies there is no justification to embark on an SOA initiative, for example, a small homogeneous IT environment gains little from SOA. ORA is as applicable to these situations as it is to any full blown SOA initiative.

ORA is documented using a modular structure that allows the pieces to be used in isolation or in combinations. Thus, for example, ORA is applicable to a company that is embracing BI but has no interest in SOA. The SOA Perspective would not be applicable, but the BI Perspective would fit perfectly. Other core topics within ORA (e.g. security, monitoring and management, application infrastructure) would likely also be directly applicable.

# 2

# Definition of a Service

The term “service” is widely used in IT to mean many different things. ORA addresses this ambiguity by providing a precise definition of a service that is part of SOA versus other uses of the term. This section includes a concrete verbal definition of a SOA Service as well as describing the meta-model for a SOA Service. This provides a precise definition for the term “SOA Service” when used in the ORA context.

## 2.1 Definition of “SOA Service”

ORA makes a distinction between the generic term service and a SOA Service. A SOA Service has been built as part of a SOA initiative i.e. it is published, discoverable, and was designed, implemented, and deployed using a service-oriented approach.

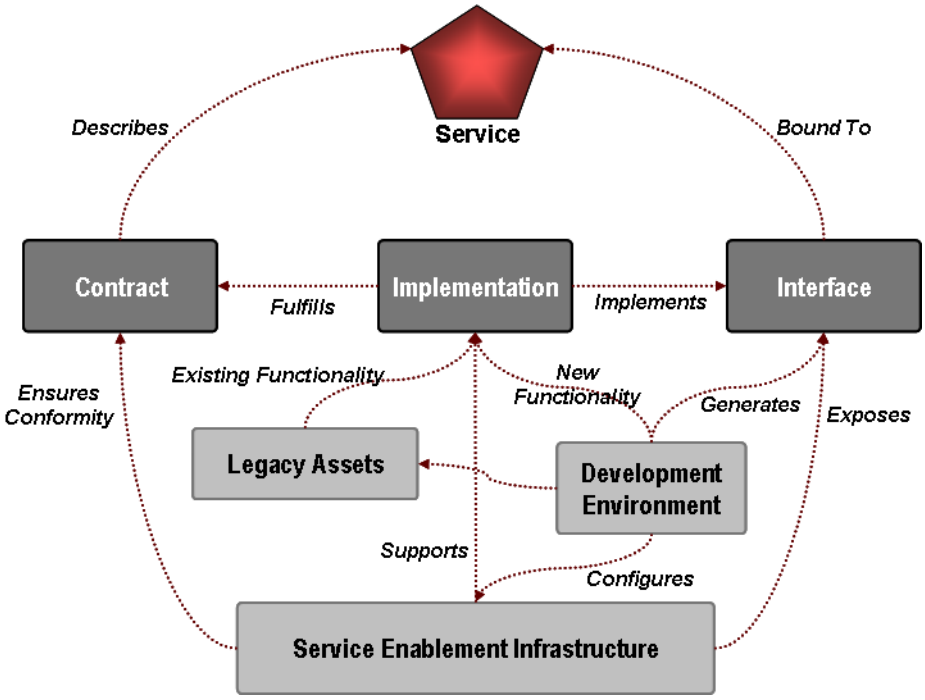
To illustrate this distinction, a web service may or may not be a SOA Service. A web service is a technology (i.e. SOAP over HTTP). This technology may be (and frequently is) used when creating and exposing a SOA Service. However, a web service may also be created when no service-oriented approach is employed i.e. the web service is simply a technical artifact of the implementation. Conversely, a SOA Service may be implemented using technologies other than web services (e.g. REST, JMS, JSON).

This chapter uses capitalization to distinguish ORA defined terms from other industry terms. When used as a noun, the word service is capitalized when it means the ORA SOA Service as defined in the first paragraph of this section. When used as an adjective, the word service is not capitalized unless it is part of an ORA defined term. For example, service consumer is not capitalized since ORA does not specifically define this term; whereas Service Contract is capitalized because ORA defines this term as an essential aspect of a SOA Service.

## 2.2 Facets of a SOA Service

ORA defines a meta-model for SOA Services. A depiction of this meta-model is provided in Figure 2–1.

*Figure 2–1 Facets of a SOA Service*




The three primary parts of a SOA Service as defined by ORA are contract, interface, and implementation.

## 2.2.1 Contract

A Service Contract describes the SOA Service in human-readable terms. It includes descriptions of both functional and non-functional capabilities. Contracts describe functional capabilities of the available operations of a SOA Service using business terms. Contracts also specify non-functional aspects of SOA Service, such as semantics, invocation style, security requirements, transaction requirements, quality of service, etc.

## 2.2.2 Interface

A Service Interface provides a means for the consumers of a SOA Service to access its functionality according to the Service Contract. The interface separates the consumer from the Service Implementation and isolates the consumer from the details of the implementation. The consumer is only able to access functions and data offered through the interface.

## 2.2.3 Implementation

The Service Implementation is the technical realization of the contract. It is responsible for fulfilling all functional and nonfunctional capabilities stated in the contract. The implementation may leverage functionality in existing systems, newly developed code, or a combination of both. Since infrastructure is often used to help satisfy certain capabilities of the SOA Service (either functional or non-functional), the infrastructure components act as part of the Service Implementation.

## 2.3 Usage Agreement

The previous section defined the constituent parts of a SOA Service. A common mistake is to assume that the Service Contract is between the service provider and the service consumer. However, this would create a point-to-point relationship between the service consumer and the service provider. Each time a new consumer wanted to use the SOA Service, a new Service Contract would need to be created. This is not a path to facilitating reuse.

The Service Contract defines what the SOA Service agrees to provide to the environment. So, for example, if the Service Contract guarantees a throughput of ten transactions per second (TPS), that is the total for all service consumers.

The service consumer Usage Agreement defines what a particular service consumer is entitled to consume. This relationship is illustrated in Figure 2–2.

***Figure 2–2 Service Consumer Usage Agreement***

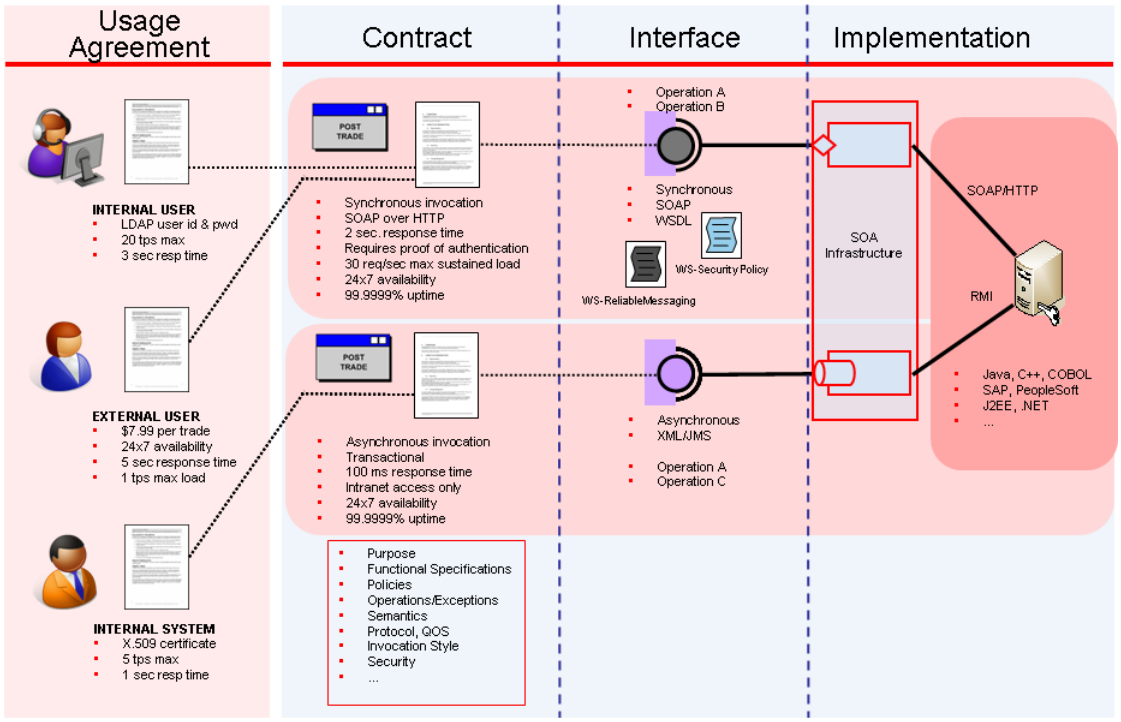



Figure 2–2 illustrates two SOA Services that share a backend system exposed via two interfaces, one synchronous and one asynchronous. There are three Usage Agreements illustrated for these two SOA Services. This is only an example and there could be different combinations of backend systems, Service Interfaces, Service Contracts, and Usage Agreements.

Having both a Usage Agreement and a Service Contract provides a decoupling between the service provider and service consumer. This not only facilitates reuse but also provides a separation of concerns. The Service Contract defines the totality of what the SOA Service guarantees to provide, and can be written and validated independent of any knowledge of specific service consumers. The Usage Agreement is service consumer specific and defines what capabilities of the SOA Service each consumer is allowed to consume.

# 3

# Combining Technology Perspectives

As described earlier in this document, the reference architecture is designed to support an expanding list of technology strategies. It is also important that the various technology perspectives can be easily combined since they are very much complementary.

Documenting how each technology perspective relates and combines with all the other technology perspectives would be onerous and unwieldy. This is the reason that ORA embraces service orientation at the core so that services provide a consistent mechanism to expose and combine various technologies and the capabilities. The following sections offer a glimpse of how service orientation is used to provide a consistent approach to combining different technology perspectives.

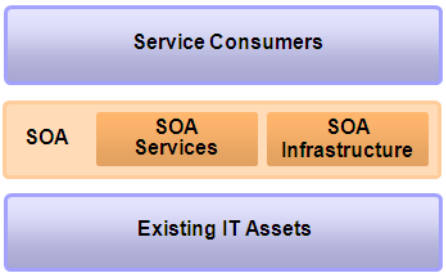
A high-level conceptual model for SOA is used to illustrate how technology perspectives consume and provide SOA Services. Additionally, the Oracle Reference Architecture from the ***ITSO Overview*** document is used to illustrate how technology perspectives leverage the common core capabilities of ORA.

## 3.1 SOA Conceptual Introduction

The intent of SOA is to provide common reusable SOA Services that can be leveraged by a variety of consumers. SOA Services are made available to various types of service consumers in order to rationalize the way business functions are performed and enterprise data is managed. Its modular architecture approach promotes reuse and business agility, and the use of widely adopted technology standards improves interoperability between business solutions.

As shown in Figure 3–1, SOA acts as a value-add layer on top of existing IT assets. It exposes these assets, as well as custom-built capabilities, as reusable SOA Services. SOA Services will typically originate from functionality and data that already exist in the enterprise - SOA Services created by “service-enabling” existing assets. As new projects are implemented, standalone SOA Services may also emerge as autonomous entities that do not have dependencies on existing systems.

*Figure 3–1 SOA Services and Infrastructure*




SOA also includes infrastructure to aid in the discovery, management, mediation, monitoring, and security of SOA Services. Consumers discover and interact with SOA Services via the SOA infrastructure.

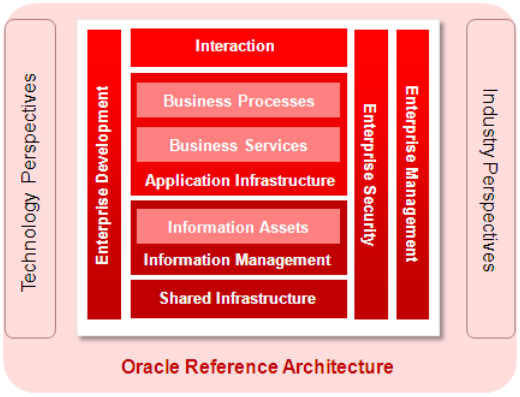
Service consumers consist of various types of business solutions. SOA Services can also act as service consumers. SOA Services invoking other SOA Services can be commonplace as each may represent the capabilities of various architecture levels. For instance, a SOA Service that performs a business function may invoke other SOA Services that provide access to data aggregations and legacy systems.

This level of detail is sufficient to describe how ORA employes SOA Services to provide the mechanism to join various technology perspectives. Greater detail about SOA is available from the SOA Perspective i.e. the ***ORA SOA Foundation*** document and the ***ORA SOA Infrastructure*** document

### 3.1.1 SOA Services and the Oracle Reference Architecture

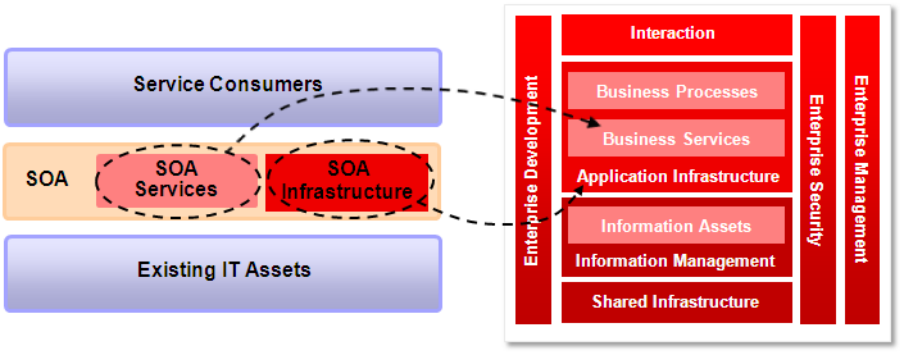
The ***ITSO Overview*** document introduces and describes the Oracle Reference Architecture. The Oracle Reference Architecture is reproduced in Figure 3–2.

*Figure 3–2 Oracle Reference Architecture*




ORA provides a framework to describe how various technology perspectives are related. For example, Figure 3–3 illustrates how SOA Services and SOA Infrastructure relate to ORA.

***Figure 3–3 Services Leverage and Expose Infrastructure***




SOA Services may be developed from a number of sources, such as exposing operations or data from custom or legacy applications, packaged applications, databases, etc. They may also be developed from scratch using popular application infrastructure platforms such as J2EE, .NET, and Tuxedo

SOA Services can function as important building blocks of enterprise-class business solutions. Therefore they must be built on a run-time infrastructure platform that delivers the RASP qualities required by mission-critical applications. In fact these qualities are often more critical for SOA Services as they can be shared by multiple applications. Concepts included in the Application Infrastructure layer, such as Grid Computing, complement SOA very well since they offer unique deployment and scalability options.

In addition, Security and Management also factor into the run-time environment for SOA Services. Each SOA Service must account for security, which includes identity validation, authorization, auditing, data confidentiality and integrity, and identity propagation. Likewise, SOA Services must be monitored for operational concerns as well as their ability to meet their stated levels of availability and performance. Administration and management infrastructure must enable SOA Services to be provisioned, versioned, and reconfigured as needed.

SOA Services can be created to expose capabilities of the underlying platform at many levels. For instance:

- Integration infrastructure can be exposed to provide messaging or connectivity as a SOA Service.
- Rules infrastructure can be exposed to provide business rule decisions as a SOA Service.
- BI infrastructure can be exposed to provide reports, graphs, etc. as a SOA Service.
- Security infrastructure can be exposed to perform authentication, access control, auditing, encryption/decryption, etc. as a SOA Service.
- Content search, publishing, and subscription capabilities can be exposed as SOA Services.
- Data aggregation and normalization SOA Services can be created based on capabilities of the MDM and integration capabilities.

SOA Services can leverage and expose infrastructure capabilities in many different ways. This is another reason why service orientation was included as a core component of ORA.

## 3.2 BPM Perspective Conceptual Interlock

This section introduces some of the basic concepts of BPM. It describes how BPM relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between BPM and SOA. Detailed information on BPM is available in the ORA BPM Perspective documentation.

### 3.2.1 BPM Solutions

BPM Solutions are composite applications in the form of business processes. Business processes are collections of activities, orchestrated in a controlled sequence, to accomplish a unit of work. The unit of work is generally related to common business functions, for example: product sales, order fulfillment, and employee benefits enrollment.

The spectrum of business processes can be defined as having two extremes: purely human-centric processes and purely system-centric processes. Quite often processes will be comprised of a mix of human interaction (manual activities) and system interaction (automated activities). Though any processes can involve both types of interaction, there are reasons to consider them separately. In addition, a third type of workflow pertaining to document management is also introduced.

#### 3.2.1.1 Human-Centric Workflow

Human-centric processes are used to orchestrate business activities that primarily involve manual tasks. It is also known as workflow, since it involves controlling the flow of work throughout the organization.

Human workflow benefits the business by defining and managing an orderly structure and flow to activities that may otherwise be ad-hoc or haphazard. It ensures order in the process and provides the ability to monitor how often processes and activities are performed, how quickly they are performed, and by whom. Workflow instances can also be monitored to track status as the process progresses.

Human workflow involves a lot of interaction with users and the scheduling and completion of tasks. Tasks are either assigned to individual users or to members of a group that perform a particular role. As such, important aspects of human-centric processes include the definition of roles, assignment of tasks to roles, management of task assignment to users, and task completion. Much of the work involved in constructing this type of process is done by the Business Analyst. IT's primary role is to provide a means to connect users to the process, which is often accomplished using a tasklist interface.

#### 3.2.1.2 System-Centric Processes

System-centric processes automate the activities and interactions between systems in order to perform business or technical functions. The intent is to drive efficiencies by automating as much, if not all, of the activities required by the process. This type of process is also known as orchestration.

System-centric process may also represent the technical aspects of a higher level business process. They may be designed as sub-processes, invoked at various points in a business process to orchestrate complex automated business activities. This approach to process architecture helps segregate the concerns of the business analyst from the concerns of IT. The business analyst can construct a process that focuses on the flow of work in terms of business activities, while IT constructs system-centric sub-processes that accomplish the supporting system interactions.

Examples of system-centric processes include data synchronization, data processing, and legacy system integration. These processes work behind the scenes to perform the technical machinations required for systems to work together harmoniously. They replace hard coded business logic and/or manual data entry, improving transparency and IT agility. They are much easier to maintain as systems are added, versioned, and retired over time.

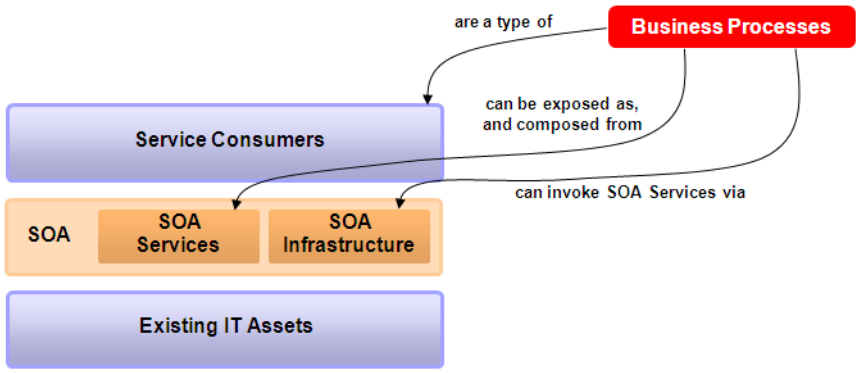
#### 3.2.1.3 Document-Centric Processes

Document-centric processes are used to manage the lifecycle of content, particularly in the stages leading up to release or publication. These processes coordinate the tasks required to advance content into its final release state. Common activities would include reviews, editing, and approvals. Since the infrastructure and tools used to manage these processes is generally associated with (and provided by) Content Management systems, this type of process will be described further in the ORA CM Perspective documentation.

### 3.2.2 BPM and SOA

BPM and SOA are often used together, as they both support a closer alignment between business and IT, and they both promote agility. BPM targets alignment and agility at the process level, while SOA applies more at the activity level. Hence, business processes and SOA Services can represent business constructs, providing a mapping between the things business does and the way IT helps get it done.

***Figure 3–4 BPM and SOA Relationship***



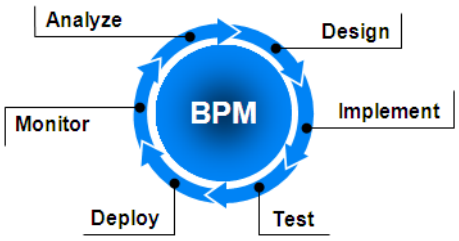

The convergence of BPM and SOA generally happens via process decomposition. That is when business processes are modeled as, (i.e. decomposed into), activities. All automated activities must be backed by some form of executable code or function call. These functions, if they are deemed worthy, can be engineered as SOA Services following service-oriented design principles. Agility at the process level is attained by changing the process model. Agility at the service level is achieved by deploying services that are loosely coupled and independently managed.

Further, BPM processes and sub-processes can themselves be exposed as SOA Services. This enables processes to be composed of SOA Services that are implemented as processes. It can be beneficial in two ways. First, it improves reuse of lower level system-centric processes (i.e. service orchestrations), and second, it offers a standard interface mechanism with which to invoke all types of business processes.

### 3.2.3 BPM and the Oracle Reference Architecture

Business processes tend to go through a development lifecycle similar to that of other applications. The initial work involves analysis of business activities related to accomplishing the goal, and the design of process models to illustrate how activities need to be orchestrated. This initial development work can be done using business process analysis (BPA) tools that are designed for the business analyst.

***Figure 3–5 Process Lifecycle***




Once the process models reach a certain level of completeness, further design and implementation must take place in order to turn it into something that is executable. The design-time tool for this is geared toward process architects and developers. It must support execution of the process and testing. Testing should address functional operation as well as load simulation.

When development and testing are complete, the process must be deployed to a production environment. This environment must provide the reliability, availability, scalability, and performance (RASP) qualities required by the business. As process instances are executed, run time statistics can be recorded. These data provide insights into the efficiency of the process, where business exceptions most often tend to occur, and where the process exceeds or fails to meet expectations. By analyzing these data business analysts and architects can determine where changes to the process can be made for optimization. Changes are introduced in subsequent versions of the process, as the lifecycle continues another revolution.

The following table describes how the Oracle Reference Architecture supports BPM throughout the process lifecycle.

***Table 3–1 BPM and the Oracle Reference Architecture***

| ORA Layer | Infrastructure capabilities provided for Business Process Management |
|---|---|
| Interaction | Supports human interaction, required for processes that include manual tasks that must be posted, accepted, assigned, completed, etc. Also supports process administration and process performance analysis. |
| Business Processes | These are the actual business processes that BPM technology is used to automate and manage. |
| Business Services | These are the SOA Services that are consumed by the business processes. Also includes the BPM processes and sub-processes that are exposed as SOA Services. |
| Application Infrastructure | Provides infrastructure to support process initiation, execution, and versions; management of all process instances in progress; service enablement, discovery, and mediation for SOA Services consumed by business processes. Also enables business processes to integrate with the many back-end systems and data stores that contribute to, or are affected by process execution. |

*Table 3–1 (Cont.) BPM and the Oracle Reference Architecture*

| ORA Layer | Infrastructure capabilities provided for Business Process Management |
|---|---|
| Information Assets | Provides data entities and content that can be consumed or updated by business processes. |
| Information Management | Handles management of process data, both operational data and process state. Also handles the process metrics that supports process analysis, process improvement, and business performance management. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the BPM infrastructure executes providing enhanced RASP capabilities. |
| Security | Ensure identity of process participants, access control of process execution, confidentiality and integrity of process data, and auditing. |
| Management | Provides process monitoring and management of BPM infrastructure. |
| Development | Includes analyst and architect level tools for process analysis, design, implementation, and test. |

## 3.3 EDA Perspective Conceptual Interlock

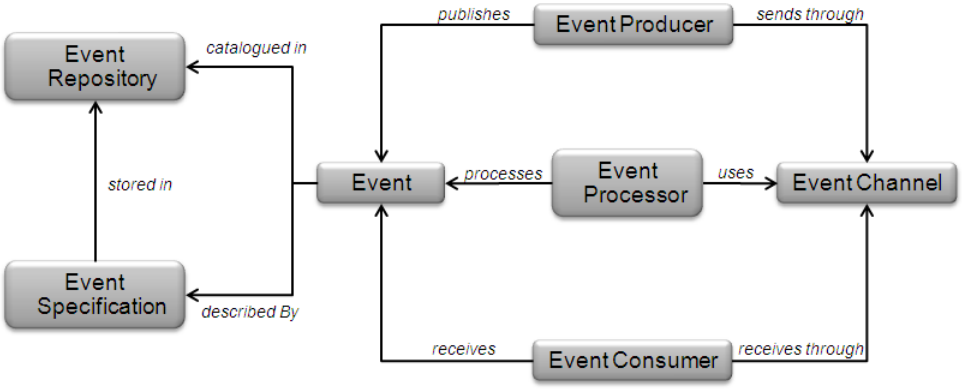
This section introduces some of the basic concepts of EDA. It describes how EDA relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between EDA and SOA. Further information on EDA is available in the ORA EDA Perspective documentation.

### 3.3.1 EDA Solutions

In order to describe what an EDA solution is, we must first look at what this technology strategy is all about. This strategy revolves around the concept of designing systems that create, process, and react to events. An event can be a change in state, a condition being met, or the detection of an action being performed. The architecture employs four main participants:

1. Event Producer - generate events (messages) representing what was detected
2. Event Consumer - receives events and takes appropriate action based on the event contents
3. Event Channel - mechanism to deliver events to interested parties
4. Event Processor - perform the necessary meaningful work in response to an event including aggregation, correlation, filtering, etc.

***Figure 3–6 EDA Relationships***




Event generation can be done in a number of ways, including database triggers, sensors, and various types of detection and measuring devices. Event processing can be accomplished using specialized infrastructure known as a Complex Event Processing (CEP). It is configuration-driven software that can process large volumes of events looking for patterns that match pre-defined conditions. A CEP can be used to process either simple or complex (multiple correlating) events. When meaningful event conditions are detected, it can notify interested parties.

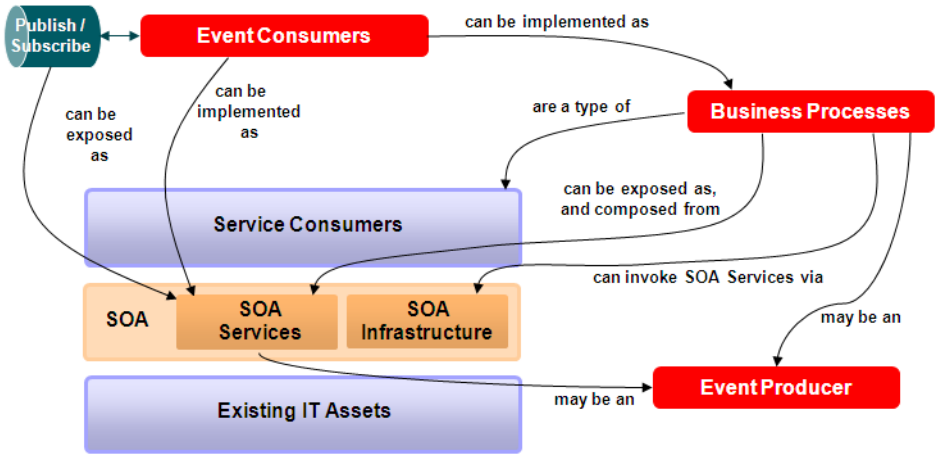
Notification can occur via a direct invocation of business logic, or via a publish and subscribe mechanism. The latter method is preferred as it establishes an architecture pattern that is loosely coupled and easily expandable. It allows multiple downstream processors to come and go based on business needs.

Since processing involves performing (orchestrating) a number of activities, the most recommended implementations are business processes. However, SOA Services could act as downstream processors in cases where simple activities need to be performed.

## 3.3.2 EDA and SOA

The intersection of EDA and SOA primarily pertains to the downstream processor. As shown below, it can take many forms, including that of a business process or SOA Service. As a result it is easy to see where EDA, SOA, and BPM might coexist. In addition, the publish / subscribe mechanism can be exposed as a set of SOA Services in order to create a universal, product-agnostic interface.

***Figure 3–7 EDA Solution in a SOA Environment***



One could argue that the other types of actors in an EDA solution could be developed as SOA Services. For example, the Event Processor could be a type of SOA Service that receives and processes events. However, there are factors to consider, including:

- The technology used to detect when an event occurs. Events can be generated by sensors, measurement devices, and other low level devices that receive inputs in ways other than through APIs.
- Such event generators may not have the capabilities required to invoke SOA Services. They may only be able to send events in very simple forms.
- The frequency of event generation can be much greater than the frequency of downstream processing. If only a small fraction of events actually require any action be taken, then a large number of SOA Services will be invoked to perform a small amount of work. This would be an inefficient architecture pattern.

### 3.3.3 EDA and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports EDA.

***Table 3–2 EDA and the Oracle Reference Architecture***

| ORA Layer | Infrastructure capabilities provided for Event-Driven Architecture |
|---|---|
| Interaction | Supports human interaction, which may be a form of downstream processing when users need to be notified and react to events. |
| Business Processes | The pusiness processes that may either consume or produce events. |
| Business Services | The SOA Services that may either consume or produce events. |
| Application Infrastructure | Provides infrastructure to support the detection, generation, processing, and distribution of events. Also provides the infrastructure for business processes and SOA Services that participate in event producing, consuming, and processing. |
| Information Assets | Provides data entities that are impacted by the events or that provide additional information that enriches the event content. |
| Information Management | Handles management of event data. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the EDA infrastructure executes providing enhanced RASP capabilities. |
| Security | Ensure confidentiality and integrity of event data, ensure source of events is authentic, and provide auditing of events. |
| Management | Provides capabilities to monitor the flow of events and manage event processing and notification. |
| Development | Includes tools to develop complex event processing rules. |

## 3.4 MDM Perspective Conceptual Interlock

This section introduces some of the basic concepts of Master Data Management (MDM). It describes how MDM relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between MDM and SOA. Further information on MDM will be available in the ORA MDM Perspective documentation.

### 3.4.1 MDM Solutions

As the number of applications and databases grows over time, so too do cases where data entities are duplicated and fragmented across the organization. This is a natural

occurrence since most legacy applications were built in a standalone fashion, with very little sharing of data. The problem is compounded enormously when companies go through mergers and acquisitions. As a result it becomes difficult, if not impossible, to merge common entities to create a holistic view and to keep everything synchronized. Copies diverge and the overall integrity of data degrades. Consumers and aggregators of data must determine which copies to use and which to ignore. Business analytics can produce different versions of truth depending on the data sources selected.

MDM solutions are designed to address this problem by creating a master "golden" source of data, and providing a mechanism to keep other copies synchronized with the golden source. New applications, processes, and services that need access to such data are encouraged to go to the master data source, as is feasible.

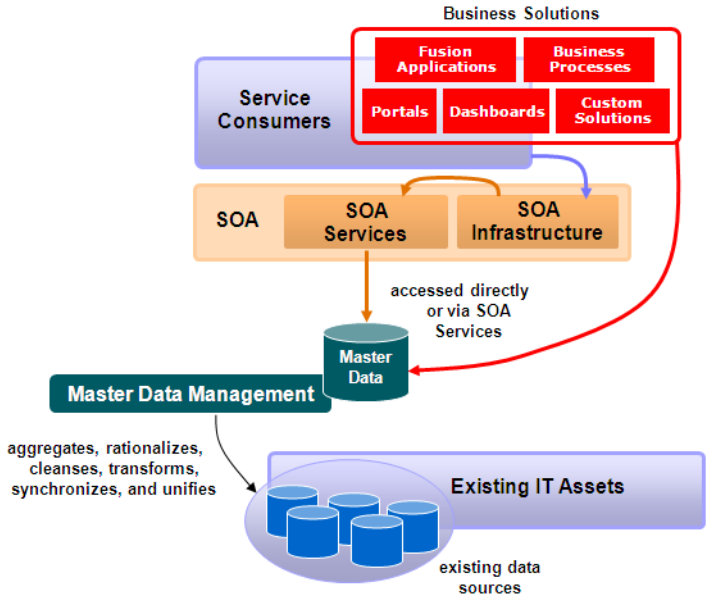
With respect to the Oracle Reference Architecture, MDM solutions can support the data needs of any type of business solution. There isn't a specific type of business solution that an MDM effort produces. Instead, it is an enabler of architecture, integration, and data management best practices.

## 3.4.2 MDM and SOA

In many ways MDM and SOA are complementary strategies that are designed to accomplish many of the same goals. Both address the issue of IT fragmentation by unifying and rationalizing existing assets. Both attempt to establish a gold source, either of function or of data. And both promote reuse and agility.

The synergy of MDM and SOA lies in the focus of each strategy. Where SOA defines a modular approach to solution engineering where SOA Services encapsulate reusable functionality and access to data, MDM provides quality data to operate on or expose. MDM enhances the value of SOA by establishing a clean, unified, master source of data. As SOA Services are leveraged beyond application silos and departmental boundaries, their applicability and integrity depend on the establishment of a consistent and universally accepted source of data.

***Figure 3–8 MDM in a SOA Environment***




As illustrated in Figure 3–8, master data can be exposed directly to consumers or via SOA Services. Direct access through database connections would yield master data in

its native form, (typically relational and normalized). The consumer would need to transform it into object or hierarchical form as needed. By developing SOA Services the transformation can be encapsulated in the service implementation. The service interface could offer master data in a hierarchical form aligned with canonical model specifications. Likewise, access to legacy data by the MDM infrastructure may or may not utilize SOA Services. Unless such SOA Services pre-date the MDM effort, it is more likely that new SOA Services will be created from master data rather than from legacy data.

### 3.4.3 MDM and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports the MDM technology strategy.

***Table 3–3 MDM and the Oracle Reference Architecture***

| Layer or Component | Infrastructure capabilities provided for Master Data Management |
|---|---|
| Interaction | UI infrastructure may be leveraged by MDM solutions in cases where human interaction is required, such as manual data reconciliation and cleansing. |
| Business Processes | Generally a consumer of MDM but may also support MDM processes such as data synchronization and cleansing |
| Business Services | Generally a consumer of MDM but may also support MDM by providing access to data sources. |
| Application Infrastructure | Generally a consumer of MDM, generally not a contributor of capabilities. |
| Information Assets | These are the data entities that feed the MDM solution as well as the "golden source" from the MDM solution. |
| Information Management | Handles persistence of master data and includes the data integration capabilities required to create an MDM solution from many disparate sources. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the MDM infrastructure executes providing enhanced RASP capabilities. |
| Security | Ensures confidentiality and integrity of, and secure access to, master data. Auditing, particularly of write operations. |
| Management | Provides capabilities to manage MDM processes such as data cleansing and synchronization |
| Development | Includes tools to develop mapping, aggregation, and transformation functions |

## 3.5 BI Perspective Conceptual Interlock

Business Intelligence (BI) is a term used to describe the collection and analysis of information that helps businesses with decision making. Infrastructure supports BI initiatives by providing capabilities to manage, query, process, report, and display information in ways that decision makers can analyze and act upon.

This section introduces some of the basic concepts of BI. It describes how BI relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between BI and SOA. Further information on BI will be available in the ORA BI Perspective documentation.

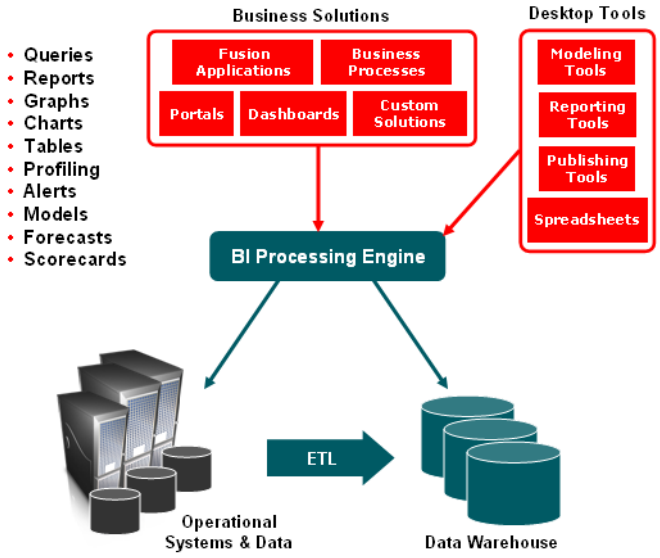
### 3.5.1 BI Solutions

A BI solution is one that provides information that enables users to make informed decisions. In its simplest form a BI solution might offer the ability to query records from an operational data store. This provides access to data but requires the end user to know what data to query, where the data are located, and how to interpret the results.

A more complete solution would abstract the user from the physical data sources, offer a logical unified view of data, provide powerful multidimensional query capabilities, alert users of events taking place, monitor key performance indicators, generate reports, and present information via multiple delivery channels. This not only adds value to the solution but makes it possible for users to acquire the greatest business intelligence with little or no understanding of all the complex technical underpinnings.

Ideally the BI user interface would be integrated into portals, dashboards, or applications in a way that provides information when and where decisions need to be made. This enables users to make informed decisions without having to switch applications, terminals, windows, etc., or having to remember where such information can be found. As depicted below, the user interface components access some form of processing engine that provides all the BI capabilities as well as integration with various data sources.

***Figure 3–9 BI Solution Concept***




Source data may come from operational systems such as CRM, ERP, customer service systems, HR, etc. The processing engine must have the ability to access these systems and/or their underlying databases.

Since organizations want to make decision based on reliable data, BI solutions must tap into data sources that are reliable. Master Data Management solutions that perform data cleansing and establish a single version of truth complement BI solutions nicely by providing that reliable data source. It is therefore worth considering MDM to some extent when undertaking BI.

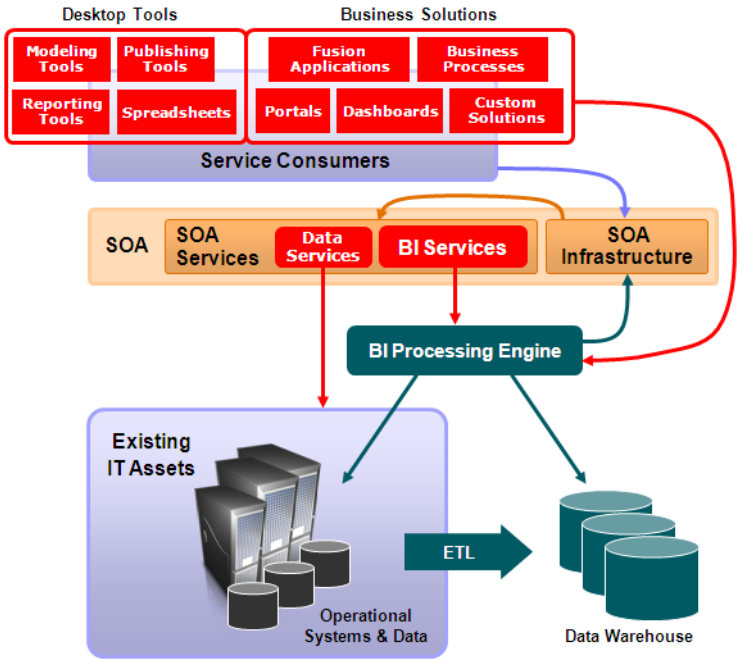
Often the data needed are not stored in a manner that efficiently supports multidimensional queries. Likewise, operational systems are seldom designed to accommodate large queries and reports. To rectify issues such as these, organizations set up data warehouses to store historical data. Data is restructured in a way that

better supports query and reporting operations. This provides faster response times and offloads the processing to resources that are dedicated to the BI solution. Data are extracted from the operational systems, transformed to match the reporting and query structure, and loaded into the data warehouse. This is known as Extract, Transform, and Load (ETL).

## 3.5.2 BI and SOA

The following graphic illustrates a typical mapping between BI solutions and SOA.

***Figure 3–10 BI in a SOA Environment***




At a very high level, the BI solution breaks down into: systems where operational data are found, BI processing capabilities, desktop tools, and various forms of information presentation. There are many links to SOA such as:

- Operational systems and data can be classified as existing IT assets, e.g. existing prior to the BI and SOA initiatives. Some of these assets may be service-enabled, offering the ability to access operational data as a SOA Service.
- The BI processing engine may access data through a variety of means including direct access to operational systems and data warehouses as well as SOA Services exposed via SOA infrastructure.
- BI capabilities may be accessed by business solutions that are service consumers. For example, BI portlets can be developed as presentation services and leveraged by multiple portal and dashboard applications. BI capabilities may also be accessed directly, without using a service-based interface.

The illustration does not draw a reference between SOA and the data warehouse. This is because data warehouses are generally populated via ETL processes, which are not a natural fit for SOA due to high data throughputs and low potential for reuse. Likewise data warehouses are not often exposed via services since the BI processing engines they are designed for can access them using means that are more efficient than typical service interfaces.

### 3.5.3 BI and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports BI.

*Table 3–4 BI and the Oracle Reference Architecture*

| Layer or Component | Infrastructure capabilities provided for Business Intelligence |
|---|---|
| Interaction | Provide user interaction capabilities via portlets, dashboards, web apps, etc. |
| Business Processes | BI solutions may interact with business processes both by providing intelligence to process participants and by incorporating metrics gathered via BPM into the BI solution. |
| Business Services | SOA Services that are either consumed by BI solutions or that provide access to BI capabilities. |
| Application Infrastructure | Enables BI solutions to interact with back end systems using various means such as adapters, messaging systems, etc. Also provides the ability to move data between operation stores and data warehouses. |
| Information Assets | These are the data entities that feed the BI solution as well as the information that the BI solution provides. |
| Information Management | Includes both operational data stores and data warehouses as well as the data integration capabilities needed to create a BI solution from multiple disparate sources. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the BI infrastructure executes providing enhanced RASP capabilities. |
| Security | Ensure identity and authenticity of users and controls access to data and functions. |
| Management | Provides monitoring and management of BI infrastructure. |
| Development | Includes tools to design and develop BI solutions. |

## 3.6 Enterprise 2.0 Perspective Conceptual Interlock

Enterprise 2.0 (E2.0), as a business-centric tangent of Web 2.0, is a way to categorize the emerging trend towards greater user participation and input in web-based computing. While previous systems were designed to bring content to the user, this movement is about users contributing content and knowledge to a collective intelligence environment (the new and improved web).

This section introduces some of the basic concepts in Enterprise 2.0 and describes how E2.0 relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between E2.0 and SOA. Further information on E2.0 will be available in the ORA E2.0 Perspective documentation.

### 3.6.1 E2.0 Solutions

Enterprise 2.0, also know as enterprise social computing, is a mix of social networking and business collaboration. The objective is to leverage social computing for the advantages it can provide to business. It differs from Web 2.0 in that it applies to all types of businesses where users can benefit from collaboration and knowledge sharing, rather than just businesses designed for Internet social networking, such as Facebook, YouTube, Plaxo, etc.

E2.0 solutions empower the end user. They provide tools that let users express their ideas, share insights and information, and collaborate in an ad hoc manner. They essentially take the interactions that have gone on for years on a verbal or physical

level and instantiate them on the desktop. It adds to traditional forms of computing by giving the end user the opportunity to contribute knowledge and value.

For example, E2.0 solutions include:

- Wikis to collaboratively build and organize information
- Blogs to give users the opportunity to post thoughts and ideas (and others the chance to read and respond)
- Mashups to spontaneously combine content from multiple sources to enhance one's knowledge or understanding
- Social tagging and page ranking to provide users the ability to identify content according to their perceptions of application and value
- Presence capability to help determine the best method available to contact someone
- IP-based communication support for VOIP interaction

Comparing these solutions to the Oracle Reference Architecture, we see that most are made available through Interaction infrastructure. Infrastructure provides the tools for end users to leverage as it suits their needs and job functions.

E2.0 solutions might be constructed to satisfy particular business needs. For example, mashups may be constructed by IT to meet specific business requirements, collaboration processes may be defined to support business processes, and presence/communication capabilities can enhance the way communications within the company occurs.

Other forms of social computing are constructs created by users to meet personal and group needs. Their forms and characteristics are driven more by personal preference. Therefore they are primarily classified as productivity enhancement tools rather than complete business solutions. (Admittedly, this is a grey area that may be disputed.)

### 3.6.2 E2.0 and SOA

The overlap between E2.0 and SOA is somewhat limited, as a subset of E2.0 solutions simply pertain to social interaction. Solutions such as blogs, wikis, and tagging mainly rely on users to post content in one form or another. These solutions don't have a strong, direct relationship to SOA since they are neither service consumers nor service providers. The solutions are not engineered by solution architects and therefore do not follow a service-oriented engineering process.

The main exceptions are mashups and collaboration. Mashups are client-side applications that aggregate data from multiple sources. They obtain value by creating a view of data for a given aggregate context. For example, a mashup might combine map data with office supply locations in order to help determine the closest place to purchase supplies. In order to provide data sources that are easily identified, understood, and consumed by multiple clients, it makes perfect sense to implement those data sources as SOA Services.

Some E2.0 services, such as collaboration, are offered by Interaction infrastructure as a way to surface common E2.0 capabilities across multiple client applications. Collaboration capabilities can be hosted by a common infrastructure deployment and exposed on various company portals in order to promote group activity planning, communication, and knowledge sharing.

### 3.6.3 E2.0 and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports E2.0.

***Table 3–5 E2.0 and the Oracle Reference Architecture***

| Layer or Component | Infrastructure capabilities provided for E2.0 |
|---|---|
| Interaction | Provides the end user tools to support all forms of E2.0 solutions |
| Business Processes | E2.0 may be used to support business processes especially collaborative processes. |
| Business Services | SOA Services that are incorporated into an E2.0 solutions. For example, data services that are provided to support mash-ups. |
| Application Infrastructure | Supports the business processes and SOA Services that are incorporated into an E2.0 solution. |
| Information Assets | The actual information that is pertinent to the E2.0 solutions. |
| Information Management | Handles persistence of content contributed by users and groups, and collaboration groups, tasks, calendars, contact information, etc. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the E2.0 infrastructure executes providing enhanced RASP capabilities. |
| Security | Provides user authentication, access to content, and management of groups and roles |
| Management | Manages group assignments, contact information, etc. |
| Development | Generally not applicable to E2.0 solutions - they don't usually involve traditional software development tools and frameworks. |

## 3.7 B2B Perspective Conceptual Interlock

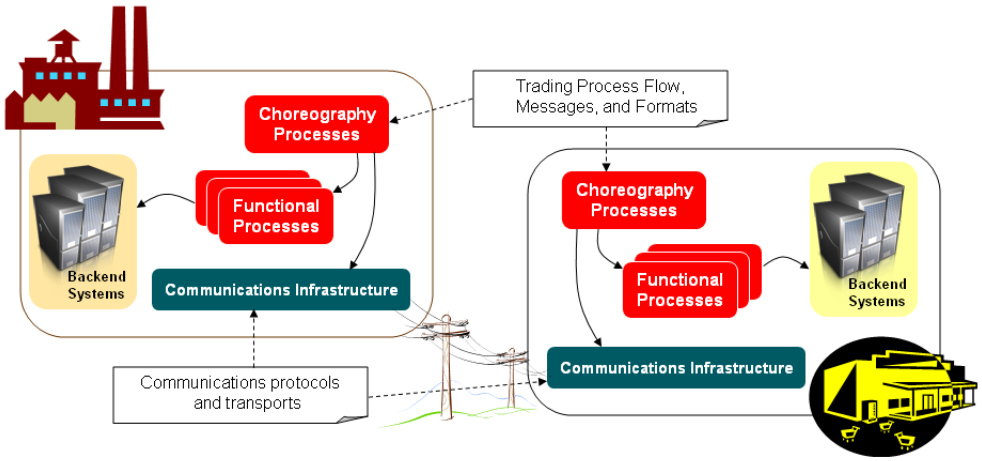
Business-to-Business (B2B) computing refers to electronic transactions between two businesses. Common scenarios are transactions between manufacturers and wholesalers, and between wholesalers and retailers. Establishing a system of electronic trading helps businesses conduct transactions quickly and efficiently. The role of humans is reduced to activities that require authority or intervention, such as final approvals, transactions involving large sums of money, exceptional cases, etc.

This section introduces some of the basic concepts of B2B computing. It describes how B2B relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between B2B and SOA. Further information on B2B will be available in the ORA B2B Perspective documentation.

### 3.7.1 B2B Solutions

A B2B solution consists of all that is necessary to achieve the goal of establishing electronic commerce between companies. Though they can be implemented in a variety of ways, (and automated to varying degrees), solutions tend to include the following components:

*Figure 3–11 B2B Solution Concept*



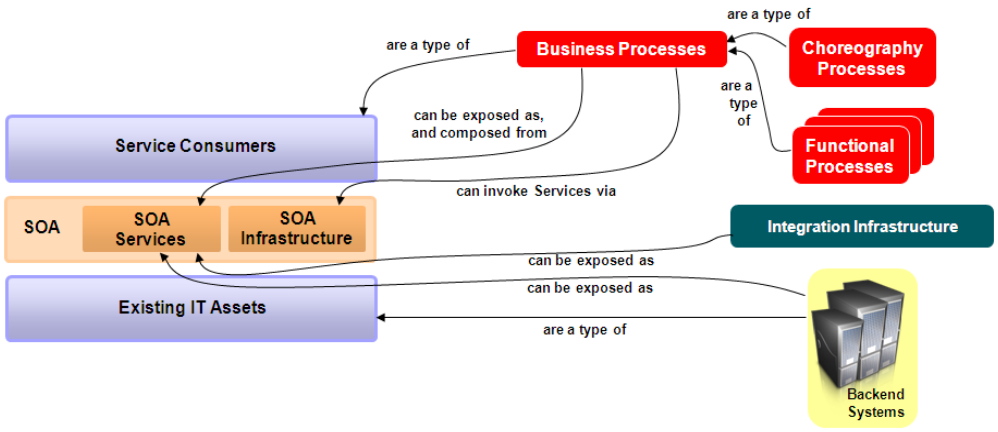

1. An agreed upon set of trading processes that describe the message exchange pattern for each type of interaction, along with the message types and formats. This defines the messages that are sent, the order they are sent, and any alternate scenarios, business exceptions, etc.
2. An agreed upon method of communication between the businesses including specifications for type of transport and protocols.
3. Each business will have some means of coordinating the exchange of messages on their end, adhering to the predefined trading process. This is represented above as Choreography Processes. They choreograph the exchange of messages and initiation of Functional Processes.
4. Each business will have a means to perform back end processing related to the exchange of messages. For example, the processing of purchase orders, sales orders, shipping notices, and billing information transmitted and received throughout the trading process. This is represented above as a set of Functional Processes interacting with the back end systems. These processes encapsulate the logic needed to process and produce or consume a particular type of message.
5. Each business will need communications infrastructure to send and receive messages.

The use of processes in this illustration represents a recommended pattern, since business processes are an ideal means to represent, coordinate, and execute sets of business activities. It is up to the business to determine which activities it needs to perform and which should be automated or performed manually. B2B solutions are often implemented as a set of business processes along with all supporting message schemas, business logic, and underlying infrastructure configurations.

## 3.7.2 B2B and SOA

There are a number of touch points between B2B and SOA as illustrated below.

*Figure 3–12 B2B in a SOA Environment*




Most notably, Choreography and Functional Processes can gain all the benefits of agility and flexibility that BPM and SOA have to offer. This is a key driver for designing the B2B architecture in this manner. It becomes a natural extension of the existing BPM+SOA environment. In fact, since many processes and activities will mirror internal processes, B2B becomes another way to apply and reuse existing resources.

In addition, interaction with the communications infrastructure and back-end systems may be implemented following SOA principles. This can be accomplished either by service-enabling legacy assets, or by engineering new systems that support B2B transactions in a service-oriented manner.

## 3.7.3 B2B and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports B2B.

*Table 3–6 B2B and the Oracle Reference Architecture*

| ORA Layer | Infrastructure capabilities provided for B2B |
|---|---|
| Interaction | Provide user interaction capabilities where manual tasks (e.g. approvals, exception processing) are performed as part of the B2B process. Provides the B2B infrastructure capabilities to support B2B interactions such as RossettaNet, EDI, ebXML, HL7, etc. |
| Business Processes | The business processes (Functional Processes) that participate in the B2B choreography. |
| Business Services | SOA Services that are consumed by B2B processes, especially those that expose back-end systems. |
| Application Infrastructure | Enables B2B processes to interact with integration infrastructure as well as the many back-end systems and data stores that contribute to, or are affected by process execution. |
| Information Assets | Provides the ability to make run-time process decisions based on external configuration-driven rules. |
| Information Management | Handles management of process data, both operational data and process state. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the B2B and BPM infrastructure executes providing enhanced RASP capabilities. |

*Table 3–6 (Cont.) B2B and the Oracle Reference Architecture*

| ORA Layer | Infrastructure capabilities provided for B2B |
|---|---|
| Security | Ensure identity and authenticity of remote systems involved in B2B transactions, provides confidentiality and integrity of transaction data, and establishes non-repudiation of transactions. |
| Management | Provides monitoring and management of B2B infrastructure |
| Development | Includes analyst and architect level tools for B2B process analysis, design, implementation, and test. |

## 3.8 CM Perspective Conceptual Interlock

Content Management (CM), in the scope of ORA, includes the collection of technologies used to maintain information and provide easy, secure access to it. This includes the management of content lifecycle as well as the ability for knowledge workers to contribute to, and locate content.

This section introduces some of the basic concepts of CM and describes how CM relates to the Oracle Reference Architecture. A section is also provided to briefly describe the relationship between CM and SOA. Further information on CM will be available in the ORA CM Perspective documentation.

### 3.8.1 CM Solutions

Content Management can be considered a very broad topic, as there are a number of types of content that can be managed, and a number of reasons for doing so. Examples include:

- Document Management. This type of solution targets the management of files such as documents, emails, reports, presentations, spreadsheets, etc.
- Digital Asset Management. Similar to document management, however it pertains to assets that are digitally encoded and not searchable, such as images, video files, etc. Also handles ***Digital Rights Management (DRM)***.
- Web Content Management. Solutions for designing and publishing web sites and managing the content from which they are created.
- Records Management. Maintaining records based on retention policies to satisfy legal obligations. Read-only warehousing of all types of documents.

CM solutions can be implemented to solve any one particular problem, or can be all inclusive. Solutions that address the broader spectrum of CM are often referred to as Enterprise Content Management (ECM) solutions.

ECM solutions, at a high level, offer six main capabilities:

1. the ability for users to locate, access, and contribute content,
2. the ability to manage the lifecycle of content,
3. the ability to manage content retention policies,
4. the ability to convert content into other formats,
5. the ability for applications to interact with the CM system, and
6. the ability to store content securely and provide structure in the form of taxonomies, policies, rules, etc.

Most of these capabilities can be realized via modern computing infrastructure. For instance, content persistence, security, taxonomy, search, lifecycle process

management, and system access can all be provided by infrastructure. As such, they are represented and realized within the scope of the Oracle Reference Architecture.

The remaining capabilities, left to construct in the form of business solutions, are the methods in which CM can effectively and efficiently be integrated into the users' environments. This is where all the benefits CM infrastructure provides are surfaced to the end user. The most common forms include:

- CM portlets for search, taxonomy navigation, and lifecycle management surfaced in enterprise portals.
- Web based applications that present content to end users
- Dashboards that expose content access and usage metrics in support of business intelligence
- Mashups that merge content in user-defined ways

### 3.8.2 CM and SOA

The intersection of CM and SOA most naturally occurs as CM capabilities are exposed as SOA Services. This opens up the infrastructure capabilities to a wide variety of users and delivery channels. SOA Services that support internal solutions over an intranet can be leveraged for external users via Internet and wireless network devices.

To accomplish this, one might implement the logical interface between infrastructure components and business solutions as a collection of SOA Services. Example services might include: search, content publish and subscribe mechanisms, lifecycle management services, and report generation and statistics.

### 3.8.3 CM and the Oracle Reference Architecture

The following table describes how the Oracle Reference Architecture supports CM.

***Table 3–7 CM and the Oracle Reference Architecture***

| Layer or Component | Infrastructure capabilities provided for Content Management |
|---|---|
| Interaction | Provide user interaction capabilities such as search and content delivery |
| Business Processes | Content-related processes (document workflow, content approval workflow, content publishing process, etc.) as well as business processes that access and update content. |
| Business Services | SOA Services that expose CM capabilities. |
| CM | Content lifecycle management, conversions, retention policies, maintains taxonomy, accumulates metrics, generates reports |
| Application Infrastructure | Offers a standard set of APIs to access CM functions. |
| Information Assets | The actual content stored in the CM solution. |
| Information Management | Content lifecycle management, conversions, retention policies, maintains taxonomy, accumulates metrics, generates reports, and persists the content. |
| Shared Infrastructure | Run-time network, storage, and computing platform on which the CM infrastructure executes providing enhanced RASP capabilities. |
| Security | Ensure identity and authenticity of users and grants access to content and functions. Auditing, particularly of content updates. |
| Management | Monitor and manage CM infrastructure deployments. |

*Table 3–7 (Cont.) CM and the Oracle Reference Architecture*

| Layer or Component | Infrastructure capabilities provided for Content Management |
|---|---|
| Development | Provides tools and utilities for file conversions, web content generation, etc. |

# 4

# Summary

This document introduced and detailed the fundamental concepts that underpin the Oracle Reference Architecture (ORA). ORA is a unified reference architecture that covers the Oracle technology space. ORA is a modular structure that allows new "perspectives" focused on particular technologies or industries to be added as needed to address the changing technical landscape.

Each ORA technology perspective is associated with an Enterprise Technology Strategy. Example ETSs include SOA, BPM, BI, etc. This document examined the relationship that ORA has with the SOA ETS. Since ORA incorporates service orientation as a core tenet, the relationship between ORA and the SOA ETS is tighter than with other ETSs. This document also discussed and illustrated how the core tenet of service orientation facilitates composition of ETSs.