Solution Architecture

This document presents an outline for a Solution Architecture document. The term solution, in this context, refers to the business processes and the manner in which the participants (people and systems) collaborate to bring those business processes to life..

**Table of Contents**

[1 Business Objectives and Constraints 4](#_Toc316046138)

[1.1 Quantified Business Expectations 4](#_Toc316046139)

[1.2 Business Constraints 4](#_Toc316046140)

[1.3 Business Risks 4](#_Toc316046141)

[2 Solution Context 4](#_Toc316046142)

[3 Business Process Inventory 4](#_Toc316046143)

[4 Domain Model 4](#_Toc316046144)

[5 Solution Architecture Pattern 4](#_Toc316046145)

[6 Business Process 1 5](#_Toc316046146)

[6.1 Business Process Design 5](#_Toc316046147)

[6.2 Process-Pattern Mapping 5](#_Toc316046148)

[7 Business Process 2 5](#_Toc316046149)

[8 Business Process n 5](#_Toc316046150)

[9 Addressing Non-Functional Solution Requirements 5](#_Toc316046151)

[9.1 Performance and Scalability 5](#_Toc316046152)

[9.2 Availability within a Data Center 5](#_Toc316046153)

[9.3 Site Disaster Recovery 6](#_Toc316046154)

[9.4 Security 6](#_Toc316046155)

[10 Component/Service A 6](#_Toc316046156)

[10.1 Business Process Involvement 6](#_Toc316046157)

[10.2 Interfaces 7](#_Toc316046158)

[10.3 Observable Architecture 7](#_Toc316046159)

[10.4 Observable State 7](#_Toc316046160)

[10.5 Coordination 7](#_Toc316046161)

[10.6 Constraints 7](#_Toc316046162)

[10.7 Non-Functional Behavior 7](#_Toc316046163)

[10.7.1 Performance and Scalability 7](#_Toc316046164)

[10.7.2 Availability within a Data Center 7](#_Toc316046165)

[10.7.3 Site Disaster Recovery 7](#_Toc316046166)

[10.7.4 Security 7](#_Toc316046167)

[11 Component/Service B 8](#_Toc316046168)

[12 Component/Service n 8](#_Toc316046169)

[13 Deployment 8](#_Toc316046170)

[13.1 Deployment Environment Migration 8](#_Toc316046171)

[13.2 Development Configuration 8](#_Toc316046172)

[13.3 Test Configuration 8](#_Toc316046173)

[13.4 Production Configuration 8](#_Toc316046174)

[14 Integration and Testing Requirements 8](#_Toc316046175)

[14.1 Integration Strategy 8](#_Toc316046176)

[14.2 Behavioral Testing 8](#_Toc316046177)

[14.3 Performance Testing 8](#_Toc316046178)

[14.4 Performance Testing 9](#_Toc316046179)

[Appendix A: Common Data Format Specifications 9](#_Toc316046180)

[Appendix B: Message Format Specifications 9](#_Toc316046181)

[Appendix C: Service Interface Specifications 9](#_Toc316046182)

[Appendix D: Data Storage Specifications 9](#_Toc316046183)

# Business Objectives and Constraints

## Quantified Business Expectations

Indicate, in measurable terms, the business objectives for this solution. If this is an incremental change to an existing solution, the quantification should reflect the expectations for the change being made.

## Business Constraints

Indicate the business constraints being placed upon the project implementing the solution (or the change to the solution). Cost and schedule expectations are common constraints, but there may be others as well. Typically these are constraints on the business processes that comprise the solution.

## Business Risks

Indicate the risks to the business that result from each of the impacted business processes failing to execute properly. Indicate the impact from a single failure and the impact from the process becoming unavailable for some period of time. This information is used by the architects to determine the level of fault tolerance to be built into each business process.

# Solution Context

Most solutions are fragments of a larger architecture. Provide an overview of how the solution fits into the larger architecture.

# Business Process Inventory

Provide an overview of the business processes that comprise this solution. Provide an overview of the artifacts that are exchanged between the business processes. Include all business processes involved in the exchange of artifacts. Even if a business process is not within the scope of this solution, if it is involved in an artifact exchange it must be shown.

# Domain Model

Present an overview of the information used in the business processes. The focus should be on the concepts and their relationships, with particular emphasis on the multiplicities of the relationships. Only major attributes – identifiers and values that drive business process decision making – should be modeled. The domain model serves as an overview and guide to the detailed data structures used during the execution of the business processes.

# Solution Architecture Pattern

Present the architecture pattern supporting the solution. The pattern should show the participants in the business processes (both human and machine) and their communications channels. Any constraints on the participants should be shown as well.

# Business Process 1

Each major business process is described in its own chapter of the architecture document.

## Business Process Design

Show the structure of the business process – its activities and the interactions between them. Show all artifacts required for the execution of each activity and all artifacts produced by each activity. This is typically done with a UML Activity Diagram or BPMN diagram.

## Process-Pattern Mapping

Map the business process design onto the architecture pattern. The intent is to show how the components of the architecture pattern collaborate to execute the business process. All interactions between participants should be shown, and all artifacts exchanged between components should be identified and modeled. This is typically done with a UML Activity Diagram and one or more UML Class Diagrams to model the artifacts being exchanged. Interfaces should be identified and detailed.

# Business Process 2

Document the second business process involved in the solution.

# Business Process n

Document the last business process involved in the solution.

# Addressing Non-Functional Solution Requirements

Describe how the solution architecture addresses the non-functional requirements of the solution. The outline has sections for the most common categories, but additional sections should be added as required to address additional categories.

## Performance and Scalability

Describe how the overall performance and scalability requirements for the solution were used to derive the performance and scalability requirements for the individual components. Where analysis indicates that more than one instance of a component may be required to handle the load, describe the mechanisms to be used for load distribution.

## Availability within a Data Center

Describe how the availability requirements for the overall solution were used to determine the availability requirements for the individual components. Where meeting availability requirements result in the need for multiple components operating in a fault-tolerant or high-availability group, describe the manner in which the group will be managed and how work-in-progress will be handled.

## Site Disaster Recovery

Describe how the site disaster recovery requirements (recovery point objective, recovery time objective) for the overall solution were used to determine the site disaster recovery requirements for the individual components. Be specific about how component state information is replicated between the sites and how work-in-progress will be handled.

## Security

Describe how the security requirements for the overall solution were used to determine the security requirements for the individual components. Be specific about the roles of authentication and authorization servers and the points at which they interact with components. Be specific about how credentials will be managed, particularly when credentials can expire.

# Component/Service A

There should be a chapter for each component involved in the architecture. The chapter summarizes the participation of the component in each of the business processes. The chapter is essentially the specification for the component’s participation in the solution.

The level of detail in the chapter will depend somewhat on whether the component or service already exists.

**Existing Components and Services:**

For existing components that require no modification or integration to be used as part of the solution, the chapter will simply provide an overview of the component’s utilization. To the extent that documentation on the component or service already exists, the sections of this chapter can be filled in with references to that documentation. This chapter should focus on the specifics of the utilization, identifying the interfaces to be used along with the volumes of utilization and required SLAs for those interactions.

**New Components:**

For new components, this chapter should be treated as the specification for the component unless it is desired to create a separate specification document.

**New Services:**

For new services, a separate service specification document should be created for the service. This chapter then references that document where appropriate and establishes the utilization contract between the solution and the service. In particular, this chapter should focus on the specifics of the utilization, identifying the interfaces to be used along with the volumes of utilization and required SLAs for those interactions.

## Business Process Involvement

Indicate how the component participates in the solution’s business processes. In most cases, this is detailed in the process-pattern mappings of the business process chapters. In such cases, a list of references to those diagrams is sufficient.

## Interfaces

Document the component interfaces used by the solution. Use UML Class diagrams here, place textual details (WSDLs, etc.) in the appendices.

## Observable Architecture

Document the observable architecture of the component (components upon which this component depends for proper operation).

## Observable State

If the component has stateful information impacting its observable behavior, document the state and the manner in which it impacts the observable behavior.

## Coordination

Indicate the patterns used to coordinate this component’s activities with other solution activities. This is particularly important when these patterns involve multiple interactions.

## Constraints

If there are constraints on when certain operations may be invoked, document them here. Some constraints are obvious: a cancelOrder() call is invalid when a placeOrder() call has not been made for that order. Others are less obvious: cancelOrder() cannot be called for an order that has already shipped.

## Non-Functional Behavior

Document the non-functional behaviors required of this component. These may be partially specified in the chapter Addressing Non-Functional Solution Requirements. Use this section to add component-specific details, as needed.

### Performance and Scalability

Describe how this component will achieve its performance and scalability requirements. Where analysis indicates that more than one instance of a component may be required to handle the load, describe the mechanisms to be used for load distribution.

### Availability within a Data Center

Describe how the availability requirements for this component will be achieved. Where meeting availability requirements result in the need for multiple components operating in a fault-tolerant or high-availability group, describe the manner in which the group will be managed and how work-in-progress will be handled.

### Site Disaster Recovery

Describe how the site disaster recovery requirements (recovery point objective, recovery time objective) for this component will be achieved. Be specific about how component state information is replicated between the sites and how work-in-progress will be handled.

### Security

Describe how the security requirements for this component will be achieved. Be specific about the roles of authentication and authorization servers and the points at which they interact with components. Be specific about how credentials will be managed, particularly when credentials can expire.

# Component/Service B

Document the component and its participation in the solution.

# Component/Service n

Document the component and its participation in the solution.

# Deployment

## Deployment Environment Migration

Describe how the solution will be migrated from one environment to another. Pay specific attention to how code and configuration information will be migrated. Describe the role that source control and deployment scripts will play. Describe the configuration steps that are required once the solution has been migrated.

## Development Configuration

Describe the development environment and the manner in which the solution will be assembled in this environment. How many machines of which type and what components are installed on each. Describe the network topology.

## Test Configuration

Describe the environment. How many machines of which type and what components are installed on each. Describe the network topology.

## Production Configuration

Describe the environment. How many machines of which type and what components are installed on each. Describe the network topology.

# Integration and Testing Requirements

## Integration Strategy

Describe the sequence in which the components and services will be integrated and define the test harnesses and test data required at each step. Describe how the solution will be deployed in this environment.

## Behavioral Testing

Describe the how functional testing will be performed, including test cases and test data (may reference another document). Describe how the solution will be deployed in this environment. Pay particular attention to the test harness required and the mechanics of how test data are repeatably loaded so that test scenarios can be exactly repeated.

## Performance Testing

Describe how failures testing will be performed. Identify the failure scenarios that will be tested and the environment into which the solution will be deployed for the test.

## Performance Testing

Describe how the system’s ability to meet performance goals will be established. Describe the additional requirements on components that may be necessary to support this testing (logging, time stamps, etc.). Describe how the solution will be deployed in this environment. If production-scale testing is not possible in this environment, describe how the testing results can be extrapolated to predict production performance.

Appendix A: Common Data Format Specifications

Place the textual versions of data format specifications (e.g. .xsd definitions of XML schema) for common (shared) data formats here.

Appendix B: Message Format Specifications

Place the textual versions of message format specifications (e.g. .xsd definitions of XML schema) here.

Appendix C: Service Interface Specifications

Place WSDLs and other textual interface specifications here.

Appendix D: Data Storage Specifications

Place textual definitions of database schema here.