Project Name

Software Architecture Document

Prepared for

Customer Name

Wednesday, 5 October 2005

Version .1 Draft

Prepared by

Arnon Rotem-Gal-Oz

Software Architect

arnon@rgoarchitects.com

Contributors

[Document Contributors]

Revision and Signoff Sheet

Change Record

Date	Author	Version	Change reference
		.1	Initial draft for review/discussion

Reviewers

Name	Version approved	Position	Date

Table of Contents

1 SCOPE

1.1 Identification

This Software Architecture Document establishes the top-level design and overall architecture for the [Subsystem] of [System].

The documents explain the design details from different views – where a view is a description of the entire sub-system from a single perspective.

1.2 System Overview

[The section shall briefly state the purpose of the system and shall identify and describe the role, within the system, of the Subsystem to which this document applies.]

1.3 Document Overview

This section describes the structure and contents of the software architecture documentation.

1.3.1 Document Structure

[Edit this according to the actual views used]

The document contains the following chapters:

- Chapter 1- lists and outlines the contents of the documentation and explains how the documentation can be used by various stakeholders.

- Chapter 2 - includes the reference to all the applicable documents used in designing and describing the system's architecture.

- Chapter 3 – describes the top-most view of the architecture, i.e. the main goals behind the design; what are the major decisions affecting the solution; the major alternatives weighted and why they were abandoned etc.

- Chapter 4 – details the technology decisions that support the implementation of the architecture.

- Chapter 5 – overviews the main functional requirements - identifying the requirements that are the core of the system (or pose the most risk). This is complimentary view to the quality attributes described in chapter 3.

- Chapter 6 – is part of the logical/static view of the system, describing the partitioning or breakdown of the system into services [or components/packages in case of non-SOA]

- Chapter 7 – details the main classes of the software infrastructure (or framework) on which the solution is built. The infrastructure class view is also part of the logical view of the system

- Chapter 8 – the process view - details the partitioning of the solution into processes and threads of execution, including any timing issues.

- Chapter 9 - details the main packages and classes of the solution from the "business" perspective, that is, the major classes that were derived from the use case realizations detailed in chapter 5.

- Chapter 10 – describes the deployment of the software system unto hardware nodes. The view also describes any infrastructure used to support eh solution (e.g. database servers, active directory etc.). [for solutions where the infrastructure plays a major part it is probably worthwhile having additional views on the subject]

- Chapter 11 – describes the development environment used for developing the solution. The view also includes the rationale for OS selection as well as the rationale for selecting any COTS tools or components.

- Chapter 12 – describes the system from the security perspectives – that is, both communication security (COMMSEC) and computerization security (COMPUSEC).

- Chapter 13 – describes the safety related issues of the solution.

1.3.2 How Stakeholders can use the document

[Edit this to reflect the project's stakeholders]

This section lists the major stakeholder roles of the [Subsystem] and how they may use the SAD to address their concerns.

- Someone new to the project: Read the documentation roadmap (this chapter) to understand how views are documented and what the different views are. Read the system overview (Chapter 3) to understand the big-picture and system-level rationale. Next, read the package view to understand the top level of the solution as well as the deployment view for understanding of the hardware environment.

- Project manager: To help with project planning / on-going management concentrate on the views explaining the major requirements and the logical view (service, software infrastructure and domain views). Read the deployment view to understand the hardware requirements (which will help understand testing requirements)

- Security/Safety Analyst: Read the deployment view to understand the physical environment in which the system operates as well as the COMMSEC & COMPUSEC and Safety views that explain the important aspects of the system in their respective areas.

- Maintainer: Read the system-level rationale as well as the package view for an overview of the system. Read the development environment view for an understanding of what is needed to set up a working environment and the deployment view for an overview of the hardware environment. Read the various parts of the logical view (service, software infrastructure and domain views) for details on the software structure.

- Customer/Acquirer: Read the system-level overview and rational. Read the requirement view and the package view to gain a broad understanding of how the system is structured to carry out its mission and to gain an appreciation of the effort that must be made to build it.

- Users: Users will usually not be very interested in the architecture documentation; however they can read the package view to gain understanding of the system functionality.

- System Engineers: Read the system-level rationale and the package view to gain understanding of the top level structure of the software. Read the deployment view for the mapping of software and hardware components. Additionally read the safety view to understand the impact of safety issue on the overall system design.

- Developers: Developers (and designers) are the primary target of the architecture document and as such most if not all of the views are relevant reading. A top-down reading is to start with the system-level rationale followed by the requirements overview, service view, process view and deployment view – reading all these will help them gain understanding of the top-level architecture of the system. To gain a more detailed understanding also read the infrastructure and class views as well as the development environment view.

1.3.3 View Template

The [Subsystem] architecture is described as a number of related "view items" or views. Each view is a relatively small, self-contained (as much as possible) bundle of information about the system or a particular part/aspect of the system.

This section described the standard organization that the documentation for the different views obeys (when applicable):

1. A **Primary Presentation** that shows the elements (and their relationships) that composes the view. The primary describes the elements in the vocabulary of that viewpoint. Usually only the primary elements and relations are depicted in the view, but under some circumstances it may contain all of them (e.g. only few elements exist in a particular view). When views are particularly large the view may contain only a subset of the primary element or relation, in those cases, for example only the elements and relations that come into play during normal operation and relegating the error handling and exception processing to supporting documentation. The primary presentation is usually graphical. If so it will include a reference to the notation (such as UML). When the notation is informal, the key will say so and define the symbology used and its meaning.

2. **Element Catalogue** detailing at least all the elements depicted in the primary presentation. The Catalogue usually includes (it may contain additional sections as needed):

   a. **Scope** or Description - The name of each element in the view and its main properties or usage.

   b. **Relations** – Each view has specific types of relations it depicts among the elements in the view. This section describes the relations that are depicted in the primary presentation as well as others that were omitted for readability propos.

3. **Architectural Background** explaining why the design reflected in the view came to be. The goal of this section is to explain why the design is as it is and to convince that the design is sound and robust. Architectural Background includes:

   a. **Rationale** – The section explains why the design decisions reflected in the view were made as well as reflecting some (or all) of the rejected alternatives and why they were rejected.

   b. **Analysis results** – The section documents the results of the analysis that have been performed that helped shape the design to its current state.

   c. **Assumptions** – The section documents any assumptions made by the architecture/design team that affected the design. As well as the reasons for the assumptions made.

4. **Additional Information**. The section, as its name implies, contains information relevant to the view that is not included in any of the former sections.

2   APPLICABLE DOCUMENTS

2.1   External Standards and Applicable Documents

1. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems (ISBN 0-7381-2518-0)

2. Documenting Software Architectures; Clements et al (ISBN 0-201-70372-6)

2.2   Internal Procedures and Documents

1. [Subsystem SRS]

2.3   Customer Documents

1.

3 RATIONALE, BACKGROUND AND CONSTRAINTS

3.1 System Context

[Describe the system boundary and relations to "the world" (other systems)]

3.2 Architecture Principles

3.2.1 [Principle Name]

3.2.1.1 Description

[What does it mean]

3.2.1.2 Rationale / Benefits

[Why do we want to apply this principle?]

3.2.1.3 Implications

[What does it mean to use it?]

3.2.1.4 Alternatives

[What else - What are the other options we considered and why we didn't use them.]

3.2.1.5 Scope/Exceptions

[When and where does it apply?]

3.3 Constraints

3.3.1 [Constraint Name]

3.3.1.1 Definition

3.3.1.2 Scope

3.3.1.3 Implications

[What does it mean for the architecture? what are the limitations it places?]

3.3.1.4 Origin

[Who placed this constraint and why]

3.4 Utility Tree

Quality Attribute	Refinement	Scenario

3.5 Architecture Overview

[Describe architectural styles used and main components of the architecture]

3.6   Key Alternatives

[Generally describe alternative architecture and/or alternative if one was weighted –
point to an appendix for the details of the architectural evaluation]

4   TECHNOLOGY MAPPING

[List the technology decision that support the implementation of the architecture. While this is not a part of the architecture per se - it is very important since a technology mapping can greatly affect the ability to actually create an implementation that adheres to the architecture]

4.1   Primary Presentation

4.2   Element Catalogue

4.2.1   [Element #1]

4.2.1.1   Rationale

[Why the technology/tool/component was chosen]

4.2.1.2   Compliance

[How does it support the architecture]

4.2.1.3   Assumptions

4.3   Additional Information

5   REQUIREMENTS (OVER)VIEW

The requirements overview is not a part of the architecture description, it is however a further refinement of the system context in chapter 3.

Listed below are the main use cases and other requirements which had the

[This is only necessary for large projects where the use case model is so large you want to high-light the important ones]

5.1   Primary Presentations

[Key Use Case Diagrams]

5.2   Elements Catalogue

5.2.1   Use Cases

[Table including: Name, Scope, Key Scenario, ref to SSR]

5.2.2   Key Non-Functional Requirements

5.2.3   Use Case realizations

[The business processes that enable the fulfilment of the use cases

5.3   Additional Information

6 SERVICE VIEW

[For non architectures which are not SOA replace this with component/package view]

6.1 Primary Presentation

[List the main services

6.2 Element Catalogue

6.2.1 [Service #1..]

6.2.1.1 Scope

[Description of the package content]

6.2.1.2 Interfaces

[Main interfaces of the package]

6.3 Architectural Background

6.3.1 Rationale

6.3.2 Analysis Results

6.3.3 Assumptions

6.4 Additional Information

7 SOFTWARE INFRASTRUCTURE VIEW

[list the "framework" level classes, enablers etc. (e.g. logger, db access etc.). This may only be applicable at the system level and not in the sub-system level]

7.1 Primary Presentation

[Major Classes diagram]

7.2 Element Catalogue

7.2.1 [Class #1..]

7.2.1.1 Scope

[Description of the Class content]

7.2.1.2 Roles

7.2.1.3 Relations

7.3 Architectural Background

7.3.1 Rationale

7.3.2 Analysis Results

7.3.3 Assumptions

7.4 Additional Information

8 PROCESS VIEW

8.1 Primary Presentation

[Task/Thread/process diagrams]

8.2 Element Catalogue

8.2.1 [Task/Process #1]

8.2.1.1 Scope

8.2.1.2 Creation

8.2.1.3 Class Allocated to Task/Process

8.2.1.4 Timing

8.3 Architectural Background

8.3.1 Rationale

8.3.2 Analysis Results

8.3.3 Assumptions

8.4 Additional Information

9 BUSINESS DOMAIN VIEW

9.1 Primary Presentation

[Key Class/package/entities diagrams per service (or of the service if service==subsystem)]

9.2 Element Catalogue

9.2.1 [Class #1..]

9.2.1.1 Scope

[Description of the class/package content]

9.2.1.2 Services

9.2.1.3 Relations

9.3 Architectural Background

9.3.1 Rationale

9.3.2 Analysis Results

9.3.3 Assumptions

9.4 Additional Information

10 DEPLOYMENT VIEW

10.1 Primary Presentation

[Hardware components and interface diagrams]

10.2 Element Catalogue

10.2.1 [Hardware Node #1..]

10.2.1.1 Scope

[Description of the Hardware]

10.2.1.2 Process allocation to node

10.2.1.3 Packages allocation to node

10.3 Architectural Background

10.3.1 Rationale

10.3.2 Analysis Results

10.3.3 Assumptions

10.4 Additional Information

11 DEVELOPMENT ENVIRONMENT VIEW

[This can be part of the Software Development Plan (SDP) if one exists]

11.1 Primary Presentation

[OS, Tools, COTS etc.]

11.2 Element Catalogue

11.2.1 [Tool #1..]

11.3 Architectural Background

11.3.1 Rationale

11.3.2 Analysis Results

11.3.3 Assumptions

11.4 Additional Information

12   COMMSEC & COMPUSEC VIEW

12.1   Primary Presentation

[Security related components]

12.2   Element Catalogue

12.2.1   [Aspect #1..]

12.3   Architectural Background

12.3.1   Rationale

12.3.2   Analysis Results

12.3.3   Assumptions

12.4   Additional Information

13 SAFETY VIEW

13.1 Primary Presentation

[Safety related components]

13.2 Element Catalogue

13.3 [Aspect #1..]

13.4 Architectural Background

13.4.1 Rationale

13.4.2 Analysis Results

13.4.3 Assumptions

13.5 Additional Information