For Illustration Purposes Only



Solution Architecture

Project <Name>

(Design Template)

Author: Daljit R Banger

Version: 1.0

Classification : Public



Delete this note when using. $\mathbf{\Sigma}$

This page is left intentionally Blank.



AMindset

Table of Contents

1.	l	Document Management	. 4
•)	Control Sections	.4
		 Approvers and/or RACI Matrix 	.4
		o Distribution	.4
•)	Change History	.4
•)	Table of Figures	.4
2.	I	Executive Summary	. 5
•)	Problem Statement / Opportunity	. 5
C)	Business Context	. 5
C)	Technical Views	. 5
C)	Impact Analysis	. 5
•)	Proposed Solution	. 5
•)	Notional Costs	. 5
•)	Next Steps	. 5
3.	l	Introduction	. 6
•)	Business Views	. 6
C)	Problem Statement	. 6
C)	Capability / Service View	. 6
•)	Scope	.6
C)	In Scope	. 6
C)	Out Scope	. 6
4.	l	Requirements	. 7
•)	Business Requirements (<i>If required</i>)	. 7
•)	Functional Requirements	.7
•)	Non-Functional requirements	.7
ι	Js	er Stories (<i>If required</i>)	. 8
5.	l	High-Level Components	. 8
•	I	Presentation Tier	.9
•	I	Application and Business Logic	.9
•	I	Data and Information Flows	.9
•	I	Integration	.9

•	Infrastructure9	
0	Platforms9	
0	Network Elements and Flows9	
6.	Technical Design	
7.	Impact Analysis	
8.	Security Architecture	
9.	Implementation Approach11	
	Timeline, Sequencing11	
	Migration Approach and Dependencies11	
•	Implementation Approach (Optional)12	
10.	Risks, Mitigation & Treatment12	
11.	Appendices	
•	Referenced Documents13	
•	Compliance Matrix (optional)13	
•	Glossary of Terms / Acronyms13	

1. Document Management

Control Sections

The control section of the document provides the metadata for the Solution Design (SD) document i.e., the date of document creation, location, fast updated and by whom, which is often represented as a set of tables, examples of possible content is listed below.

o Approvers and/or RACI Matrix

A list of Approvers with Name, function, approval date and any comments.

A RACI Matrix may also be included in a simple **tabular format** which clarifies individuals or groups responsible for contribution or approval of the document as shown below.

Name	Responsible	Accountable	Consulted	Informed
Joe Blogs	\checkmark		\checkmark	
Fred Blogs			\checkmark	\checkmark
Sam Blogs		\checkmark	\checkmark	

The acronym RACI stands for the different responsibility types: *Responsible, Accountable, Consulted, and Informed*

o Distribution

This represents the distribution list for the document - If your organisation operates a classification marking scheme for distribution of documents then one must mark the document and distribute subject to the classification which may impact the distribution list especially when individuals are outside of your organisation.

Name	Contact details (email)	Function
Joe Blogs	joe@blogs.com	Business Sponsor
Fred Blogs	fred@blogs.com	Chief Technology Officer
Sam Blogs	sam@bloggs.com	Lead Architect

• Change History

Date	Version	Change Made by	Change
17/11/2023	0.01	Daljit Banger	Document Created
18/11/2023	0.02	Daljit Banger	Content Added
28/12/2023	1.0	Tech Author	Content Validation

• Table of Figures

In Most Word processing packages, you will find option to insert a Table of Figures. In Microsoft Word this is found in the Top Menu-> References ->Insert Table of Figures (shown below)

File Home Insert Draw Design Layout References Mailings Review View Help						
Table of Contents ~	ab ¹ Insert Endnote Insert Ab ¹ Next Footnote ~ Footnote Show Notes	i) Search Researcher	Insert Manage Sources	Insert Table of Figures	Mark Entry	Mark Citation
Table of Contents Footnotes 5 Research Citations & Bibliography Captions Index Table of Authorities				Table of Authorities		

2. Executive Summary

This section of the document summarizes key points found in this document and provides a snapshot of the context and salient points addressed.

Below are some key headlines to emphasize, however this section should be modified to address the information needs of the target audience and address the specific problem, issue, or opportunity **(PIO)**.

• Problem Statement / Opportunity

Here we Introduce the context surrounding the POI that the solution seeks to address, remediate, or solve. There are three short sections you may wish to pass through, and these are.

o Business Context

The elements which highlight the **Business view** of the POI which the Organisation faces and if not addressed will result in organisational pain in terms of financials, efficiency, or debt i.e., the key driver for the change.

o Technical Views

A high-level **System view** of the issues or 'choke points' which align to POI e.g., capabilities, capacity, performance security or absence of the services or enablers required to address a new business offering.

o Impact Analysis

The **Impact statement** should summaries the **consequences** (Compliance, financial and business processes) of not doing the required change or introducing the new solution.

• Proposed Solution

This short section expresses how the above POI will be resolved and highlights changes required for existing systems or business process – It would be prudent to include a solution (level 0) image and text explained from proceeding sections of the document.

Notional Costs

Here we highlight **key financials** e.g., *direct / indirect capital or operational expenditure, revenue,* software licence costs etc or any potential savings and hidden costs e.g., technical debt.

• Next Steps

This summary presents the known immediate next steps e.g. Delivery of a Minimum Viable Product (MVP). This can be discussed as technical and project next steps depending on the project e.g.

- **Technical** Summaries e.g., buy/build components sprint definitions or the initiation of a Proof-of-concept.
- o Project related Info Activity, milestones, and key deliverables for each phase

3. Introduction

This is the main body of the Solution Design Document and should **contain as much 'known' information as practically possible** to provide sufficient information for stakeholders (developers, testers etc). If there are any gaps that have been identified, then this section should highlight this and provide any mitigating actions to be confirmed (TBC)

Business Views

This section should highlight the authors appreciation of the business domain and include a current state business centric **capability map** and **value drivers**_that identifies the capabilities in relation to the business value being addressed and or any bits that are missing?

o Problem Statement

This brief section should frame and identify any **weaknesses**, **shortcomings from existing business or system processes and or platforms**. In essence this section should specify the driver for the change and the impact or effect of not making the change or introducing the new system / components.

o Capability / Service View

These sections should highlight the Capabilities that the solution is dependent on and will address together with any supporting Servicing which when orchestrated deliver the desired outcomes.

It is important to demonstrate that you understand the problem domain being addressed with all enablers captured in other sections of the document.

Scope

o In Scope

list the elements which are in scope for this project or for this phase of the project – maybe represent as a set of simple bullets?

o Out Scope

list all elements of the project or requirements that are not in scope of this project or if have moved to subsequent later phases of the programme.

Assumptions and Prerequisites

Any assumptions that you have made in the choice of the solution should be listed together with dependencies e.g., lack of resources or 3rd party required activities.

4. Requirements

This section of the document highlights and references the main requirements for the solution and should be presented in a format that meets the organisational standard.

Requirements should <u>map and trace</u> back to the Capabilities, Services and Enablers leveraged to realize the Solution.

To **avoid long unstructured lists**, requirements will often be arranged and group in relation to *essential and desirable items and to achieve this* several approaches exist with the most popular being the MoSCOW¹ classification.

If the organisation is following an agile approach to delivery, requirements and desired features may be presented as a set of **User Stories** which represent the informal explanations of **features** written from the perspective of the end user or client.

Requirements are essentially the artefacts produced by one or more project Business Analyst (BA) assigned to the project and can be simple or complex depending on the style adopted by the BA. In most cases the SD will focus on the functional and non-functional requirements the system must deliver against.

The requirements section can be split into three distinct areas., which are.

• Business Requirements (*If required*)

The Business Requirements represent the high-level wants relating to **goals, objectives or needs** of the Organisation e.g. "We need an accurate monthly sales forecast in all regions" here the requirement is for a <u>sales forecast system</u>.

If the Solution is based on a COTS product it would be advantageous to highlight the product roadmap if any exists.

• Functional Requirements

Functional requirements specify how the system will **responds and process** specific functions and inputs e.g. "When a Sale is booked all ledgers should be updated and the rolling forecasts updated against budget with variances if red shown as -ve" - here the requirement is for <u>ledger updating</u> in in <u>'real time'</u> and screens designed to <u>show the variances</u>.

• Non-Functional requirements

Non-Functional Requirements specify various **system characteristics** and **attributes** which should be observed during the operation of the system, examples of which are.

- Performance: i.e. The response time to requests e.g. "screen updates should be no less than 5 milliseconds when updating from the time of input." Often these requirements have <u>quantifiable performance indicators</u> to design against.
- Scalability: How well a system can handle an increase in users or workload e.g. the inclusion of a new country to the system and users will <u>scale without any impact on</u> <u>the overall system</u> component design

- Security: How well a system protects against unauthorized access and data breaches or <u>attacks</u> from friendly or hostile actors, encryption keys management, also the operation of secure transfers and hardening of endpoints etc.
- Usability: How easy a system is to use e.g. moving between features, functions, and screens
- Maintainability: How easy it is to update and modify the system including <u>documentation for support</u> during the life of the system.

User Stories (*If required*)

If the organisation has adopted a Agile approach to systems development, then requirements will surface as a set of Stories from the perspective of the User and follow a standard pattern which is 'As a <**Type of User**> I want <**Function**> with the Goal <**Outcome**> i.e. who wants the functionality, what they want and why they want it.

Example of a User story is "As a **sales operations manager** I want **real time updates** of sales so I can **adjust incentives**".

If User Stories are adopted, then it would be prudent to highlight the most 'significant' user stories in the SD.

5. High-Level Components

In this section of the document, we seek to present a view which encapsulates a set of **high-level components** which form part or all the solution, these views can represent the **services and enablers** which are required for the realisation of the solution and the section should state both <u>internal</u> and <u>external</u> boundaries of the system².

Using the organisational notational standards, for example UML, ArchiMate, BPMN, C4 etc. It would be prudent to show, at a minimum, the following architectural layers as depicted below i.e. visually with context explaining how the components service the solution against the various Use Cases where.



² It is important to define the boundaries both at the system and organizational level, as this will aid in the definition of trusted zones for the security controls required, to manage both data and systems in these zones.

Presentation Tier

The Presentation layer represents the <u>user interaction</u> specification and controls, where a User can be a Human or Machine? This section allows stakeholders and testers to understand the UI/UX features and with the user stories can allow for the creation of test cases and hence should highlight the system channel interactions.

Application and Business Logic

Here one can show the software components, which orchestrate processes and support workflows together, with any triggers, alerts, and notifications during the core orchestrations.

Data and Information Flows

In this section we highlight major data flows between components within the system and those of the inbound and outbound. If there are any data transformations

Suggestion should be made to Ingested 'raw. Data, Transformations, Master Data, Static and Dynamic Reference data etc.

Integration

In this section we highlight the use of any API Gateways, How APIs are published exposed together with any schemas and version controls.

If data is flowing in or out of your system using a secure FTP service, this should be documented and how the files are moved from a public facing area to the trusted areas of your system.

Infrastructure

I like to refer to this section as the 'Tin & String' section which highlights of servers, virtual machines, and any other devices e.g. firewalls. which form part of the solution. Whilst this will be subsequently documented in a Technical Design or production acceptance documentation it is prudent to highlight or reference any material here.

o Platforms

Platform is often the aggregation of multiple systems that ultimately provide a service or set of services which enable the realisation of business capabilities. If the Solution leverages a platform, then references to the platform documentation should be made and highlight how they fit into the overall solution.

Network Elements and Flows

The Network and any configuration changes should be documented in any subsequent Technical Design documents; however, it would be prudent to show the overall fit of the solution in respect to the Network (MPLS, LANS, VLANS) and any bandwidth Segmentation or routing which can affect overall network traffic.

NB: It musted be stressed that IP Addresses, Open Ports, Passwords etc should be restricted in diagrams unless the document is classified for internal use only.

If you reference any repeatable patterns (internal or external), in any of the above, it is necessary to provide any URLs and a description of how the patterns are instantiated or realized.

AMindse

6. Technical Design

In some cases, it may be necessary, for clarity, to show the next level of decomposition for the components of the solution (we normally refer to this as the level 2 set of attributes). Whilst this section of the document often presents a high-level view of the services and enablers and associated technical components you may wish to use the appendices to show the lower level of detail and use this section to keep it at a high abstract level.

NB: System configurations should be specified elsewhere e.g. in release documents or in a version control system.

Additional flowcharts, pseudo-code, and process diagrams may be used when highlighting lower levels of details.

Any reference to bespoke code, data models, Database tables, SQL or functions should be moved to the appendices or referenced and documented elsewhere.

7. Impact Analysis

The Impact Analysis (IA) section can be brief or extended subject to the complexity of the solution. The IA should highlight any major impact of any proposed changes in terms of interactions, process, people, or organisational changes.



If the Solution is based on an existing COTs product and proposed 'workarounds' may result in the creation of Technical Debt and possible degradation or interruption of service after the update. then this must be captured and documented

The Impact assessment can take the form of a <u>bidirectional traceability matrix which</u> links business requirements to the value from ode and specific test cases and user documentation.

If the solution will utilise, store or forward any personal information then an organisational Privacy Impact Assessments (PIAs) should be completed and referenced here.

8. Security Architecture

In most cases, your Organisation will have specific control, policies and or guardrails which the Solution must comply with, and documents completed or complied before any solution can be deployed into production.

In this section you may consider placing some UML type sequence diagrams which illustrate authentication, security tokens, certificates exposed in handling identify and session management between the User and the System highlighting the Components used to validate a user's credentials and provide the resource access rights.

This section will have several aspects, examples of which are.

- Any Modifications to the Organisational Trust Model
- Identify & Access Management Single Sign-On and Access for applications and services.
- Controls for **Single Sign on** If enforced
- Alerting and Logging Processes and any interactions with the Security Operations Centre (SOC)
- Managing the **Confidentiality, Integrity** of Data in transit and at rest.
- **Compliance** How the Solution will comply with the organisational guardrails.

9. Implementation Approach

This section is normally light weight in content and will reference the project plan, backlogs and defined sprints.

You may wish to include.

• Timeline, Sequencing

Reference to the Delivery Dates, if phased any Gant Charts showing Milestones and Deliverables

• Migration Approach and Dependencies

This should reference the following.

- Data / Users How Data from Legacy systems will be migrated with users to the new system.
- Application Any configuration Data which needs to be considered (Time Zones etc)
- Infrastructure If moving to a new hosting model e.g. 3rd party cloud services, highlight the approach e.g. lift and shift, phased migration etc
- Decommissioning How components and especially any data will be removed and destroyed or retained with the period for retention.

EAMindset

• Implementation Approach (Optional)

This is optional and will reference the methodology adopted e.g. classic waterfall, Agile or mix of both.

Specify and 'rinse and repeat' processes or patterns that can be adopted by the organisation.

10. Risks, Mitigation & Treatment

Risks may be identified in the Security or Impact section, however if there is a set of specific high level of risk associated with the project (timelines, dependencies etc) then this should be documented. Below are some considerations.

- External **3**rd **party** dependencies on the availability of components e.g. software frameworks, modules, specifications, feedback and or resource availability.
- Any roll-back issues introduced due to work on legacy, poorly documented, or badly designed or implemented systems.
- **Incomplete requirements** and decisions to be deferred until after the project has started which is often found in Agile type programmes.
- Any missing or unavailable **prerequisites** for coders, testers, or operations such as environments, tools etc.

11. Appendices

This will obviously be dependent on the content in the SD however, common information forum in appendices are shown below;

• Referenced Documents

Links to texts and URLs referenced in this document and may include Enterprise Reference Models, Patterns and Standards that the design must follow to remain compliant.

Doc Ref	Document Name	Source Location	Comments
1	Reflecting on Error Management and listing HTTP Status Codes	https://dalbanger.blogspot.com/202 3/03/reflecting-on-error- management-and.html	Example URL
2	Enterprise Systems Architecture: Aligning Business Operating Models to Technology Landscapes	ISBN: 978-1484286456	Example Link to 3rd Party Resource

• Compliance Matrix (optional)

If your organization or client has a set of policies, standards and principles that must be followed then you may wish to show a simple table referencing the obligation and if the solution is compliant with a simple Y/N.

• Glossary of Terms / Acronyms

The following table lists some of terms and acronyms used throughout this book.

Acronym	Label	Description
ΑΡΙ	Application Programming Interface	A set of shared definitions and protocols for building and integrating application software between systems.
COTs	Commercial off-the-shelf	Pre-packaged software or hardware purchased, leased or even licensed commercially to multiple clients, - Often features, functions can be configured or adapted to meet most of the needs of the organization
LLM	Large language models	A LLM is a type of artificial intelligence (AI) algorithm that uses deep learning techniques and extremely large data sets to understand, summarize, generate and predict new content
ETL	Extract, Transform and Load	Three core Enterprise Data Management functions that when combined seek to pull data out of one or more data stores and place it into another data store.

TAMindse

This page is left Intentionally Blank