Template - Design

- Problem Statement
- Expected Benefits of Solution
- Key Use Cases
  - Use Case - 1
  - Use Case - 2
  - Use Case - 3
- Context and Requirements
- Logical Architecture
- Physical Architecture
- Sequence Diagrams of Key Use Cases
- Availability and Fault Tolerance
  - Strategies
  - Maintenance
- Scalability
  - Transaction Volume
  - Data Volume
  - Scaling Strategy
- Performance
- Security Model
- Verification and Monitoring
- Risk Summary
- Cost
- Resources

Status: DRAFT

Problem Statement

<1-3 sentences describing the problem and how it impacts customers, business, or organization

Extra: Wardley Map of the current state>

Expected Benefits of Solution

<1-3 sentences describing the benefits of the proposed solution in the context of the problem

be clear if/when the solution does not solve significant portions of the problem statement

Required: Wardley Map of the target state>

Key Use Cases

<describe the ~3 most important use cases, max 5>

Use Case - 1

Use Case - 2

Use Case - 3

Context and Requirements

<Briefly Describe:

- the current solution to this problem
- significant constraints within the environment
- major business requirements
- major technical requirements

>

Logical Architecture

<block diagram illustrating logical architecture of solution>

Physical Architecture

<describe how the system will be implemented, including aspects such as:

- description of how components implementing the system will interact
- key data models, structures, and wire formats
- when and where data is persisted

>

Sequence Diagrams of Key Use Cases

<optional, but often very useful to provide sequence diagrams for key use cases>

diagramming tool: https://www.websequencediagrams.com

Availability and Fault Tolerance

<Describe Availability Requirements of the System in terms of ‘Nines’>

<Describe who and how dependents of system will be affected>

Strategies

<Describe how the system achieves fault tolerance, e.g.
circuit breakers
failover
rate limits
retry-with-backoff
feature flags
>

Maintenance

<Can the system be maintained while online or is will a ‘maintenance’ window be required? How frequently?>

Scalability

Transaction Volume

<describe peak and ‘normal’ throughput requirements>

Data Volume

<describe peak and ‘normal’ data volume requirements, incl data retention>

Scaling Strategy

<describe how system if and how system will scale up to handle more work; can system scale down? what are the minimum and maximum footprints?

Unit of Service Scalability

processes, databases?

Scaling of Reads, Writes

Limits

what limits does the system impose to facilitate scaling and fault tolerance, e.g. message size, per-customer/tenant limits

>

Performance

<describe performance requirements for most important and lowest-latency use cases & transactions, e.g. p95 & p99 response time @ peak throughput>

<how will performance requirements be validated?>

Security Model

<

Describe Key Aspects of the Security Model for this design:

Authentication & Authorization of Access

Classification of System’s Data

Standard/Sensitive? Regulatory requirements: PHI/PCI

Data Protection

- in-flight
- at-rest

>

Verification and Monitoring

<

How will operators know the system is operating correctly?
Does the system have built-in mechanisms to detect errors?
will this data be propagated to monitoring systems?
how is data distributed and is there a way to verify its consistency?
How can the system be tested while online?
can this be done continuously?
>

Risk Summary

The following table summarizes effects of the most important failure modes in this design (FMEA).

Scales are relative with:

- 1 - best: no impact, occurs less than once per year, detected immediately
- 5 - worst: system unsafe to use, occurs monthly, detection longer than 1 hour

Failure Mode	Severity	Occurrence	Detection	RPN

Top 3-5 Failure Modes present in this design

Cost

These are the rough, expected costs to develop and operate the system:

Develop System

< Describe the costs required to develop the system in people and money.

1 month of a Cloud Architect and Engineer to implement an end-to-end prototype

1 month of a Market Analyst to gather feedback on the prototype

2-3 months for a Cloud Architect and Engineer develop features of MVP

1 month for a Cloud Engineer to implement v1 of automated system provisioning process for Cloud and application resources

2 weeks for a Cloud Engineer to implement end-to-end monitoring of system

Total: $NNN

>

Operate System

<Describe the costs required to operate the system in people and money.

Compute

Storage

Total: $NNN / month

>

Resources

- Helpful Link 1
- Helpful Link 2