# Writing Technical Design Docs, Revisited

Engineering Insights

Talin Follow
Dec 4, 2019 · 4 min read

In the past ten months I've received a lot of interest for my essay, Writing Technical Design Docs. In this article I'd like to present an example of a more refined design doc template. This is the template we use in the Creative Tools division at Amazon Web Services. It was created by my co-worker Graeme McHale, with input and suggestions by myself and others on the Creative Tools team. It is used here with his permission.

## Technical Design Document Template

### 1. Description of the Problem

Give a brief (one paragraph) summary of the problem you are solving. Work to keep away from technical detail. Try instead to talk about this in terms of the problem as it pertains to your customers. At the end of reading this document, any team member should be able to understand the problem, how you intend to solve it, and who are the stakeholders.

### 2. Solution Requirements

Briefly describe what is required of a solution which addresses the problem. Try to steer clear of the how (implementation detail) and concentrate on what is required any solution in order to address the problem outlined above.

### 3. Glossary

Link to the service wide glossary, and define any new terms used in this document.

### 4. Out of Scope (Non-goals)

Explicitly call out what is not in scope (Sometimes articulating what you are not going to do is an easier way to define scope than to talk about what you are going to do.)

### 5. Assumptions

What are you assuming will be true or in place to make your solution successful?

### 6. Solution

Your solution goes here.

Start by including a high level diagram and decompose from there. Please diagram (where possible) how your solution interacts with other subsystems and services (including sequence diagrams for complex interactions).

Aim to answer:

- What are you going to deliver?
- What are your upstream and downstream dependencies?
- How does it fit in to the broader service?
- How will it scale?
- What are the limits of your solution?
- How will you ensure fault tolerance and quick recovery after failure?
- How might your solution evolve to meet future requirements?

## 7. Security Considerations

What are the security implications of your solution? If your solution is large enough in scope to warrant its own threat model, please add it here, otherwise please describe how your solution impacts existing threat models.

## 8. Cost Analysis

A high level analysis of the costs that will be incurred in running your chosen solution on a day-to-day basis.

## 9. Cross-region Considerations

If applicable, how does your solution optimize or is compatible with cross-region requirements. This includes data transfer costs between regions, availability of the service in different data centers, latency issues, etc.

## 10. Operational Readiness Considerations

Discuss how your solution will support operational excellence, ensuring customer satisfaction with a frugal level of support.

Aim to answer:

- How your chosen solution will be deployed?

- What metrics and alarms will be key to monitoring the health of your solution?
- How are your solution limits enforced?
- Will there be any throttling or blacklisting mechanisms in place?
- Will there be any data recovery mechanisms in place?
- If this is a multi-tenant solution, how are you dealing with noisy neighbor issues?
- How will your solution be debugged when problems occur?
- How will your solution recover in case of a brown-out?
- Are there any operational tools required for your solution?

## 11. Risks and Open Issues

If there are any risks or unknowns, list them here. Are there any open questions which could impact your design for which you do not currently have answers? How are you going to get answers? Will any required team members be loaned to other teams during the time slated for implementation? Are all of required dependencies available in all the regions you need them? What are the one way doors, and are we sure we want to go through them?

## 12. Solutions considered and discarded

What alternatives have you have considered and discarded? Why don't these work? Be **brief**, linking to other documents for details is ok, but always provide a summary inline.

Only alternative solutions that an impartial observer would deem credible need be documented.

If an alternative solution is not appropriate now, but may be in the future, please discuss potential migration paths.

## 13. Work Required

Include a high level breakdown of the work required to implement your proposed solution, including t-shirt size estimates (S, M, XL) where appropriate. Also, specifically call out if this solution requires resources from other teams to be completed (away teams, dependencies etc.)

## 14. High-level Test Plan

At a high level, describe how your chosen solution be tested.

## 15. References

Links to any other documents that may be relevant, or sources you wish to cite.