Author: You
Reviewer: Designate a reviewer right at the start. The reviewer needs to be able to solve the problem and will provide sign-off when the design is ready for implementation (just like GitHub).
Stakeholders: DRIs and advisors are a good place to start. It's better to overshare than to undershare, people can always opt-out. Don't worry about adding them before things are done, remember this is all about collaborating to the best solutions possible.
Project: What project is this design associated with?

Background & Requirements

What information do people need to understand the problem and approaches? Are there specific requirements for the approaches? This section should have enough information to empower others to participate, but not so much that they swim in it. You may also consider including:

1. Technical or non-technical constraints
2. Prior or concurrent work (links to projects or tech designs are fine!)

Problem statement

This is the most important section. What is the problem that this design attempts to solve? Try to keep it short. If the problem is really complicated consider how it can be broken up (*remember: one tech design per problem*). If new problems arise in implementation make a comment to get the reviewer involved and spin up an additional design.

Once you're satisfied get the reviewer involved for validation before moving on to the approaches.

Approaches

What are feasible approaches for solving the problem? No one has all the answers and sometimes you need to experiment with different ideas. That's fine, but *don't go alone*! Post a comment or sync up with the reviewer before going too deep. The whole idea is to <u>not</u> disappear into a vacuum.

1. Describe the approach and state the reason why this approach is, *or isn't*, being considered. A diagram may be useful if the approach is complicated.

Deployment Plan

This section should describe what steps need to be taken to deploy.

Testing

Does the approach introduce new functionality that needs to be tested? Are there critical algorithm changes? This section should describe what needs to be tested and how it will be done.

Have you gotten initial review from

- ❏ Architecture Team
- ❏ Performance Team
- ❏ DBA Team
- ❏ Security Team

Additional considerations (if applicable)

1. **Rollback plan:** Describe how this plan can be safely deployed and safely undeployed in the event of breakages
2. **Experimentation:** Describe how this implementation could be validated through experimentation.
3. **Monitoring**: Describe how this can be monitored (and include a SRE team member).
4. **Operational**: Describe how the implementation will scale with predicted growth. Include any special production/hardware considerations.
5. **Privacy:** Describe potential privacy risks with the proposed implementation and how they will be mitigated. If there are unaddressed concerns or open questions, cross post this design to the Privacy project under the *Triage* section. Examples:
6. If using a client side library, describe its interactions with external services.
7. When doing client requests, explain how they will be proxied through our servers.
8. If sensitive user information will be shared with upstreams, explain how it will be obfuscated.