# RFCs

An “RFC” literally means a “Request for Comments”.

RFCs give us a way to write down ideas and plans so we can communicate, collect thoughtful feedback from others on the team, and align expectations. An RFC can be thought of as an asynchronous conversation.

# When to use an RFC

## Use an RFC when…

* You want to frame a problem and propose a solution.
* You want thoughtful feedback from team members on our globally-distributed remote team.
* You want to surface an idea, tension, or feedback.
* You want to define a project or design brief to drive project collaboration.
* You need to surface and communicate around a highly cross-functional decision with our formal decisionmaking process.

## Don’t use an RFC when…

* You want to discuss personal or sensitive topics one-on-one with another team member.
* You want to make a decision to change something where you are the decider. In the vast majority of cases, creating an RFC to explain yourself will be overkill. RFCs should only be used if a decision explicitly requires one of the bullets in the section above.

How RFCs are different from ADRs (Architecture Decision Record)?

* RFCs drive *discussion*, ADR drive *decision;*
* RFCs are *mutable* - via Errata section, ADRs are *immutable* - each change requires a new ADR;
* In most cases teams create an RFC first that in turn might or might not result in an ADR

# References

<https://en.wikipedia.org/wiki/Request_for_Comments>  
<https://www.ietf.org/standards/rfcs/>  
<https://works.hashicorp.com/articles/rfc-template><https://handbook.sourcegraph.com/company-info-and-process/communication/rfcs/><https://github.com/tensorflow/community/blob/master/rfcs/yyyymmdd-rfc-template.md><https://github.com/tensorflow/community/tree/master/rfcs><https://cwiki.apache.org/confluence/display/GEODE/RFC+Template><https://blog.pragmaticengineer.com/rfcs-and-design-docs/>