Architecture Decision Record

- Documents a decision that addresses a requirement (functional/non-functional) that has significant impact on the application's architecture.
- An architecturally significant requirement is something that has a measurable effect on the application architecture's qualities such as
  - Usability
  - Reliability
  - Availability
  - Portability
  - Testability
  - Scalability
  - Flexibility
  - Reusability
  - Maintainability
  - Supportability
  - Interoperability
  - Performance
  - Security
- The solution to a problem/requirement can start as an RFC to facilitate discussion and collaboration among the community members. The outcome of the RFC must be recorded as an ADR.
- ADR can also be used to document patterns/standards that forms naturally. Such patterns can be identified when doing code reviews.
- Applicable Status
  - DRAFT
  - ACCEPTED
  - SUPERSEDED by ADR-xxx
  - OBSOLETE
- Add label to confluence pages that has an ADR. Available tags
  - Database
  - Cloud
  - Languages
  - Java
  - JS
  - Spring
  - NodeJS
  - Security
  - Performance
  - API
  - Streaming
  - CI/CD

Resources
https://github.com/npryce/adr-tools

Questions
- Is there any value in maintaining a decision log at the repo level? Won't be an ADR.