What is an architecture decision record (ADR), and how do you write one?

20 November 2026 · 4 min read · blogs.iictp.org/architecture-decision-records-explained

An architecture decision record is a short document that captures one significant technical decision: the situation, what was decided, and what follows from it. Kept next to the code, a set of ADRs tells future engineers not just what the system looks like but why.

TL;DR The short version

- An architecture decision record captures one significant decision: the context, the decision and its consequences.
- Write one for decisions that are expensive to undo or likely to be questioned, and keep it in the repository it affects.
- Never edit an old record when a decision changes; supersede it with a new one, and draft ADRs as part of the work.

Read online, listen to it, or change the text size: https://blogs.iictp.org/architecture-decision-records-explained

What goes in an ADR?

Michael Nygard's widely used format has a title, a status such as proposed, accepted or superseded, the context that forced the decision, the decision itself, and its consequences, good and bad. Many teams add the options they considered and why they were rejected. One or two pages is enough.

Which decisions deserve one?

Those that are expensive to undo or that someone will later question: choosing a database, splitting a service, adopting a framework, changing how authentication works. A useful test is whether a new team member would ask "why did we do it this way?".

Where should ADRs live?

In the repository the decision affects, in a folder such as docs/adr, numbered in order. They are reviewed in pull requests like code, versioned with it, and found by the people who change it.

What happens when a decision changes?

Do not edit the old record. Write a new ADR that supersedes it, and mark the old one as superseded with a link. The history of changed decisions is itself useful, because it shows what the team learned.

How do you keep the habit going?

Make writing an ADR part of the work, not a separate task: draft it when the decision is being discussed, use it as the agenda for the review, and link it from the pull request that implements it. Short, plain records get written; long templates get skipped.

> Related credential: Certified Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Nygard, M. Documenting Architecture Decisions. Cognitect blog, 2011.
2. Keeling, M. Design It! From Programmer to Software Architect. Pragmatic Bookshelf, 2017.

Published by IICTP, an independent technical credential body, on 20 November 2026. The current version of this post is at https://blogs.iictp.org/architecture-decision-records-explained. More guides: blogs.iictp.org.