| Decision: # | Decision Title |
| --- | --- |
| Issue | Describe the architectural design issue you’re addressing, leaving no questions about why you’re addressing this issue now. Following a minimalist approach, address and document only the issues that need addressing at various points in the life cycle. |
| Decision | Clearly state the architecture’s direction – that is, the position you’ve selected. |
| Status | The decision’s status, such as pending, decided, or approved. |
| Group | You can use a simple grouping – such as integration, presentation, data and so on – to help organize the set of decisions. You could also use a more sophisticated architecture ontology. |
| Assumptions | Clearly describe the underlying assumptions in the environment in which you’re making the decision – cost, schedule, technology and so on. Note that environmental constraints (such as accepted technology standards, enterprise architecture, commonly employed patterns, and so on) might limit the alternatives you consider. |
| Constraints | Capture any additional constraints to the environment that the chosen alternative (the decision) might pose. |
| Positions | List the positions (viable options or alternatives) you considered. These often require long explanations, sometimes even models and diagrams. This isn’t an exhaustive list. However you don’t want to hear the question “Did you think about …?” during a final review; this leads to loss of credibility and questioning of other architectural decisions. This section also helps ensure that you heard other’s opinions; explicitly stating other opinions helps enroll their advocates in your decision. |
| Argument | Outline why you selected a position, including items such as implementation cost, total ownership cost, time to market, and required development resources, availability. This is probably as important as the decision itself. |
| Implications | A decision comes with many implications. For example, a decision might introduce a need to make other decisions, create new requirements, or modify existing requirements; pose additional constraints to the environment; require renegotiating scope or schedule with customers; or require additional staff training. Clearly understanding and stating your decision’s implications can be very effective in gaining buy-in and creating a roadmap for architecture execution. |
| Related Decisions | It’s obvious that many decisions are related; you can list them here. However, we’ve found that in practice, a traceability matrix, decision trees, or metamodels are more useful for showing complex relationships. |
| Related Requirements | Decisions should be business driven. To show accountability, explicitly map your decisions to the objectives or requirements. You can enumerate these related requirements here, but we’ve found it more convenient to reference a traceability matrix. You can assess each architecture decision’s contribution to meeting each requirement, and then assess how well the requirement is met across the decisions. If a decision doesn’t contribute to meeting a requirement, don’t make that decision. |
| Related Artifacts | List the related architecture, design, or scope documents that this decision impacts. |
| Related Principles | If the enterprise has an agreed-upon set of principles, make sure the decision is consistent with one or more of them. This helps ensure alignment along domains or systems. |
| Notes | Because the decision-making process can take weeks, we’ve found it useful to capture notes and issues that the team discusses during the socialization process. |