

0001-use-clean-architecture

- [ADR 0001: Use Clean Architecture](#)
 - [Status](#)
 - [Context](#)
 - [Decision](#)
 - [Consequences](#)

ADR 0001: Use Clean Architecture

Status

Accepted

Context

We need a solution structure that supports maintainability, testability, and clear separation of concerns for our .NET 9 workshop. The codebase should be easy to extend, refactor, and onboard new contributors, while supporting modern patterns like DDD and CQRS.

Decision

We will use Clean Architecture as the foundation for our solution. This means:

- Organizing code into Domain, Application, Infrastructure, and Api layers.
- Enforcing dependencies: Domain → Application → Infrastructure → Api.
- Keeping business logic in the Domain and Application layers only.
- Using feature-oriented folders to group related code.
- Applying DDD modeling rules and CQRS for business logic.

Consequences

- The solution will be easier to maintain and scale.
- New features can be added with minimal impact on existing code.
- Testing is simplified by clear boundaries and dependency injection.
- Onboarding is faster due to consistent structure and documentation.
- Some initial overhead in learning the architecture, but long-term benefits outweigh this cost.

See also: [Sample Solution Architecture](#), [Glossary](#)