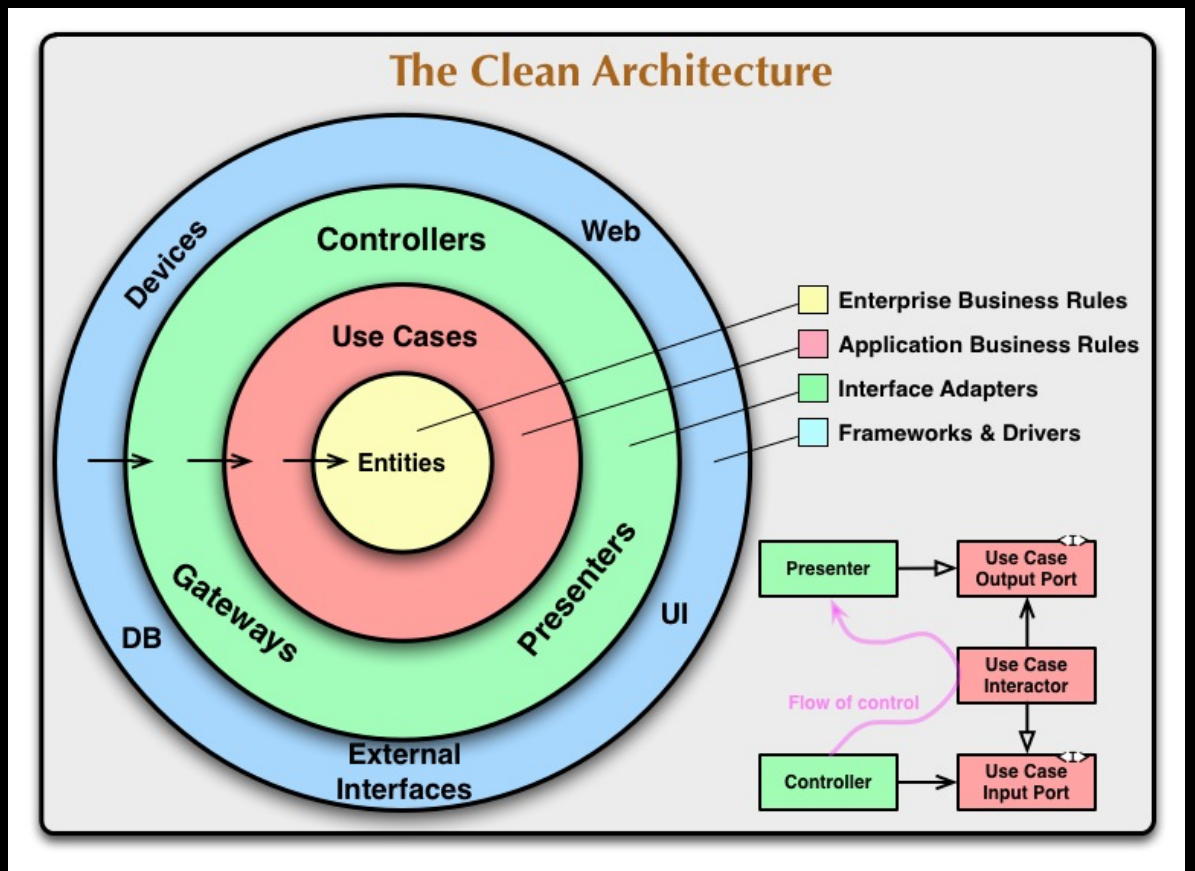
The Clean Architecture

The Dependency Rule



- Source code dependencies must point only inward, toward higher-level policies
  - Inner circles are policies, and outer circles are mechanisms
  - Inner circles knows nothing about outer circles
- Interface adapters - convert data used by use cases and entities into that for external agency like database or web
- Crossing boundaries
  - align the flow of control and source code dependencies
  - Dependency Inversion Principle, dynamic polymorphism
- The data passed across a boundary always in the form of inner circles

Presenters and Humble Objects

Data pattern across architectural boundaries

- The Humble Object Pattern
    - Split the behaviors into hard-to-test behaviors (e.g., views) and testable behaviors (e.g., presenters)
    - Hubble object contains all hard-to-test behaviors
- Presenters and Views