🎯 What it covers:

- Deep dive into **software architecture**:
  - Architectural genres.
  - Architectural styles and patterns.
  - Architectural decisions and rationale.
  - Representing architectures with diagrams.

Architectural styles explained:

- **Data-centered architectures**:
  - Central repository (database, blackboard).
  - Clients interact with shared data store.
- **Data-flow architectures**:
  - Pipe & filter systems (transform data through stages).
- **Call and return architectures**:
  - Hierarchical structure; main program calls submodules.
- **Object-oriented architectures**:
  - System built around interacting objects.
- **Layered architectures**:
  - System divided into layers (e.g. UI, business logic, data).

Architectural considerations:

- Performance vs. maintainability.
- Security layers.
- Scalability and fault tolerance.
- Architectural decisions are documented for future developers.

Why it matters:

- Architectural decisions shape:
  - Non-functional requirements (performance, security, maintainability).
  - Long-term cost and adaptability of the system.
- For AI:
  - Choosing architecture impacts data flows, model serving, real-time constraints.
  - Helps balance scalability and performance for high-load inference services.