

《Software Architecture 4+1》模型由Philippe Kruchten提出，旨在通过多视角描述复杂软件系统架构，以满足不同利益相关者的需求。该模型包含四个核心视图和一个场景视图，具体如下：

1. 逻辑视图 (Logical View)

• **目的**：描述系统的功能组件及其关系，关注如何满足功能需求。• **目标用户**：系统分析师、设计师、开发人员。• **关键内容**：类、包、组件、接口、交互关系等。• **模型工具**：类图 (UML)、对象图、领域模型。• **示例**：用户管理模块中的类（如 `User`、`AuthService`）及其继承或组合关系。

2. 开发视图 (Development View / Implementation View)

• **目的**：组织代码结构，管理模块化开发与构建。• **目标用户**：开发团队、维护人员。• **关键内容**：模块划分、子系统、代码库、构建工具、依赖管理。• **模型工具**：组件图 (UML)、包图、构建系统配置（如Maven、CMake）。• **示例**：将系统分为前端（React模块）、后端（Spring Boot服务）和数据库（MySQL驱动）的子模块。

3. 进程视图 (Process View / Concurrency View)

• **目的**：处理运行时行为，如并发、同步、通信和性能。• **目标用户**：系统架构师、性能优化工程师。• **关键内容**：进程、线程、任务调度、IPC（进程间通信）、锁机制。• **模型工具**：活动图 (UML)、序列图、通信图。• **示例**：电商系统中订单处理与库存更新的异步消息队列（如Kafka）设计。

4. 物理视图 (Physical View / Deployment View)

• **目的**：描述系统在物理环境中的部署，如硬件和网络配置。• **目标用户**：运维团队、基础设施工程师。• **关键内容**：服务器、节点、网络拓扑、存储设备、容灾策略。• **模型工具**：部署图 (UML)、云架构图（如AWS/Azure）。• **示例**：Web服务器集群部署在AWS EC2，数据库采用RDS主从复制。

+1. 场景视图 (Scenarios View / Use Case View)

• **目的**：通过典型用例验证其他视图的一致性，确保架构满足需求。• **目标用户**：所有利益相关者（包括最终用户）。• **关键内容**：用户故事、关键业务流程、端到端交互。• **模型工具**：用例图 (UML)、序列图、用户旅程图。• **示例**：“用户登录”场景涉及前端提交、后端鉴权、数据库查询等跨视图协作。

核心价值

• **多视角覆盖**：每个视图解决特定问题，避免单一视角的局限性。• **利益相关者对齐**：不同角色（如开发者、运维、业务方）通过各自关注的视图达成共识。• **架构验证**：场景视图作为粘合剂，确保各视图协同工作。

通过4+1模型，复杂系统的架构设计可被全面描述，同时保持清晰性和可维护性。