**Middleware (1)**

- How do we build distributed systems with heterogeneous components?
  - Software layer between application and network
  - Network is “invisible” to application

- Middleware requirements
  - Network communication
    - Marshalling and unmarshalling data between hosts
  - Coordination
    - Synchronous or asynchronous
  - Reliability - tradeoff with performance
  - Scalability - transparency

**Middleware (2)**

- Middleware Solutions
  - Transactional - distributed transactions using two-phase commit protocol
  - Message-oriented - message passing over the network
  - Procedural - RPC
  - Object and Component - CORBA, DCOM, Java RMI

- Unanswered questions
  - How do we reconcile the relatively high overhead of middleware with the growing need for distributed objects in embedded systems?
  - How do we scale middleware up to the Internet?
  - How can we make middleware work better with low bandwidth and wireless networks?

**Software Architecture (1)**

- High level organization of the software system
  - Software components
  - Interactions between components
  - Component interfaces
  - System structure

- Abstraction to help:
  - Understand and analyze system properties
  - Identify and organize parts of the system for reuse
  - Maintain and evolve system

**Software Architecture (2)**

- Architectural Ideas
  - Product Lines - Architecture for a family of similar systems
  - Architectural styles - pipe and filter, blackboard, etc
  - Architectural Description Languages (ADL) - formalize architectural descriptions

- Challenges
  - How do we change “buy-versus-build” balance?
  - How do we reuse architectures and components?
  - Architectures for embedded/low resource systems?