**Architectural Styles for Software Architecture Systems**

- Pipes and Filters Architectural Style
- Client-Server Architectural Style
- 3-Tier Architectural Style
- Layered Architectural Style
- Repository (Blackboard) Architectural Style

**Pipes-and-Filters Architectural Style**
- Aka. Data-flow style
- Filters are independent
  - Don’t know of the others’ existence
- Reuse (1) by re-wiring filters
- Reuse (2) by replacing filters
- Concurrency (implicit)
  - E.g. concurrent branches, pipelining
- System can be analyzed
  - E.g. throughput
- Batch processing only
  - No interactive applications
- Some redundancy in filter functionality, e.g. sanity checks of data, may affect performance

**Client-Server Architecture**
- 2 separate roles: Client, Server
  - 2 layers
  - Clients connect dynamically to server
  - (Remote) Method invocation on server
  - Common access point of service

**3 - Tier Architecture**
- Presentation layer (GUI)
- Business logic layer
- Database access layer
- E.g. SAP R/3

**Layered (Onion) Architectural Style**
- No upwards calls
  - (except callbacks – anticipated)
- Strict layering
  - Calls only from Layer i to i-1
    - Layer i is server for Layer i+1
    - and client for Layer i-1
- Layer interfaces
  - Upwards exposed (provided)
  - Downwards exposed (expected)
- Examples
  - Network processing (OSI 7 layers)
  - Layered OS (Unix, Windows, …)
  - …

**Repository (Blackboard) Architecture**
- Examples
  - Linda Tuple Space
  - Jini
  - Component based compiler frameworks e.g. CoSy
- Consistency of shared data by synchronizing common access point
- Performance bottleneck