



Architectural Styles for Software Architecture Systems

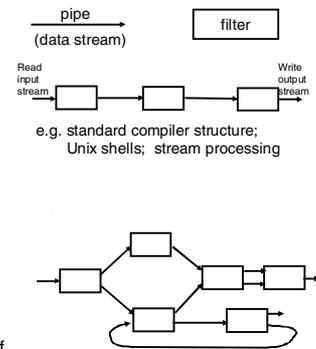
- Pipes and Filters Architectural Style
- Client-Server Architectural Style
- 3-Tier Architectural Style
- Layered Architectural Style
- Repository (Blackboard) Architectural Style

Christoph Kessler, IDA,
Linköpings universitet, 2010.

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



C. Kessler, IDA, Linköpings universitet.

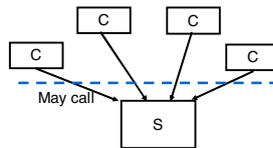
2

TDDD05 Component-Based Software, 2010

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

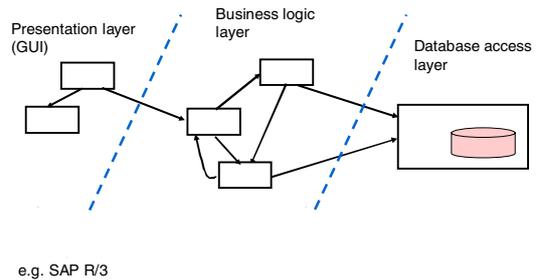


C. Kessler, IDA, Linköpings universitet.

3

TDDD05 Component-Based Software, 2010

3 - Tier Architecture



C. Kessler, IDA, Linköpings universitet.

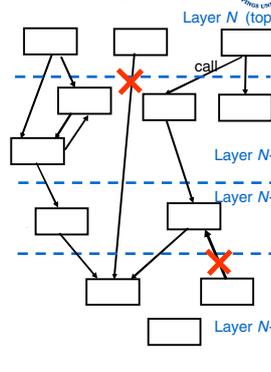
4

TDDD05 Component-Based Software, 2010

Layered (Onion) Architectural Style



- No upwards calls
 - (except callbacks – anticipated)
- Strict layering
 - Calls only from Layer i to $i-1$ for $i = N, \dots, 1$
 - 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, ...)
 - ...



C. Kessler, IDA, Linköpings universitet.

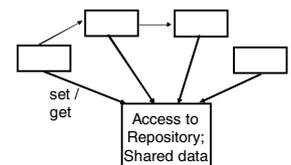
5

TDDD05 Component-Based Software, 2010

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



C. Kessler, IDA, Linköpings universitet.

6

TDDD05 Component-Based Software, 2010