Announcements
Review Github usage for Assignment 1
ARCHITECTURAL PATTERNS Cont
Architectural Patterns

- MVC
- Layers
- Microservices
- Peer-to-Peer
- Pipe and Filter
- Service-Oriented architecture
- Event-Driven Architecture
- Blackboard
The Layers architectural pattern helps to structure applications that can be decomposed into groups of subtasks in which each group of subtasks is at a particular level of abstraction.
Layers

Benefits:
- Makes reuse easier
- Makes individual layers interchangeable
- Layers interact clearly defined
Layers

Drawbacks:
Possibly less efficient than monolithic solution
Layers sometimes introduce unnecessary work
Microservices

A pattern where software systems provide minimal functional core which then can be expanded with extended functionality.
Microservices

Microservices

http://techblog.netflix.com/2013/01/announcing-ribbon-tying-netflix-mid.html
Benefits:
Strong Module Boundaries
Independent deployment
High diversity of technology
Microservices

Drawbacks:
Distribution adds complexity
Eventual Consistency must be managed
Higher operational complexity
A group of peer nodes where every peer simultaneously functions as a client and a server.
Peer-To-Peer
Peer-To-Peer

Benefits
All resource shared
More reliable with no single point of failure
Costs less to build and maintain the network
Peer-To-Peer

Drawbacks
Less Secure
Hard to Backup
No centralized authority
Pipe and Filter

Provides a structure for systems that process a stream of data. Each step (filter) processes the data and passes it on to the next step. Can be composed in any order.
Pipe and Filter

```
cat input.txt | grep "text" | sort > output.txt
```
Pipe and Filter

Scanner → Parser → Semantic Checker → Bytecode Generator
Pipe and Filter

Benefits
Flexible behavior
Filters can be reused
Ease of debugging
Pipe and Filter

Drawbacks
Error handling can be difficult
Cannot share state between filters
Event Driven Architecture

An external change in state (event) causes the application to respond to the change in events. Built with component models with no direct connection with each other.
Event Driven Architecture

Benefits:
Broadcast Communications
Asynchrony built in
Events distributed in timeliness
Event Driven Architecture

Drawbacks:
Can be a little Brittle
Different understanding of events can lead to problems
Credits

Special thanks to all the people who made and released these awesome resources for free:
✖ Presentation template by SlidesCarnival
✖ Photographs by Unsplash