Patterns for Today

- Prototype
- Publish-Subscribe
Prototype

Prototype
- factory: Factory
  #clone()
  +makeInstance(type:String)

Factory
- registry: Map
  +makeInstance(type:String)

Prototype1
  #clone()

Prototype2
  #clone()
Prototype vs. OO Languages

- No concept of “class” or type, only instances
- No inheritance or polymorphism
- Create instances by cloning a prototypical instance
- Often implemented in dynamic languages
- Redefine instances at runtime
- Focus on functionality through instances, rather than focus on class hierarchy
Publish-Subscribe
Topics and Queues

• **Topics**
  – One-to-many
  – Not persistent

• **Queues**
  – Typically point-to-point
  – May be persistent
Publish-Subscribe with Comet

- Asynchronous push of data from server to client
- Long-polling HTTP connection
- Uses Bayeux protocol

From http://alex.dojotoolkit.org/?p=545

Ashwin J Mathew  Spring 2008
School of Information, UC Berkeley
Stateful vs. Stateless Protocols

• **Stateful**
  – Client and server track connection state
  – Higher overheads, may not scale well
  – e.g. Comet tunnels over HTTP 1.1 persistent connection

• **Stateless**
  – Neither client nor server track connection state
  – Lower overheads, typically scales well
  – e.g. AJAX functions over stateless HTTP connection
Using Publish-Subscribe

- Messages and channels are more important than publishers or subscribers
- Stateful connections
- Number of publishers and subscribers are relatively small
  - But Comet proponents claim it scales well
- In a distributed system, subscribers will miss messages when they go offline
  - Subscriber recovery must be handled separately