



View Source:  
Design Patterns  
in the Wild

Thinking in  
Patterns

Someone invented or developed the code that contains the patterns, but that's largely irrelevant: It's that the same solution or style has been used and found habitable... Patterns are discovered because they are well-used, they form a vein that anyone could find.

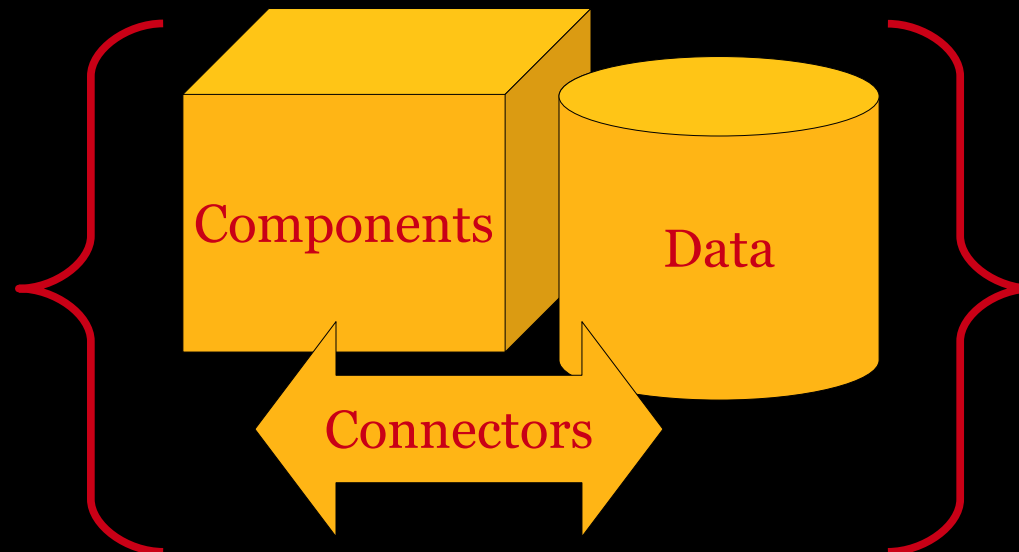
Richard P. Gabriel  
“Repetition, Generativity and Patterns”

# Software Architecture

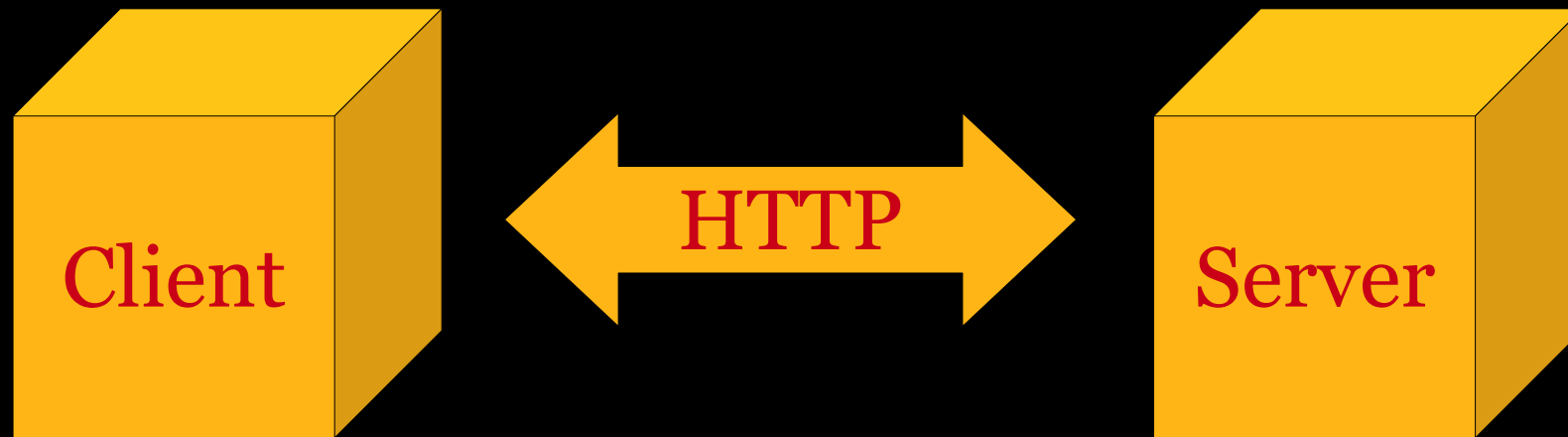
A software architecture is defined by a configuration of architectural elements - components, connectors, and data - constrained in their relationships in order to achieve a desired set of architectural properties.

Roy Fielding

“Architectural Styles and the Design of Network-based Software Architectures”

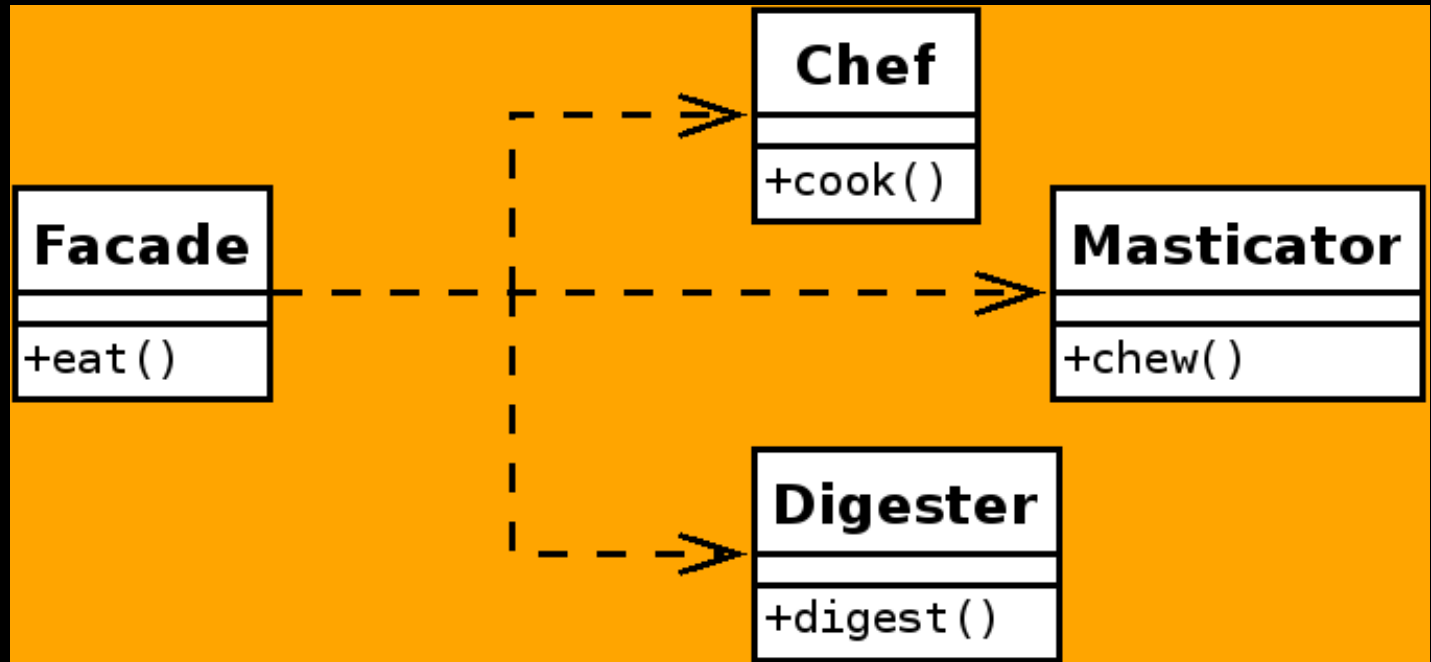


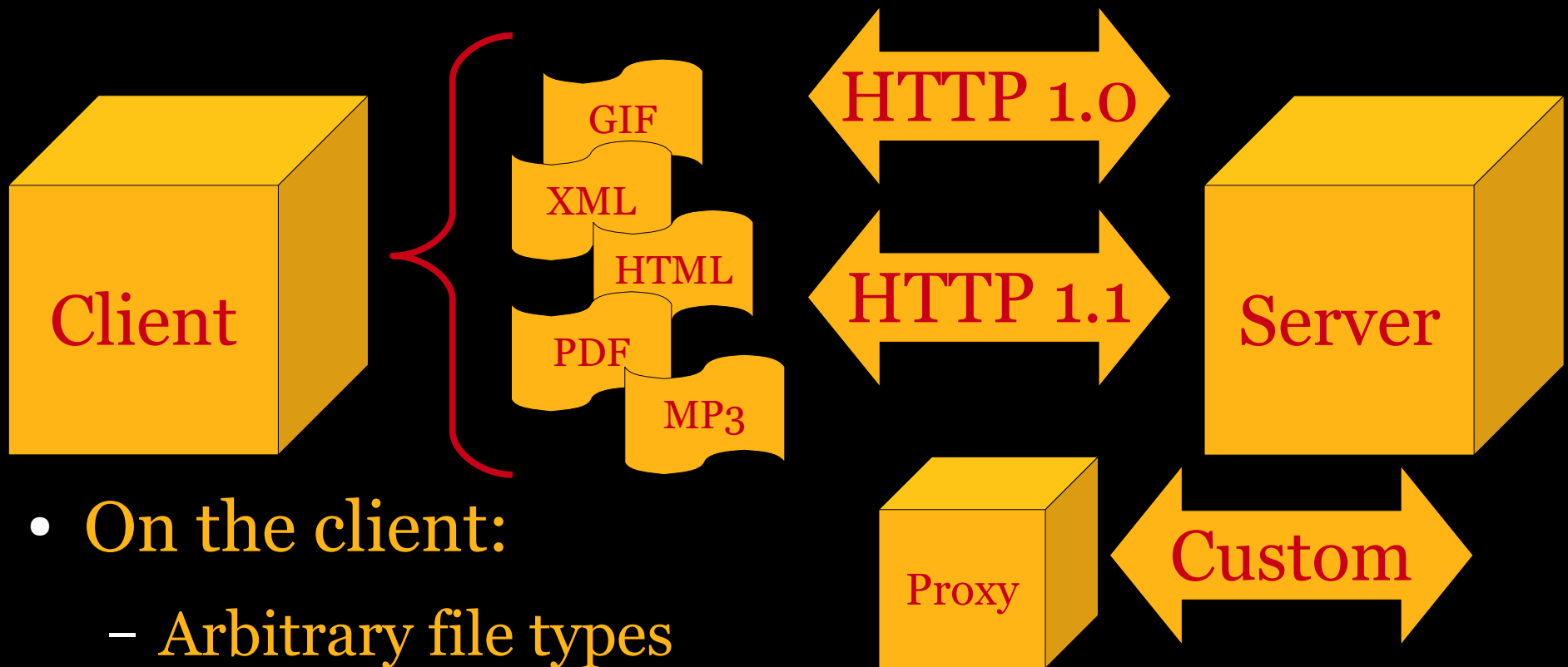




- Separation of concerns
- Encapsulation of functionality
- Standardized interface

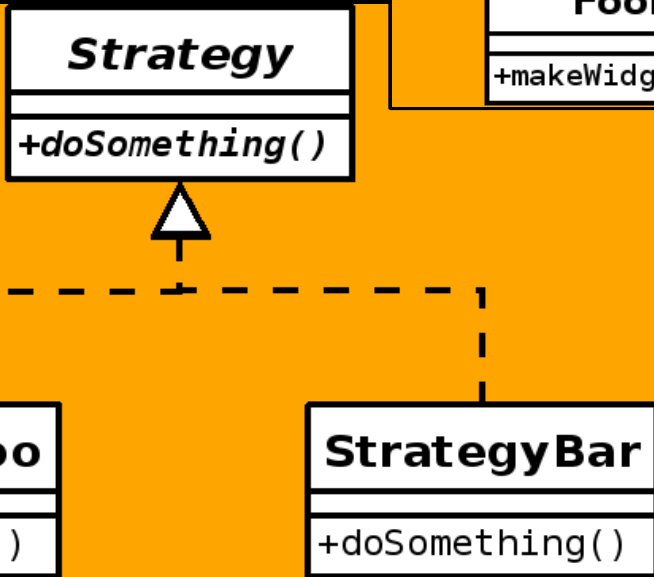
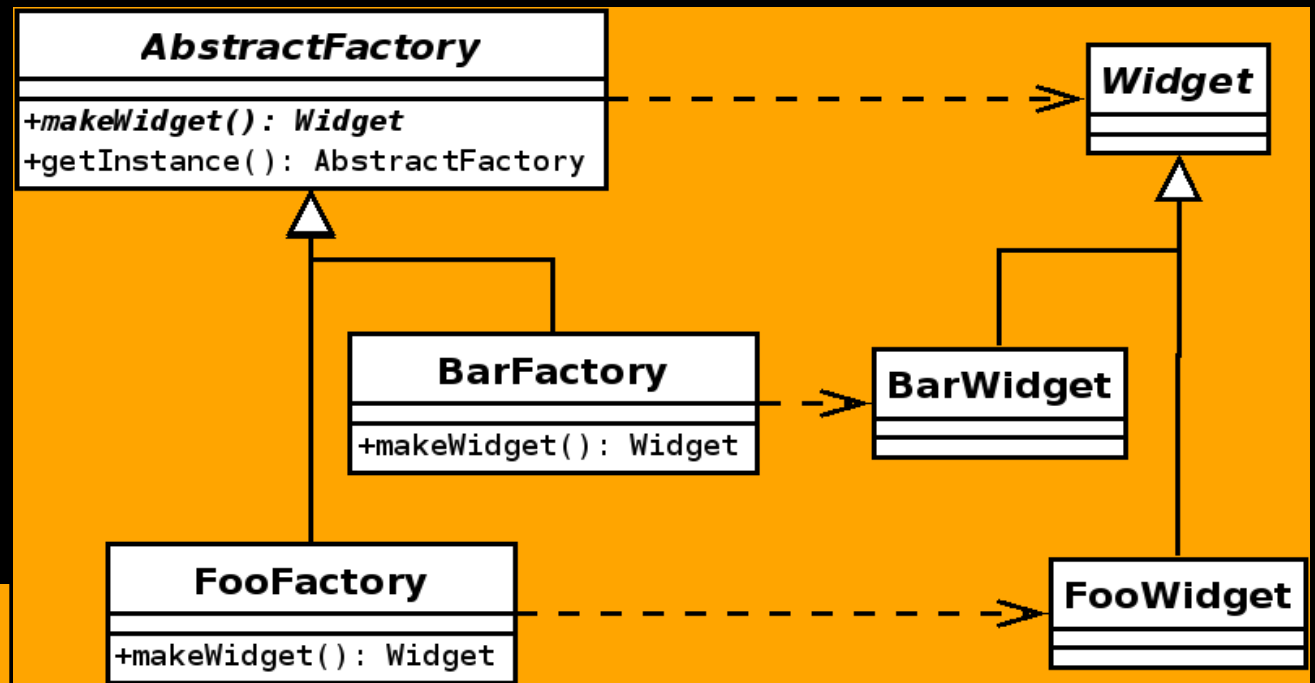
# Facade





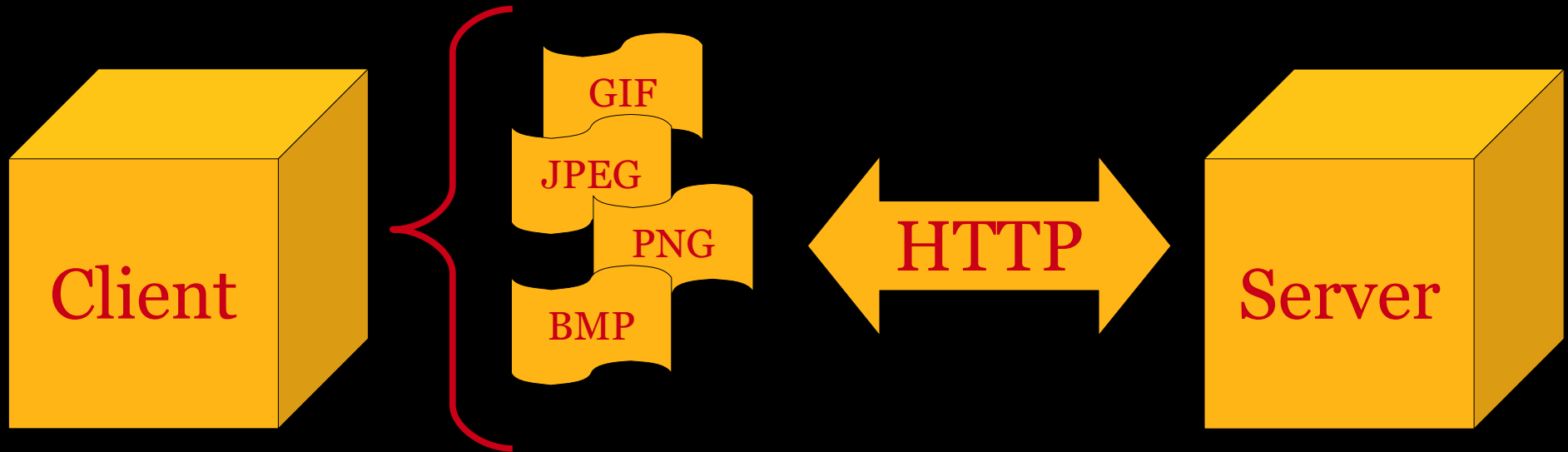
- On the client:
  - Arbitrary file types
  - Unique handler for each file type
- On the server:
  - Arbitrary protocols
  - Unique handler for each protocol

# Abstract Factory



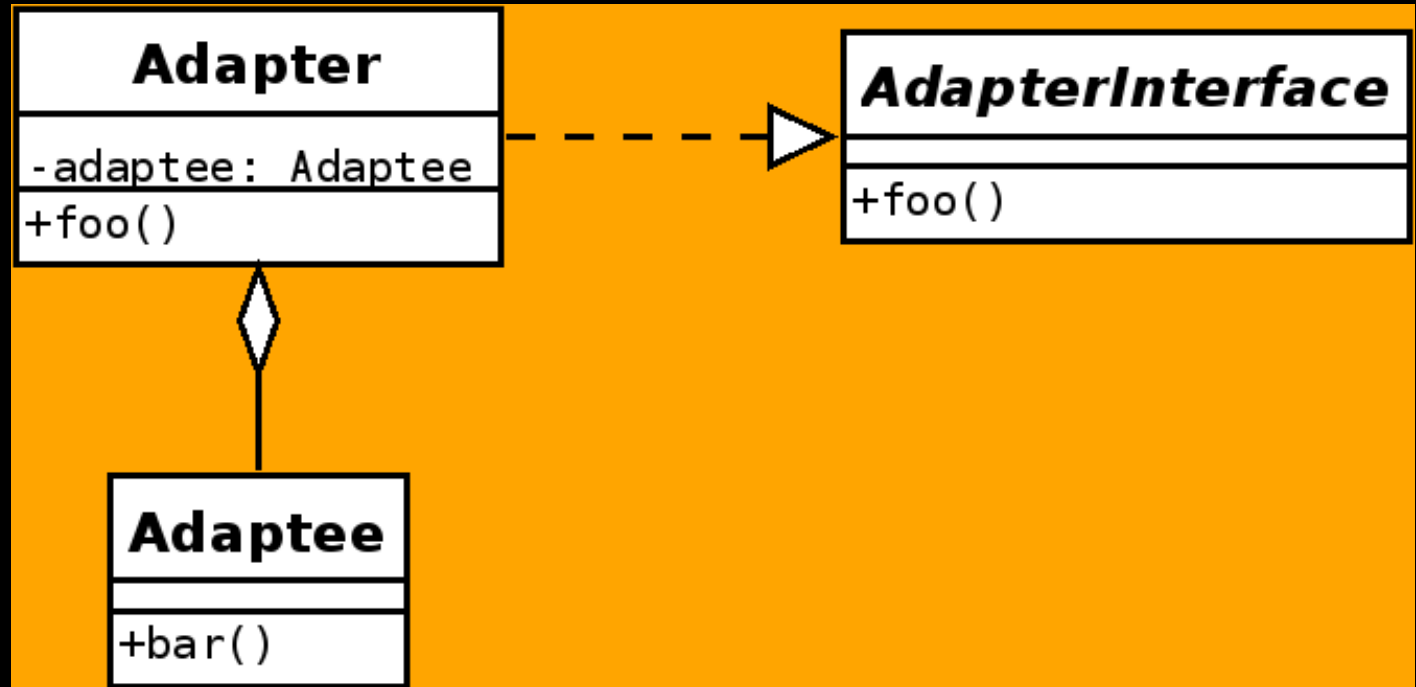
Strategy  
... or Dependency Injection

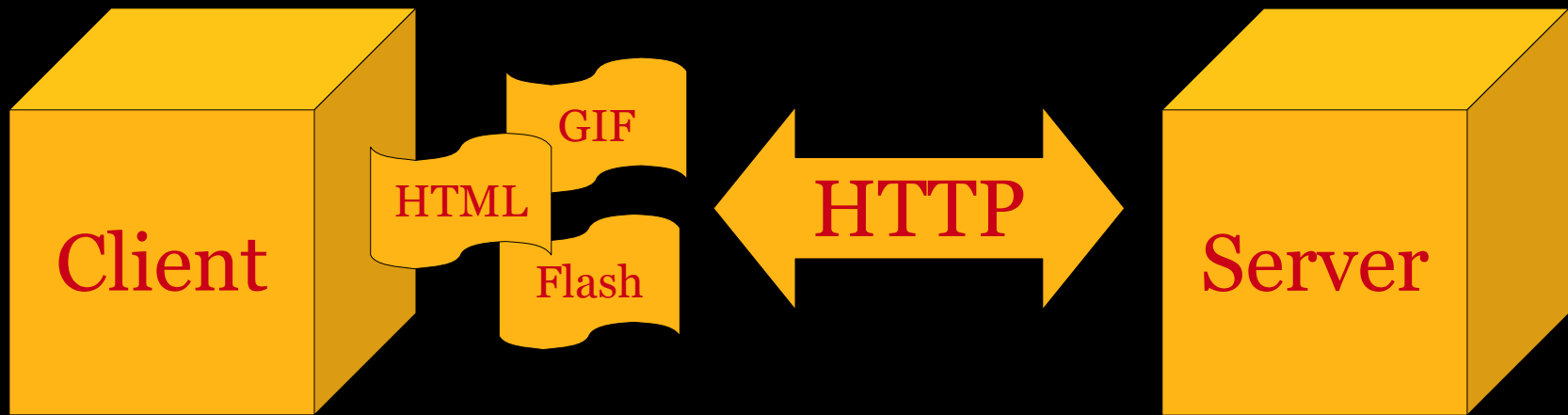




- Handle many types of images
- Render each in a similar fashion

# Adapter

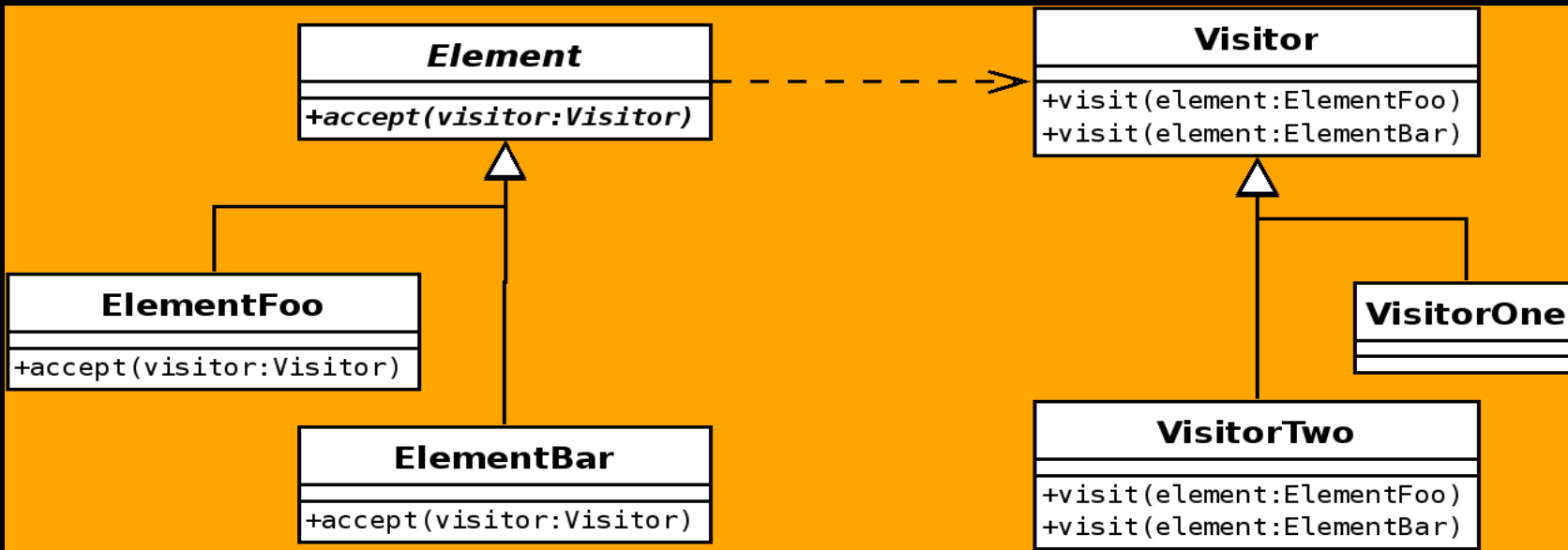




- Render different formats in a single page

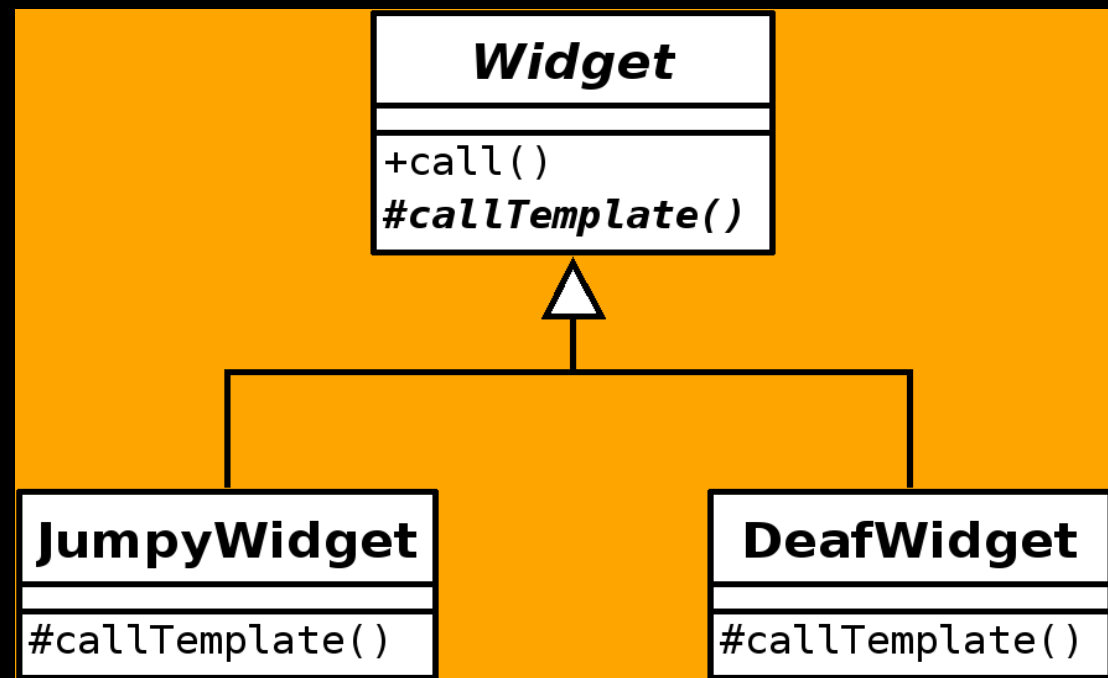
... `<img src='/images/me.gif' />` ...

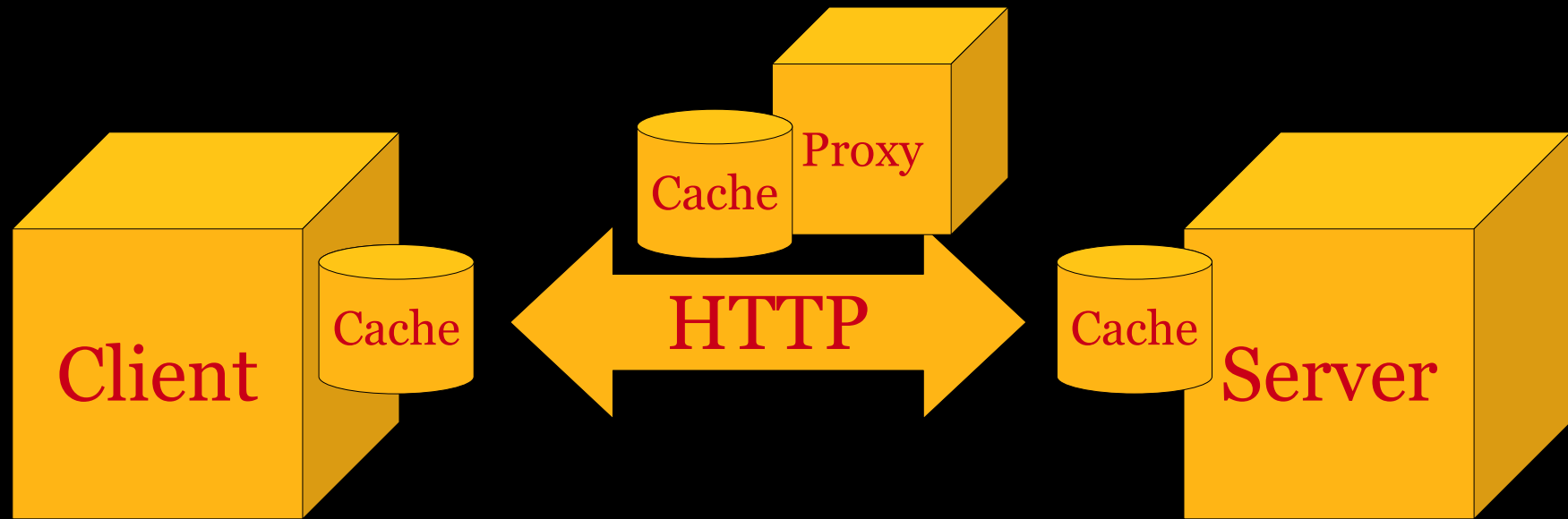
`<embed type='application/x-shockwave-flash'  
src='/myapp.swf' />` ...



## Visitor

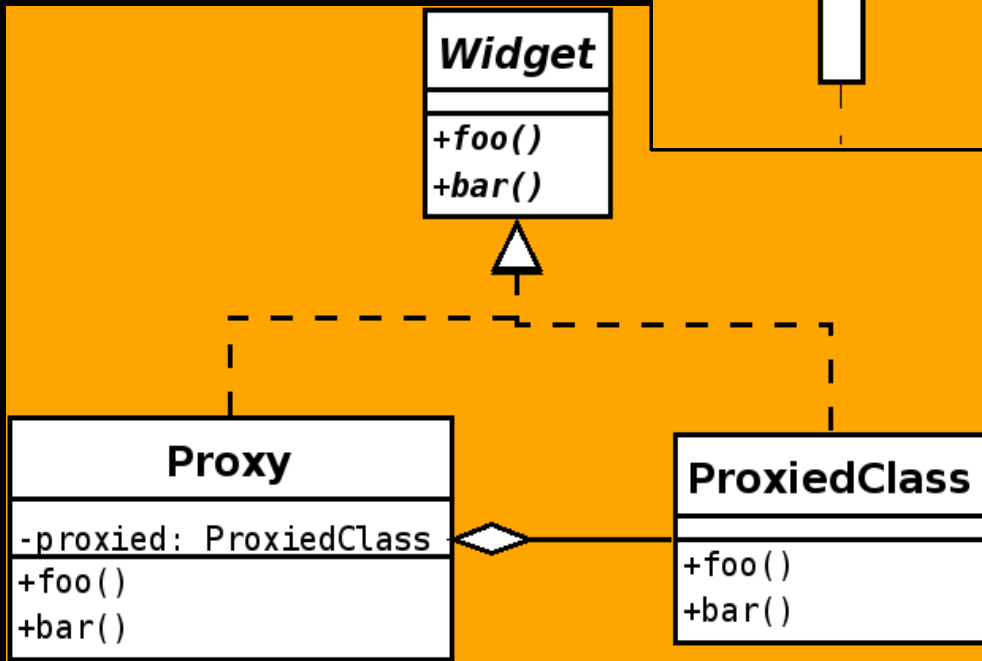
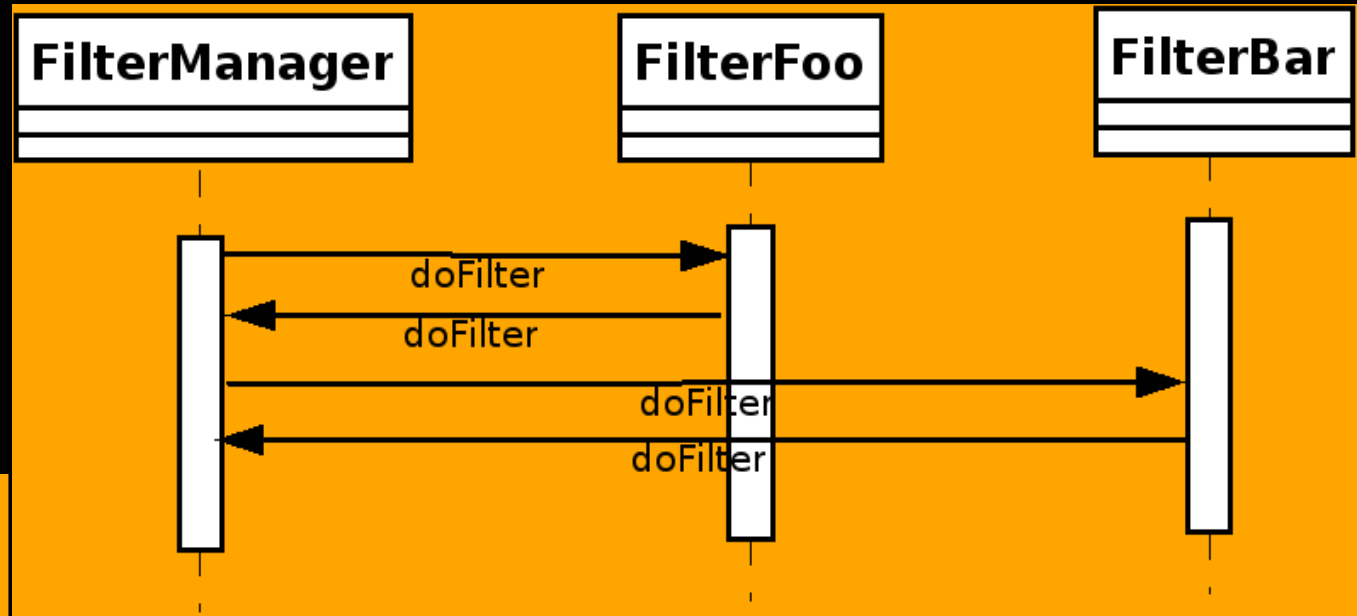
## Template Method





- Caching built into HTTP
- Maintain standard interface
- Cache is an *optional* feature

# Intercepting Filter



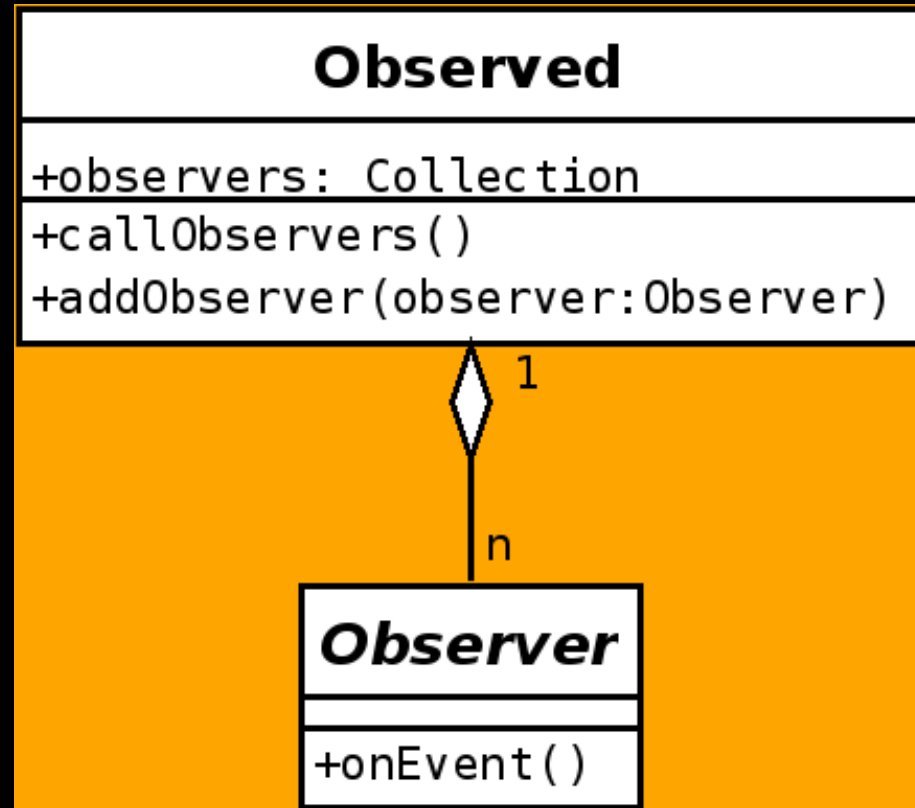
## Proxy

+ Prevalence for storage

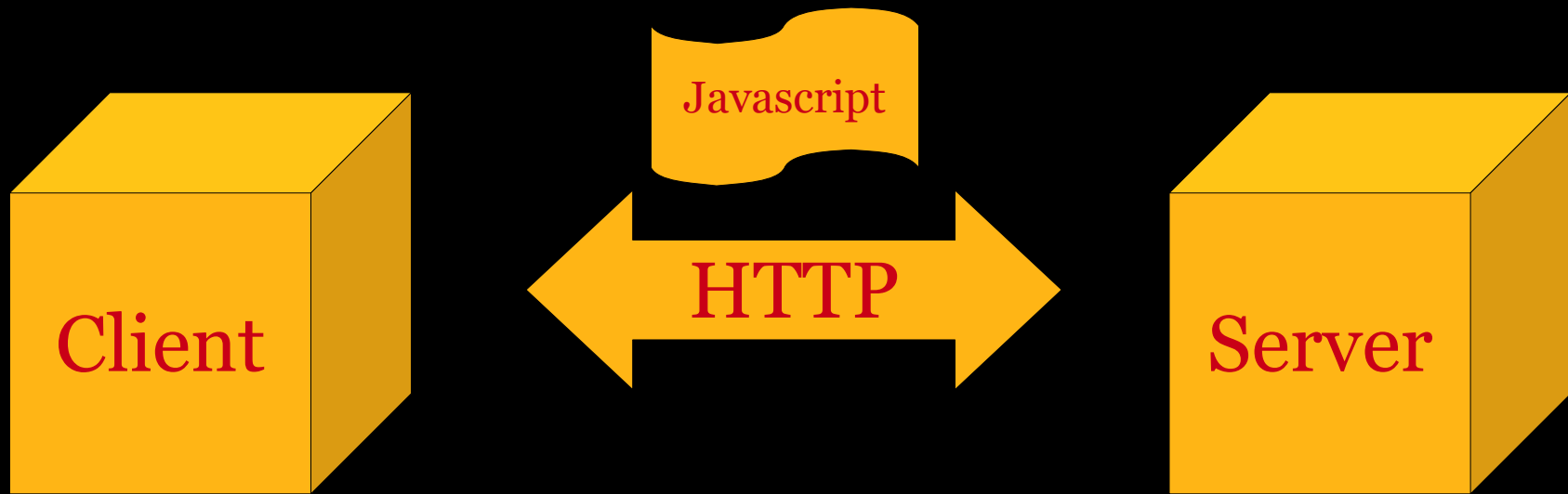


- Execute logic when a HTML page loads

# Observer

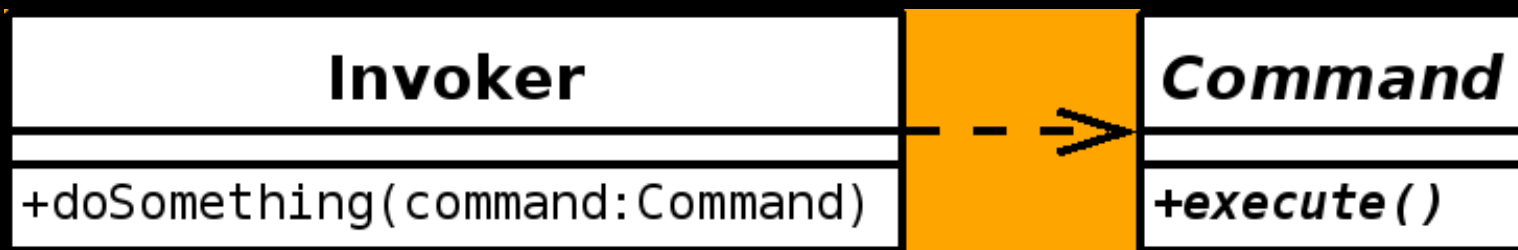
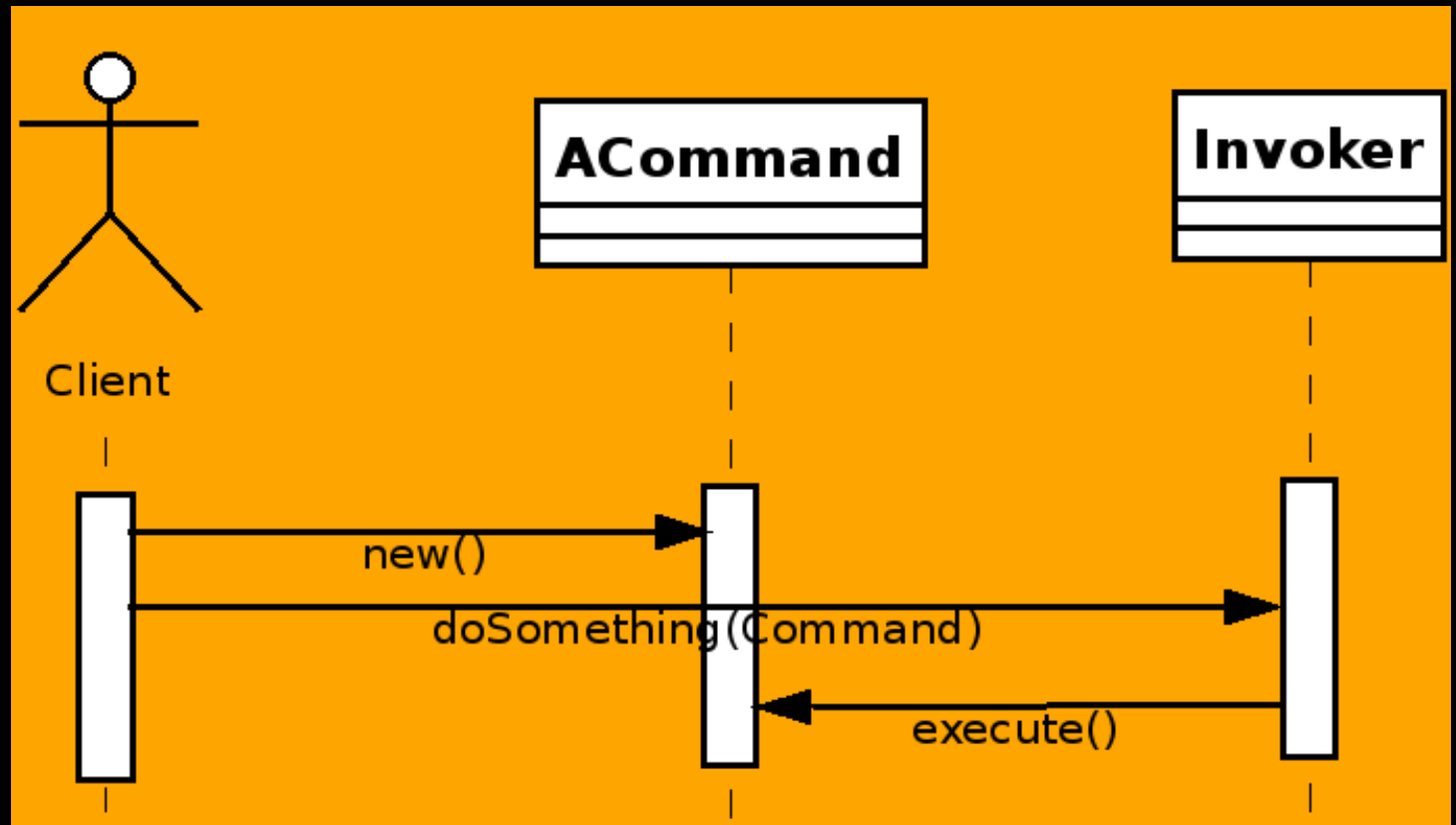


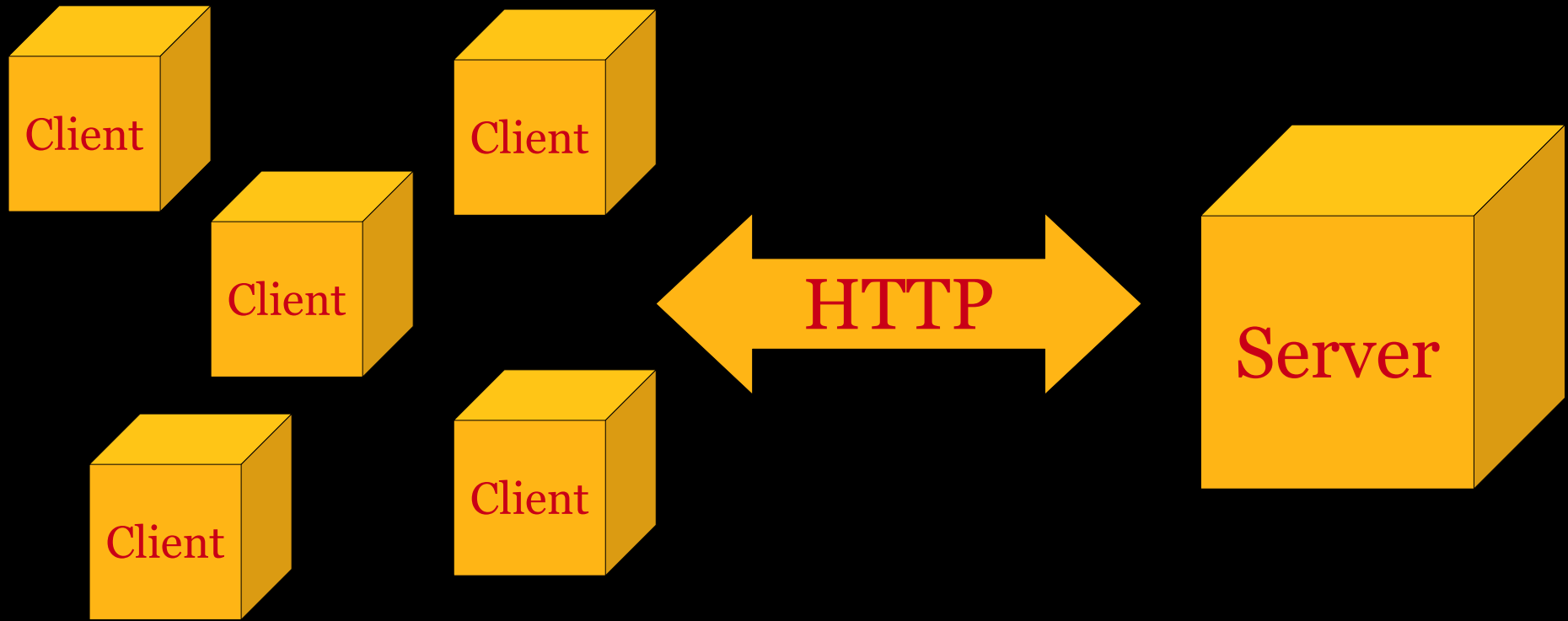




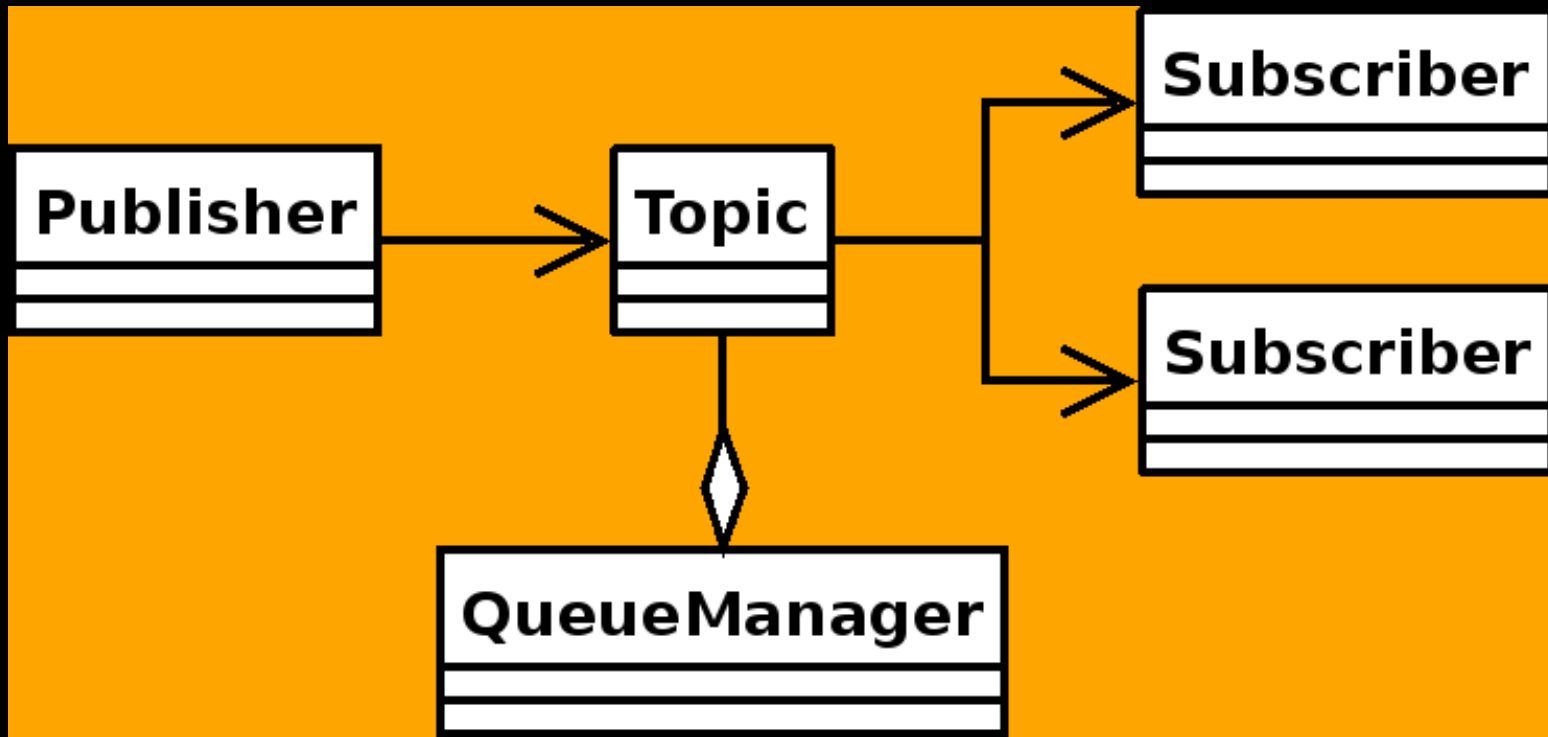
- Javascript generated on server
- Sent to client for execution

# Command





- Broadcast to many interested clients

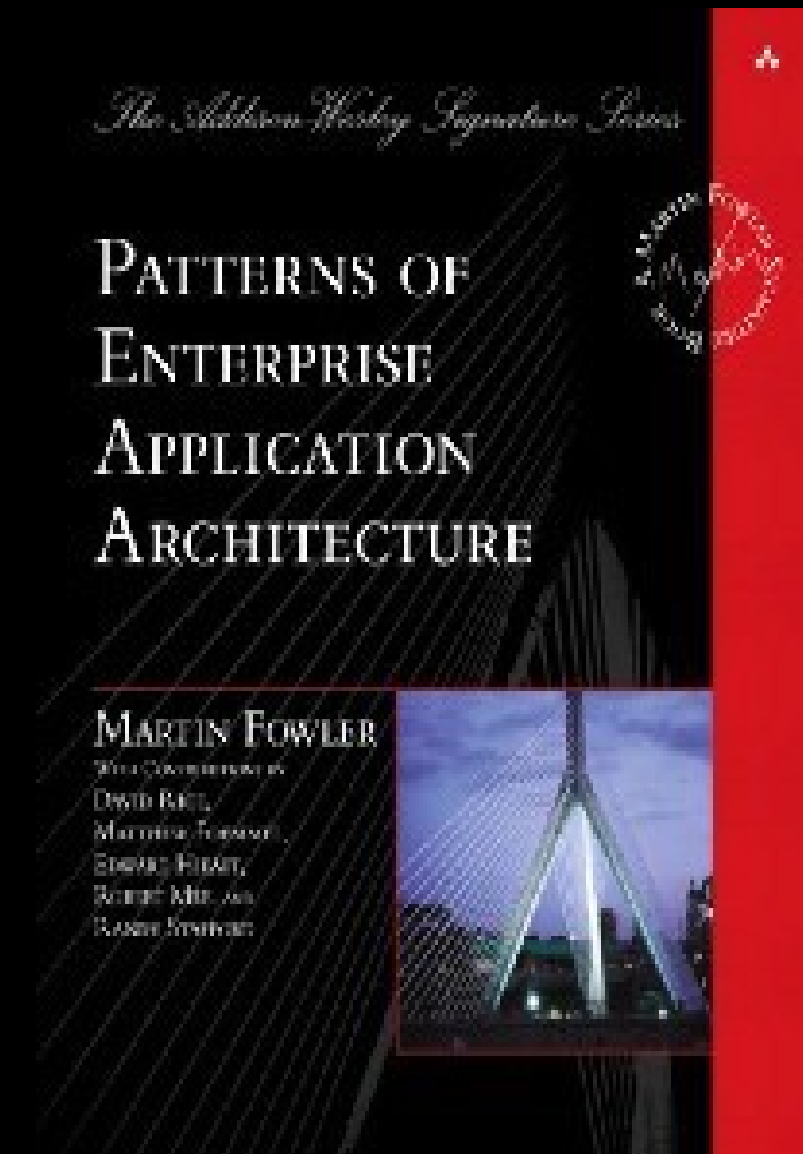
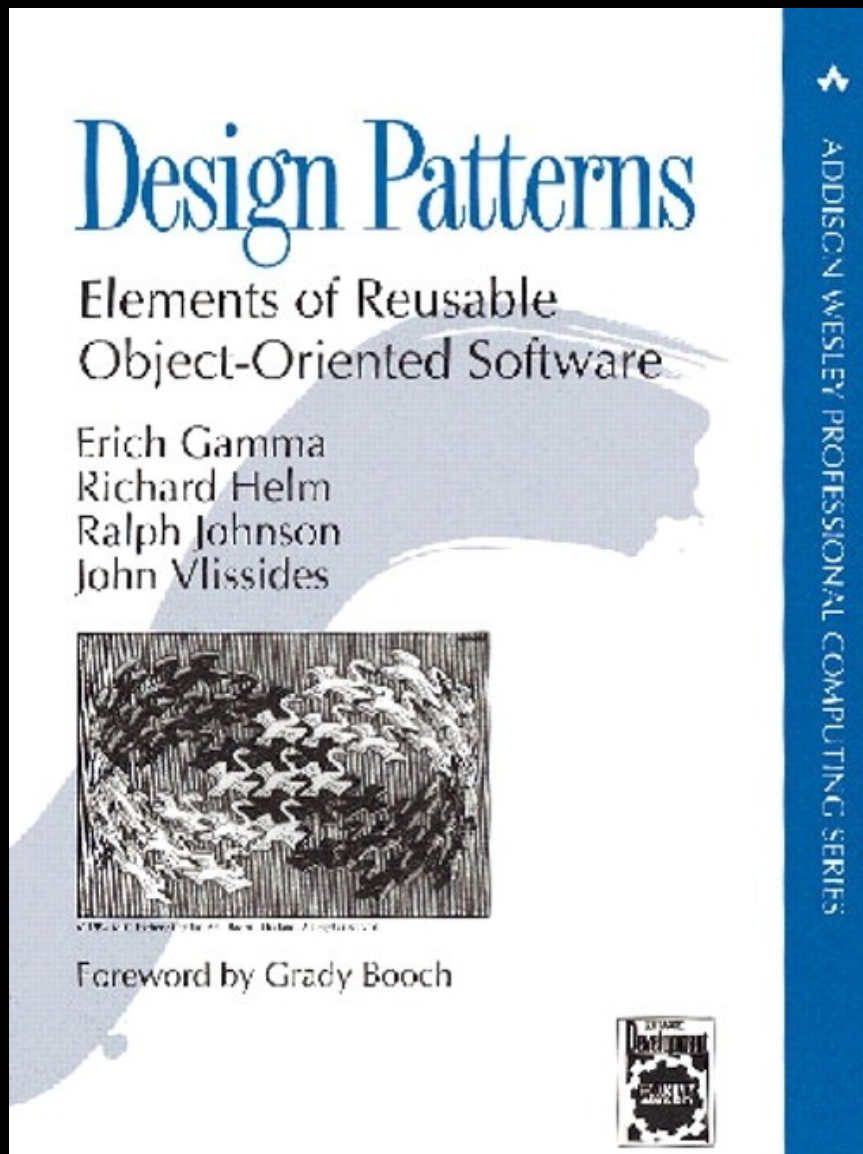


## Publish/Subscribe

# Other Patterns

- Mediator
- Prototype
- Map/Reduce
- Model-View-Controller
- Data Access Object, Active Record

# ... and more



# No Silver Bullet



- Essential Complexity  
“difficulties inherent in the nature of software”
- Accidental Complexity  
“difficulties that attend to its production ... but are not inherent”

Between order and chaos, interesting and unexpected combinations come about and last long enough to have repercussions. Trends can be observed. Patterns emerge.

Richard P. Gabriel and Ron Goldman  
“Mob Software: The Erotic Life of Code”



Whereas the difference between poor conceptual designs and good ones may lie in the soundness of design method, the difference between good designs and great ones surely does not. Great designs come from great designers. Software construction is a creative process.

Fred Brooks

“No Silver Bullet: Essence and Accidents of Software Engineering”