

Patterns You Already Use

Chain of Responsibility and Facade

Jeremy Clark
www.jeremybytes.com
jeremy@jeremybytes.com



pluralsight 
hardcore developer training

Chain of Responsibility Pattern

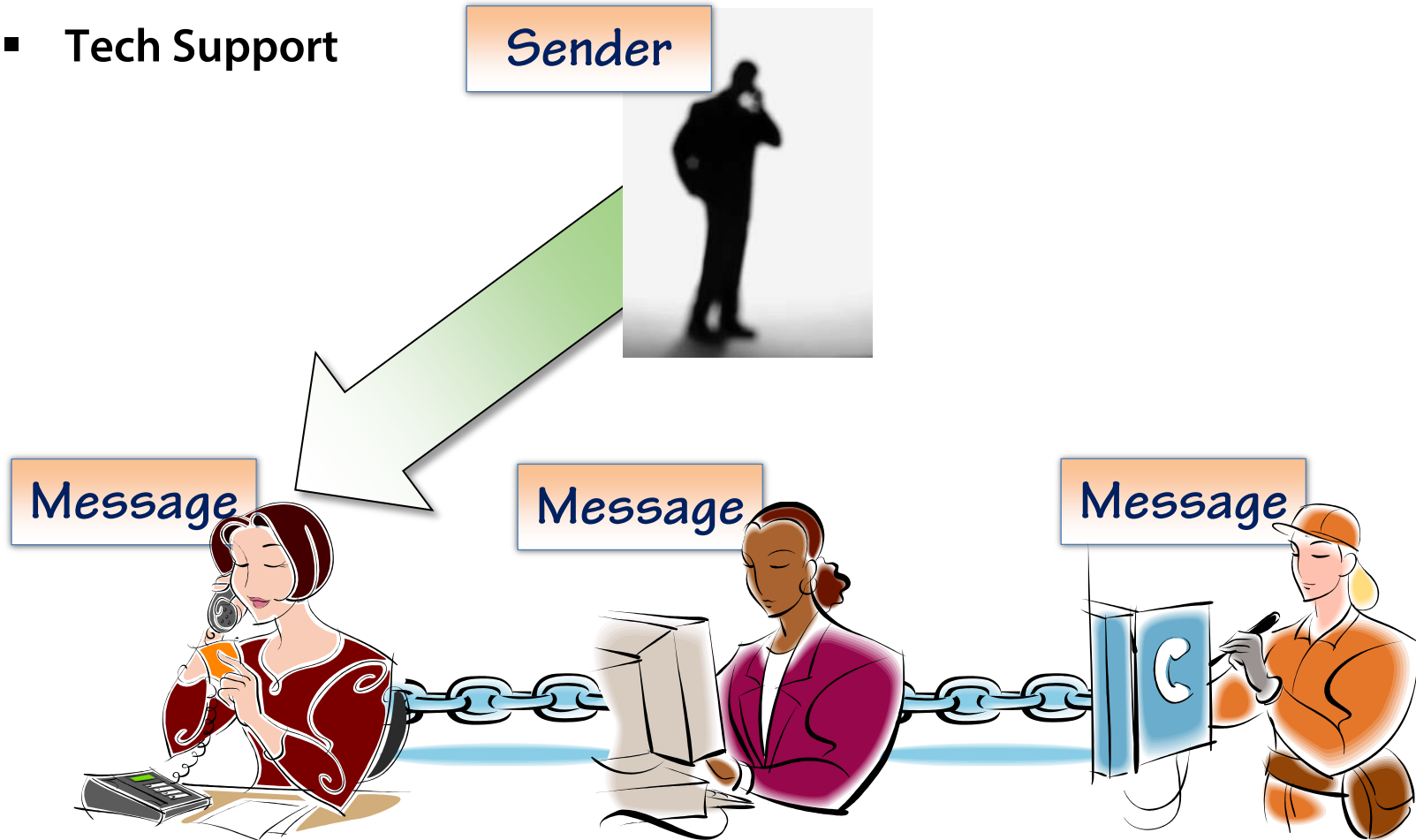


Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it.

Gamma, et al, *Design Patterns*. Addison-Wesley, 1994.

Real World Chain of Responsibility

- Tech Support



Chain of Responsibility Pattern

Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. Chain the receiving objects and pass the request along the chain until an object handles it.



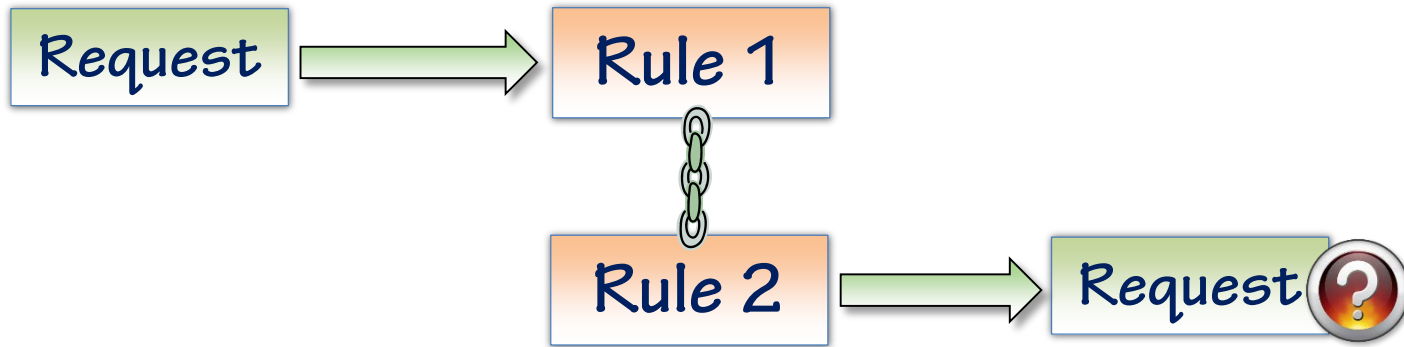
Chain of Responsibility Consequences

- **If none of the receivers handle the message, then the message will fall off the end of the chain.**

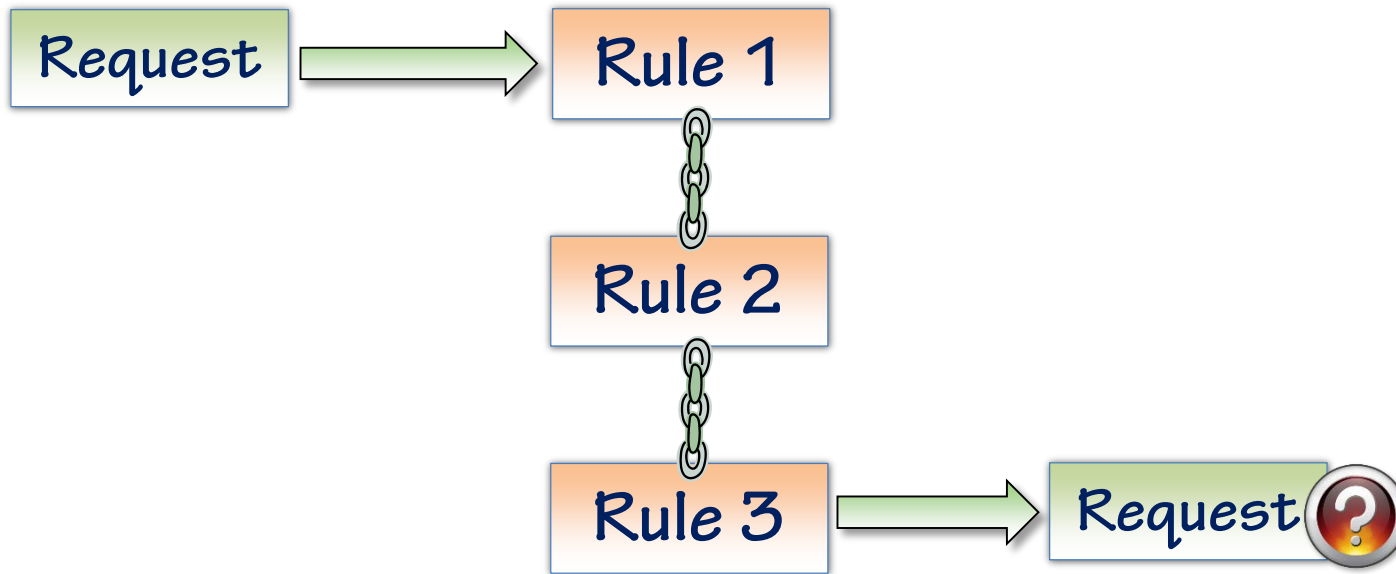
Approval Engine



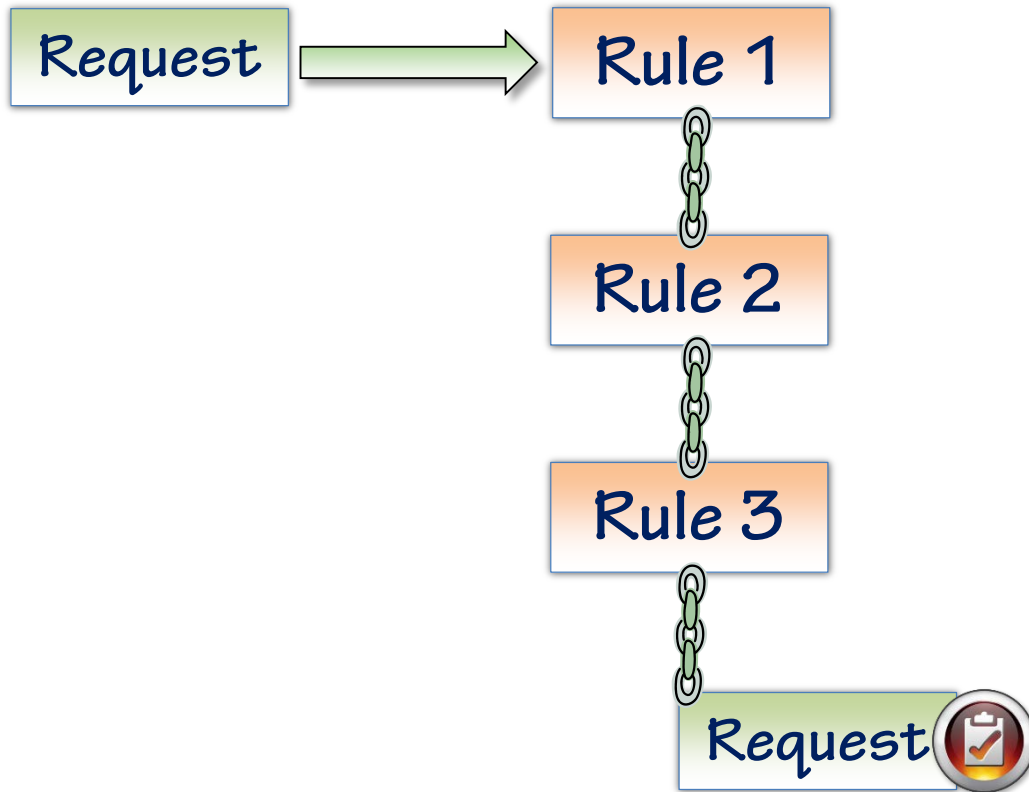
Approval Engine



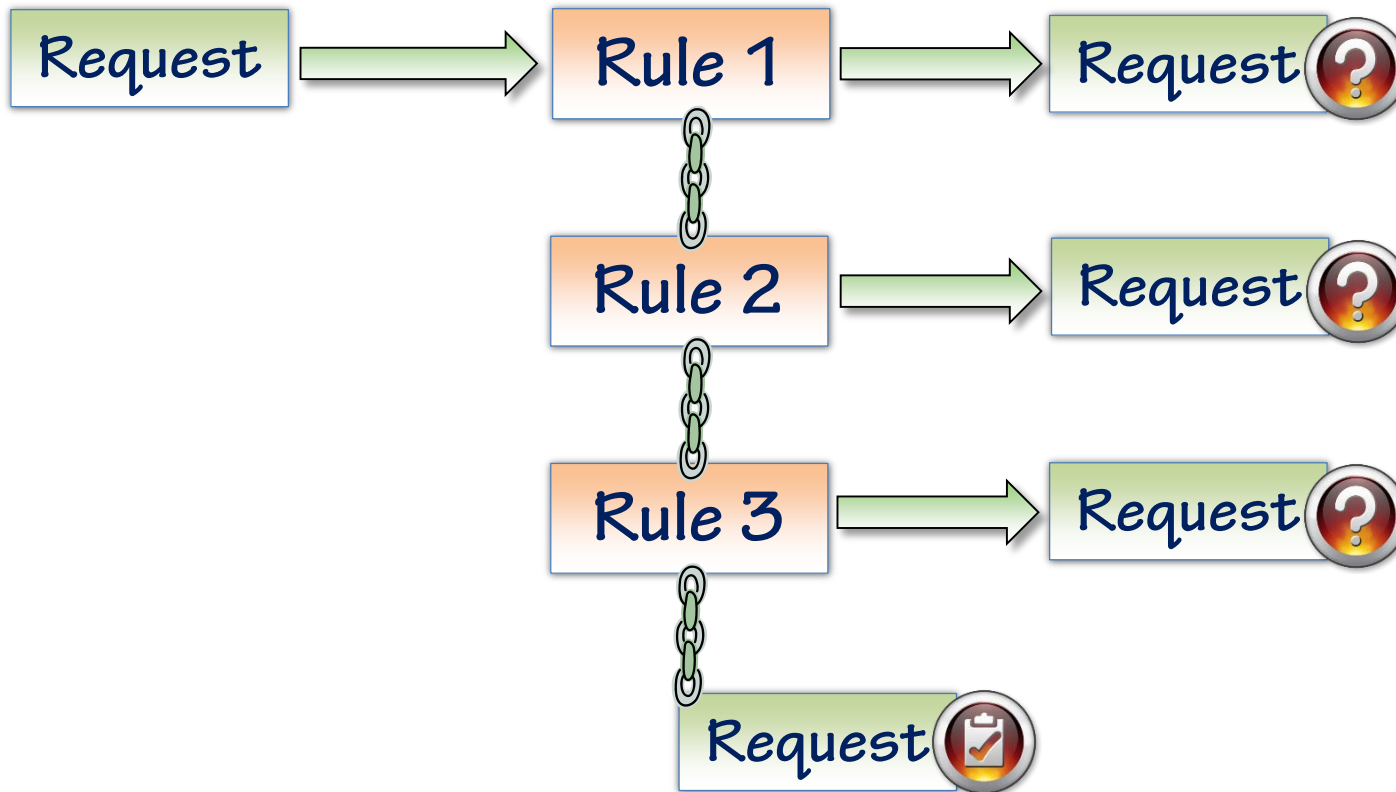
Approval Engine



Approval Engine



Approval Engine



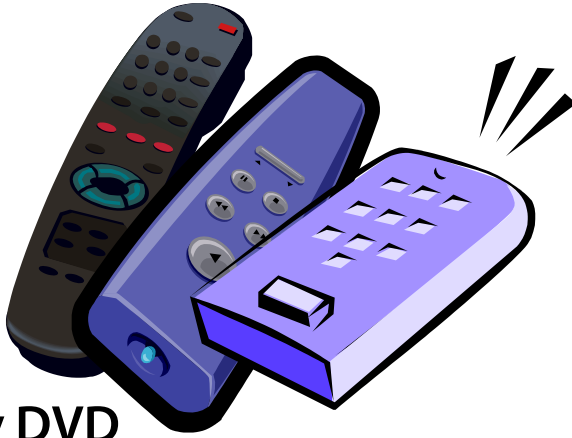
Facade Pattern



Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.

Gamma, et al, *Design Patterns*. Addison-Wesley, 1994.

Real World Facade



■ Play DVD

- Turn on TV
- Set TV to Component Input
- Turn on Stereo
- Set Stereo to Aux Input
- Turn on DVD Player
- "Play" on DVD Player



Facade Translation

Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.

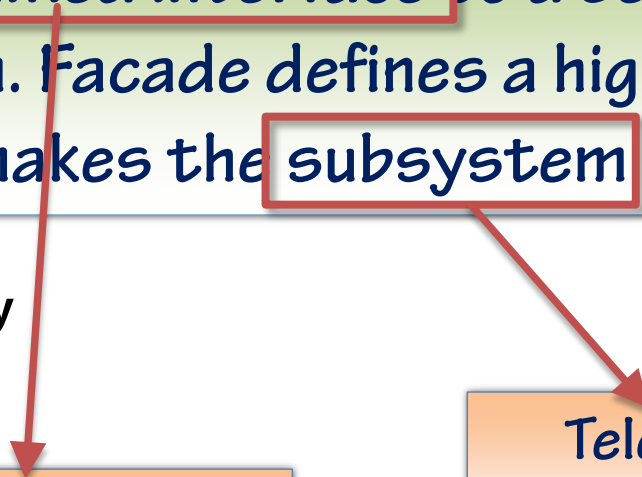
Hiding Complexity

Universal Remote
(Interface)

Television
(Subsystem)

Stereo
(Subsystem)

DVD Player
(Subsystem)



BackgroundWorker Component

Methods

RunWorkerAsync
ReportProgress
CancelAsync

Properties

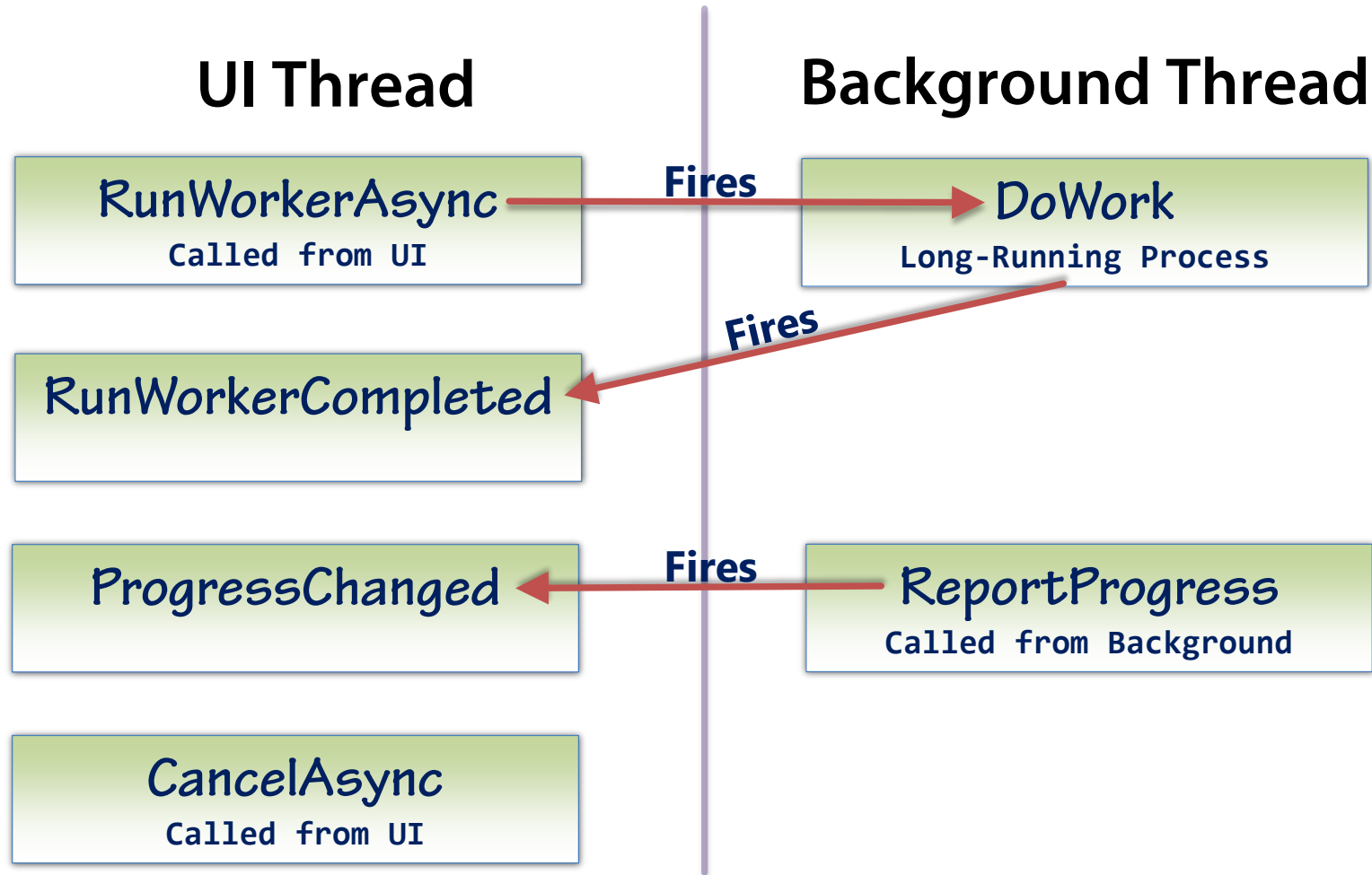
WorkerReportsProgress
WorkerSupportsCancellation
CancellationPending
IsBusy

Events

DoWork
RunWorkerCompleted
ProgressChanged

That's All There
Is To It!

UI and Background Threads



Facade Consequences

- If we want to expose more functionality from the subsystem, we may need to update or create a new facade.
- Since the facade hides complexity, the entire API is often not available.

Summary

■ Chain of Responsibility

- Message Handling
- Everyday usage: try/catch blocks
- Approval Engine



■ Facade

- Hiding Complexity
- Everyday usage: foreach
- BackgroundWorker and other components



■ Next Up: Useful Patterns Factory Method, Decorator, and Adapter

