

Asynchronous Messaging Patterns with WCF

Rob Daigneau
Chief Architect, SynXis

Pre-requisites for this presentation:

- 1) Intermediate to Advanced C#
- 2) Some Exposure to WCF

Level: Intermediate

Overview

- The Problem with Synchronous Calls
- Asynchronous Patterns Part 1
 - The Polling Client
 - Proxy Callback
 - Service Initiated Callback
- One-Way Messaging and Event Driven Architecture
- Asynchronous Patterns Part 2
 - Split Request/Response
 - Notification Relay

About Me

- Chief Architect
 - A Sabre Holding Company



- Director of Architecture, Monster.com



- Addison Wesley author
 - Currently working on *SOA Design Patterns* book
- Host of *www.DesignPatternsFor.Net*

The Problem with Synchronous Calls

The Request/Response Message Exchange Pattern

Request/Response is usually assumed to be synchronous



What Problems do we have with Synchronous Calls?

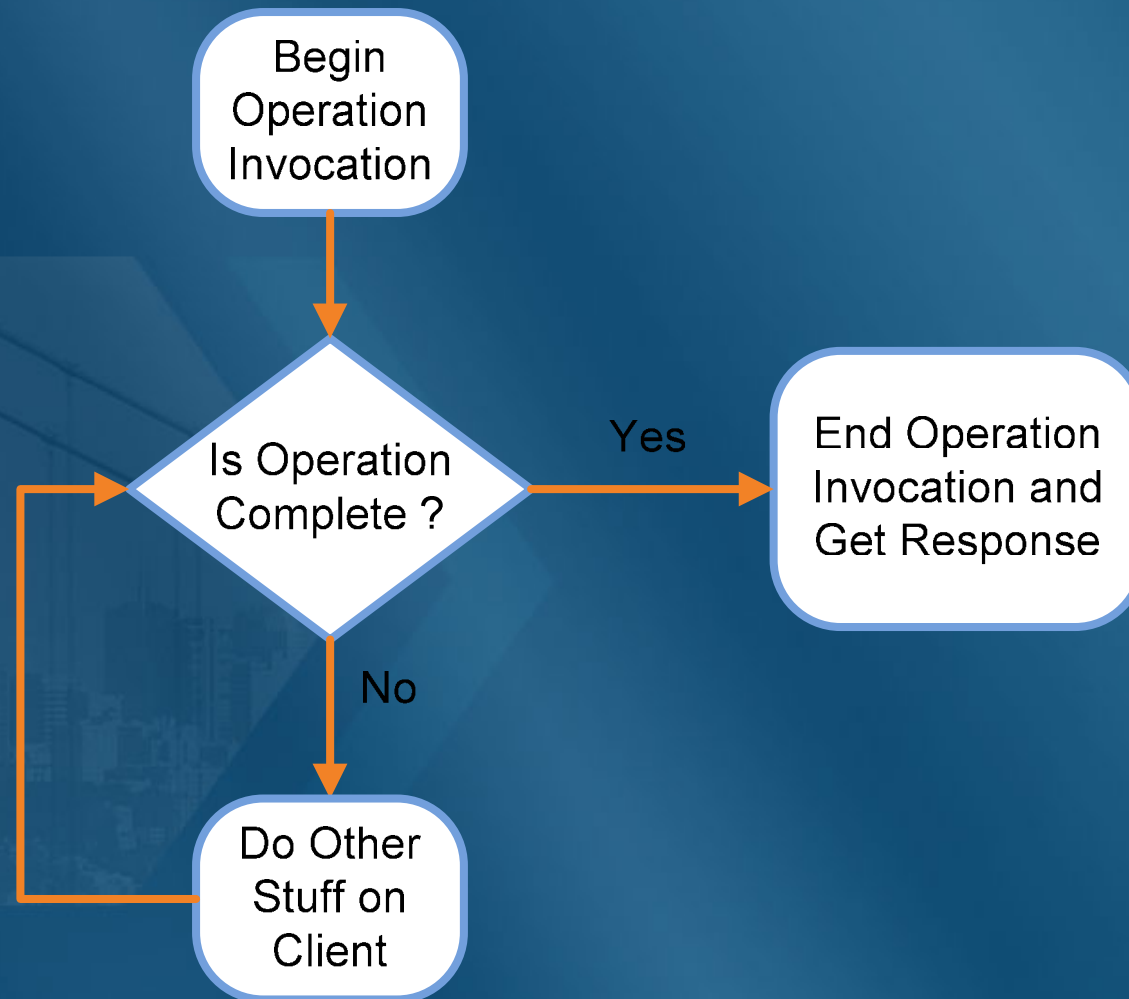
1. Client is blocked; this is a problem if ...
 - Client could do other work
 - Server logic takes a long time to complete
2. Systems behind the service façade must be available
3. Services usually aren't designed to efficiently handle concurrent incoming requests
 - Service could be bombarded and overwhelmed with requests
 - Messages are processed in a "first-come, first-served" style

Asynchronous Patterns Part 1

Variations on Request/Response

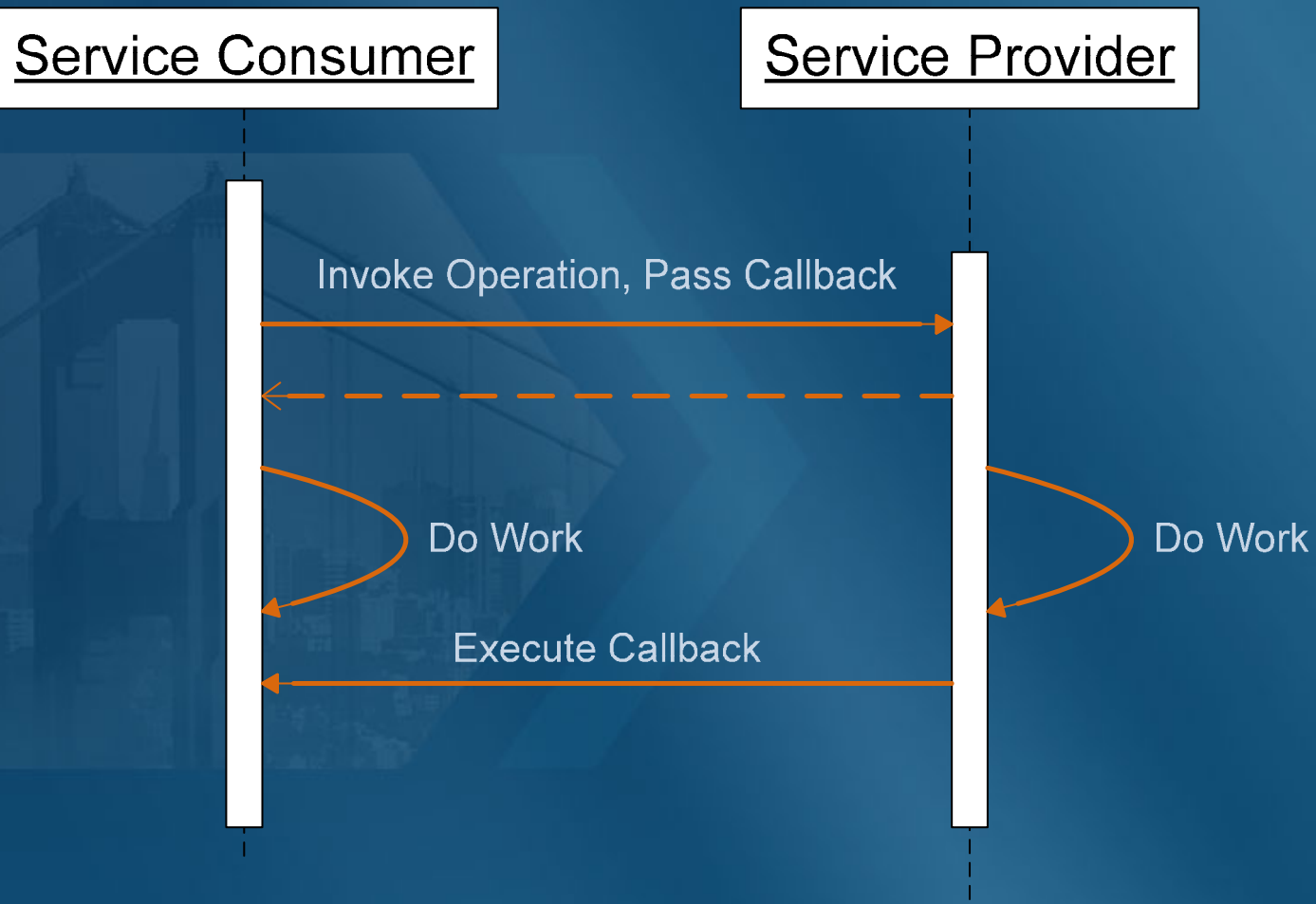
The Polling Client

Addresses
problem of
client being
blocked



The Callback Client

Also Addresses problem of client being blocked



One-Way Messaging and Event Driven Architecture

Addressing the problems of
system availability and
throttling of inbound requests

SOAP and Asynchronous Messaging

- SOAP provides the foundation for creating one-way message exchanges
 - aka In-only style messages

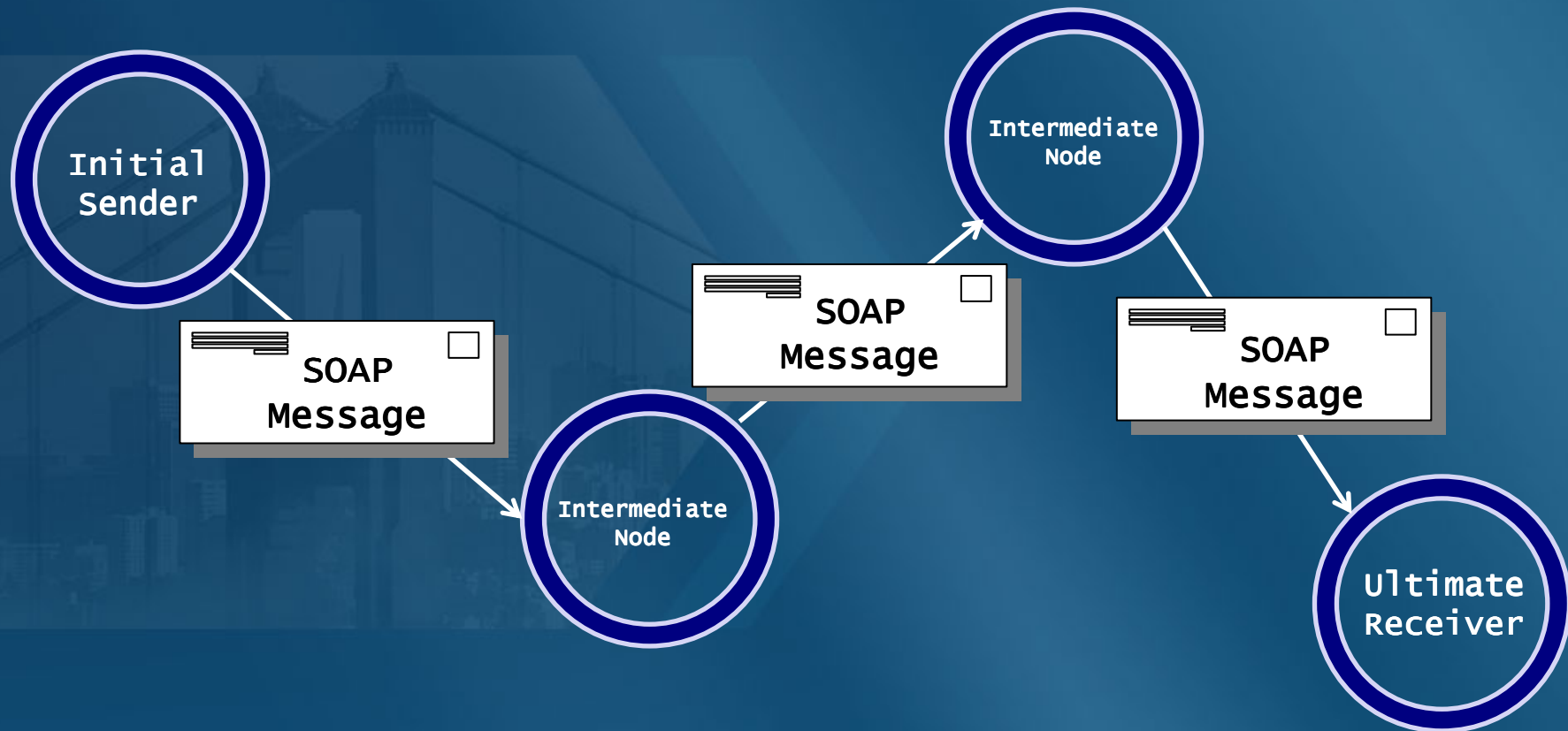
– “Fire and Forget”



- Keep in mind
 - Client may still be blocked until request can be queued (just like synchronous calls)
 - Server can't return faults to client
 - Client only knows that the message was sent
 - **Doesn't know if it was received**

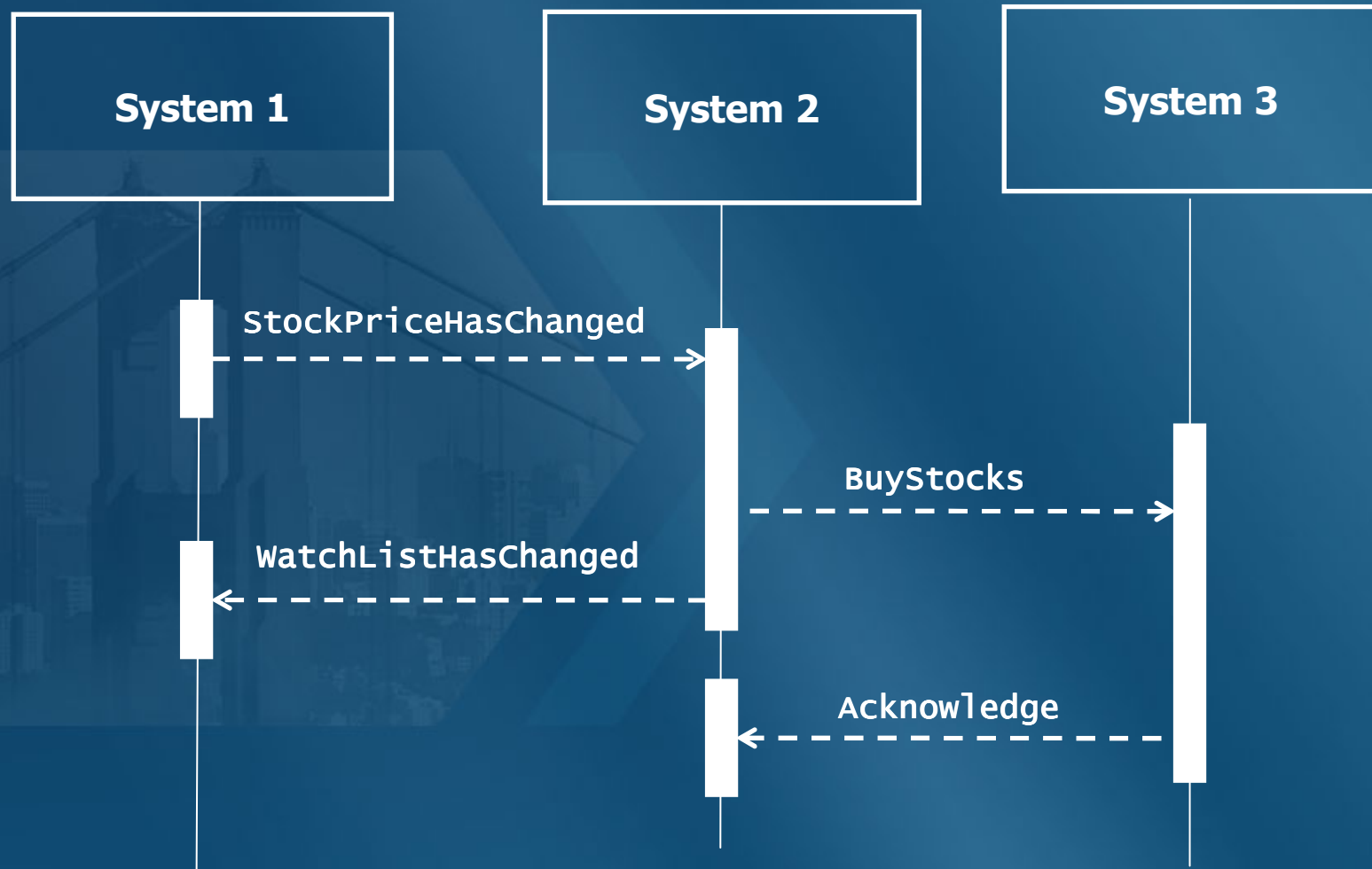
Building Up from One-Way Message Exchanges

More complex conversations are possible when SOAP messages are routed across intermediary nodes



Event Driven Architecture

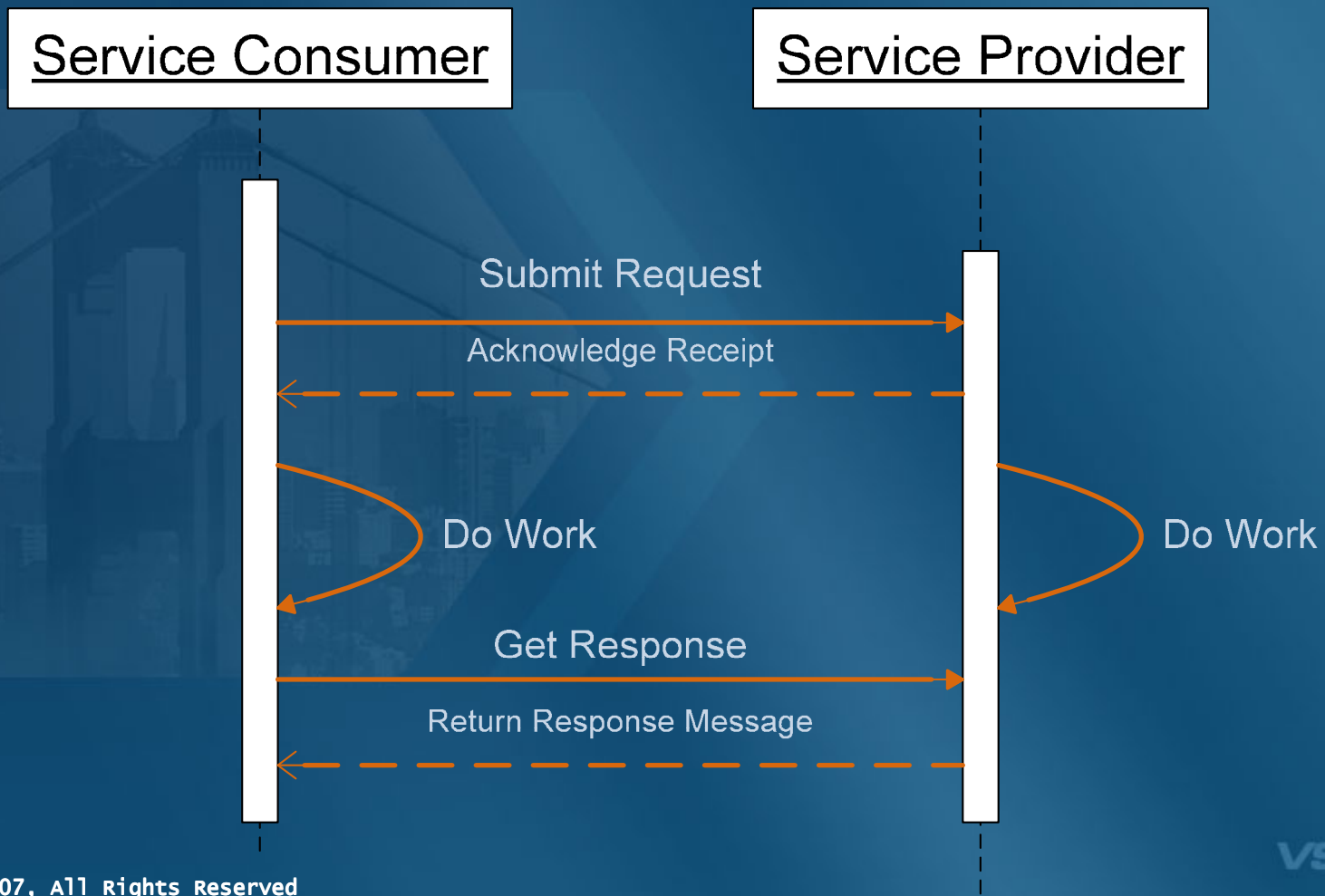
One-way message exchanges are the foundation of Event Driven Architectures



Asynchronous Patterns Part 2

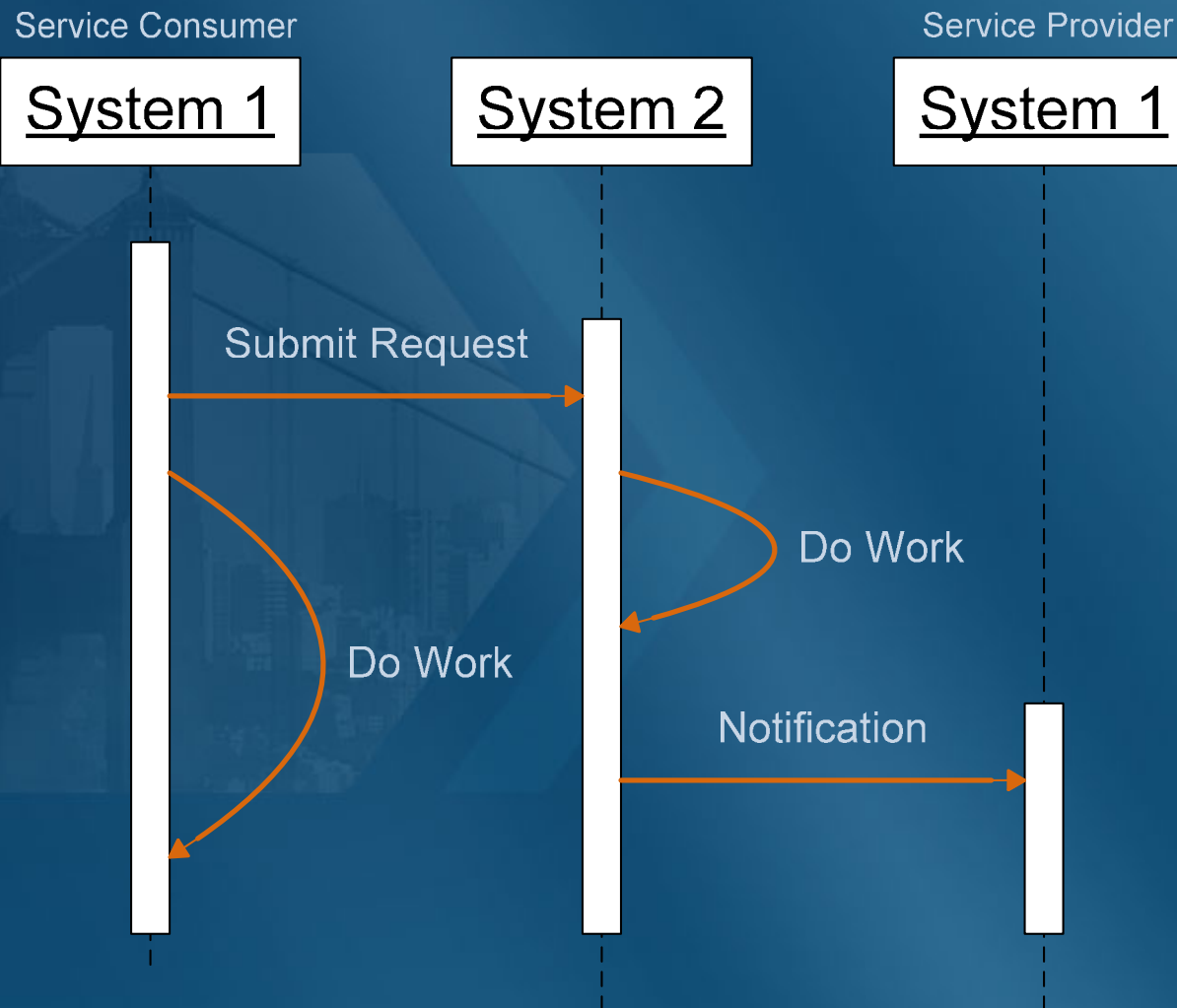
The Split Request/Response Pattern

- The consumer may send a request message in one exchange and retrieve the response in a subsequent exchange
- Addresses all problems associated with synchronous calls



Notification Relay

- Sometimes responses cannot be sent back to the same "call context"
- The original client can direct the service to RELAY a NOTIFICATION on to another address



Conclusion

- The Problem with Synchronous Calls
- Asynchronous Patterns Part 1
 - The Polling Client
 - The Callback Client
- One-Way Messaging and Event Driven Architecture
- Asynchronous Patterns Part 2
 - Split Request/Response
 - Notification Relay