Publish-subscribe Messaging

Alan Smith
ACTIVE SOLUTION

@alansmith   www.cloudcasts.net

Overview

Publish-subscribe Messaging

Subscription Rules

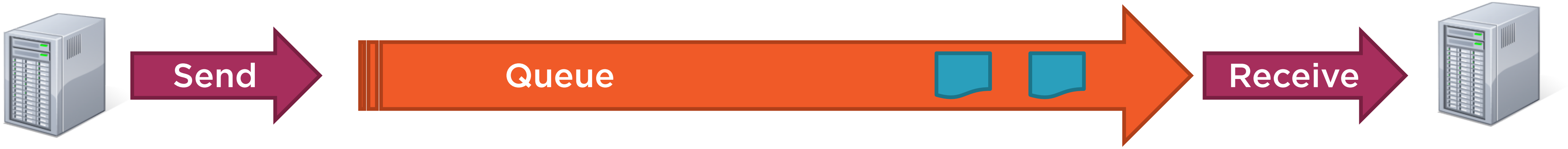
Demo: Subscription Rules

Implementing a Wire Tap

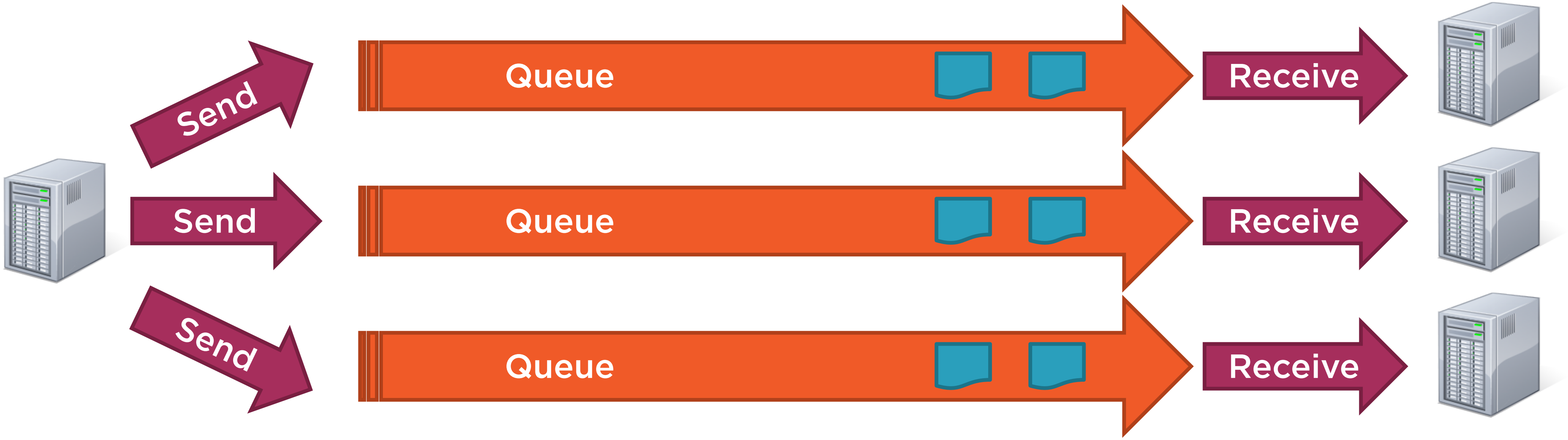
Demo: Implementing a Wire Tap

Publish-subscribe Messaging

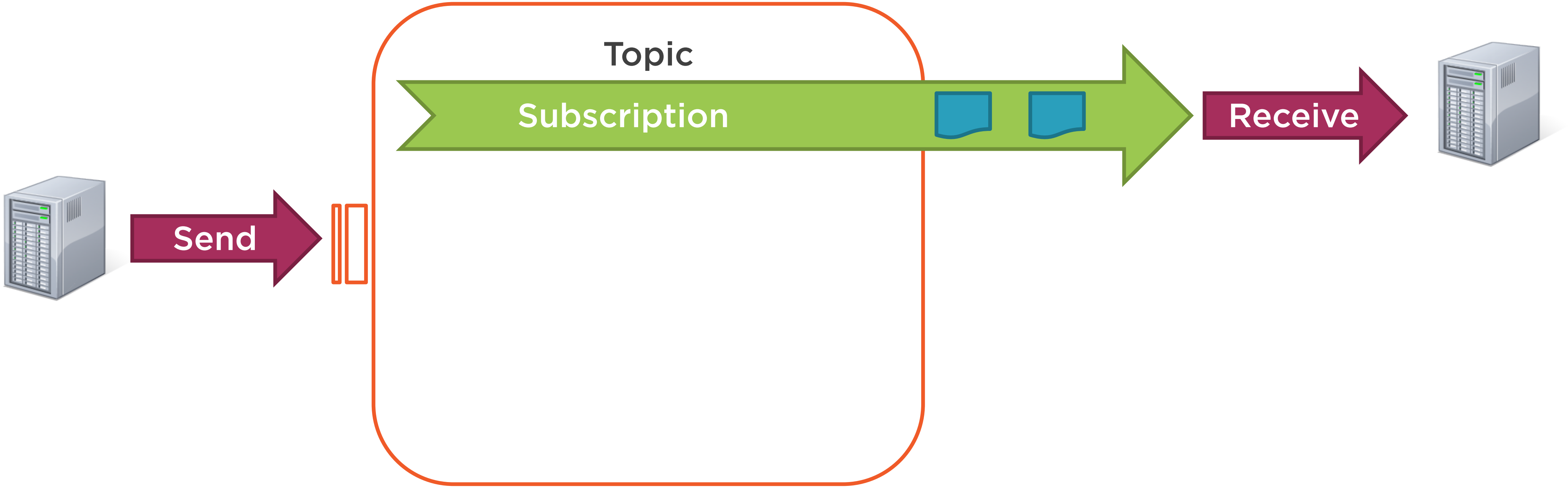
Point-to-Point Messaging



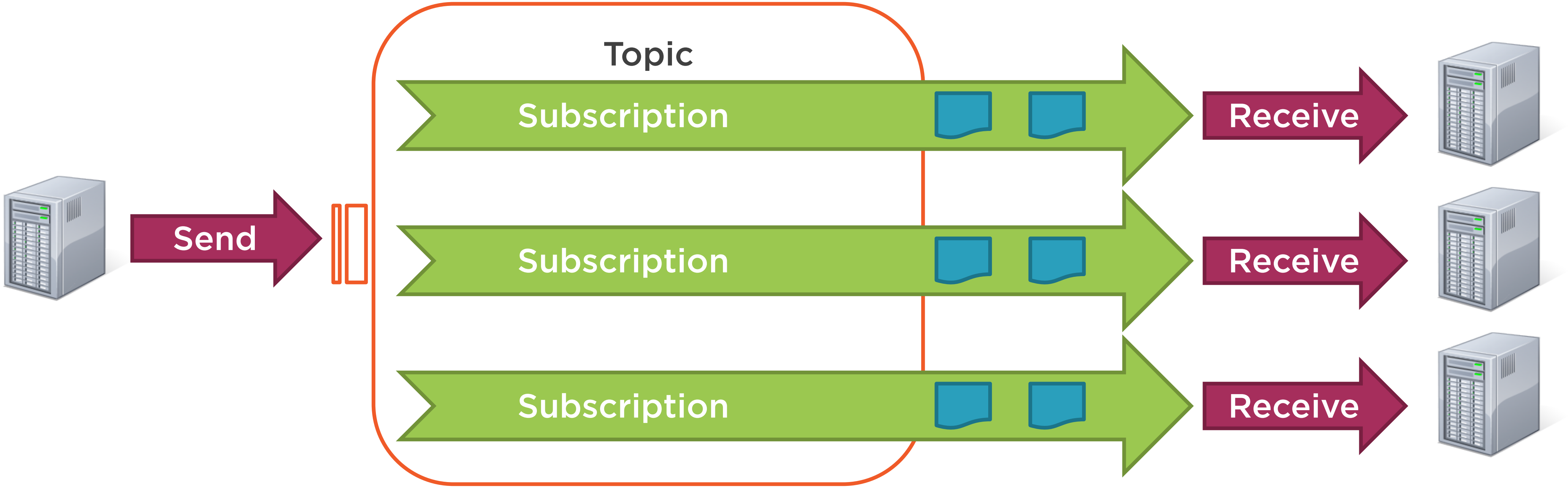
Point-to-Point Messaging



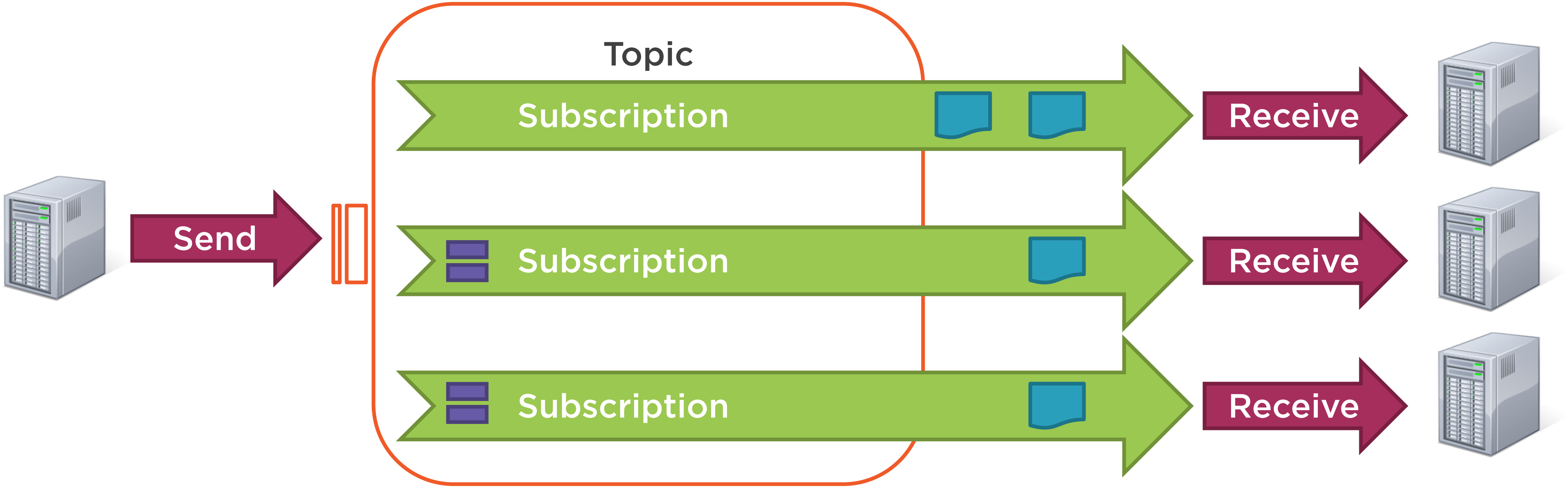
Publish-subscribe Messaging



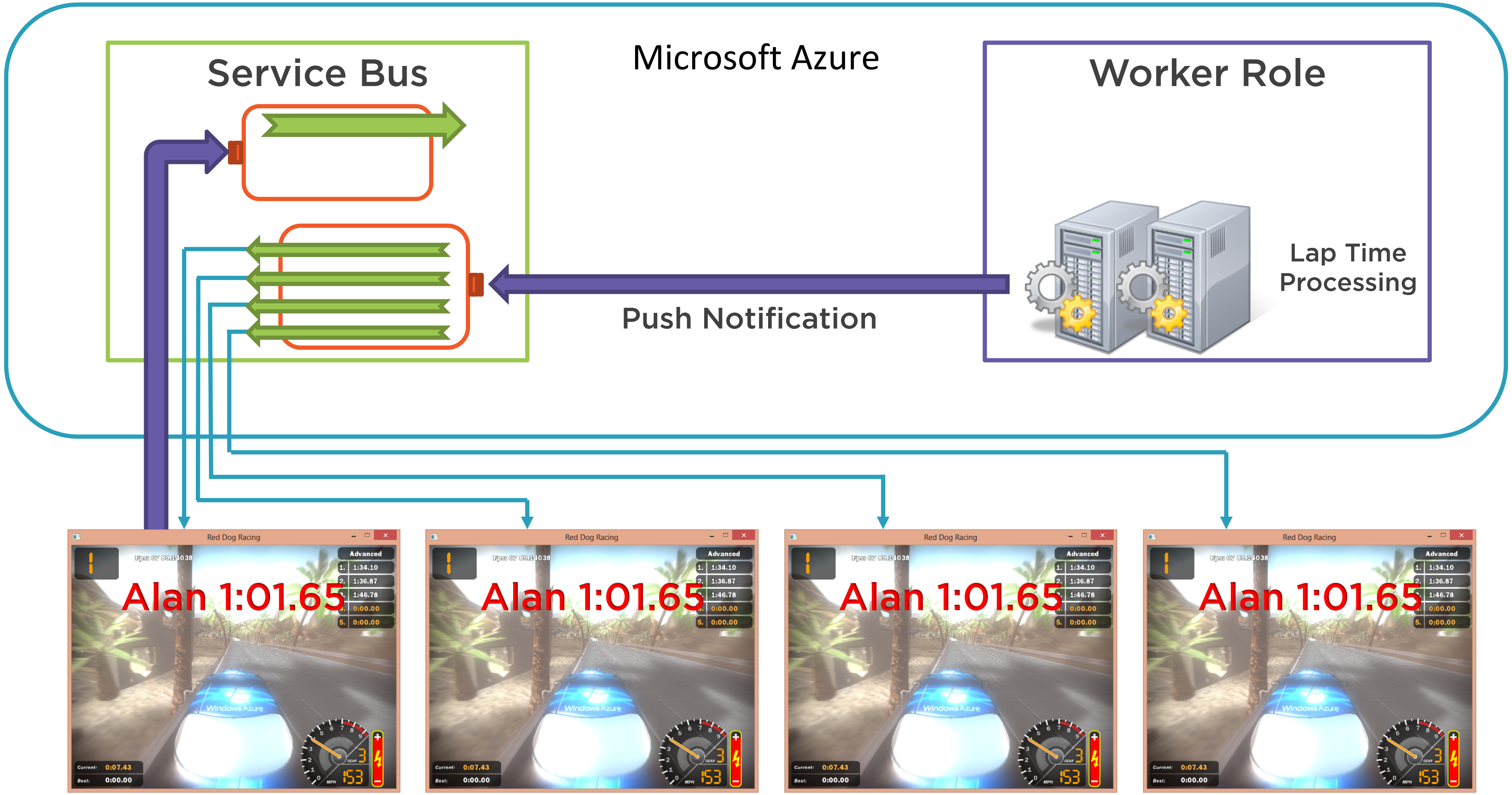
Publish-subscribe Messaging



Publish-subscribe Messaging



Scenario: Push Notifications



Microsoft Azure

Service Bus

Worker Role

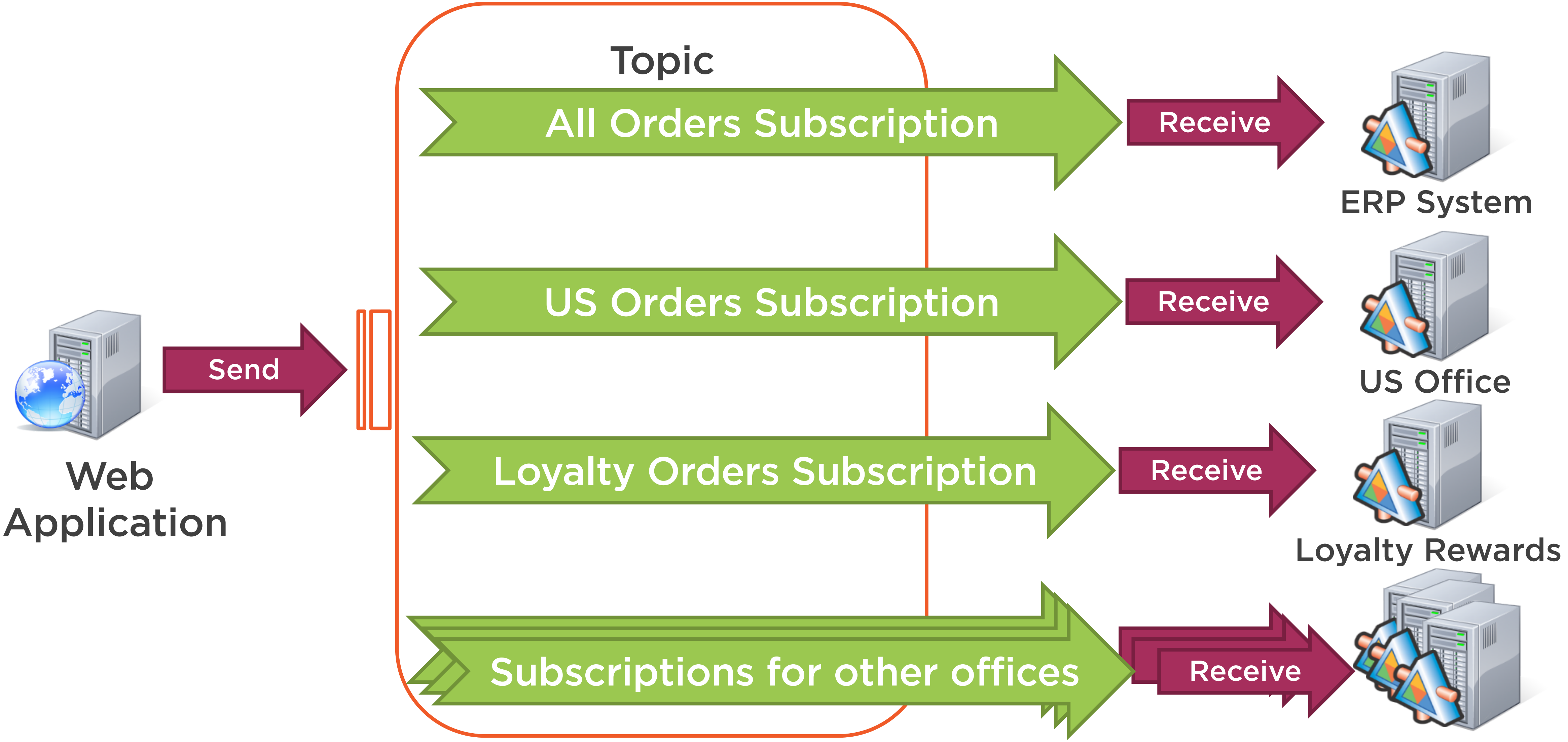
Lap Time Processing

Push Notification

PLACEHOLDER

Subscription Rules

Publish-Subscribe Messaging



Filter Types

SQL Filter

- Expression defined in “TSQL Like” expression
- Message properties dictionary used in expression evaluation

Correlation Filter

- Only supports equals operation
- Improved performance for basic filtering
- Can be applied to many properties in message header

Creating Topics with Default Subscriptions

```
// Add a default subscription
await managementClient.CreateSubscriptionAsync("ordertopic", "allOrdersSubscription");
```

Default filter subscription: 1 = 1

Creating Topics with Default Subscriptions

```
// Add a default subscription
namespaceMgr.CreateSubscription("ordertopic", "allOrdersSubscription");
```

Creating SQL Filtered Subscriptions

Subscriptions are added with SQL filter specified

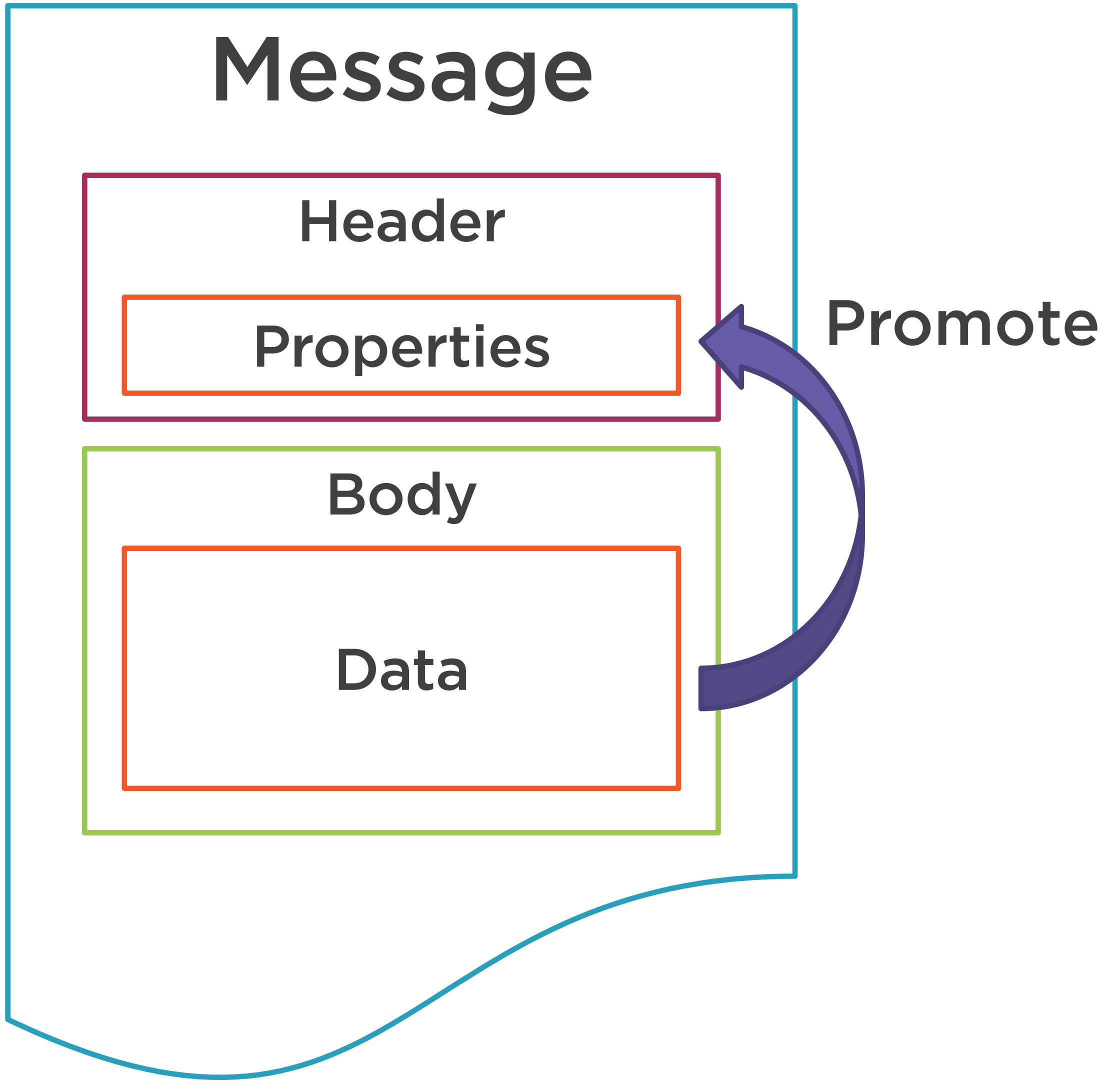
Values used in filters must be available in message header properties collection

```
// Add subscription for USA region
var ruleDescription = new RuleDescription("region", new SqlFilter("Region = 'USA'"));
var subscriptionDescription = new SubscriptionDescription("ordertopic", "allOrdersSubscription");
await managementClient.CreateSubscriptionAsync(subscriptionDescription, ruleDescription);
```

```
// Add subscription for large orders
var ruleDescription = new RuleDescription("items", new SqlFilter("Items > 30"));
var subscriptionDescription = new SubscriptionDescription("ordertopic", "largeOrderSubscription");
await managementClient.CreateSubscriptionAsync(subscriptionDescription, ruleDescription);
```

```
// Add subscription for loyal customers in the USA region
var ruleDescription = new RuleDescription("loyalty", new SqlFilter("Loyalty = true AND Region = 'USA'"));
var subscriptionDescription = new SubscriptionDescription("ordertopic", "loyaltySubscription");
await managementClient.CreateSubscriptionAsync(subscriptionDescription, ruleDescription);
```

Promoting Properties



Message body data cannot be used to route messages

Fields used for message routing are promoted to message header

Promoting Properties

```
class Order
{
    public string Name { get; set; }
    public DateTime OrderDate { get; set; }
    public int Items { get; set; }
    public double Value { get; set; }
    public string Priority { get; set; }
    public string Region { get; set; }
    public bool HasLoyltyCard { get; set; }
}
```

```
// Serialize the order to JSON
var orderJson = JsonConvert.SerializeObject(order);

// Create a message containing the serialized order Json
var message = new Message(Encoding.UTF8.GetBytes(orderJson));

// Promote properties...
message.UserProperties.Add("Region", order.Region);
message.UserProperties.Add("Items", order.Items);
message.UserProperties.Add("Value", order.Value);
message.UserProperties.Add("Loyalty", order.HasLoyltyCard);
```

Using Correlation Filters

Subscriptions are added with correlation filter specified

- Only equals predicate is possible

Various properties can be used

- CorrelationId, ContentType ,Label ,MessageId, ReplyTo, ReplyToSessionId, SessionId, To

```
// Add correlation subscription for UK orders.
var ruleDescription = new RuleDescription("region", new CorrelationFilter("UK"));

var subscriptionDescription = new SubscriptionDescription("ordertopic", "ukSubscription");
await managementClient.CreateSubscriptionAsync(subscriptionDescription, ruleDescription);
```

```
// Add correlation subscription for test messages.
var ruleDescription = new RuleDescription("testmessages", new CorrelationFilter() { Label="test" });

var subscriptionDescription = new SubscriptionDescription("ordertopic", "loyaltySubscription");
await managementClient.CreateSubscriptionAsync(subscriptionDescription, ruleDescription);
```

Demo

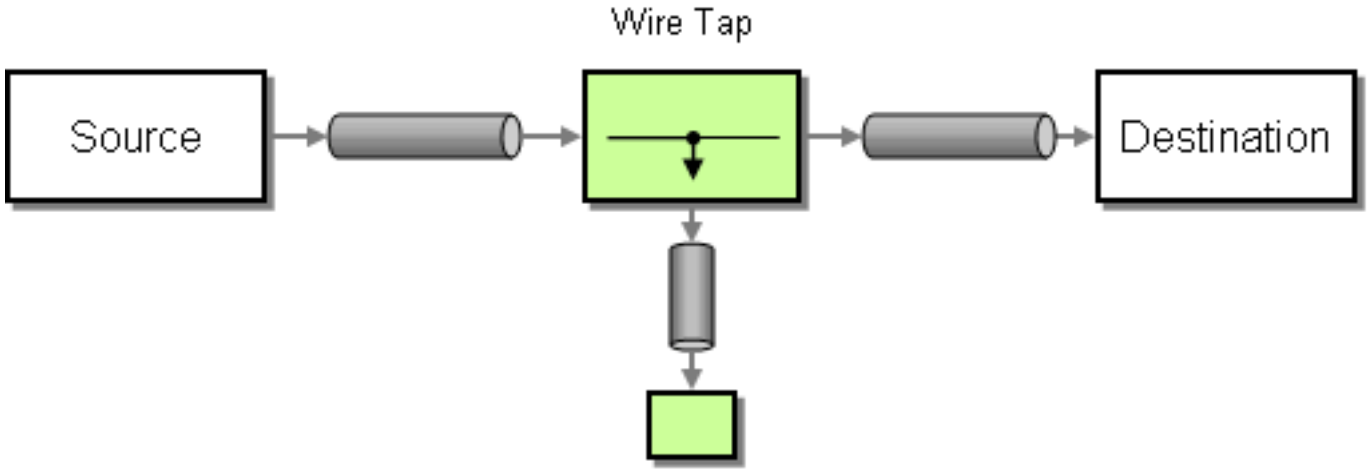
Subscription Rules

- Creating subscriptions using SQL filters
- Creating subscriptions using correlation filters
- Promoting properties and sending messages
- Receiving messages from subscriptions

Implementing a Wire Tap

Wire Tap Pattern

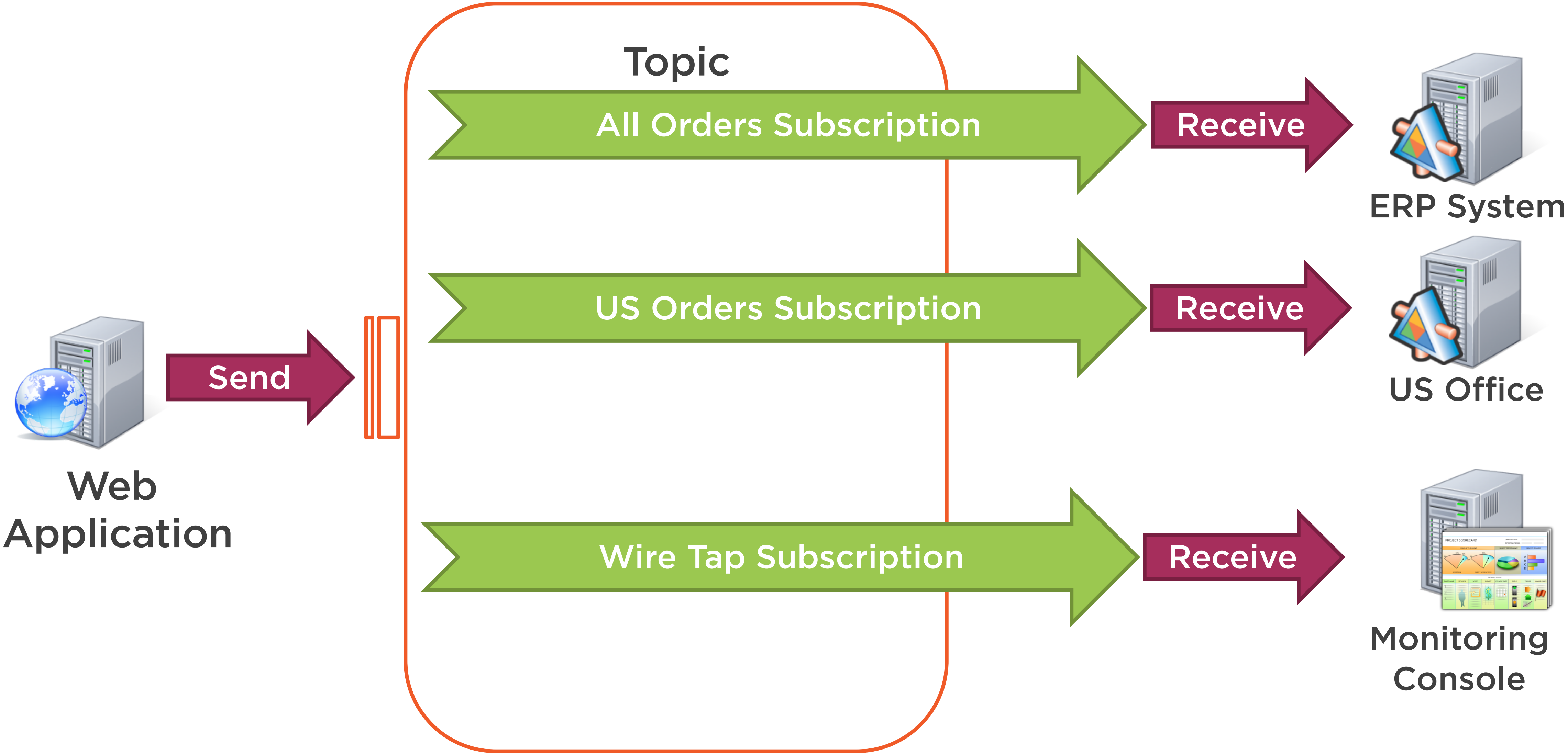
How do you inspect messages that travel on a point-to-point channel?



Insert a simple Recipient List into the channel that publishes each incoming message to the main channel and a secondary channel.

Enterprise Integration Patterns – Gregor Hohpe
www.enterpriseintegrationpatterns.com

Publish-subscribe Messaging



Demo

Implementing a Wire Tap

- Creating a wire tap
- Inspecting messages

Summary

Topics and Subscriptions provide publish-subscribe messaging

Filter subscriptions can be used to route messages

- SQL filters
- Correlation filters

Routing is based on message properties

If no filter subscription is specified, a default filter is created

- 1 = 1
- Subscribes to all messages

A wire tap can be used to monitor message flow through a topic