

Advanced NServiceBus Deployment

NSBCon NYC 2014

Kijana Woodard / @kijanawoodard

Background

- From Dallas, TX
- Professional develop since 1996
- .net since 1.0
- Independent contractor since 2010

Around the web

- kijanawoodard.com
- twitter.com/kijanawoodard
- github.com/kijanawoodard
- linkedin.com/in/kijanawoodard

- particular forum
- ravendb forum
- ddd/cqrs forum
- hypermedia forum

Slides are on GitHub

github.com/kijanawoodard/NSBCon-NYC-2014/

Mitigation
all
the
way
down

Big Ball of Mud

An application which defies modification

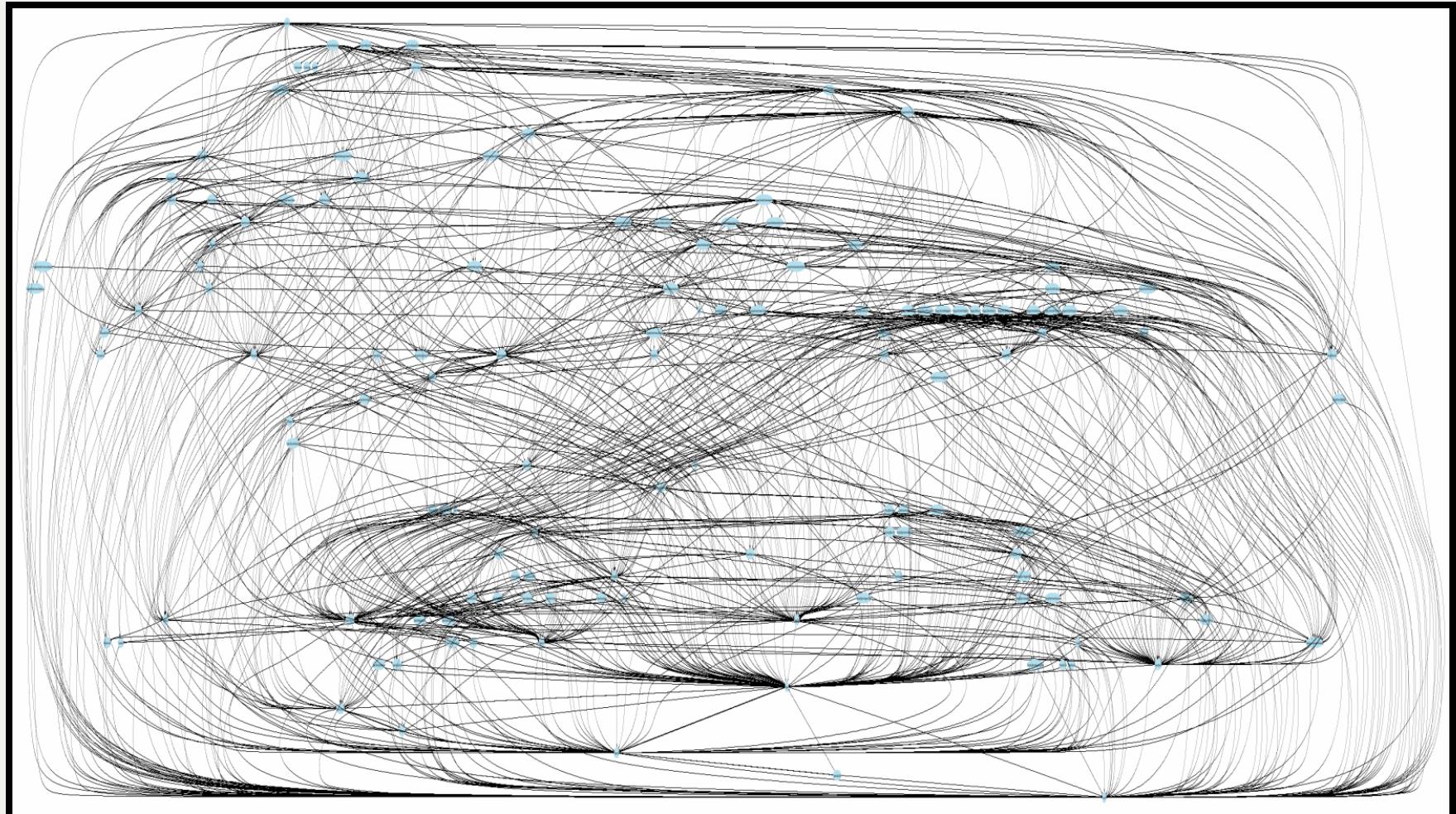
BBOM Formal Definition

```
SELECT currentAssignment  
FROM workHistory  
WHERE originalDeveloper != me
```

Defense Mechanisms

- Change control committee
- Source control committee

BBOM Architecture



- thedailywtf.com Comment On The Enterprise Dependency

BBOM Implementation

```
public class OrderController
{
    ...

    public OrderController(
        IOrderRepository orders,
        IBillingService billing,
        IShippingService shipping)
    {
        _orders = orders;
        _billing = billing;
        _shipping = shipping;
    }

    public HttpStatusCode Post(Order order)
    {
        _orders.Save(order.OrderDetails);
        _billing.Charge(order.BillingDetails);
        _shipping.Ship(order.ShippingDetails);

        return HttpStatusCode.OK;
    }
}
```

Mediator

- kijanawoodard.com/introducing-nimbus
- lostechies.com/jimmybogard/2014/09/09/tackling-cross-cutting-concerns-with-a-mediator-pipeline/

Decouple through Events

Events Implementation

```
public class OrderController
{
    public IBus Bus { get; set; }

    public HttpStatusCode Post(ProcessOrder order)
    {
        Bus.Send(order);
        return HttpStatusCode.OK;
    }
}

...

public class OrderHandler
    : IHandleMessages<ProcessOrder>
{
    public IBus Bus { get; set; }

    public void Handle(ProcessOrder message)
    {
        //save order...
        Bus.Publish<IOrderAccepted>(x => x.Id = message.Id);
    }
}
```

Bliss

- Encapsulation
- Single Responsibility
- Open Closed Principle
- Interface Segregation Principle
- Dependency Inversion Principle

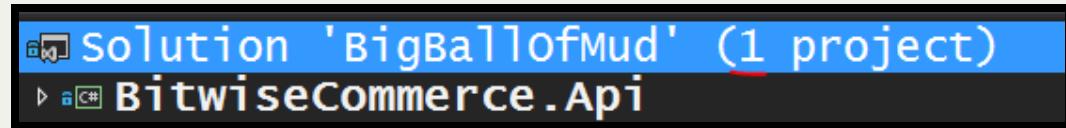
It's almost as if...

OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things.

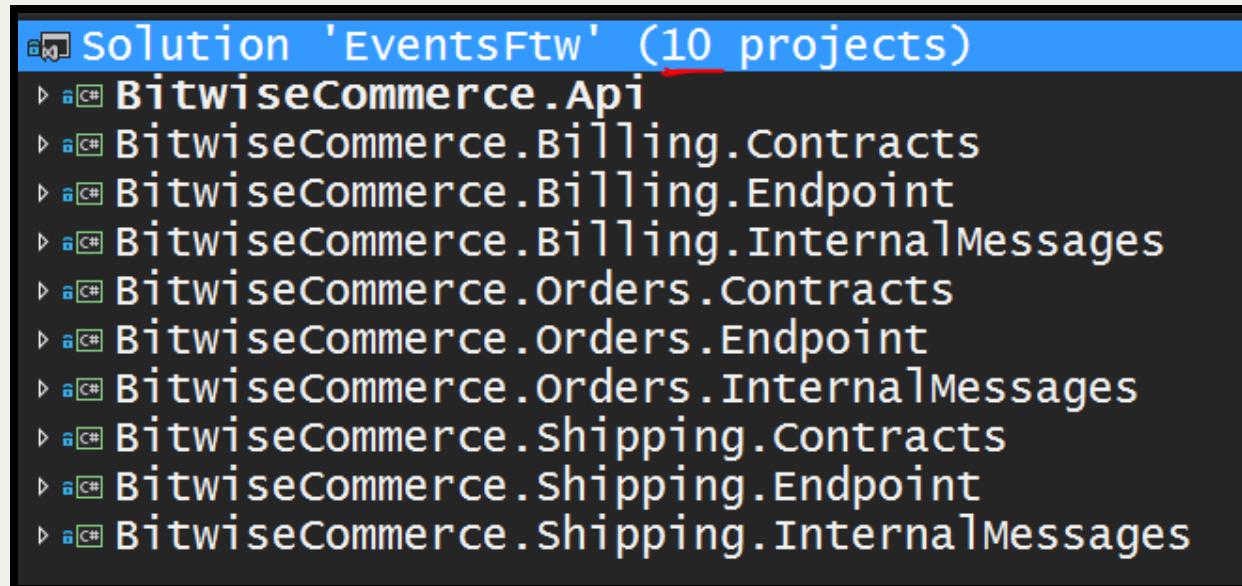
Alan Kay - 2003

What's the Problem?

BBOM Solution



Events Solution



Events Dev UX

Friction

- Message Conventions
- Serialization Format
- Persistence Configuration

```
public class EndpointConfig : IConfigureThisEndpoint, AsA_Server, IWantsCustomInitial
{
    public void Init()
    {
        Configure.with()
            .DefaultBuilder()
            .DefiningCommandsAs(t =>
                t.Namespace != null &&
                t.Namespace.Contains("InternalMessages"))
            .DefiningEventsAs(t =>
                t.Namespace != null &&
                t.Namespace.Contains("Contracts"))
            .UseInMemoryTimeoutPersister()
            .InMemorySagaPersister()
            .InMemorySubscriptionStorage()
            .InMemoryFaultManagement();

        Configure.Serialization.Json();
    }
}
```

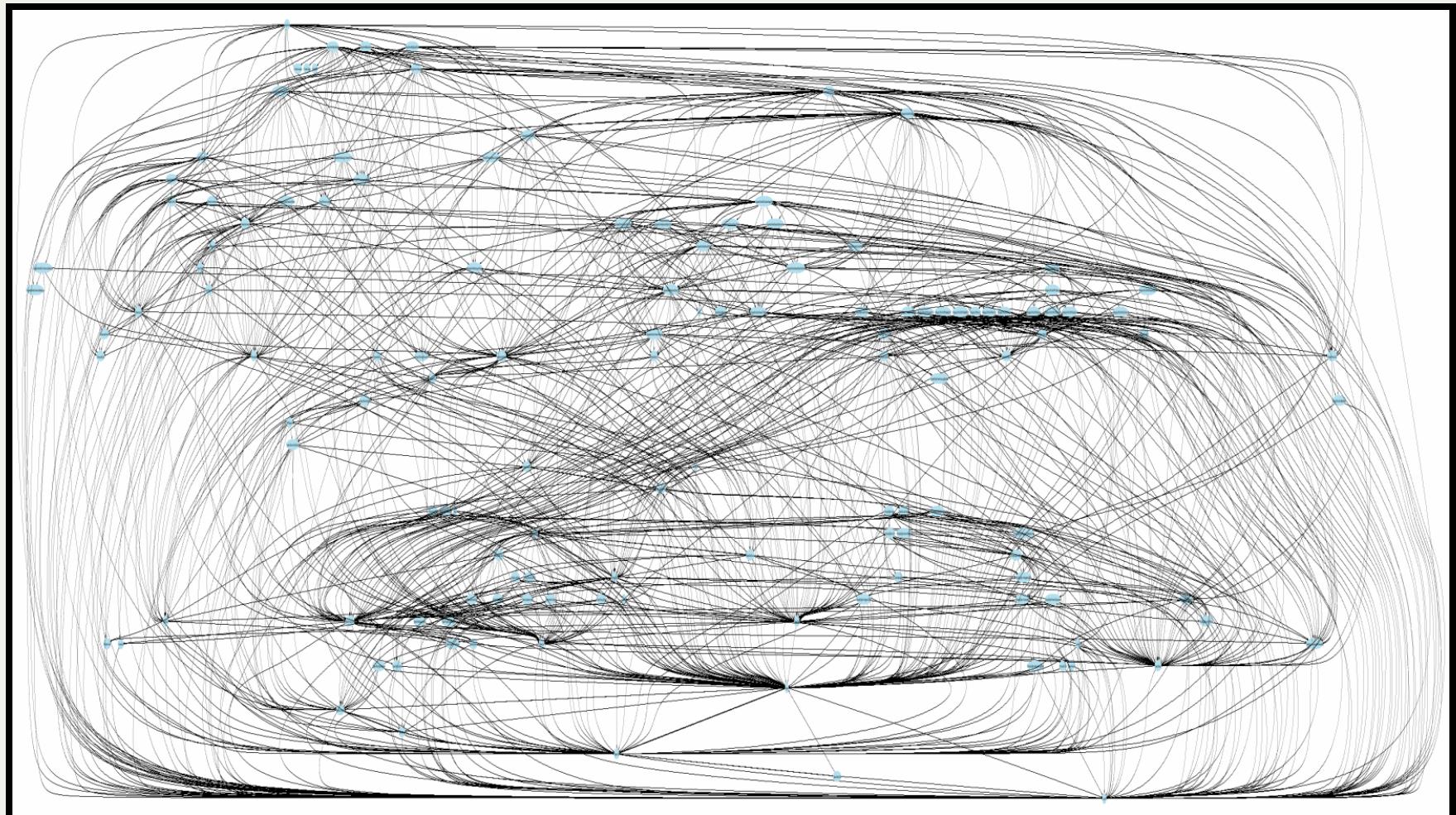
Harder to Deploy

- Many Endpoints vs One Application

Decreased Legibility

ribbonfarm.com/2010/07/26/a-big-little-idea-called-legibility/

Developer's Opinion of BBOM



- thedailywtf.com Comment On The Enterprise Dependency

Executive Summary of BBOM

**Important Software
Known**

Executive Summary of Messaging

Messages | Endpoints | Queues |
Brokers | Services | Autonomy |
Agile | Standups | TDD | BDD | Git
Hub | HipChat | Bamboo | Blah | Bla
h | Blah | Blah | Blah | Blah | Bla
h | Blah

\$?????????????????

Combat Friction

Don't Call It a Rewrite

```
public class OrderController
{
    ...

    public OrderController(
        IOrderRepository orders,
        IBillingService billing,
        IShippingService shipping)
    {
        _orders = orders;
        _billing = billing;
        _shipping = shipping;
    }

    public HttpStatusCode Post(Order order)
    {
        _orders.Save(order.OrderDetails);
        _billing.Charge(order.BillingDetails);
        _shipping.Ship(order.ShippingDetails);

        return HttpStatusCode.OK;
    }
}
```

Strangler Pattern



- Entry on Strangler Fig Wikipedia

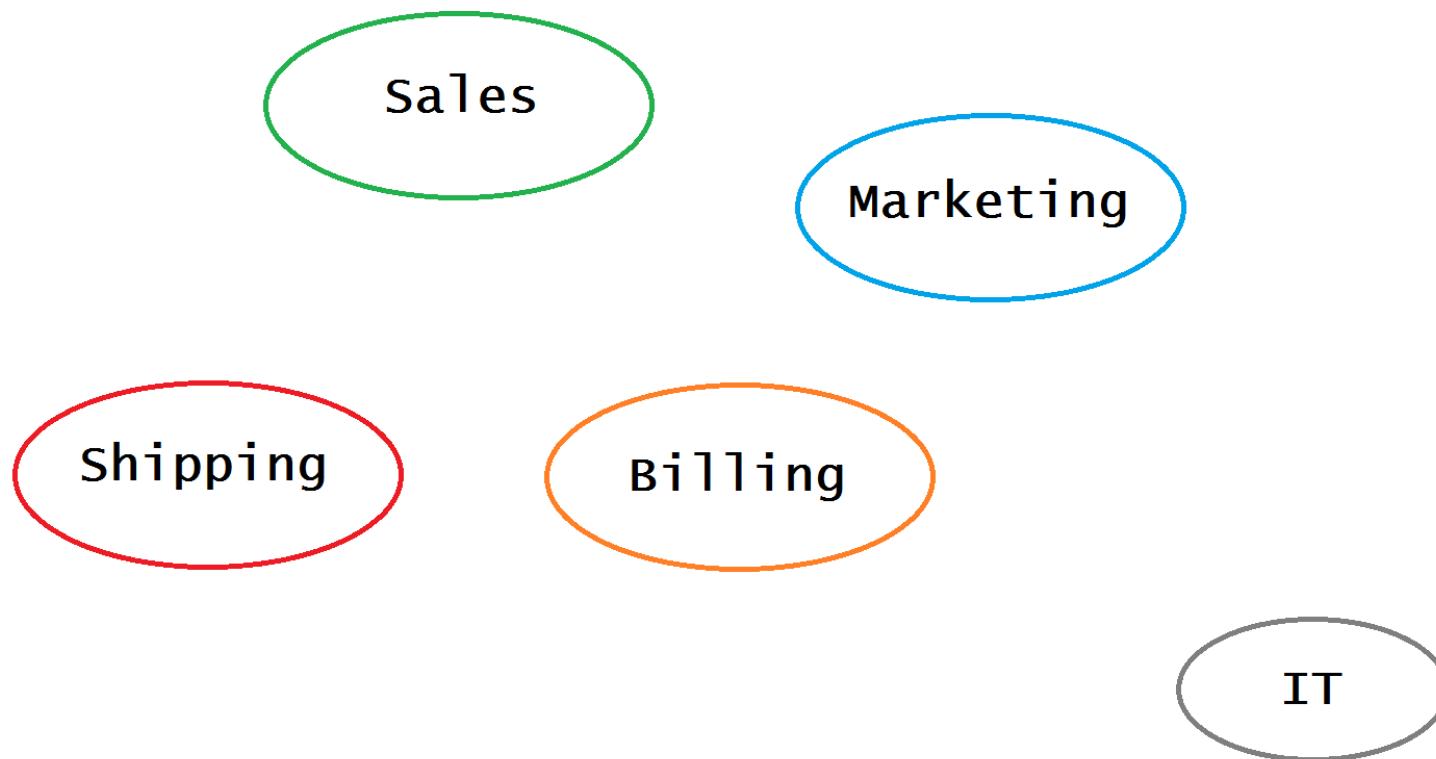
<http://www.martinfowler.com/bliki/StranglerApplication.html>

Context

- Remember that time when we couldn't deploy anything for 6 weeks...
- Remember that time when the strategic marketing initiative was held up due to a Shipping upgrade...

Restore some Legibility

Important Software



Have No Illusions

The Big Ball of Mud will not die easily



- Advanced Life Skills

Automate

Automate

Automate

- Conventions: Web vs Endpoint
- Separate configuration repo
- Days to Minutes
- Smaller, more frequent releases
- Confidence
- Repeatability

Configuration

	db	alert
local	http://localhost:8080	alerts@devnull.example.com
dev	http://localhost:8080	alerts@dev.example.com
qa	http://qa:8080	alerts@qa.example.com
prod	http://p532977:8080	alerts@example.com

Don't Do an "Automation Project"

Over time, layer in...

- Build scripts
 - powershell / msbuild
 - Team City / Bamboo
- Deploy scripts
 - powershell
 - Octopus Deploy
- Monitoring and Metrics
 - Service Insight / Control
 - SCOM

Discussion on DevOPS Experience on devopslive.org

Bus Stop

github.com/andreasohlund/BusStop

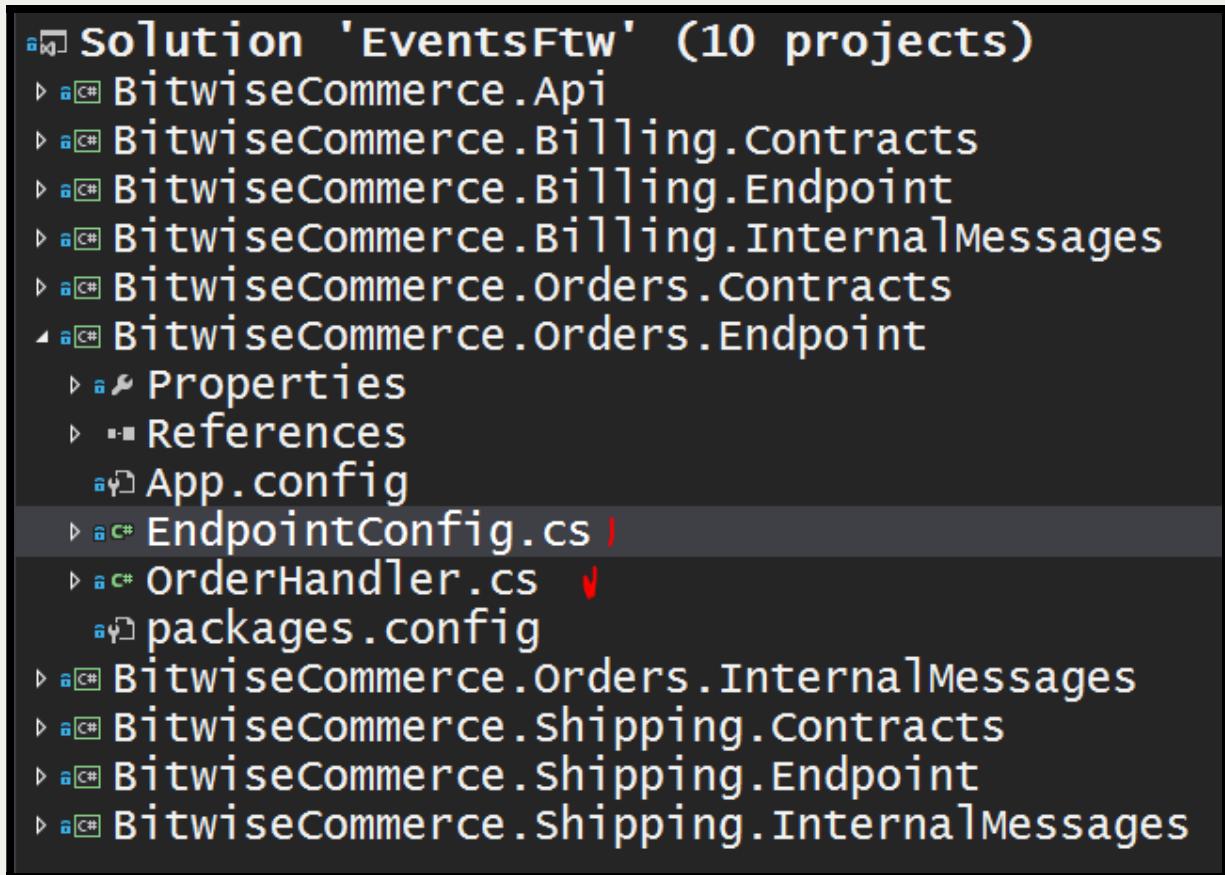
Bus Stop

```
namespace BusStop.Config
{
    public static class MyConvention
    {
        public static Configure MyMessageConventions(this Configure config)
        {
            config.DefiningEventsAs(t => t.Namespace != null &&
                                      t.Namespace.EndsWith(".Contracts"));
            return config;
        }
    }
}
```

github.com/andreasohlund/BusStop

Endpoints vs Handlers

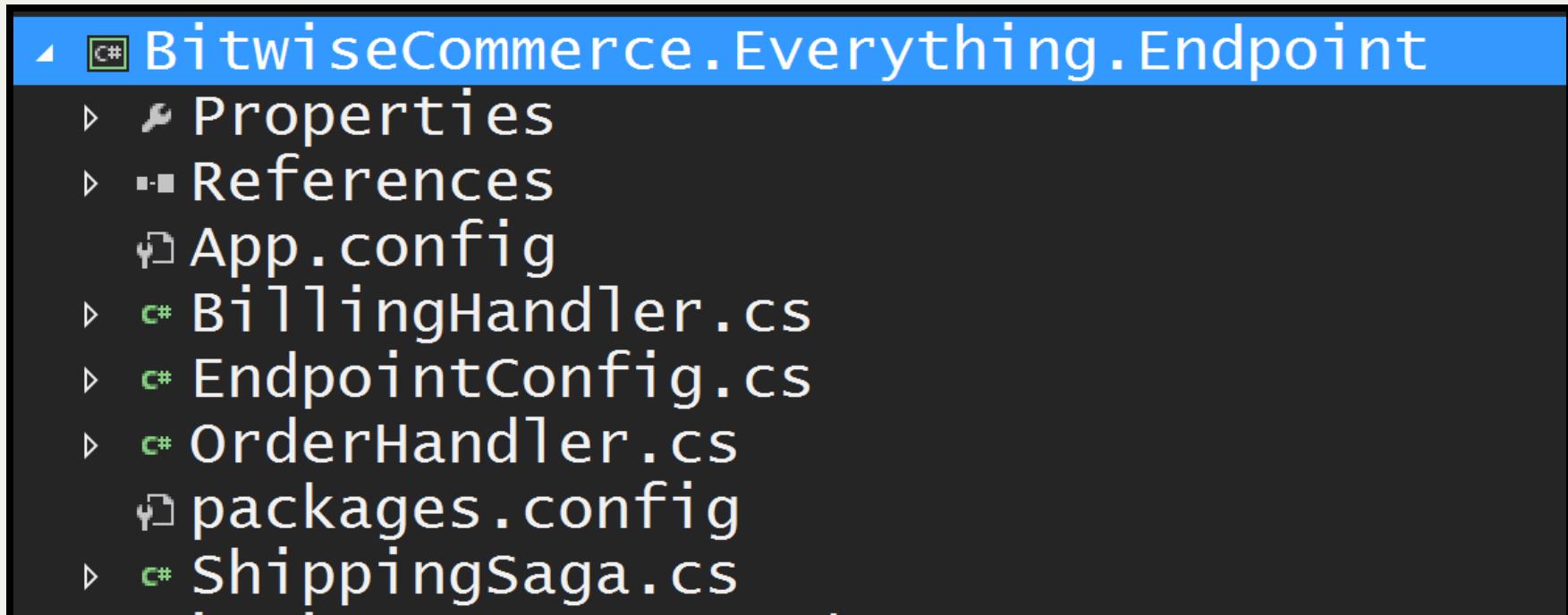
Before



The screenshot shows a Solution Explorer window for a solution named 'EventsFtw' containing ten projects:

- BitwiseCommerce.Api
- BitwiseCommerce.Billing.Contracts
- BitwiseCommerce.Billing.Endpoint
- BitwiseCommerce.Billing.InternalMessages
- BitwiseCommerce.Orders.Contracts
- BitwiseCommerce.Orders.Endpoint
 - Properties
 - References
 - App.config
 - EndpointConfig.cs (selected)
 - OrderHandler.cs
 - packages.config
- BitwiseCommerce.Orders.InternalMessages
- BitwiseCommerce.Shipping.Contracts
- BitwiseCommerce.Shipping.Endpoint
- BitwiseCommerce.Shipping.InternalMessages

All Together



The screenshot shows a file explorer window with a blue header bar. The header bar contains the text "BitwiseCommerce.Everything.Endpoint" in white, with a small "C#" icon to its left. Below the header, the project structure is listed:

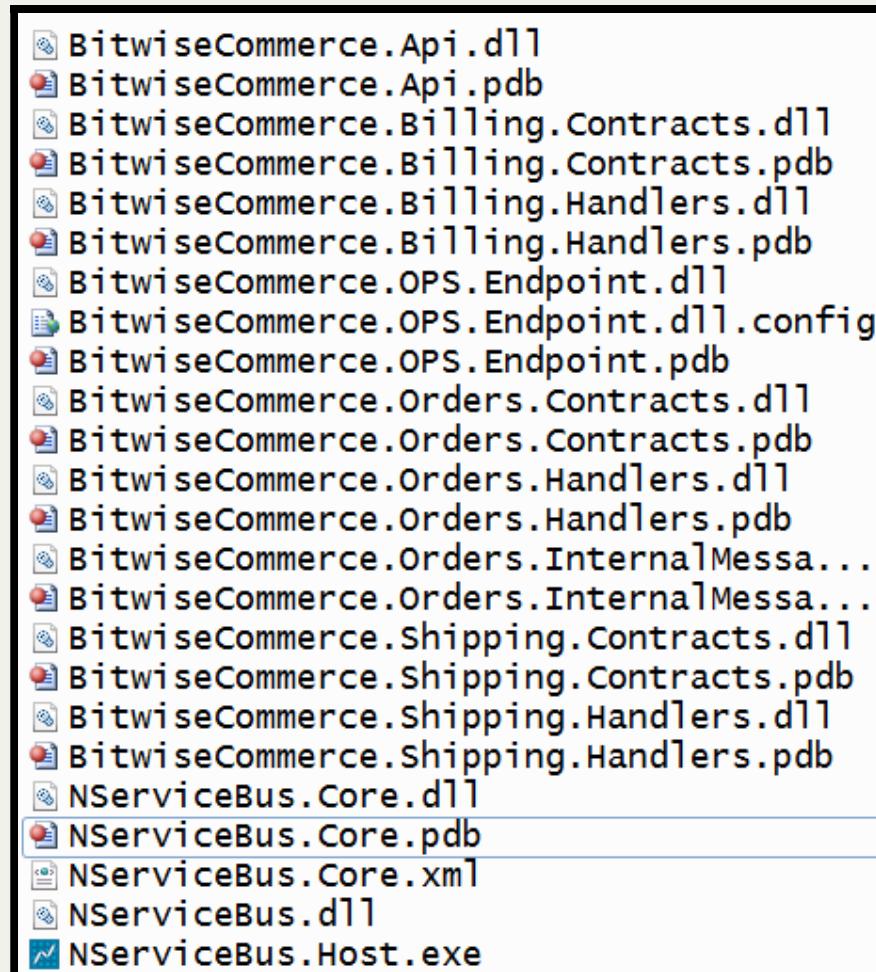
- ▶ C# BitwiseCommerce.Everything.Endpoint
 - ▷ Properties
 - ▷ References
 - ▷ App.config
 - ▷ C# BillingHandler.cs
 - ▷ C# EndpointConfig.cs
 - ▷ C# OrderHandler.cs
 - ▷ packages.config
 - ▷ C# ShippingSaga.cs

Combined

```
Solution 'Combined' (11 projects)
    BitwiseCommerce.Api
    BitwiseCommerce.Billing.Contracts
    BitwiseCommerce.Billing.Handlers
    BitwiseCommerce.Billing.InternalMessages
    BitwiseCommerce.OPS.Endpoint
        Properties
        References
            App.config
        EndpointConfig.cs
        packages.config
    BitwiseCommerce.Orders.Contracts
    BitwiseCommerce.Orders.Handlers
        Properties
        References
        OrderHandler.cs
        packages.config
    BitwiseCommerce.Orders.InternalMessages
    BitwiseCommerce.Shipping.Contracts
    BitwiseCommerce.Shipping.Handlers
    BitwiseCommerce.Shipping.InternalMessages
```

Dev UX

Build Artifacts



Mix and Match

Api	Orders	Shipping	Billing
BitwiseCommerce.Api.dll BitwiseCommerce.Api.pdb BitwiseCommerce.OPS.Endpoint.dll BitwiseCommerce.OPS.Endpoint.... BitwiseCommerce.OPS.Endpoint.pdb BitwiseCommerce.Orders.Internal... BitwiseCommerce.Orders.Internal... NServiceBus.Core.dll NServiceBus.Core.pdb NServiceBus.Core.xml NServiceBus.dll NServiceBus.Host.exe	New folder BitwiseCommerce.OPS.Endpoint.dll BitwiseCommerce.OPS.Endpoint.dll.c.. BitwiseCommerce.OPS.Endpoint.pdb BitwiseCommerce.Orders.Contracts.dll BitwiseCommerce.Orders.Contracts.pdb BitwiseCommerce.Orders.Handlers.dll BitwiseCommerce.Orders.Handlers.pdb BitwiseCommerce.Orders.InternalMes.. BitwiseCommerce.Orders.InternalMes.. NServiceBus.Core.dll NServiceBus.Core.pdb NServiceBus.Core.xml NServiceBus.dll NServiceBus.Host.exe		
		BitwiseCommerce.Billing.Contrac... BitwiseCommerce.Billing.Contrac... BitwiseCommerce.Billing.Handler... BitwiseCommerce.Billing.Handler... BitwiseCommerce.OPS.Endpoint.dll BitwiseCommerce.OPS.Endpoint.dl... BitwiseCommerce.OPS.Endpoint.pdb BitwiseCommerce.Orders.Contract... BitwiseCommerce.Orders.Contract... BitwiseCommerce.Shipping.Contra... BitwiseCommerce.Shipping.Contra... BitwiseCommerce.Shipping.Handle... BitwiseCommerce.Shipping.Handle... NServiceBus.Core.dll NServiceBus.Core.pdb NServiceBus.Core.xml NServiceBus.dll NServiceBus.Host.exe	

Run

```
2014-09-27 21:47:27,909 [13] INF
O NServiceBus.Scheduling.DefaultScheduler [(null)] <(null)> - S
cheduled task <>c__DisplayClass1
    run for 00:00:00.0022638
2014-09-27 21:47:30,900 [29] INF
O NServiceBus.Scheduling.DefaultScheduler [(null)] <(null)> - S
tart executing scheduled task <>
c__DisplayClass1
Order 3 sent.
2014-09-27 21:47:30,906 [11] INF
O NServiceBus.Scheduling.DefaultScheduler [(null)] <(null)> - S
cheduled task <>c__DisplayClass1
    run for 00:00:00.0018225
^ .MessageDrivenSubs
] <(null)> - Subsc
eCommerce.Orders.C
ntracts, Version=1
Order 1 accepted.
Order 2 accepted.
Order 3 accepted.
□
ntracts, Version=1
Order 1 billed.
Order 1 shipped.
Order 2 billed.
Order 2 shipped.
Order 3 billed.
Order 3 shipped.
□
```

NServiceBus.Host.exe -endpointName=orders

Decouple Dev from OPS

- Redeploy without recompile
- Tie to Platform Tools

Pushing Forwards

A 2nd Look at Events Impl

```
public class OrderController
{
    public IBus Bus { get; set; }

    public HttpStatusCode Post(ProcessOrder order)
    {
        Bus.Send(order);
        return HttpStatusCode.OK;
    }
}

...

public class OrderHandler
    : IHandleMessages<ProcessOrder>
{
    public IBus Bus { get; set; }

    public void Handle(ProcessOrder message)
    {
        //save order...
        Bus.Publish<IOrderAccepted>(x => x.Id = message.Id);
    }
}
```

One Choice

```
public class OrderController
{
    private IFactory<Duck<Mockable<ProcessOrder>>> _something;
    public IBus Bus { get; set; }

    public OrderController(IFactory<Duck<Mockable<ProcessOrder>>> something)
    {
        _something = something;
    }
    public HttpStatusCode Post(ProcessOrder order)
    {
        _something.Process(order);
        return HttpStatusCode.OK;
    }
}

...
public class OrderHandler
    : IHandleMessages<ProcessOrder>
{
    private IFactory<Duck<Mockable<ProcessOrder>>> _something;
    public IBus Bus { get; set; }

    public OrderHandler(IFactory<Duck<Mockable<ProcessOrder>>> something)
    {
        _something = something;
    }
    public void Handle(ProcessOrder message)
    {
```

Exploit Decoupling

```
public class OrderController
{
    private readonly IHandleMessages<ProcessOrder> _handler;
    public IBus Bus { get; set; }

    public OrderController(IHandleMessages<ProcessOrder> handler)
    {
        _handler = handler;
    }

    public HttpStatusCode Post(ProcessOrder order)
    {
        _handler.Handle(order);
        return HttpStatusCode.OK;
    }
}

...

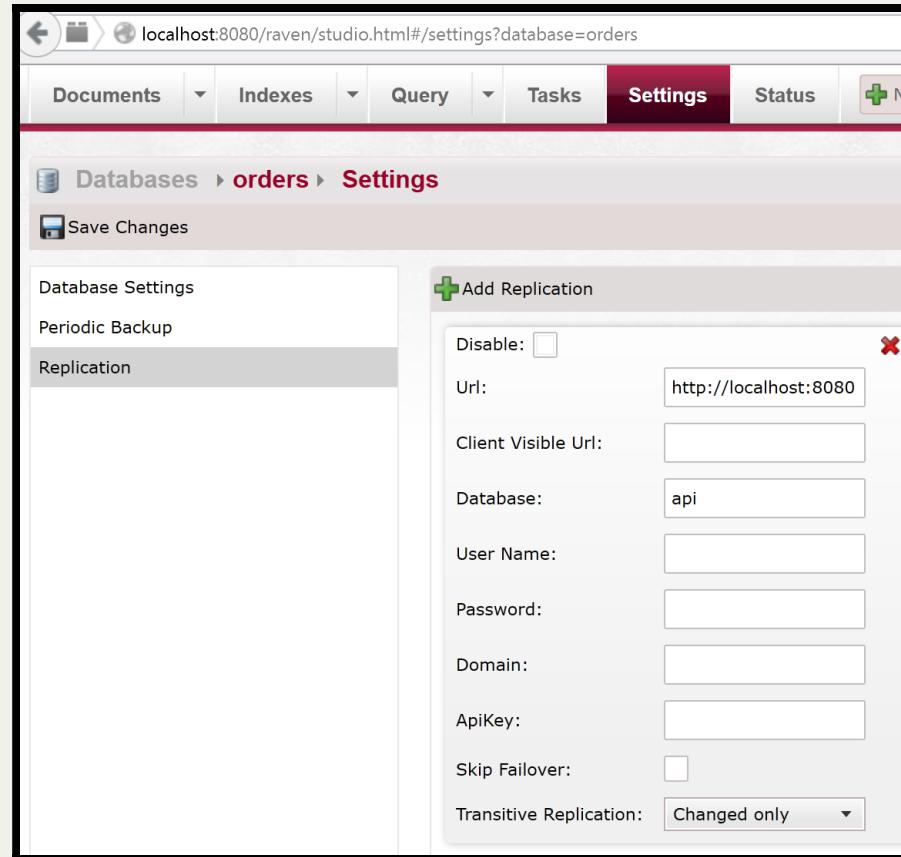
public class OrderHandler
    : IHandleMessages<ProcessOrder>
{
    public IBus Bus { get; set; }

    public void Handle(ProcessOrder message)
    {
        //save order...
        Bus.Publish<IOrderAccepted>(x => x.Id = message.Id);
    }
}
```

Run

```
2014-09-27 21:51:42,254 [21] : eduling.DefaultSchedule executing scheduled task Order 2 accepted.  
2014-09-27 21:51:42,267 [20] : eduling.DefaultSchedule executing scheduled task <>c__DisplayC  
020  
2014-09-27 21:51:45,258 [21] : eduling.DefaultSchedule executing scheduled task Order 3 accepted.  
2014-09-27 21:51:45,268 [20] : eduling.DefaultScheduler executing task <>c__DisplayCl  
90  
2014-09-27 21:51:32,705 [21] : iicast.Subscriptions.MessageDrivenSubscriptionManager - Subscribing shipbill@WYLDMA01 BitwiseCommerce.Orders.Controller, BitwiseCommerce.Orders.Controller, 0, Culture=neutral, PublicKeyToken=null  
OrderController(IHandler.Handle(order));  
2014-09-27 21:51:31,251 [20] : iicast.Subscriptions.MessageDrivenSubscriptionManager - Subscribing shipbill@WYLDMA01 BitwiseCommerce.Billing.Controller, BitwiseCommerce.Billing.Controller, 0, Culture=neutral, PublicKeyToken=null  
HttpHandler.Handle(order);
```

Subscriptions



Subscribed

```
2014-09-27 21:53:24,335
eduling.DefaultSchedule
executing scheduled ta
Order 36 accepted.
2014-09-27 21:53:24,527
eduling.DefaultSchedule
uled task <>c__DisplayC
587                                         Database:
2014-09-27 21:53:27,352
eduling.DefaultSchedule
executing scheduled ta
Order 37 accepted.
2014-09-27 21:53:27,376
eduling.DefaultSchedule
uled task <>c__DisplayC
145                                         ApiKey:
[REDACTED]                                         Skip Fai
2014-09-27 21:51:32,705
icast.Subscriptions.Mess
ssageDrivenSubscriptionM
- Subscribing shipbill@W
e BitwiseCommerce.Orders
, BitwiseCommerce.Orders
.0, Culture=neutral, Pub
[REDACTED]                                         api
[REDACTED]
e BitwiseCommerce.Billin
BitwiseCommerce.Billing
.0, Culture=neutral, Pub
Order 36 billed.
Order 36 shipped.
Order 37 billed.
Order 37 shipped.
[REDACTED]
```

Direct

```
2014-09-27 21:54:06,380
eduling.DefaultSchedule
executing scheduled ta
Order 50 accepted.
2014-09-27 21:54:06,393
eduling.DefaultSchedule
uled task <>c__DisplayC
659                               Database:
2014-09-27 21:54:09,380
eduling.DefaultSchedule
executing scheduled ta
Order 51 accepted.
2014-09-27 21:54:09,395
eduling.DefaultSchedule
uled task <>c__DisplayC
253                               ApiKey.
[REDACTED]                               Skip Fai
▼
```

```
icast.Subscriptions.MessageDr
ssageDrivenSubscriptionManager
- Subscribing shipbill@WYLDMA
e BitwiseCommerce.Orders.Cont
, BitwiseCommerce.Orders.Cont
.0, Culture=neutral, PublicKey
Order DIRECT ORDER accepted.
□    api
```

```
Order 49 shipped.
Order DIRECT ORDER billed.
Order DIRECT ORDER shipped.
Order 50 billed.
Order 50 shipped.
Order 51 billed.
Order 51 shipped.
□
```

Future exploration

Break Apart the API?

```
{  
    "submit-order" : "http://api.example.com/orders"  
}
```

Later....

```
{  
    "submit-order" : "http://orders.example.com/"  
}
```

HAL - Hypertext Application Language

What about queries?

- N fanout queries vs 1 for command
- Volatile queues
- Could work with proper circuit breakers
- Easier configuration with mulitple databases

Assembly Neutral Interfaces

davidfowl.com/assembly-neutral-interfaces/
Consumer Driven Contracts

Roslyn

What if source files were distributed....

THE END

Advanced NServiceBus Deployment Questions?

- slides @ github.com/kijanawoodard/NSBCon-NYC-2014/
- kijanawoodard.com
- twitter.com/kijanawoodard