Ditching BizTalk for NServiceBus

Sam Ritchie

Case study

- Manufacturing organisation
- iSeries, SCO Unix, Windows, Linux
- Siebel, JD Edwards, LANSA, .NET
- BizTalk integration - HIS, SQL, WCF

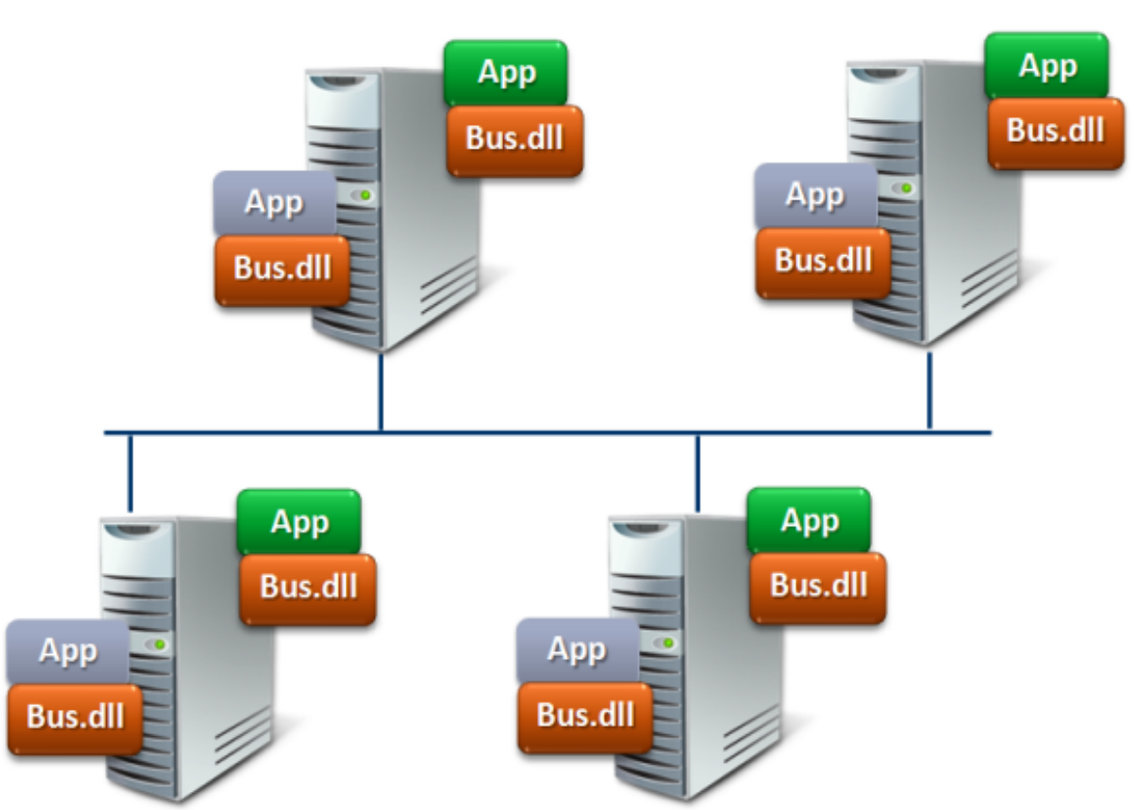
What was wrong with BizTalk?

- Skills availability - dev & admin
- Development/deployment complexity
- Inflexible - throttling, transactions, errors
- Over-reliance on HTTP/SOAP (RPC)
- Custom code at both ends anyway

What is NServiceBus?

- Open-source Service Bus on top of MSMQ
- Typically deployed as Windows Services
- Simple .NET API - plain C# code
- Asynchronous one-way messaging model
- Automatic pub/sub functionality

BizTalk vs NServiceBus



Show me the code!

```
public class CarrierUpdateHandler : IHandleMessages<CarrierUpdate>
{
    public IBus Bus { get; set; }
    public IRepository Repository { get; set; }
    public IJdeGateway JdeGateway { get; set; }

    public ILog Log
    {
        get { return LogManager.GetLogger(GetType()); }
    }

    public void Handle(CarrierUpdate message)
    {
        SalesOrderDetail salesOrder = Repository.GetNextSalesOrderDetail(message.OrderCompany, message.OrderNumber, message.OrderT
        if (salesOrder == null)
        {
            Log.WarnFormat("The Sales Order with the key {0}, {1}, {2} does not have any unshipped sales order lines.", message.Or
            return;
        }
        if (salesOrder.CarrierNumber != message.CarrierNumber)
        {
            salesOrder.CarrierNumber = message.CarrierNumber;
            Repository.UpdateSalesOrder(salesOrder);
        }
    }
}
```

Handling messages

```
public class SalesOrderChangedEventListener : EventListener
{
    public SalesOrderChangedEventListener(IRepository repository, IBus bus) : base(repository, bus) { }

    public override void CheckForEvents()
    {
        IEnumerable<SalesOrderDetailIdentifier> notifications = Repository.GetSalesOrderChangedEvents();
        foreach (var id in notifications)
        {
            try
            {
                using (TransactionScope transaction = new TransactionScope())
                {
                    SalesOrderDetail detail = Repository.GetSalesOrderDetail(id);
                    SalesOrderChanged message = SalesOrderMapper.Map(detail);

                    Bus.Publish(message);

                    Repository.DeleteSalesOrderChangedEvent(id);
                    transaction.Complete();
                }
            }
            catch (Exception exception)
            {
                Log.WarnFormat("Error attempting to publish SalesOrderChanged for {0}: {1}.", id, exception.Message);
            }
        }
    }

}
```

Publishing messages (via a DB trigger)

```
bus = Configure.With()
    .Log4Net()
    .SpringFrameworkBuilder()
    .RunCustomAction(() => Configure.Instance.Configurer.ConfigureComponent<Repository>(ComponentCallModelEnum.Singlecall))
    .RunCustomAction(() => Configure.Instance.Configurer.ConfigureComponent<ReportGateway>(ComponentCallModelEnum.Singlecall))
    .MsmqSubscriptionStorage()
    .XmlSerializer("http://tempuri.com/ServiceBus")
    .MsmqTransport()
        .IsTransactional(true)
        .PurgeOnStartup(false)
    .UnicastBus()
        .ImpersonateSender(false)
        .LoadMessageHandlers()
    .CreateBus()
    .Start();
```

```
<MsmqTransportConfig
  InputQueue="jdeinput"
  ErrorQueue="error"
  NumberOfWorkerThreads="1"
  MaxRetries="5"
/>
<MsmqSubscriptionStorageConfig Queue="jdesubscriptions" />
<UnicastBusConfig>
  <MessageEndpointMappings>
    <add Messages="Messages.Labware" Endpoint="labwareinput" />
    <add Messages="Messages.AppointmentBook" Endpoint="appointmentbookinput" />
    <add Messages="Messages.Siebel" Endpoint="siebelinput" />
  </MessageEndpointMappings>
</UnicastBusConfig>
```

Bus Configuration

Outcome

- BizTalk solution migrated in 5 man-weeks
- Significantly more reliable
- Mainstream .NET skills
- Better monitoring = more proactive fixing

Key Lessons

- Request/response is evil!
- Security limited to MSMQ ACLs
- Avoid large, complex messages
- Be prepared to roll your own admin tools
- Monitoring is still critical

Questions?

sam.ritchie@lateralwa.com.au