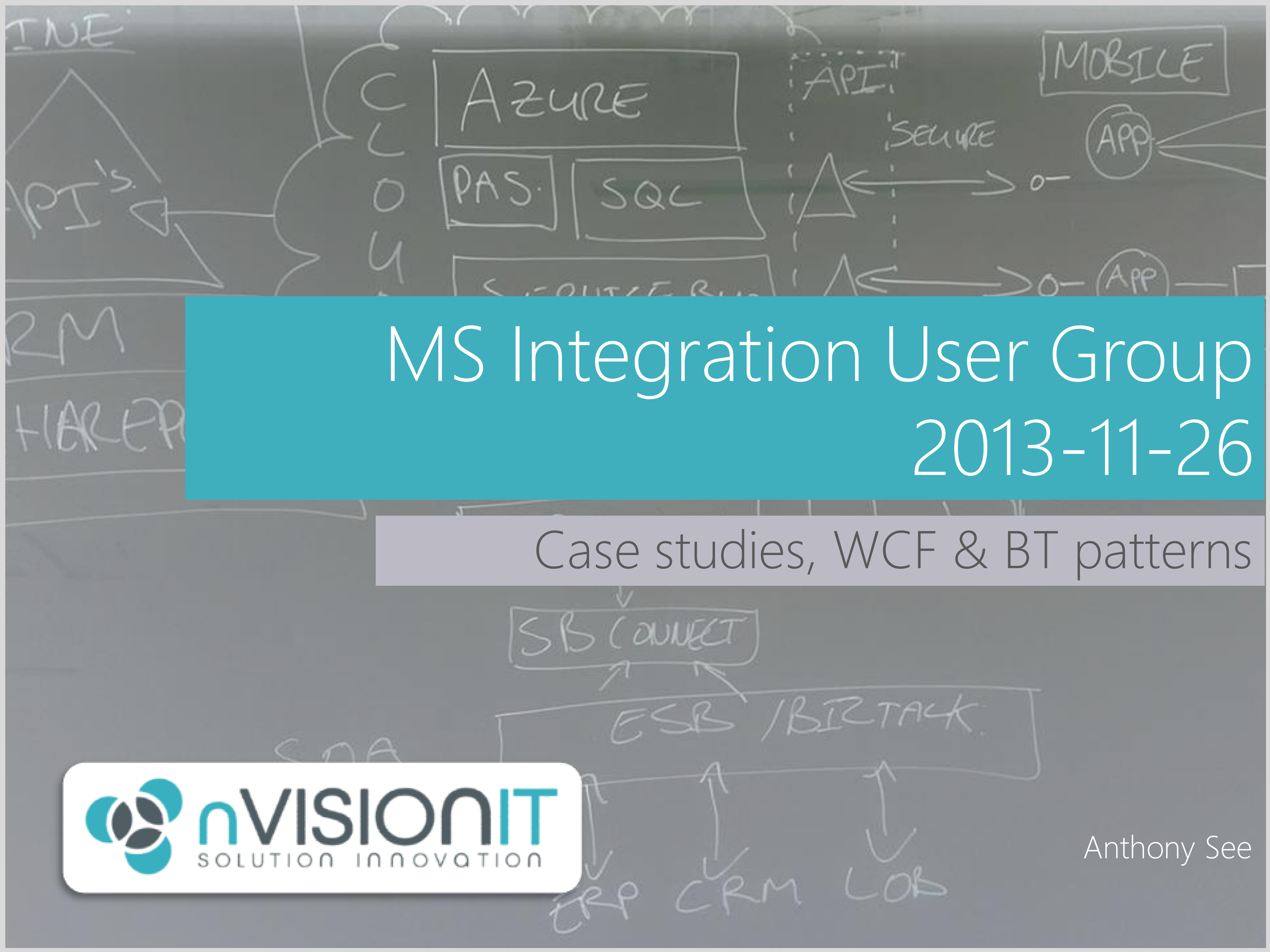


MS Integration User Group 2013-11-26

Case studies, WCF & BT patterns

nVISIONIT
SOLUTION INNOVATION

Anthony See

Introduction: nVisionIT

Time	Description
3:00 - 3:15	Welcome
3:15 – 3:45	Anthony See: Case Study and demo showcasing async message processing, schema inheritance, composite operations
3:45 – 4:00	Break
4:15 – 4:45	Moosa Gani: Utilizing Biztalk 2013 Enterprise capabilities in conjunction with Windows Communication Foundation Services to take advantage of scaling, load and message tracking
4:45 – 5:15	Deepul Sharma: BT Pattern is a tool that has pre-defined BT patterns based on best practices
5:15 – 5:30	Q &A
5:30 -	Drinks, pizza

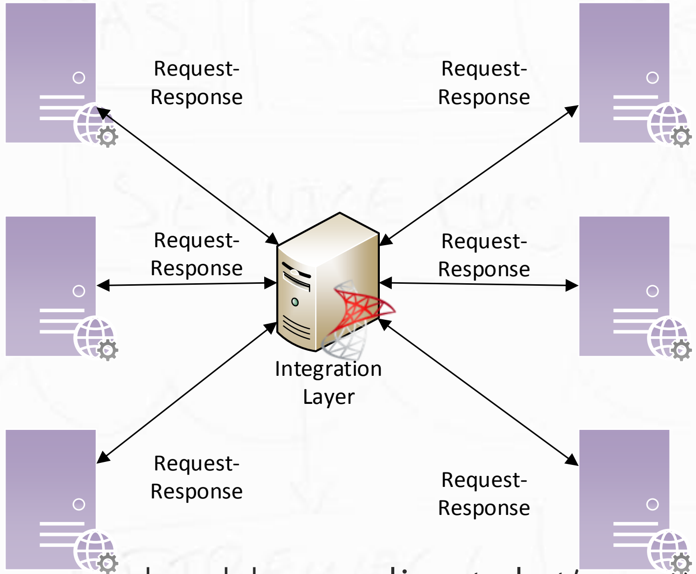
Agenda

- Part 1: Asynchronous (Async) Architecture Case Study
- Part 2: Schema Inheritance
- Part 3: Composite Operations in SQL (BizTalk)

Part1 – Agenda

- Async Architecture Case Study
  - Scenario
  - Observations
  - Issues
  - When to use?
    - Sync
  - Sync To Async Integration Pattern
  - When to use?
    - Async

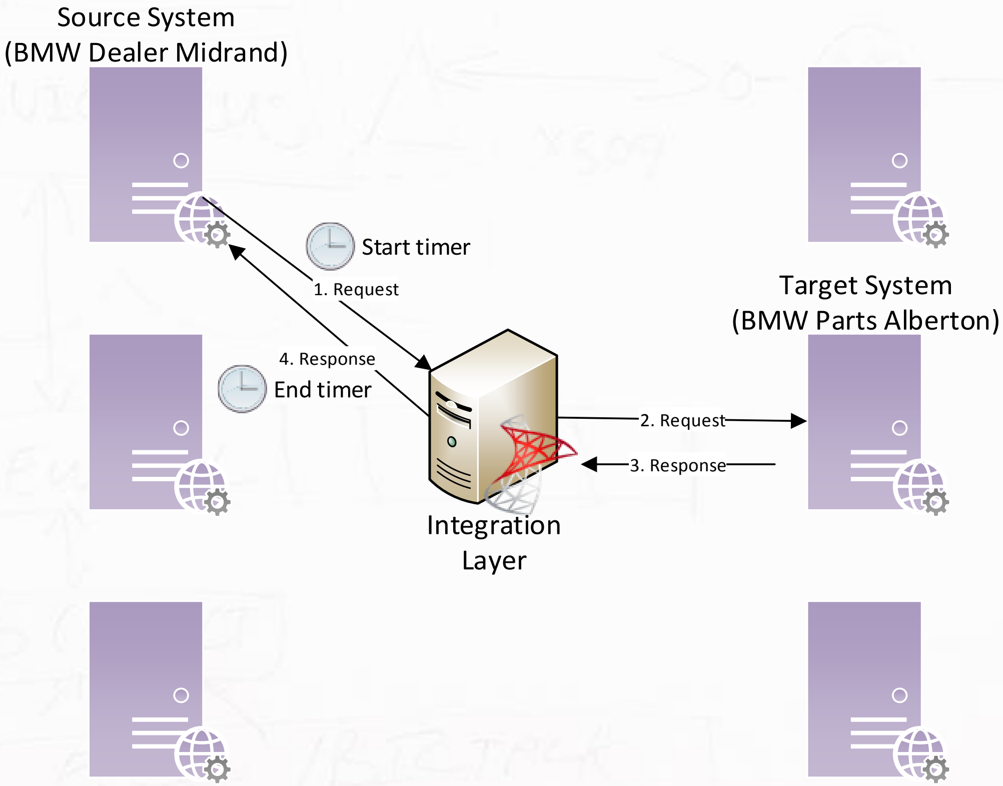
Case Study



- Scenario
  - nVisionIT was approached by a client, let's call them BMW.
  - Many external vendors sending ***different types*** of messages to many other external vendors through ***web services.*** (Many to many)
    - Message Types
      - Offer to purchase (Invoice Number)
      - Car Parts (Job Card)
  - Each request from *source system* to *target system*, requires a response from that *target system*.

Case Study

- Observations
  - Already an existing implementation of BizTalk 2006 using Synchronous Architecture
  - Total Response time (I.e. turn around time) varied from short to extremely long
    - Total Response time: Sum of (1, 2, 3, 4)
      - (1) Time taken of the request message from the source system to BizTalk.
      - (2) Time taken from BizTalk to the target system.
      - (3) Time taken for the response from the target system to BizTalk.
      - (4) Time taken of the response from BizTalk to the source system.



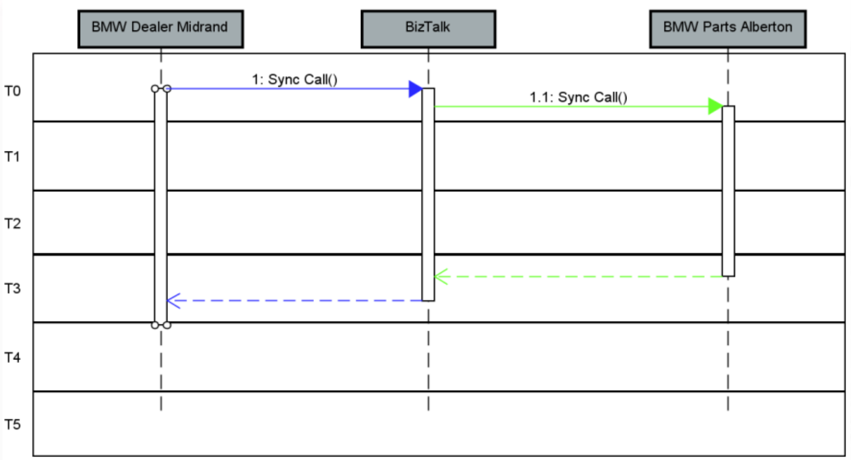
Sync Architecture Pattern

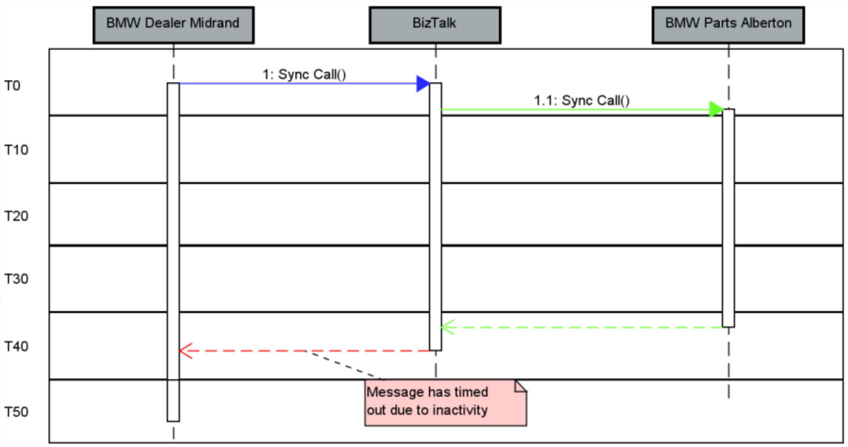
- Fast sync request and response scenario
  - Target system processes the request and returns the response quickly (< 30 seconds)
- Slow sync request and response scenario
  - Target system processes the request and takes long to process the response (> 30 seconds)
  - Web service times out

Synchronous Pattern

Fast sync request-response

Slow sync request-response





Synchronous Architecture Pattern

Client and External Vendors unhappy ☹

- Issues
  - Performance extremely bad
    - Total response time (Turnaround time) > 5 minutes
  - Unresponsive calls
    - Timeouts
    - *Apparent* lost messages
  - Tracking information - not available
    - Date and time a message received from a source system?
    - Date and time a message delivered to the target system?
    - Performance bottlenecks
      - Which processes are taking long?
- How above solved can be found at the end of the presentation

Synchronous Pattern Summary

When to use

- Real-time information is needed (E.g. lookups)
- Response time is < 30 seconds

When NOT to use

- Response time is unknown
- Target system has long running processes
  - Processing request message > 30 seconds
  - **may** be due to
    - processes that **take** long
    - dependent processes that **take** long
- Complex processes
  - Many target systems (External Providers) need to respond to a request

Advantages	Disadvantages
Easy to create and manage	Not suitable for long running processes
Simple - Only one web service required	Cannot handle more than 1 end point

Sync to Async Architecture Pattern - Agenda

- What is Async?
- 2 Methods of Implementing an Async Integration Architecture
- Common misconceptions about Async

Sync to Async Architecture Pattern

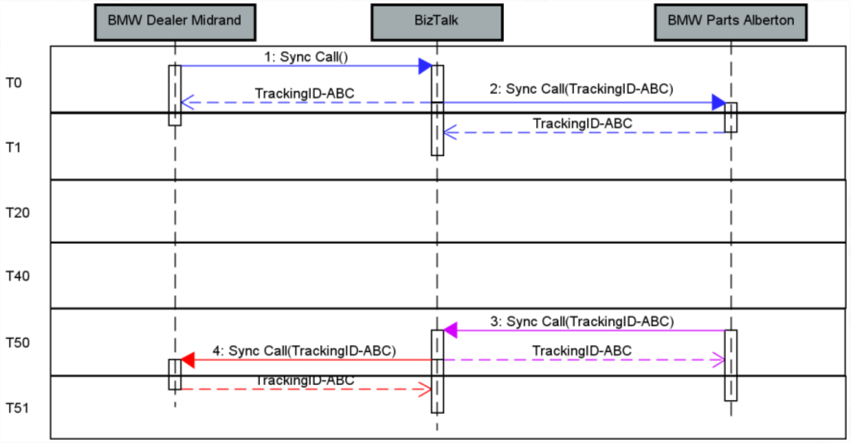
- What is an Async message?
  - Messages are ***not*** transmitted at a regular interval
  - At an unknown ***time*** a response will later be received

Sync to Async Pattern

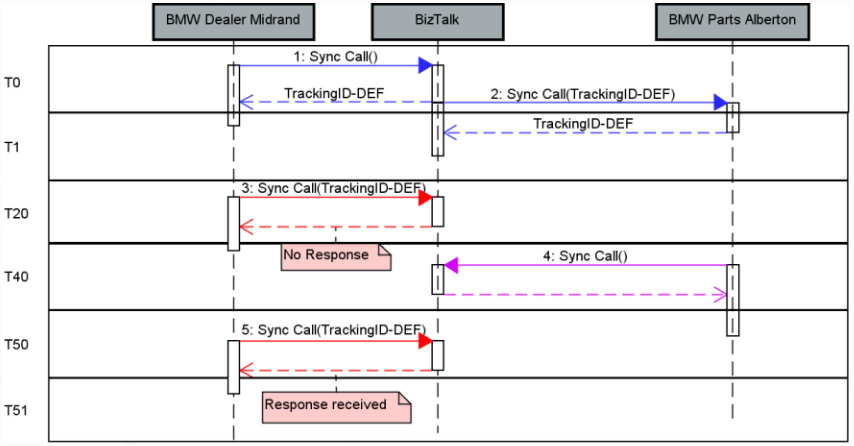
- Why is it Sync to Async?
  - Turning a sync call into an async call
    - Serves following purposes
      - TrackingID
      - Proof of receipt (Consisting of date and time)
- Two methods of implementing a Sync to Async
  - Callback (Code is executed and returned at some convenient time)
    - Requires a web service from the source system (BMW Dealer Midrand)
  - Polling (Source system/ Client makes continuous queries to determine the state of its request)
    - Requires a web service on the integration layer to handle the request for a particular status

Async types

Sync to Async Integration Pattern with call back



Sync to Async Integration Pattern with polling



Async Pattern

- Note
  - Sync to Async Integration Pattern is not a feature that comes out of the box with BizTalk
  - Architectural design decision that affects how integration between various systems will work

Async Pattern Summary

When to use

- Response time is unknown
- Bulk data processes
- Target system has long running processes
- Complex processes
  - Multiple vendors need to respond to the request

When NOT to use

- Simple architecture
- Real time responses (Lookups)

Advantages	Disadvantages
Processes can take as long as *necessary*	May take longer to implement as all parties need to be involved
Can handle sending to multiple systems	Complicated

Misconceptions

- Async is ***not*** always faster
  - Overheads with state management
    - New request
    - Processed
    - Response received yet?
    - Error
  - Source and Target System
    - Queuing mechanism
  - Web services required (Source / Target / Integration Layer)
    - Request
    - Response
    - Errors
  - Managing time and scheduling

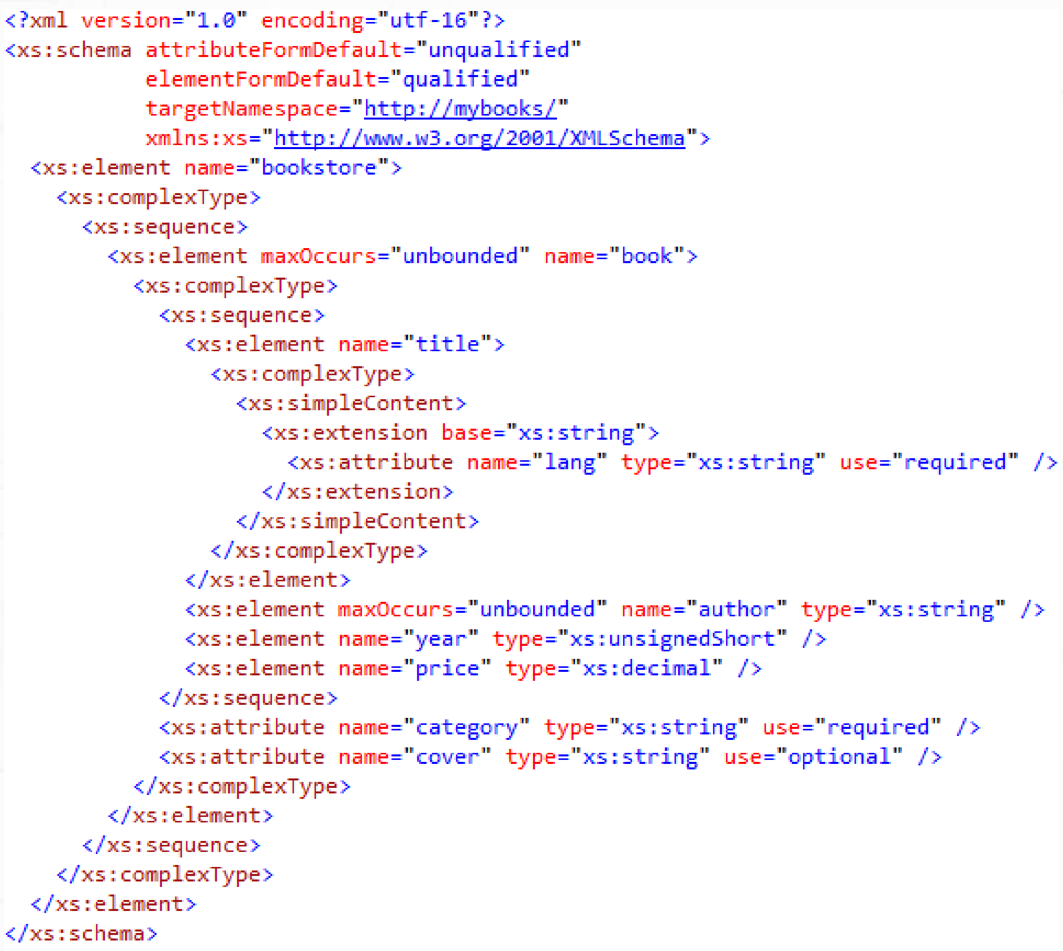
Part 2 - Schema Inheritance – Agenda

- Definition of a schema (W3C standard)
- Target namespace
- Schema Inheritance
    - Why should you use them?

Schema - Definition

W3C Standard

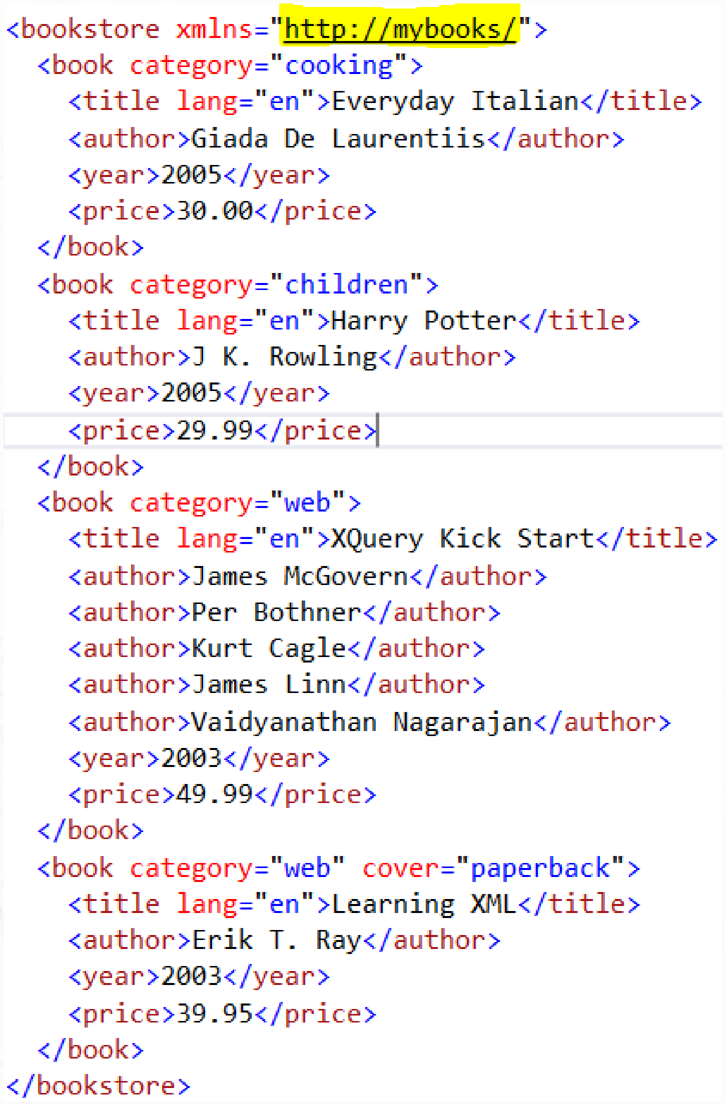
- Define the contract of a XML document
- Data types
  - string
  - int
  - decimal
  - Etc.



```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://mybooks/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bookstore">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="book">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="lang" type="xs:string" use="required" />
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element maxOccurs="unbounded" name="author" type="xs:string" />
              <xs:element name="year" type="xs:unsignedShort" />
              <xs:element name="price" type="xs:decimal" />
            </xs:sequence>
            <xs:attribute name="category" type="xs:string" use="required" />
            <xs:attribute name="cover" type="xs:string" use="optional" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Schema - Target namespace

- The target namespace identifies the namespace that components in this schema belongs to.
- If no target namespace is provided, then the schema components do not belong to any namespace.



```
<bookstore xmlns="http://mybooks/">
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

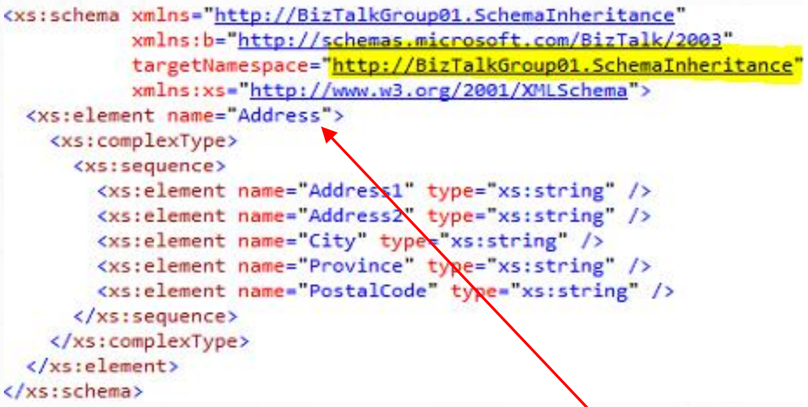
Schema Inheritance

Schemas allow you to make use of other schemas (similar to inheritance in programming)

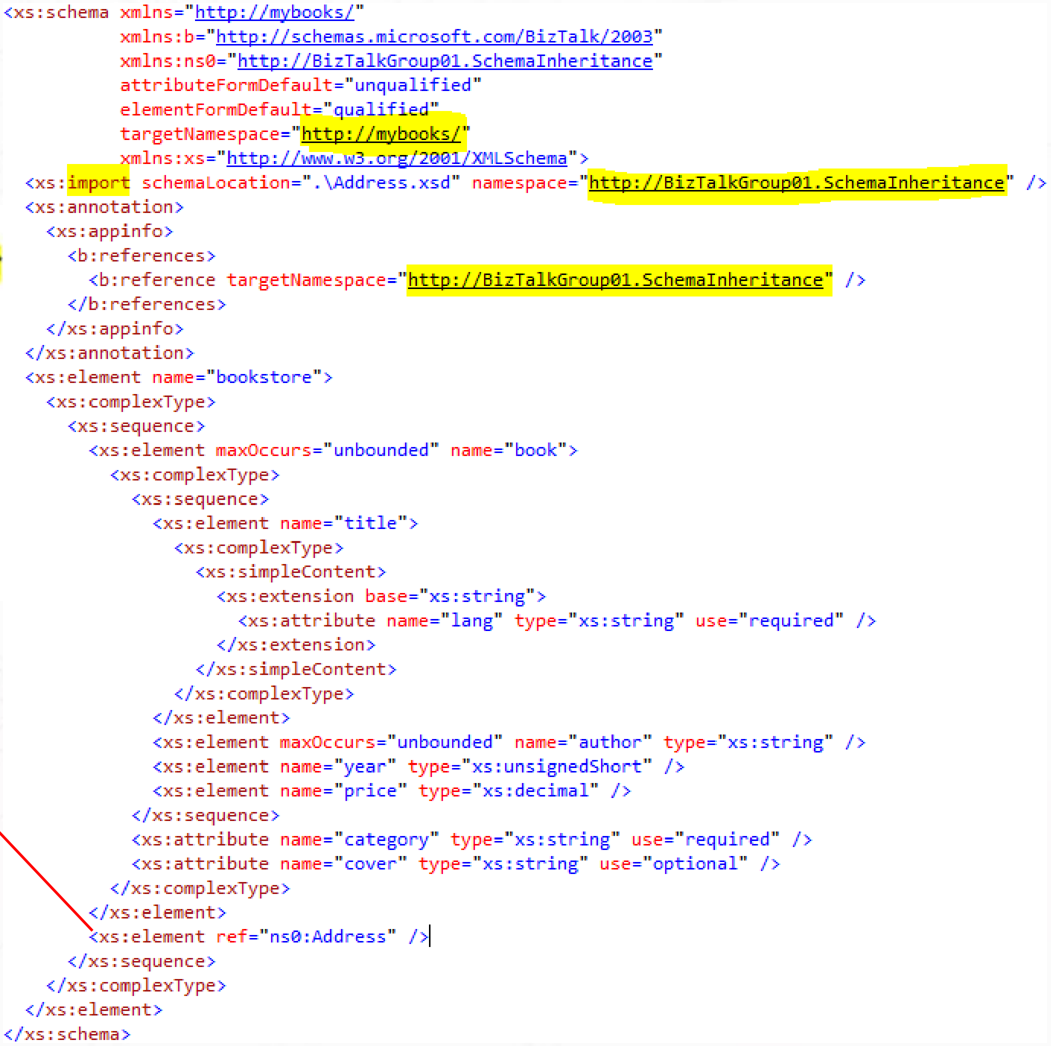
- The following are types of inheritance that schemas can perform. (Not BizTalk specific)
  - Schema Imports
  - Schema Include
  - Schema Redefine

Schema Import

- Target namespace must be DIFFERENT



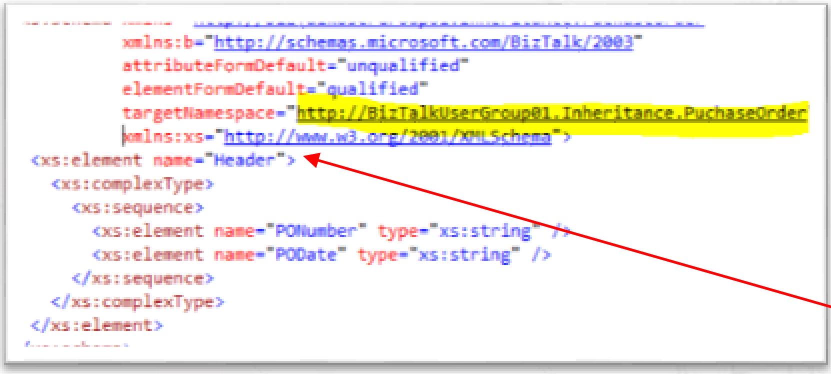
```
<xs:schema xmlns="http://BizTalkGroup01.SchemaInheritance"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           targetNamespace="http://BizTalkGroup01.SchemaInheritance"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Address1" type="xs:string" />
        <xs:element name="Address2" type="xs:string" />
        <xs:element name="City" type="xs:string" />
        <xs:element name="Province" type="xs:string" />
        <xs:element name="PostalCode" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



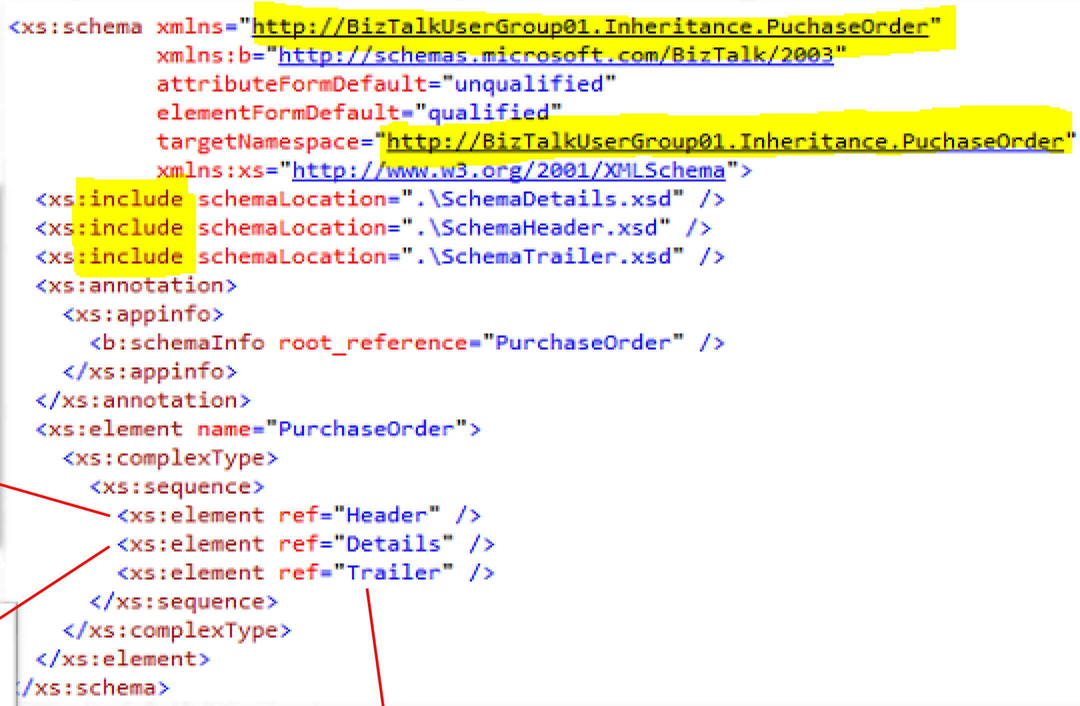
```
<xs:schema xmlns="http://mybooks/"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           xmlns:ns0="http://BizTalkGroup01.SchemaInheritance"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://mybooks/"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import schemaLocation=".\Address.xsd" namespace="http://BizTalkGroup01.SchemaInheritance" />
  <xs:annotation>
    <xs:appinfo>
      <b:references>
        <b:reference targetNamespace="http://BizTalkGroup01.SchemaInheritance" />
      </b:references>
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="bookstore">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="book">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="lang" type="xs:string" use="required" />
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element maxOccurs="unbounded" name="author" type="xs:string" />
              <xs:element name="year" type="xs:unsignedShort" />
              <xs:element name="price" type="xs:decimal" />
            </xs:sequence>
            <xs:attribute name="category" type="xs:string" use="required" />
            <xs:attribute name="cover" type="xs:string" use="optional" />
          </xs:complexType>
        </xs:element>
        <xs:element ref="ns0:Address" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Schema Include

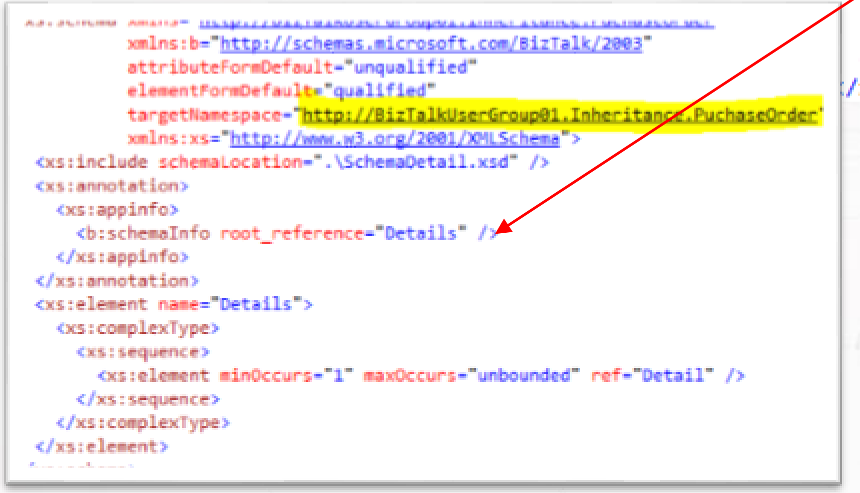
- Target namespace must be the SAME



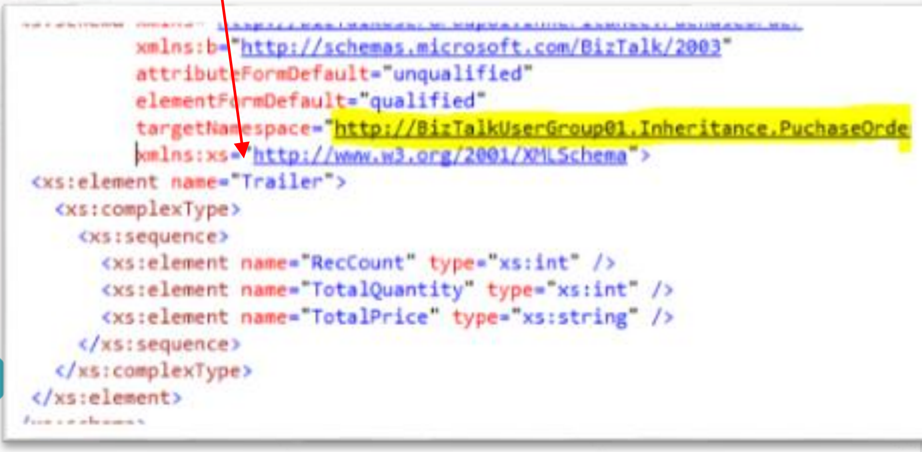
```
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.PuchaseOrder
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Header">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PONumber" type="xs:string" />
        <xs:element name="PODate" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```



```
<xs:schema xmlns="http://BizTalkUserGroup01.Inheritance.PuchaseOrder"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.PuchaseOrder"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation=".\SchemaDetails.xsd" />
  <xs:include schemaLocation=".\SchemaHeader.xsd" />
  <xs:include schemaLocation=".\SchemaTrailer.xsd" />
  <xs:annotation>
    <xs:appinfo>
      <b:schemaInfo root_reference="PurchaseOrder" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="PurchaseOrder">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Header" />
        <xs:element ref="Details" />
        <xs:element ref="Trailer" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



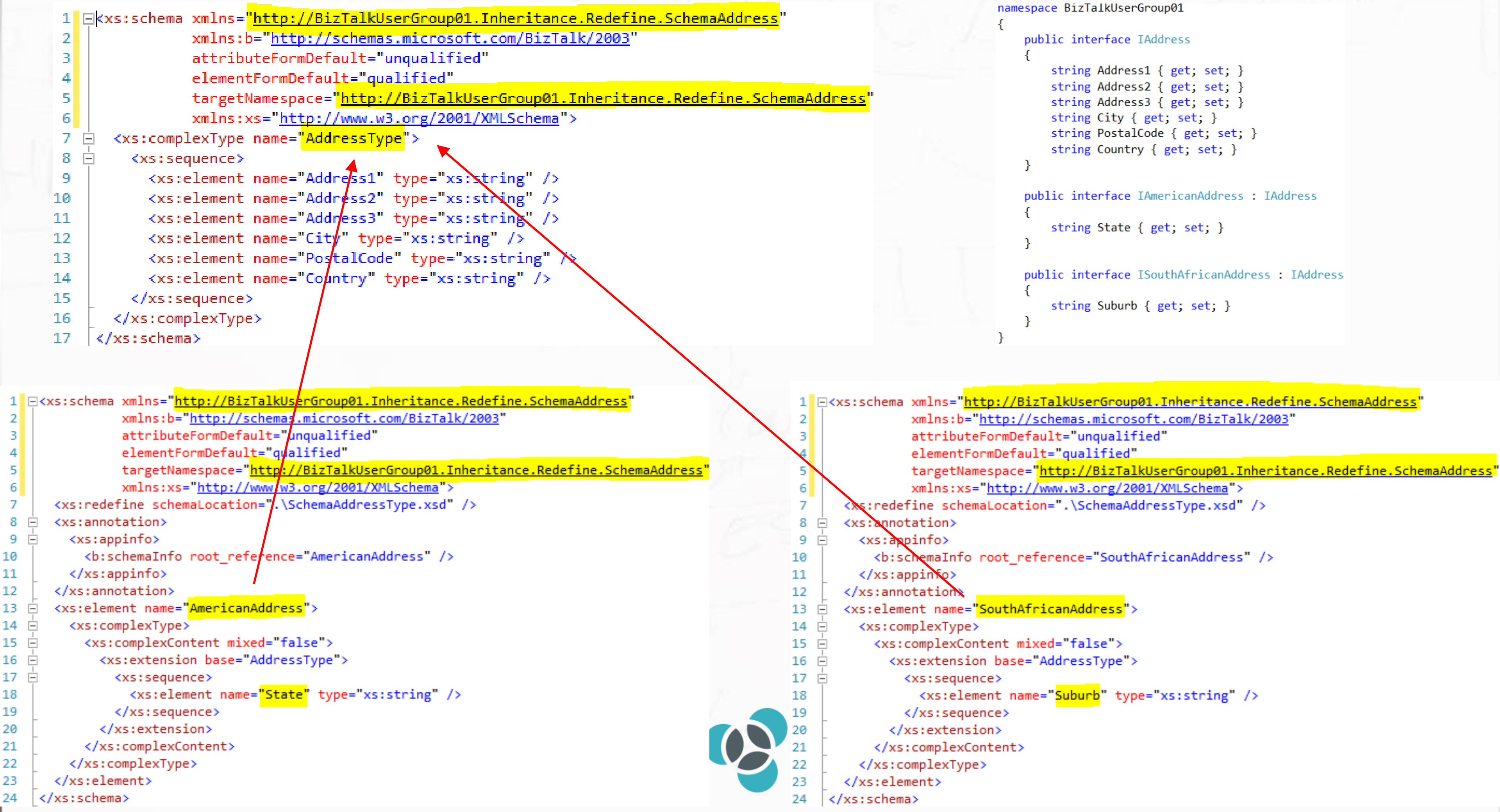
```
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.PuchaseOrder"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation=".\SchemaDetail.xsd" />
  <xs:annotation>
    <xs:appinfo>
      <b:schemaInfo root_reference="Details" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="Details">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="Detail" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```



```
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.PuchaseOrde
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Trailer">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RecCount" type="xs:int" />
        <xs:element name="TotalQuantity" type="xs:int" />
        <xs:element name="TotalPrice" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

Schema Redefine

- Target namespace must be the SAME
- Importing types from another schema with the intent on extending/overriding the types



```xml
<xs:schema xmlns="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="AddressType">
    <xs:sequence>
      <xs:element name="Address1" type="xs:string" />
      <xs:element name="Address2" type="xs:string" />
      <xs:element name="Address3" type="xs:string" />
      <xs:element name="City" type="xs:string" />
      <xs:element name="PostalCode" type="xs:string" />
      <xs:element name="Country" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

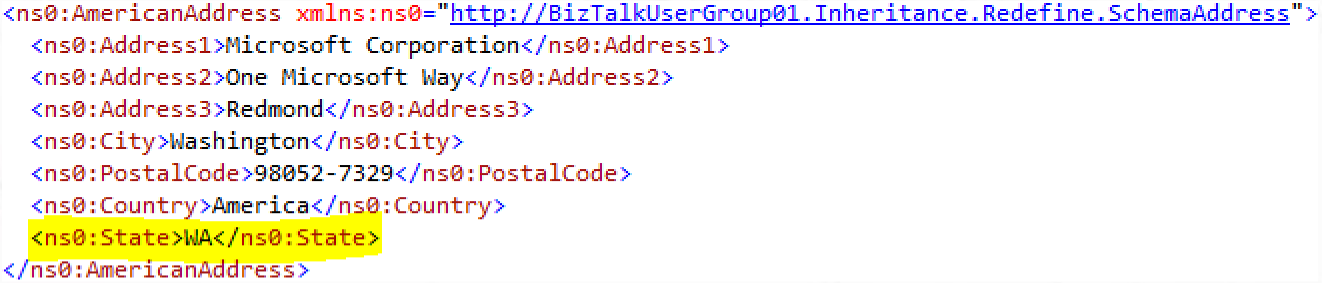
```csharp
namespace BizTalkUserGroup01
{
    public interface IAddress
    {
        string Address1 { get; set; }
        string Address2 { get; set; }
        string Address3 { get; set; }
        string City { get; set; }
        string PostalCode { get; set; }
        string Country { get; set; }
    }

    public interface IAmericanAddress : IAddress
    {
        string State { get; set; }
    }

    public interface ISouthAfricanAddress : IAddress
    {
        string Suburb { get; set; }
    }
}
```

```xml
<xs:schema xmlns="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:redefine schemaLocation=".\SchemaAddressType.xsd" />
  <xs:annotation>
    <xs:appinfo>
      <b:schemaInfo root_reference="AmericanAddress" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="AmericanAddress">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="AddressType">
          <xs:sequence>
            <xs:element name="State" type="xs:string" />
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

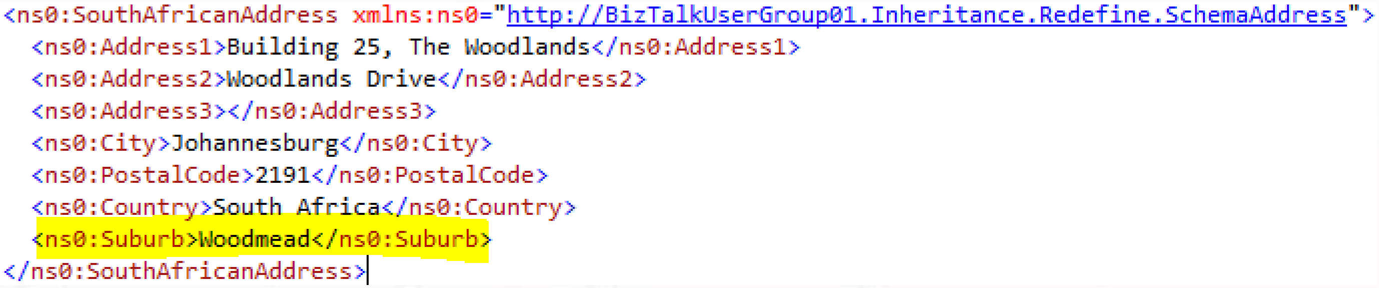
```xml
<xs:schema xmlns="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:b="http://schemas.microsoft.com/BizTalk/2003"
           attributeFormDefault="unqualified"
           elementFormDefault="qualified"
           targetNamespace="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:redefine schemaLocation=".\SchemaAddressType.xsd" />
  <xs:annotation>
    <xs:appinfo>
      <b:schemaInfo root_reference="SouthAfricanAddress" />
    </xs:appinfo>
  </xs:annotation>
  <xs:element name="SouthAfricanAddress">
    <xs:complexType>
      <xs:complexContent mixed="false">
        <xs:extension base="AddressType">
          <xs:sequence>
            <xs:element name="Suburb" type="xs:string" />
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Examples

- Schema Redefine



```
<ns0:AmericanAddress xmlns:ns0="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress">
  <ns0:Address1>Microsoft Corporation</ns0:Address1>
  <ns0:Address2>One Microsoft Way</ns0:Address2>
  <ns0:Address3>Redmond</ns0:Address3>
  <ns0:City>Washington</ns0:City>
  <ns0:PostalCode>98052-7329</ns0:PostalCode>
  <ns0:Country>America</ns0:Country>
  <ns0:State>WA</ns0:State>
</ns0:AmericanAddress>
```



```
<ns0:SouthAfricanAddress xmlns:ns0="http://BizTalkUserGroup01.Inheritance.Redefine.SchemaAddress">
  <ns0:Address1>Building 25, The Woodlands</ns0:Address1>
  <ns0:Address2>Woodlands Drive</ns0:Address2>
  <ns0:Address3></ns0:Address3>
  <ns0:City>Johannesburg</ns0:City>
  <ns0:PostalCode>2191</ns0:PostalCode>
  <ns0:Country>South Africa</ns0:Country>
  <ns0:Suburb>Woodmead</ns0:Suburb>
</ns0:SouthAfricanAddress>
```

Schema Inheritance - Summary

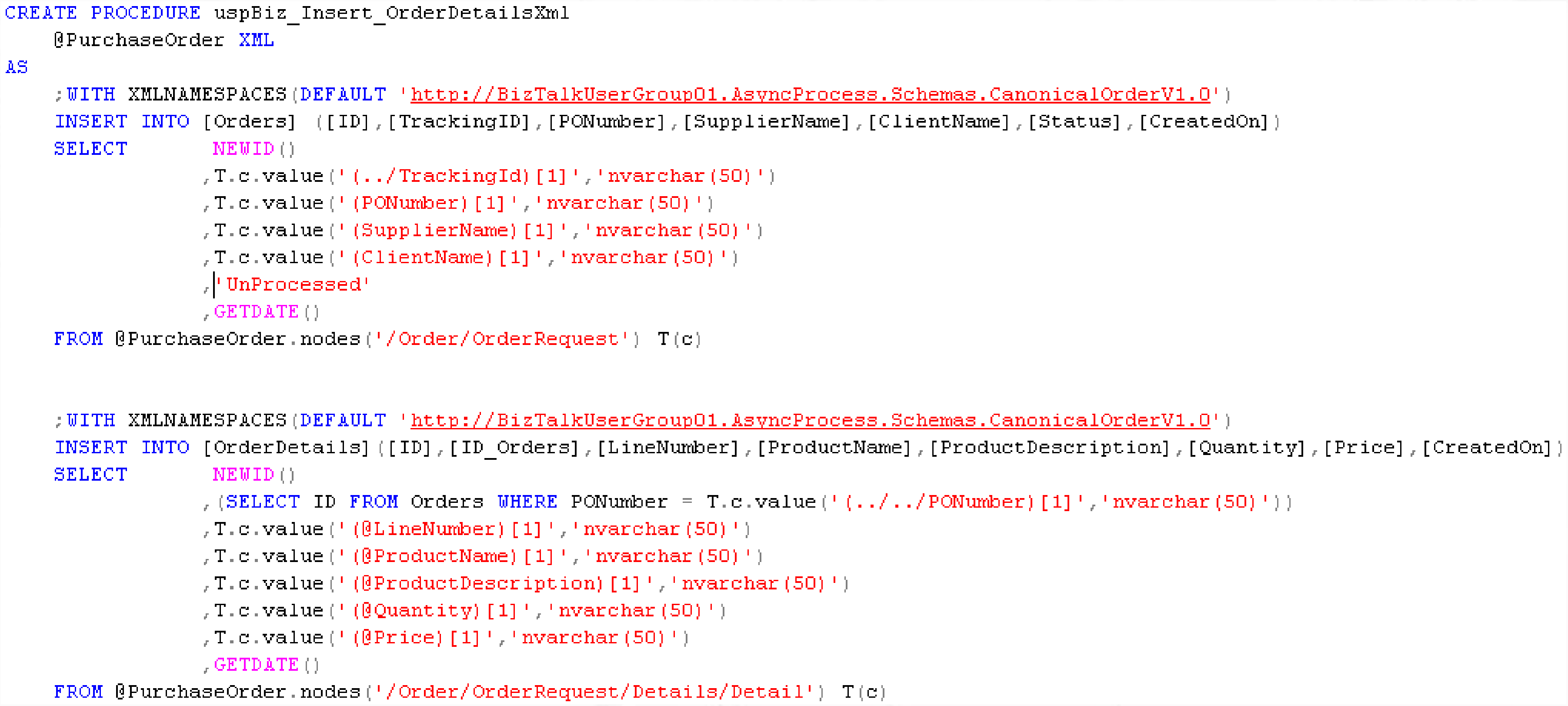
- Where should you use them?
  - Repetitive and common structures (Address, Phone Numbers etc.)
    - Similar to coding
      - base class
      - Super class
  - Imports - Complicated structures
  - Include - Envelopes
- Why should you use them?
  - Increase development time
  - Reduce mistakes (Inherited data types)
- Warning
  - Redefine does not work with XSD.EXE tool

Part 3 Composite Operations on SQL database

- Supports combination of operations as a single action
- Transactional
  - Failure
    - Rollback on all operations
- Where would you use Composite Operations?
  - Bulk inserts
  - Auditing
  - CRUD

Why use composite operations?

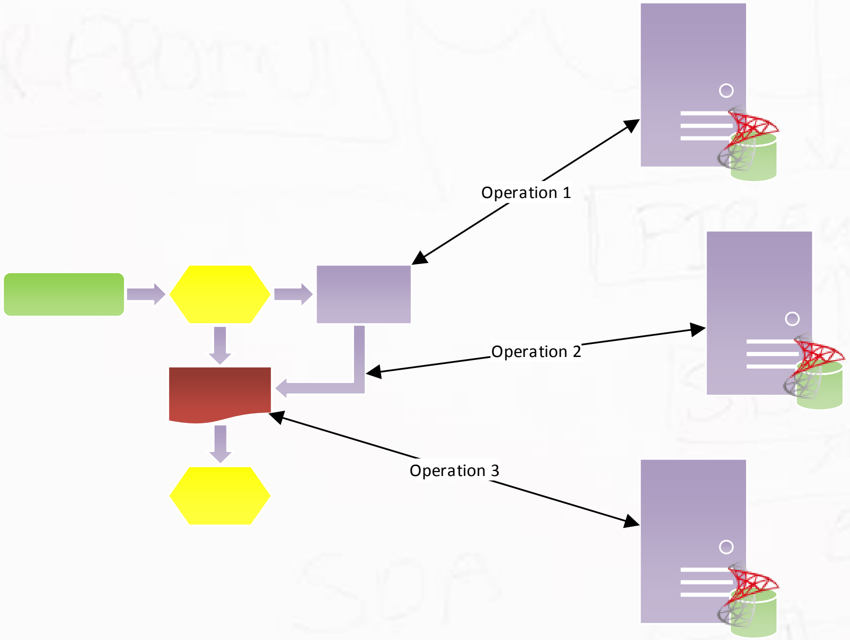
- Passing XML as a parameter to a stored procedure



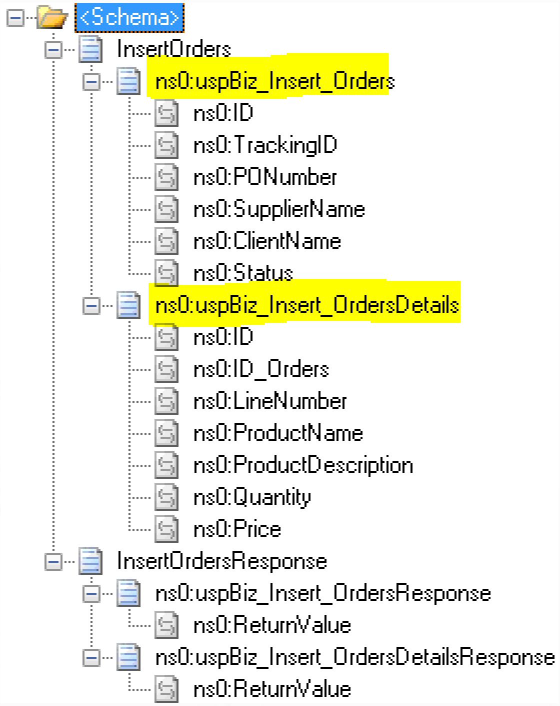
```sql
CREATE PROCEDURE uspBiz_Insert_OrderDetailsXml
    @PurchaseOrder XML
AS
    ;WITH XMLNAMESPACES(DEFAULT 'http://BizTalkUserGroup01.AsyncProcess.Schemas.CanonicalOrderV1.0')
    INSERT INTO [Orders]  ([ID],[TrackingID],[PONumber],[SupplierName],[ClientName],[Status],[CreatedOn])
    SELECT       NEWID()
                ,T.c.value('(../TrackingId)[1]','nvarchar(50)')
                ,T.c.value('(PONumber)[1]','nvarchar(50)')
                ,T.c.value('(SupplierName)[1]','nvarchar(50)')
                ,T.c.value('(ClientName)[1]','nvarchar(50)')
                ,'UnProcessed'
                ,GETDATE()
    FROM @PurchaseOrder.nodes('/Order/OrderRequest')  T(c)


    ;WITH XMLNAMESPACES(DEFAULT 'http://BizTalkUserGroup01.AsyncProcess.Schemas.CanonicalOrderV1.0')
    INSERT INTO [OrderDetails]([ID],[ID_Orders],[LineNumber],[ProductName],[ProductDescription],[Quantity],[Price],[CreatedOn])
    SELECT       NEWID()
                ,(SELECT ID FROM Orders WHERE PONumber = T.c.value('(../../PONumber)[1]','nvarchar(50)'))
                ,T.c.value('(@LineNumber)[1]','nvarchar(50)')
                ,T.c.value('(@ProductName)[1]','nvarchar(50)')
                ,T.c.value('(@ProductDescription)[1]','nvarchar(50)')
                ,T.c.value('(@Quantity)[1]','nvarchar(50)')
                ,T.c.value('(@Price)[1]','nvarchar(50)')
                ,GETDATE()
    FROM @PurchaseOrder.nodes('/Order/OrderRequest/Details/Detail')  T(c)
```

Why use composite operations?

Orchestration to handle SQL events



Single Schema to handle same logic



- <Schema>
  - InsertOrders
    - ns0:uspBiz_Insert_Orders
      - ns0:ID
      - ns0:TrackingID
      - ns0:PONumber
      - ns0:SupplierName
      - ns0:ClientName
      - ns0:Status
    - ns0:uspBiz_Insert_OrdersDetails
      - ns0:ID
      - ns0:ID_Orders
      - ns0:LineNumber
      - ns0:ProductName
      - ns0:ProductDescription
      - ns0:Quantity
      - ns0:Price
  - InsertOrdersResponse
    - ns0:uspBiz_Insert_OrdersResponse
      - ns0:ReturnValue
    - ns0:uspBiz_Insert_OrdersDetailsResponse
      - ns0:ReturnValue

Composite Operations on SQL database

- Available from BizTalk 2009 onwards
  - WCF-based SQL Adapter
  - WCF-Custom SQL Adapter
- Supports combination of
  - Insert
  - Update
  - Delete
  - Stored Procedures
  - Select (not supported)

Tools for the demo

- Tools
  - BizTalk 2010
  - SQL Server 2008R2
  - SoapUI (http://www.SoapUI.org)
    - Open source Functional Testing Tool for API Testing

- Links
  - Schema Inheritance (http://msdn.microsoft.com/en-us/library/ee254473(v=bts.10).aspx)
  - Composite Operations (http://msdn.microsoft.com/en-us/library/dd788487.aspx)

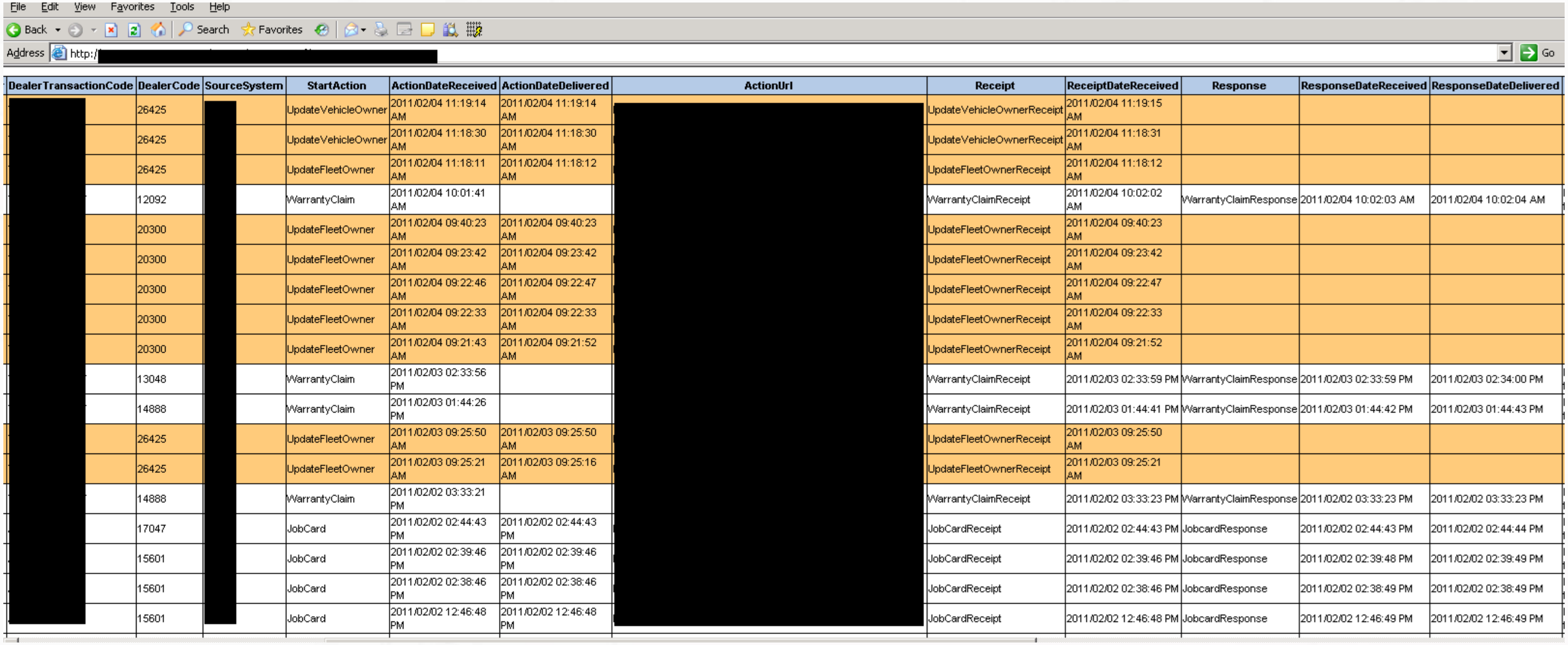
Synchronous Architecture Pattern (Solved)

Client and External Vendors unhappy ☹

- Issues
  - Performance extremely bad
    - Total response time > 5 minutes
    - Sync to Async
  - Unresponsive calls
    - Timeouts
    - *Apparent* lost messages
    - Sync to Async with Tracking ID (Receipt of message)
  - Tracking information - not available
    - Date and time a message from a source sent?
    - Date and time a message delivered to the target system?
    - Performance bottlenecks
    - Business Activity Monitoring (BAM)

Synchronous Architecture Pattern (Solved)

- Tracking

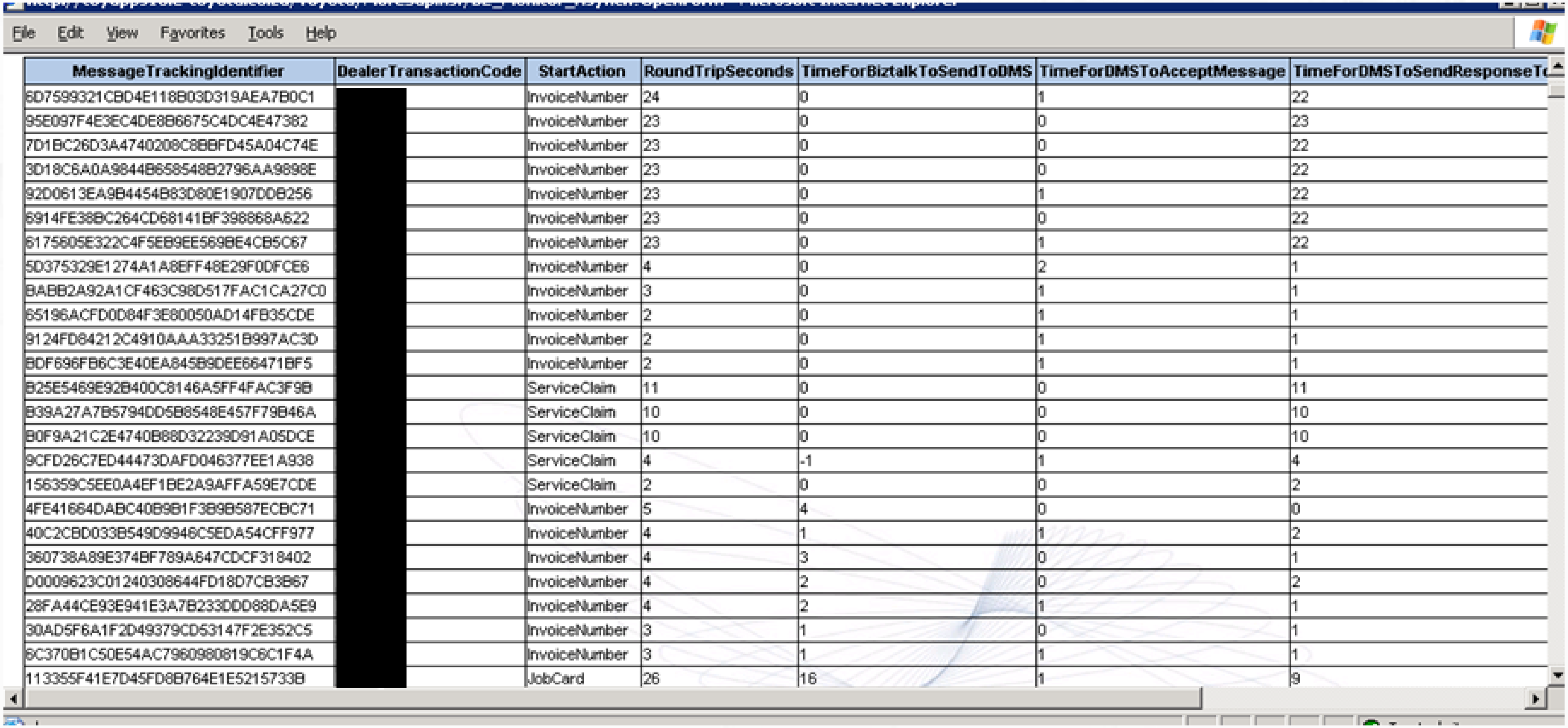


Address http:/[redacted]

DealerTransactionCode	DealerCode	SourceSystem	StartAction	ActionDateReceived	ActionDateDelivered	ActionUrl	Receipt	ReceiptDateReceived	Response	ResponseDateReceived	ResponseDateDelivered
[redacted]	26425	[redacted]	UpdateVehicleOwner	2011/02/04 11:19:14 AM	2011/02/04 11:19:14 AM	[redacted]	UpdateVehicleOwnerReceipt	2011/02/04 11:19:15 AM			
[redacted]	26425	[redacted]	UpdateVehicleOwner	2011/02/04 11:18:30 AM	2011/02/04 11:18:30 AM	[redacted]	UpdateVehicleOwnerReceipt	2011/02/04 11:18:31 AM			
[redacted]	26425	[redacted]	UpdateFleetOwner	2011/02/04 11:18:11 AM	2011/02/04 11:18:12 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 11:18:12 AM			
[redacted]	12092	[redacted]	WarrantyClaim	2011/02/04 10:01:41 AM		[redacted]	WarrantyClaimReceipt	2011/02/04 10:02:02 AM	WarrantyClaimResponse	2011/02/04 10:02:03 AM	2011/02/04 10:02:04 AM
[redacted]	20300	[redacted]	UpdateFleetOwner	2011/02/04 09:40:23 AM	2011/02/04 09:40:23 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 09:40:23 AM			
[redacted]	20300	[redacted]	UpdateFleetOwner	2011/02/04 09:23:42 AM	2011/02/04 09:23:42 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 09:23:42 AM			
[redacted]	20300	[redacted]	UpdateFleetOwner	2011/02/04 09:22:46 AM	2011/02/04 09:22:47 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 09:22:47 AM			
[redacted]	20300	[redacted]	UpdateFleetOwner	2011/02/04 09:22:33 AM	2011/02/04 09:22:33 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 09:22:33 AM			
[redacted]	20300	[redacted]	UpdateFleetOwner	2011/02/04 09:21:43 AM	2011/02/04 09:21:52 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/04 09:21:52 AM			
[redacted]	13048	[redacted]	WarrantyClaim	2011/02/03 02:33:56 PM		[redacted]	WarrantyClaimReceipt	2011/02/03 02:33:59 PM	WarrantyClaimResponse	2011/02/03 02:33:59 PM	2011/02/03 02:34:00 PM
[redacted]	14888	[redacted]	WarrantyClaim	2011/02/03 01:44:26 PM		[redacted]	WarrantyClaimReceipt	2011/02/03 01:44:41 PM	WarrantyClaimResponse	2011/02/03 01:44:42 PM	2011/02/03 01:44:43 PM
[redacted]	26425	[redacted]	UpdateFleetOwner	2011/02/03 09:25:50 AM	2011/02/03 09:25:50 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/03 09:25:50 AM			
[redacted]	26425	[redacted]	UpdateFleetOwner	2011/02/03 09:25:21 AM	2011/02/03 09:25:16 AM	[redacted]	UpdateFleetOwnerReceipt	2011/02/03 09:25:21 AM			
[redacted]	14888	[redacted]	WarrantyClaim	2011/02/02 03:33:21 PM		[redacted]	WarrantyClaimReceipt	2011/02/02 03:33:23 PM	WarrantyClaimResponse	2011/02/02 03:33:23 PM	2011/02/02 03:33:23 PM
[redacted]	17047	[redacted]	JobCard	2011/02/02 02:44:43 PM	2011/02/02 02:44:43 PM	[redacted]	JobCardReceipt	2011/02/02 02:44:43 PM	JobcardResponse	2011/02/02 02:44:43 PM	2011/02/02 02:44:44 PM
[redacted]	15601	[redacted]	JobCard	2011/02/02 02:39:46 PM	2011/02/02 02:39:46 PM	[redacted]	JobCardReceipt	2011/02/02 02:39:46 PM	JobcardResponse	2011/02/02 02:39:48 PM	2011/02/02 02:39:49 PM
[redacted]	15601	[redacted]	JobCard	2011/02/02 02:38:46 PM	2011/02/02 02:38:46 PM	[redacted]	JobCardReceipt	2011/02/02 02:38:46 PM	JobcardResponse	2011/02/02 02:38:49 PM	2011/02/02 02:38:49 PM
[redacted]	15601	[redacted]	JobCard	2011/02/02 12:46:48 PM	2011/02/02 12:46:48 PM	[redacted]	JobCardReceipt	2011/02/02 12:46:48 PM	JobcardResponse	2011/02/02 12:46:49 PM	2011/02/02 12:46:49 PM

Synchronous Architecture Pattern (Solved)

- Performance bottlenecks



MessageTrackingIdentifier	DealerTransactionCode	StartAction	RoundTripSeconds	TimeForBiztalkToSendToDMS	TimeForDMSToAcceptMessage	TimeForDMSToSendResponseT
6D7599321CBD4E118B03D319AEA7B0C1	[redacted]	InvoiceNumber	24	0	1	22
95E097F4E3EC4DE8B6675C4DC4E47382	[redacted]	InvoiceNumber	23	0	0	23
7D1BC26D3A4740208C8BBFD45A04C74E	[redacted]	InvoiceNumber	23	0	0	22
3D18C6A0A9844B658548B2796AA9898E	[redacted]	InvoiceNumber	23	0	0	22
92D0613EA9B4454B83D80E1907DDB256	[redacted]	InvoiceNumber	23	0	1	22
6914FE38BC264CD68141BF398868A622	[redacted]	InvoiceNumber	23	0	0	22
6175605E322C4F5EB9EE569BE4CB5C67	[redacted]	InvoiceNumber	23	0	1	22
5D375329E1274A1A8EFF48E29F0DFCE6	[redacted]	InvoiceNumber	4	0	2	1
BABB2A92A1CF463C98D517FAC1CA27C0	[redacted]	InvoiceNumber	3	0	1	1
65196ACFD0D84F3E80050AD14FB35CDE	[redacted]	InvoiceNumber	2	0	1	1
9124FD84212C4910AAA33251B997AC3D	[redacted]	InvoiceNumber	2	0	1	1
BDF696FB6C3E40EA845B9DEE66471BF5	[redacted]	InvoiceNumber	2	0	1	1
B25E5469E92B400C8146A5FF4FAC3F9B	[redacted]	ServiceClaim	11	0	0	11
B39A27A7B5794DD5B8548E457F79B46A	[redacted]	ServiceClaim	10	0	0	10
B0F9A21C2E4740B88D32239D91A05DCE	[redacted]	ServiceClaim	10	0	0	10
9CFD26C7ED44473DAFD046377EE1A938	[redacted]	ServiceClaim	4	-1	1	4
156359C5EE0A4EF1BE2A9AFFA59E7CDE	[redacted]	ServiceClaim	2	0	0	2
4FE41664DABC40B9B1F3B9B587ECBC71	[redacted]	InvoiceNumber	5	4	0	0
40C2CBD033B549D9946C5EDA54CFF977	[redacted]	InvoiceNumber	4	1	1	2
360738A89E374BF789A647CDCF318402	[redacted]	InvoiceNumber	4	3	0	1
D0009623C01240308644FD18D7CB3B67	[redacted]	InvoiceNumber	4	2	0	2
28FA44CE93E941E3A7B233DDD88DA5E9	[redacted]	InvoiceNumber	4	2	1	1
30AD5F6A1F2D49379CD53147F2E352C5	[redacted]	InvoiceNumber	3	1	0	1
6C370B1C50E54AC7960980819C6C1F4A	[redacted]	InvoiceNumber	3	1	1	1
113355F41E7D45FD8B764E1E5215733B	[redacted]	JobCard	26	16	1	9