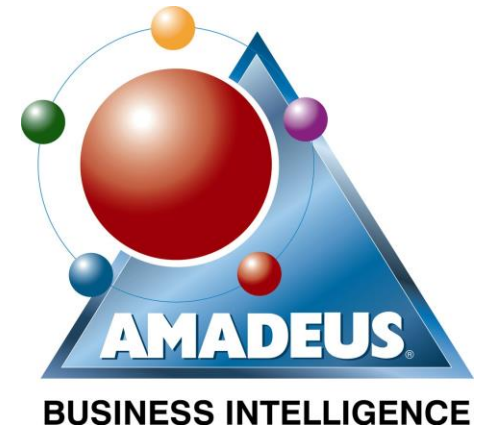




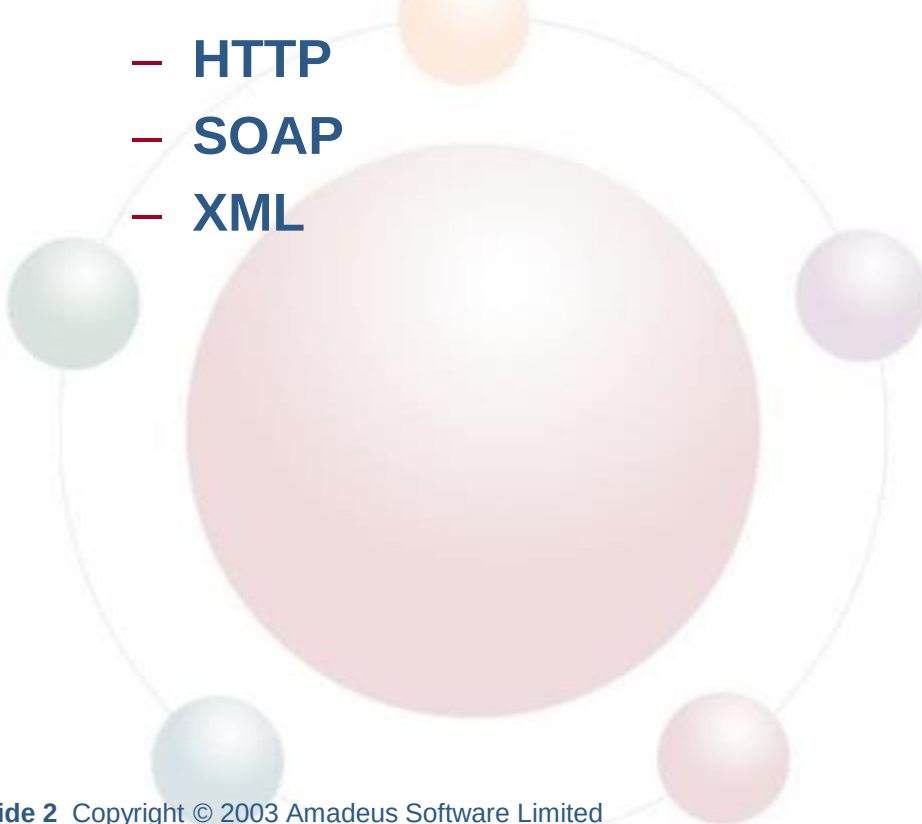
Building SAS Based Web Services

Jonathan Williamson
Amadeus Software Limited
20th May 2003



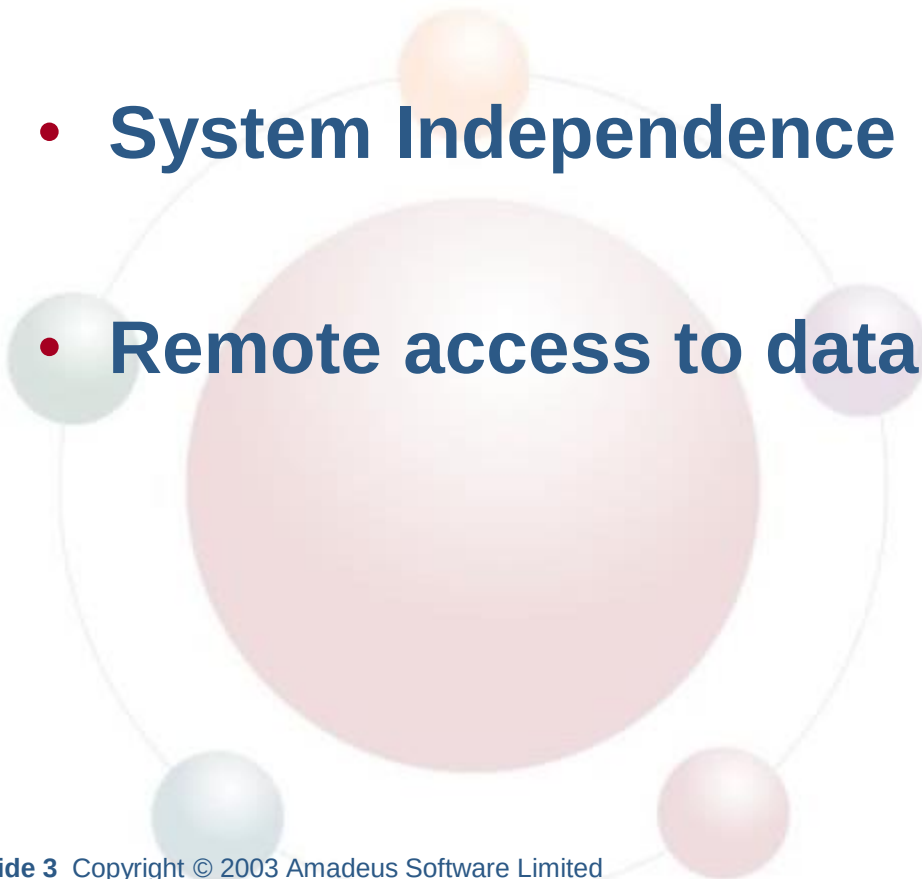
What Are Web Services?

- **Generalised Communications Model**
- **Utilise standard Internet protocols**
 - HTTP
 - SOAP
 - XML

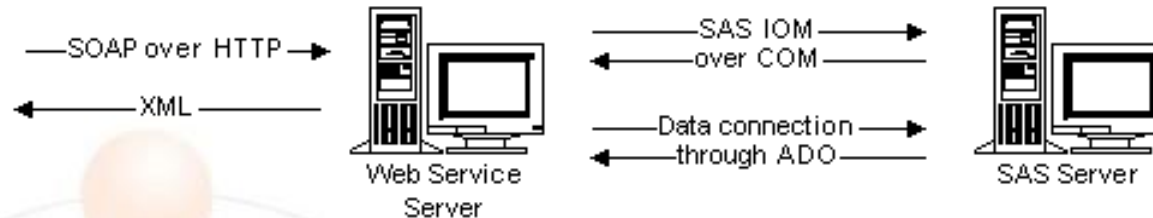


How Do They Help Us?

- **Data Storage Independence**
- **System Independence**
- **Remote access to data and processing power**



How Can We Build a SAS Web Service



- **Microsoft .NET Framework (Visual Basic)**
 - Web Service Interface
- **SAS Integrated Object Model (IOM)**
 - Access and Control SAS
- **ActiveX Data Objects (ADO.NET)**
 - Data Retrieval

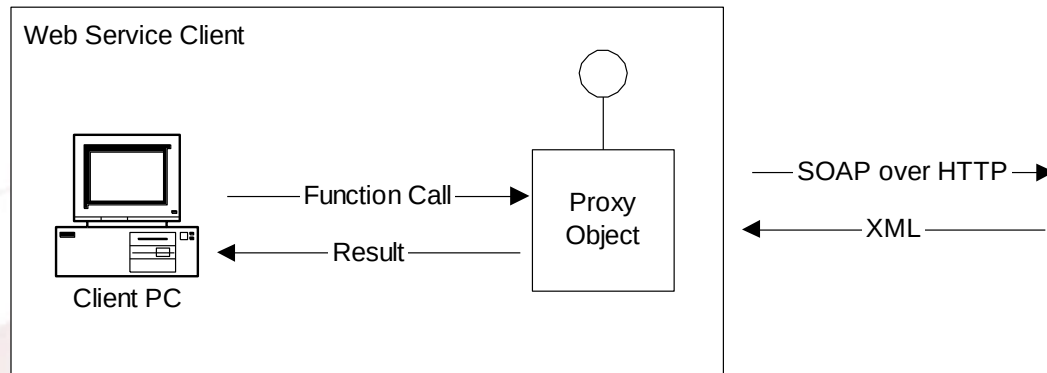
Web Service Code

```
<%@ WebService Language="vb"  
    Codebehind="Inventory.aspx.vb"  
    Class="SasStockController.SimpleInventory" %>
```

```
Public Class SimpleInventory  
    Inherits System.Web.Services.WebService  
  
    <WebMethod()> _  
    Public Function GetStockLevels( _  
        ByVal supplierRef As Integer) _  
        As DataSet  
        ' Function Implementation  
    End Function  
  
End Class
```

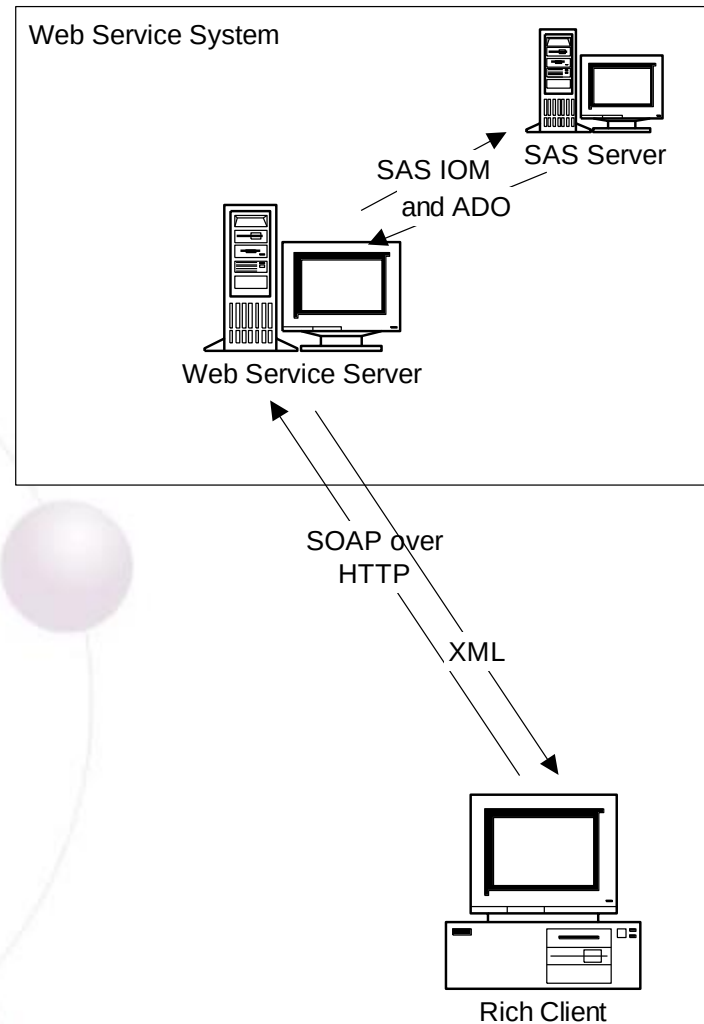
Writing Clients

- **Web Service Proxy Object**

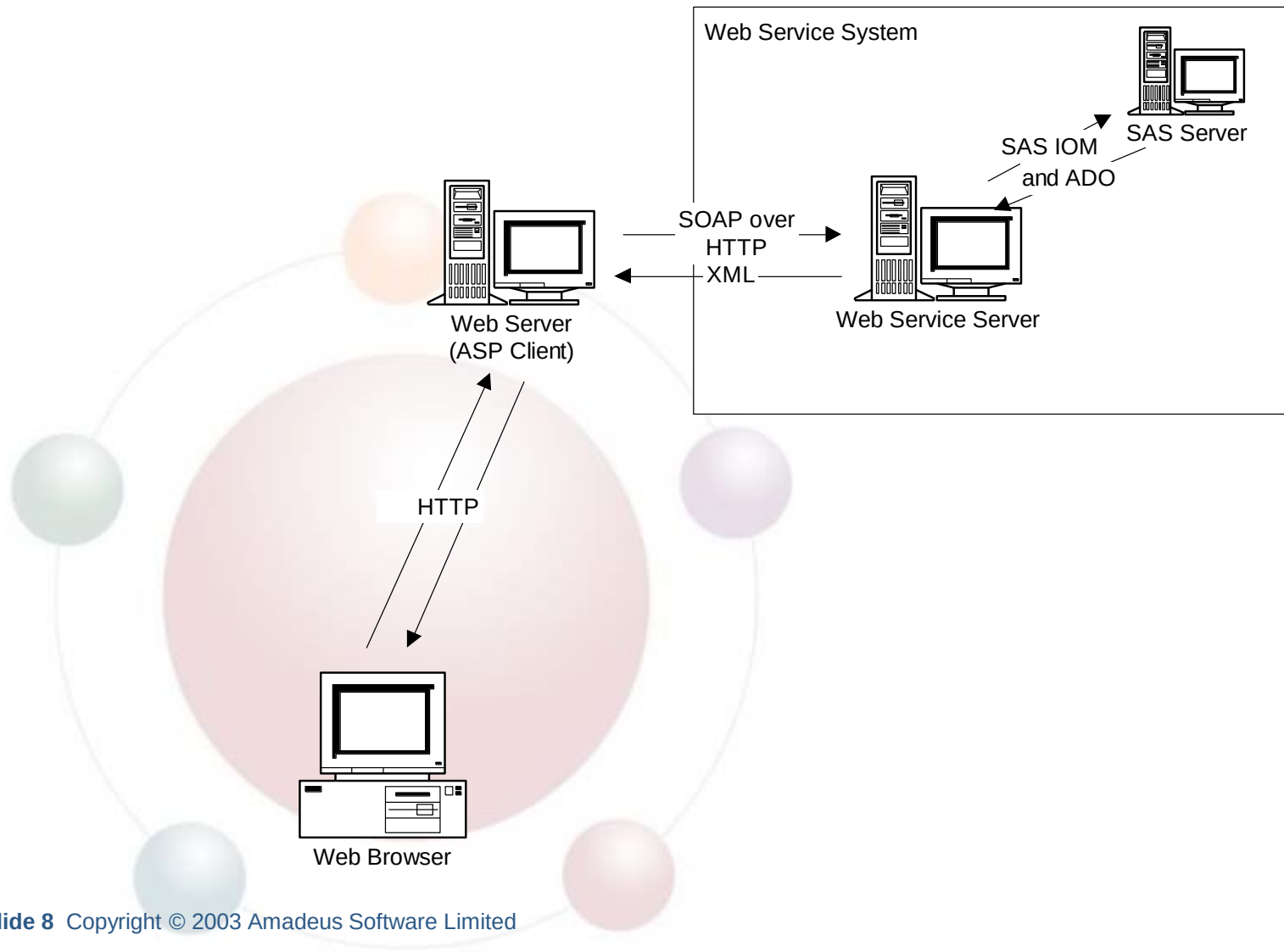


- **Built by retrieving WSDL**
- **Manages conversion of parameters to XML**
- **Creates SOAP envelope**
- **Converts response**
- **Provides asynchronous operation**

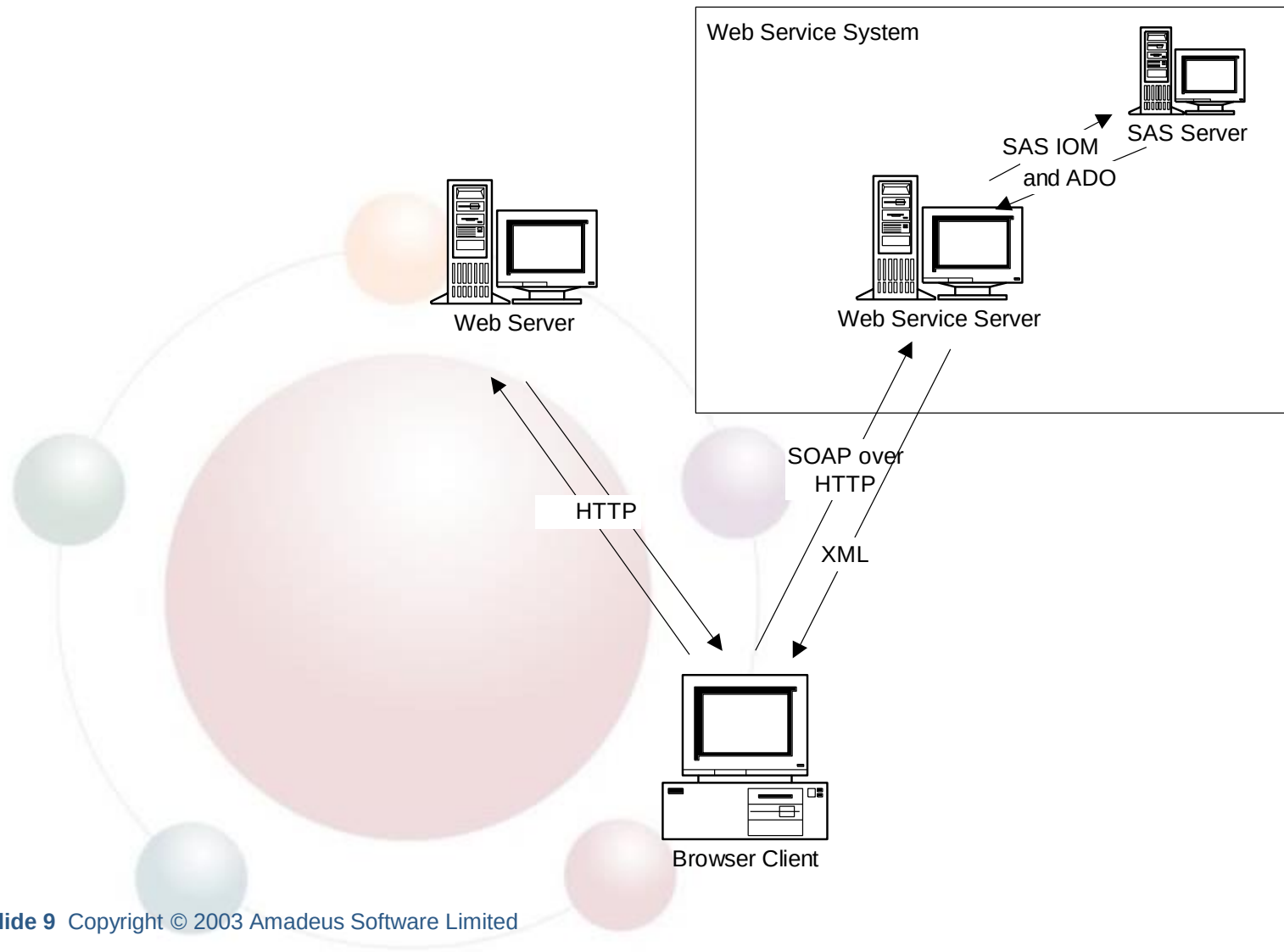
Rich Client



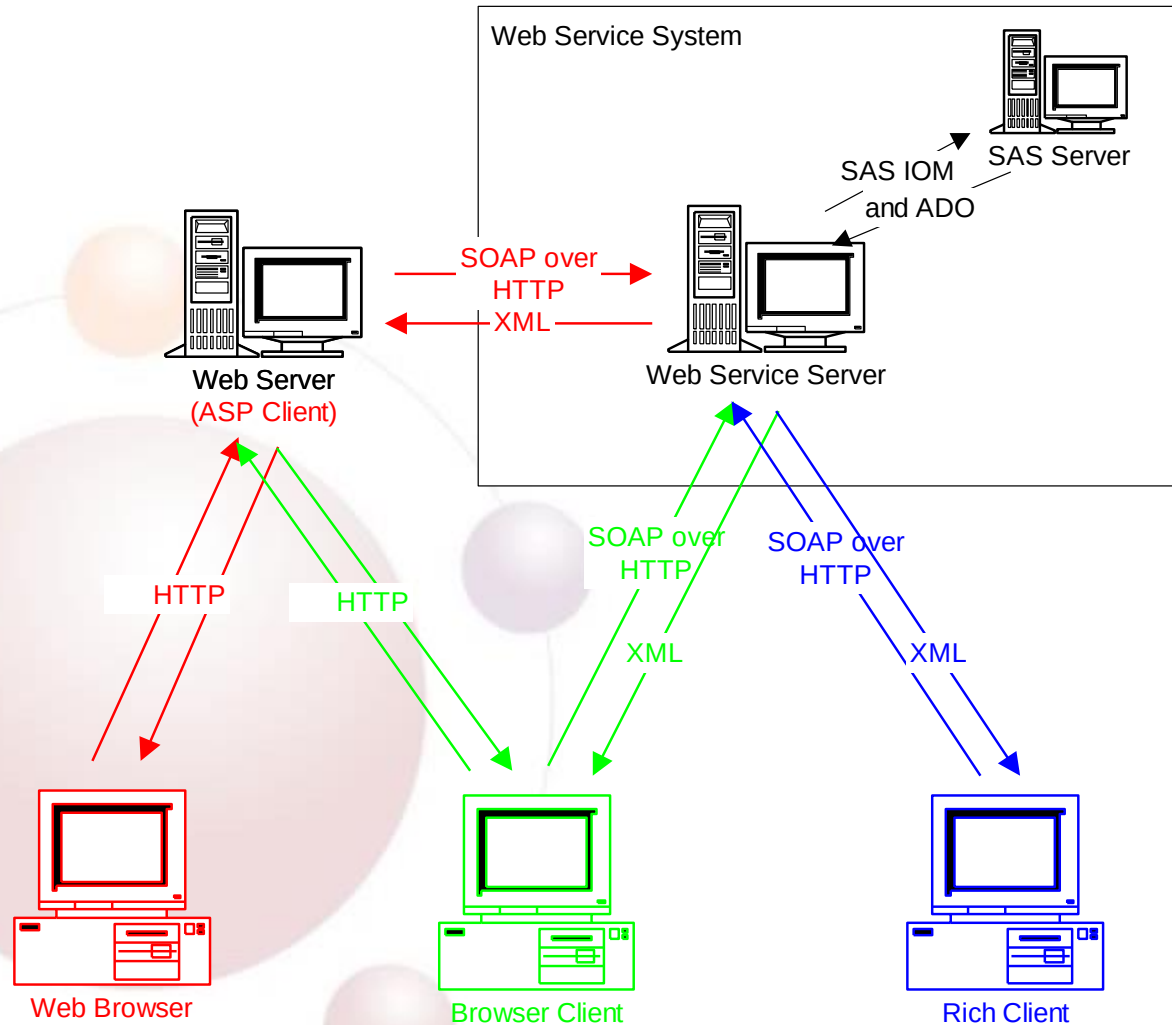
ASP.NET Client



Browser Client

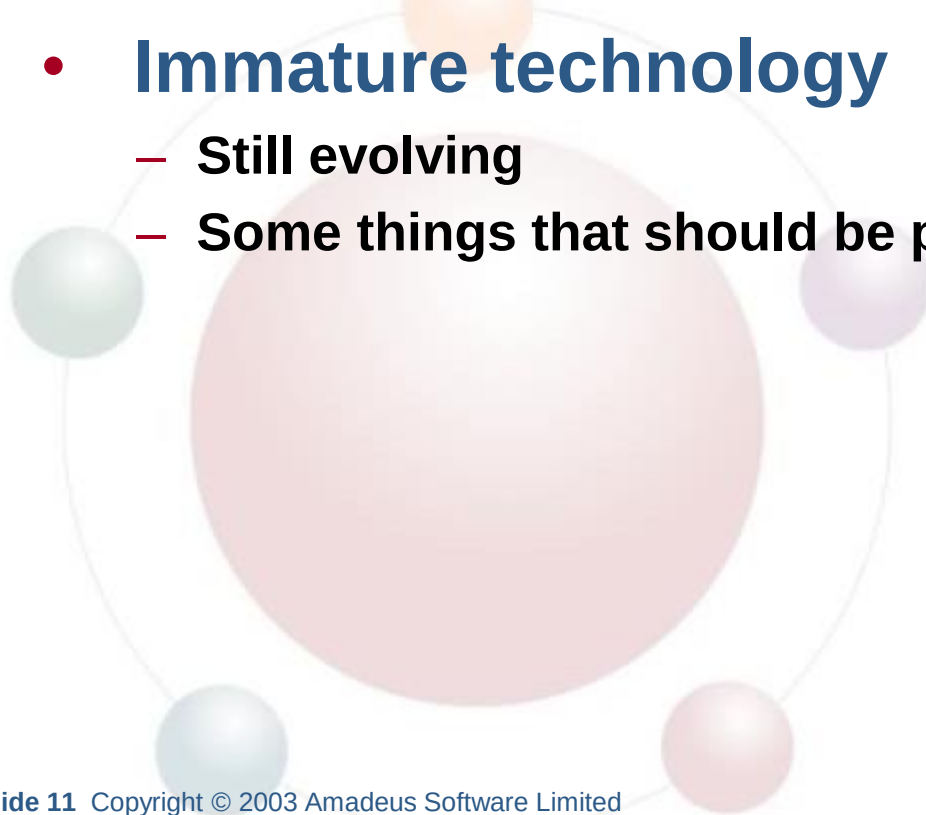


Web Service Clients

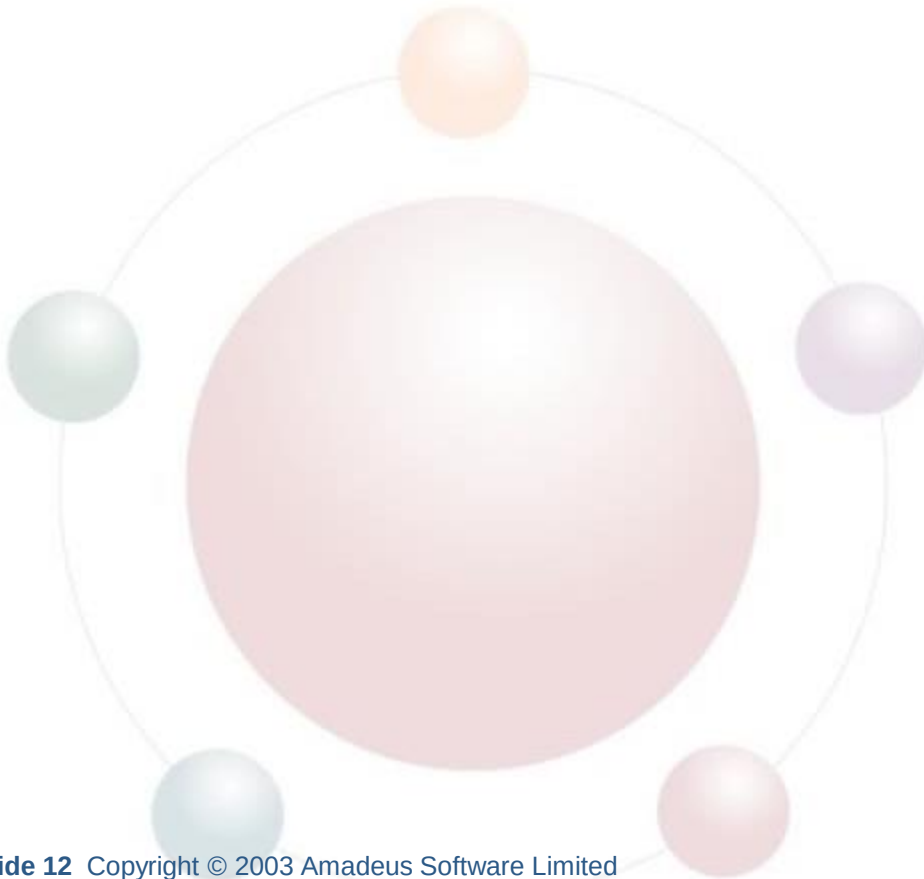


Conclusions

- **Web services allow new levels of flexibility**
- **Different clients**
- **Immature technology**
 - Still evolving
 - Some things that should be possible are not – yet



Any Questions?



Contact Information

- **Amadeus Conference Stand**
 - Come and chat
- **Email**
 - Jonathan.Williamson@Amadeus.co.uk
- **Amadeus Web Site**
 - www.amadeus.co.uk