PI Web Services Embrace New Technologies

Presented By:

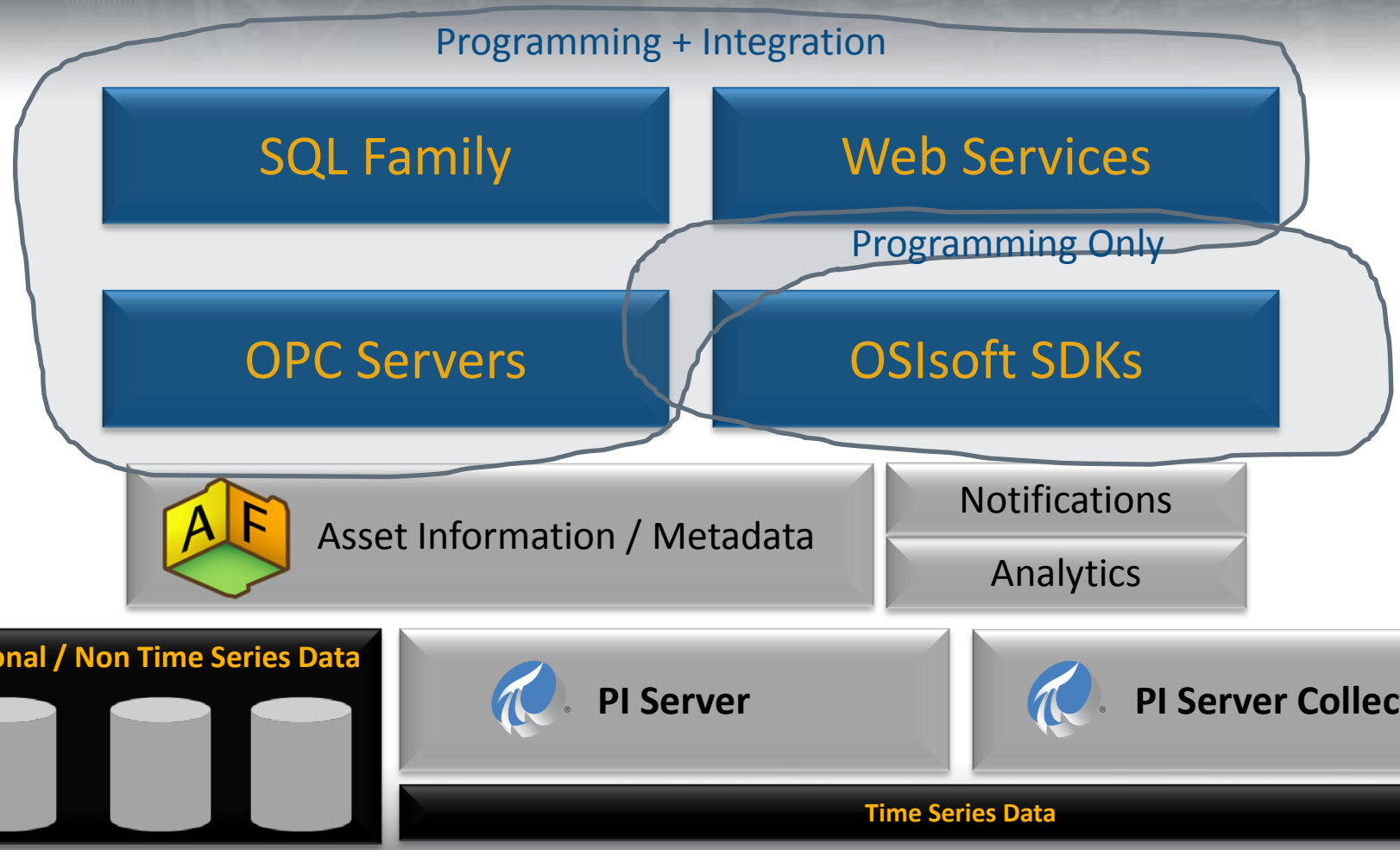
Ray Verhoeff and Stephen Mohr,
OSIsoft

OSIsoft®

Agenda

- About the Product
- Use Cases
- Architecture
- Interface
- Demos
- Security
- Roadmap

PI Data Access: The Products



PI Data Access: The 2010 Wave

Analysis/Reporting Services

PI JDBC

PI Web Services 2010

PI OLEDB Enterprise 2010

PI OPC DA/HDA Server 2010

OSIsoft SDKs

Asset Information / Metadata

Notifications

Analytics

Relational / Non Time Series Data

PI Server

PI Server Collective

Time Series Data

OSIsoft®

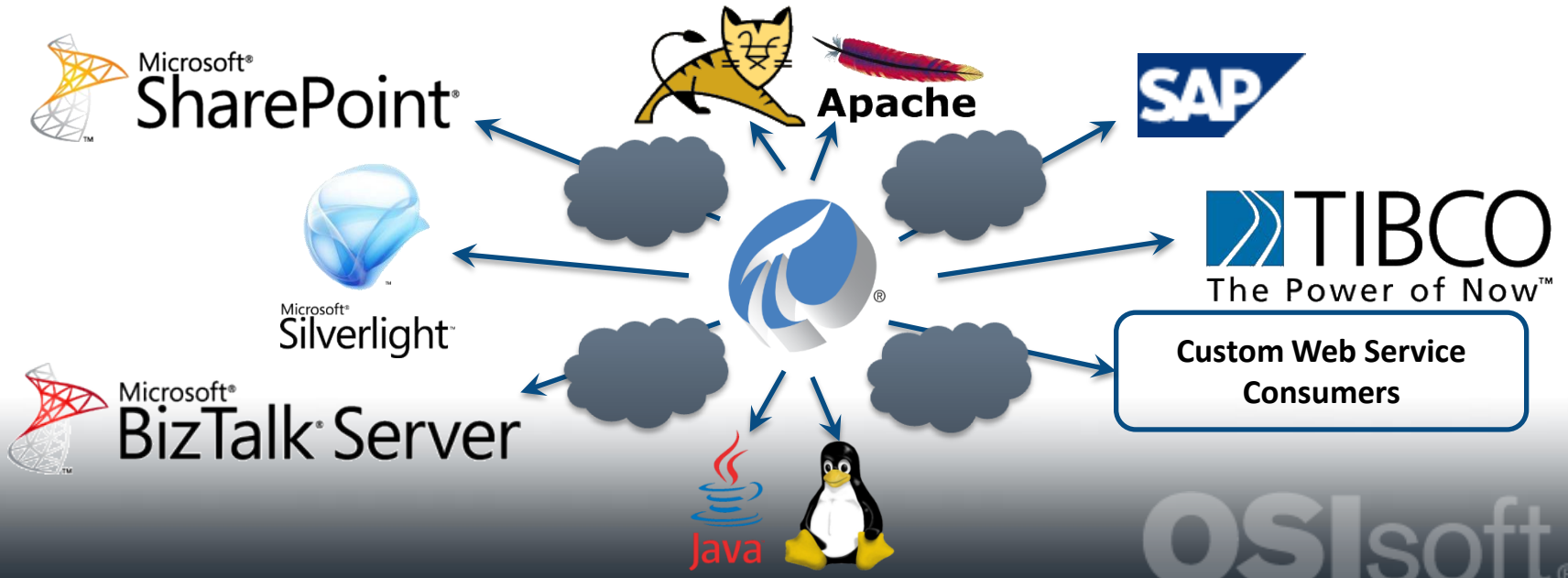
About the Product

- Provides access to **PI and PI AF** data through standard **web service** technologies
  - Many code-free integration scenarios
  - Any application, any language, any OS

- Facilitates data access over secure and distributed environments
  - Nothing required on the client machine
  - Lightweight and firewall-friendly

Use Cases

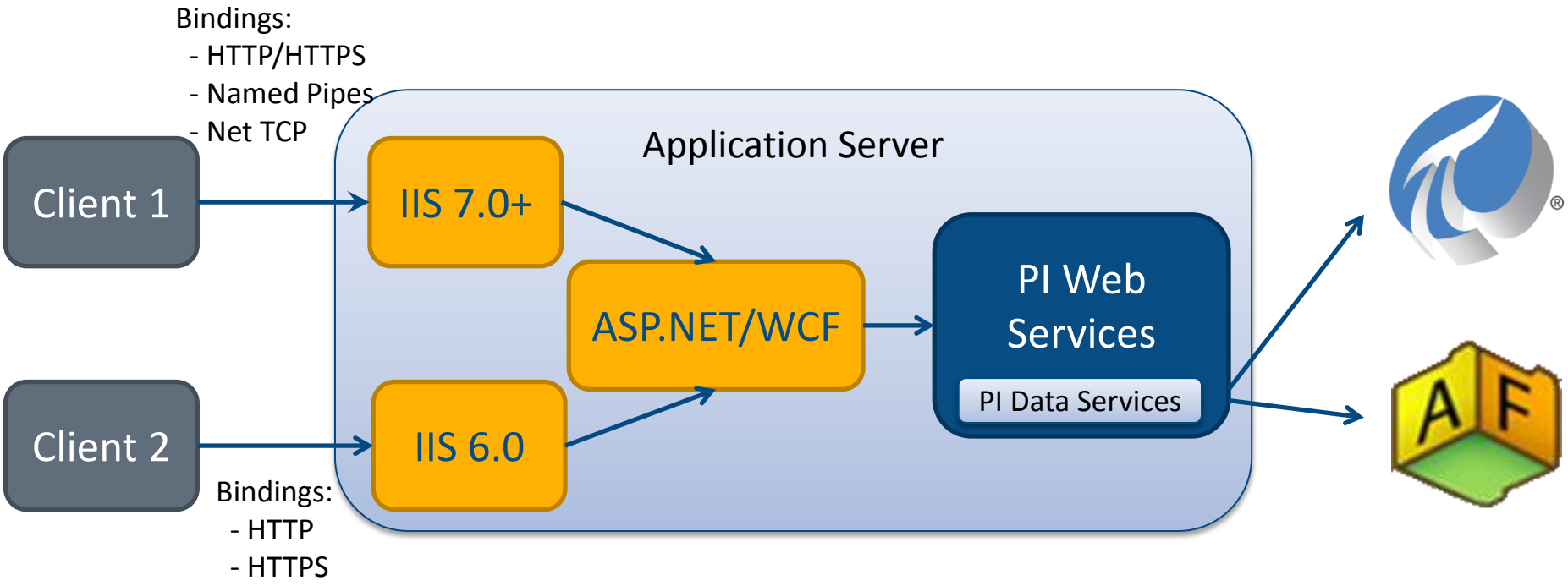
- Web-based visualization
- Integration with business systems
- Non-Windows environments



Features

- Synchronous or asynchronous
- Leverages back-end investment
  - PI System infrastructure
  - PI AF Data References
  - PE experience
- WS-* support
- Auto-describing via WSDL

Architecture



IT Friendly

- HTTP – firewall issues understood
- Web service stands between clients and back-end data sources
- Centrally managed
- Standards support many client tools

WCF

- Windows Communication Foundation included as part of .NET
- Provides a communications abstracting security and wire protocols
- Standards support – WS-*
- Provides many features for "free"

Bindings and Protocols

- SOAP
  - basicHttpBinding – SOAP 1.1
  - wsHttpBinding – SOAP 1.2

- Binary Protocols
  - netTcp
  - netNamedPipes – single machine

WCF Configuration

- XML configuration files or programming
  - web.config on the server
  - app.config or web.config on the client

- Complex but robust syntax
  - http://msdn.microsoft.com/en-us/library/dd456779.aspx

Interface

- `TimeSeries[] GetPIArchiveData(PIArcDataRequest[])`
- `TimeSeries[] GetPISummaryData(PISummaryDataRequest[])`
- `TimeSeries InsertPIData(TimeSeries[])`
- `string GetProductVersion()`

What's in a Request?

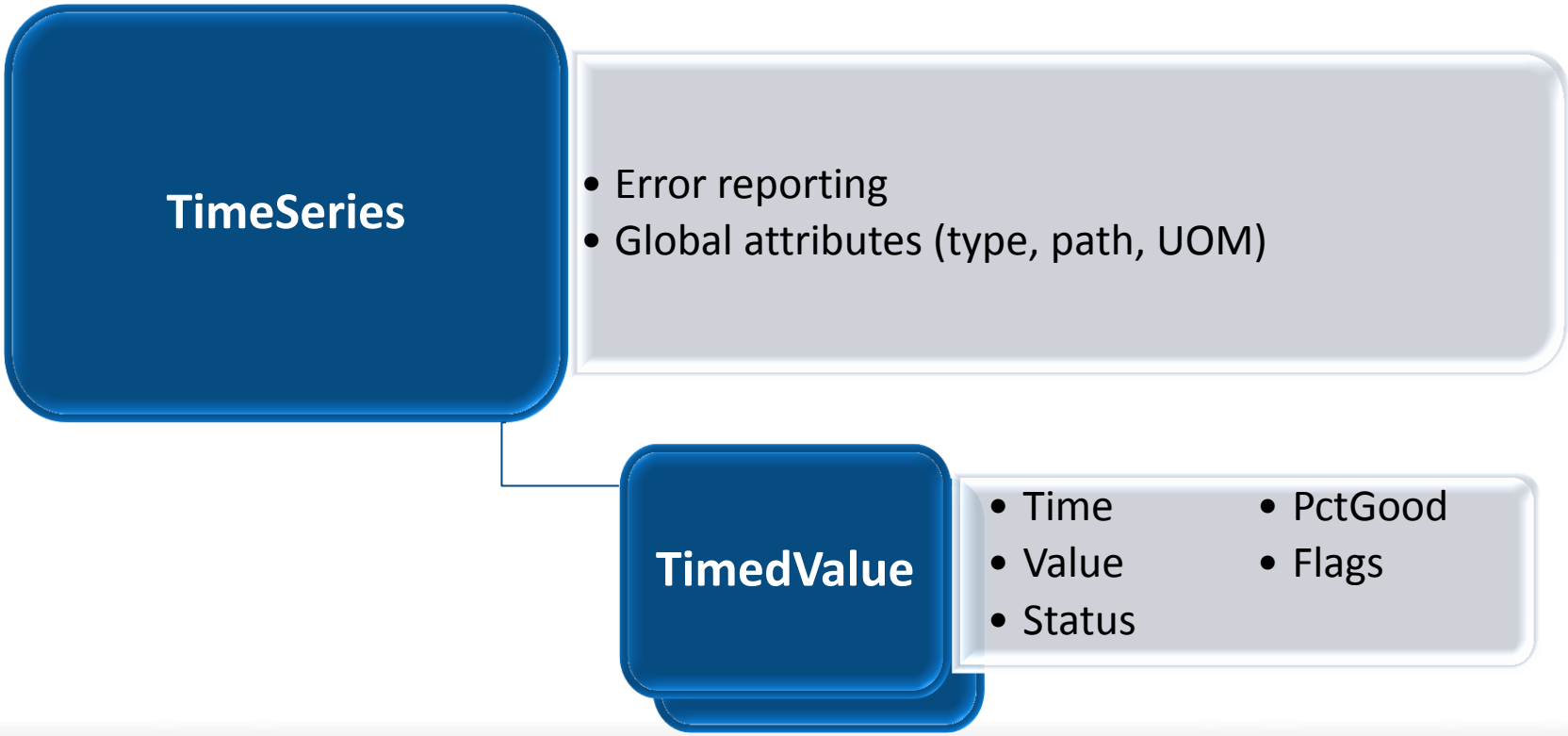
PIArcDataRequest

- Path
- TimeRange
- PIArcManner

PISummaryDataRequest

- Path
- TimeRange
- PISummaryManner

Returned Values



Paths

- String that locates a source of data
  - PI Point
    `pi:\\server\tag`
  - Performance Equation (PE)
    `pe:\\server\pe_expression`
  - AF Attribute
    `af:\\server\database\element[…]|attribute`

Specifying Times

- PI times
  - Absolute (Y, T, *)
  - Relative (*-1H)

- ISO 8601 absolute times
  - YYYY-MM-DDTHH:MM:SS[Z|+/-HH:MM]
  - 2010-09-13T13:00:00-07:00

Archive Manners

- Retrieval type
  - Compressed
  - Interpolated
  - Plotvalues
- Boundaries
  - Inside
  - Outside
  - Interpolated
- Filter
- Number of values

Summary Manner

- Summary Value
  - Average
  - Range
  - Minimum
- Weight type
  - Event-weighted
  - Time-weighted
- Intervals
  - Divide the range into “n” equal durations
- Use Start
  - Report the start or end time of the interval as the timestamp

Demo – Managed Code

- .NET WinForms with synchronous calls
- Silverlight with asynchronous calls

Demo – Java

- Eclipse + Apache Axis
  - Eclipse – SOA,
  - Axis 1.4, Tomcat 6

Demo – Unmanaged Code

- C (seriously, it still works!)
- gSoap

gSOAP “Demo”

- About the gSOAP Project
- Platforms
- How to do it
  - Generating proxies
  - Building a project
  - Calling a web service
- Memory Management
- Security

The gSOAP Project

- `http://www.cs.fsu.edu/~engelen/soap.html`
- SOAP Web services from C and C++
- Generic C/C++ XML data bindings
- Many technology companies use it:
  - Cisco, CNR Software, Adobe, AOL, BEA (now Oracle), Oasis Group, IBM, Xerox

Platforms

- **Windows** Win32/Win64 (including NT, 2000, XP, Vista, Windows 7), MS-DOS (limited), and Cygwin
- **Linux** (RedHat, SuSE, and any other "standard" Linux distro),
- **Unix** (Solaris, HP-UX, FreeBSD, Irix, QNX, AIX, 64bit TRU64, and other)
- **Apple Macintosh** OS X
- **Small and Embedded OS** (VxWorks, WinCE, Palm OS, Symbian)

Generating Proxies

- **`wsdl2h.exe`**
  - Reads WSDL, creates C or C++ header file
- **`soapcpp2.exe`**
  - Reads header file, creates proxy code
- Where?
  - `\gsoap-2.7\gsoap\bin\win32`
  - `\gsoap-2.7\gsoap\bin\linux386`

wsdl2h.exe

```
C:\>wsdl2h.exe -o piws.h -t \gsoap-2.7\gsoap\WS\typemap.dat
      http://my.server.com/PIDataService/PITimeSeries.svc?wsdl

**  The gSOAP WSDL/Schema processor for C and C++, wsdl2h release 1.2.17
**  Copyright (C) 2000-2010 Robert van Engelen, Genivia Inc.
**  All Rights Reserved. This product is provided "as is", without any warranty.

**  The wsdl2h tool is released under one of the following two licenses:
**  GPL or the commercial license by Genivia Inc. Use option -l for more info.

Saving piws.h

Reading type definitions from type map file '\gsoap-2.7\gsoap\WS/typemap.dat'

Connecting to 'http://my.server.com/PIWebServices/PITimeSeries.svc?wsdl' to retrieve
WSDL/XSD...
Connected, receiving...

Connecting to 'http://my.server.com/PIWebServices/PITimeSeries.svc?xsd=xsd0' to retrieve
schema...
Connected, receiving...
Done reading 'http://my.server.com/PIWebServices/PITimeSeries.svc?xsd=xsd0'
Done reading 'http://my.server.com/PIWebServices/PITimeSeries.svc?wsdl'

To complete the process, compile with:
> soapcpp2 piws.h
or to generate service proxy/object classes:
> soapcpp2 -i piws.h
```

soapcpp2.exe

```
C:\>soapcpp2.exe -i -C -1 -pPIWSTimeSeries -I\gsoap-2.7\gsoap\import    piws.h

**  The gSOAP code generator for C and C++, soapcpp2 release 2.7.17
**  Copyright (C) 2000-2010, Robert van Engelen, Genivia Inc.
**  All Rights Reserved. This product is provided "as is", without any warranty.
**  The soapcpp2 tool is released under one of the following three licenses:
**  GPL, the gSOAP public license, or the commercial license by Genivia Inc.

**WARNING**: option -1 or -2 overrides SOAP-ENV namespace (detected at line 104 in piws.h)
**WARNING**: option -1 or -2 overrides SOAP-ENC namespace (detected at line 105 in piws.h)

Saving PIWSTimeSeriesStub.h annotated copy of the input declarations
Saving PIWSTimeSeriesH.h interface declarations
Saving PIWSTimeSeriesC.cpp XML serializers
Using ns1 service name: BasicHttpBinding_USCOREIPITimeSeries
Using ns1 service style: document
Using ns1 service encoding: literal
Using ns1 service location: http://my.server.com/PIWebServices/PITimeSeries.svc
Using ns1 schema namespace: http://xml.osisoft.com/services/PIDataService
Saving PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy.h client proxy class
Saving PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy.cpp client proxy class
Saving BasicHttpBinding_USCOREIPITimeSeries.GetPIArchiveData.req.xml sample SOAP/XML request
Saving BasicHttpBinding_USCOREIPITimeSeries.GetPIArchiveData.res.xml sample SOAP/XML response
Saving BasicHttpBinding_USCOREIPITimeSeries.GetPISummaryData.req.xml sample SOAP/XML request
Saving BasicHttpBinding_USCOREIPITimeSeries.GetPISummaryData.res.xml sample SOAP/XML response
Saving BasicHttpBinding_USCOREIPITimeSeries.InsertPIData.req.xml sample SOAP/XML request
Saving BasicHttpBinding_USCOREIPITimeSeries.InsertPIData.res.xml sample SOAP/XML response
Saving BasicHttpBinding_USCOREIPITimeSeries.GetProductVersion.req.xml sample SOAP/XML request
Saving BasicHttpBinding_USCOREIPITimeSeries.GetProductVersion.res.xml sample SOAP/XML response
Saving BasicHttpBinding_USCOREIPITimeSeries.nsmap namespace mapping table

Compilation successful (2 warnings)
```

Building your Project

- Header Files:
  - `PIWSTimeSeriesH.h`
  - `PIWSTimeSeriesStub.h`
  - `PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy.h`
  - `stdsoap2.h`

Building your Project

- Source Files:
  - `PIWSTimeSeriesC.cpp`
  - `PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy.cpp`
  - `stdsoap2.cpp`

Building your Project

- In your code:

```
#include "BasicHttpBinding_USCODEIPITimeSeries.nsmap"
#include "PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy.h"


PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy proxy;
```

Or…

```
PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy *pproxy;
pproxy = new PIWSTimeSeriesBasicHttpBinding_USCOREIPITimeSeriesProxy();
```

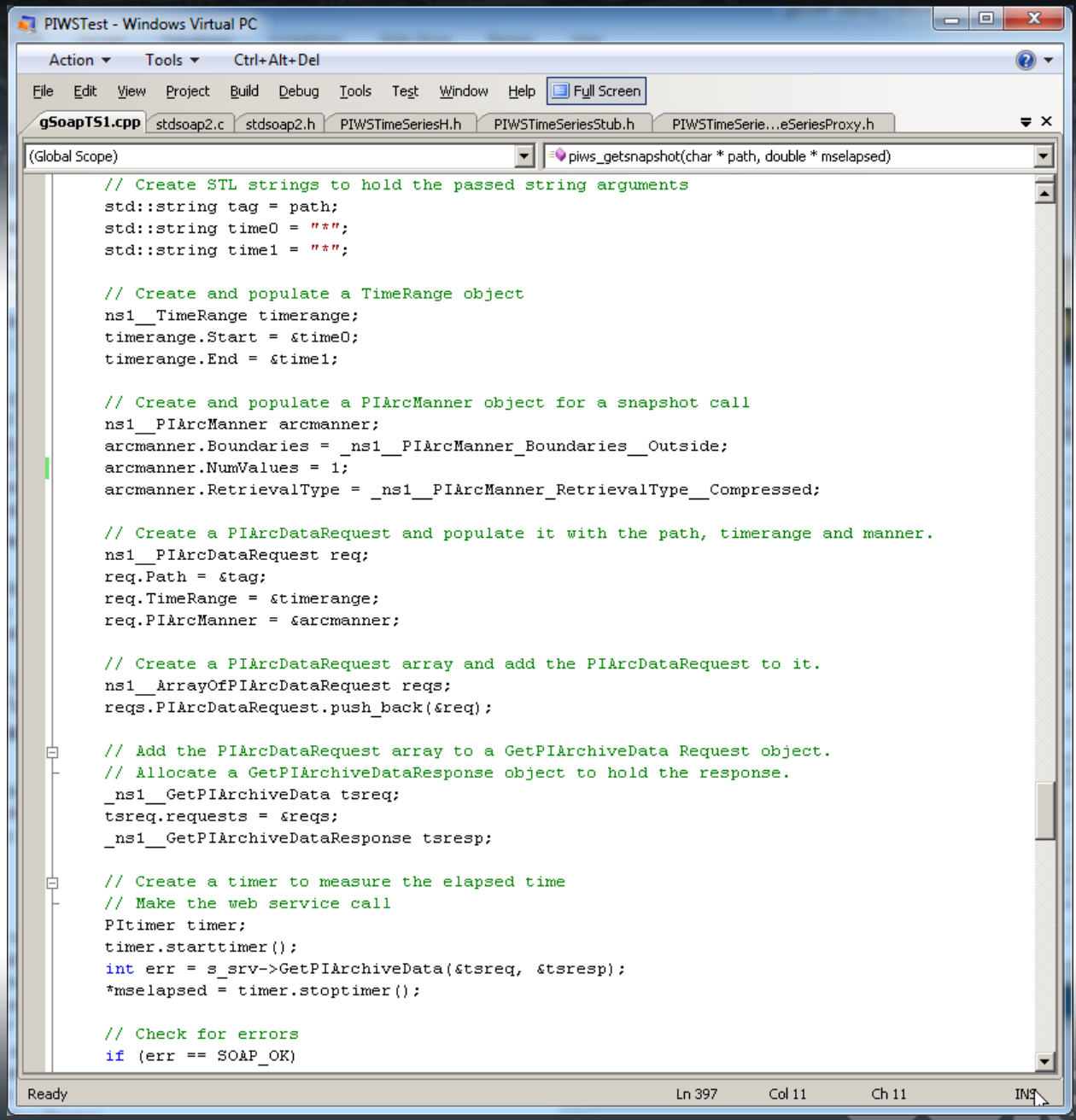
Web Method Proxies

```
virtual int GetPIArchiveData (
        _ns1__GetPIArchiveData *ns1__GetPIArchiveData,
        _ns1__GetPIArchiveDataResponse *ns1__GetPIArchiveDataResponse);

virtual int GetPISummaryData (
        _ns1__GetPISummaryData *ns1__GetPISummaryData,
        _ns1__GetPISummaryDataResponse *ns1__GetPISummaryDataResponse);

virtual int InsertPIData (
        _ns1__InsertPIData *ns1__InsertPIData,
        _ns1__InsertPIDataResponse *ns1__InsertPIDataResponse);

virtual int GetProductVersion (
        _ns1__GetProductVersion *ns1__GetProductVersion,
        _ns1__GetProductVersionResponse *ns1__GetProductVersionResponse);
```

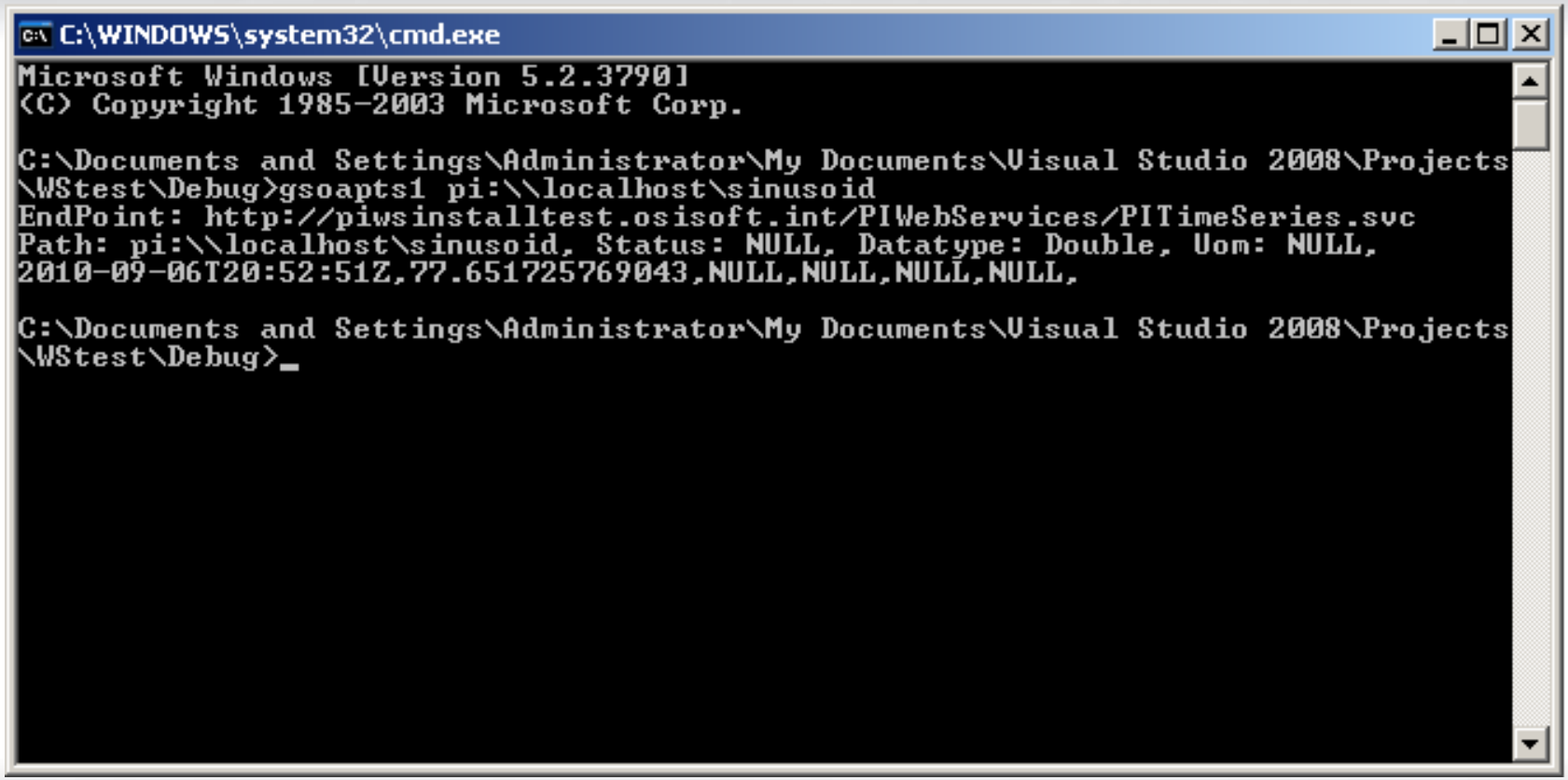
Example: Getting the Version String

```
_ns1__GetProductVersion vreq;
_ns1__GetProductVersionResponse vresp;

err = proxy.GetProductVersion(&vreq, &vresp);

printf("Version: %s\n", vresp.GetProductVersionResult->c_str());

s_srv->destroy();
```

Example: Getting a Snapshot



Snapshot Result



gSOAP Memory Management

- gSOAP allocates memory to hold results
- Free with proxy.destroy()
- You can participate:

```
ns1__TimeRange *timerange = soap_new_ns1__TimeRange(pproxy, -1);

ns1__PIArcDataRequest *request  = soap_new_ns1__PIArcDataRequest(pproxy, -1);

std::string *Path = soap_new_std__string(pproxy, -1);
```

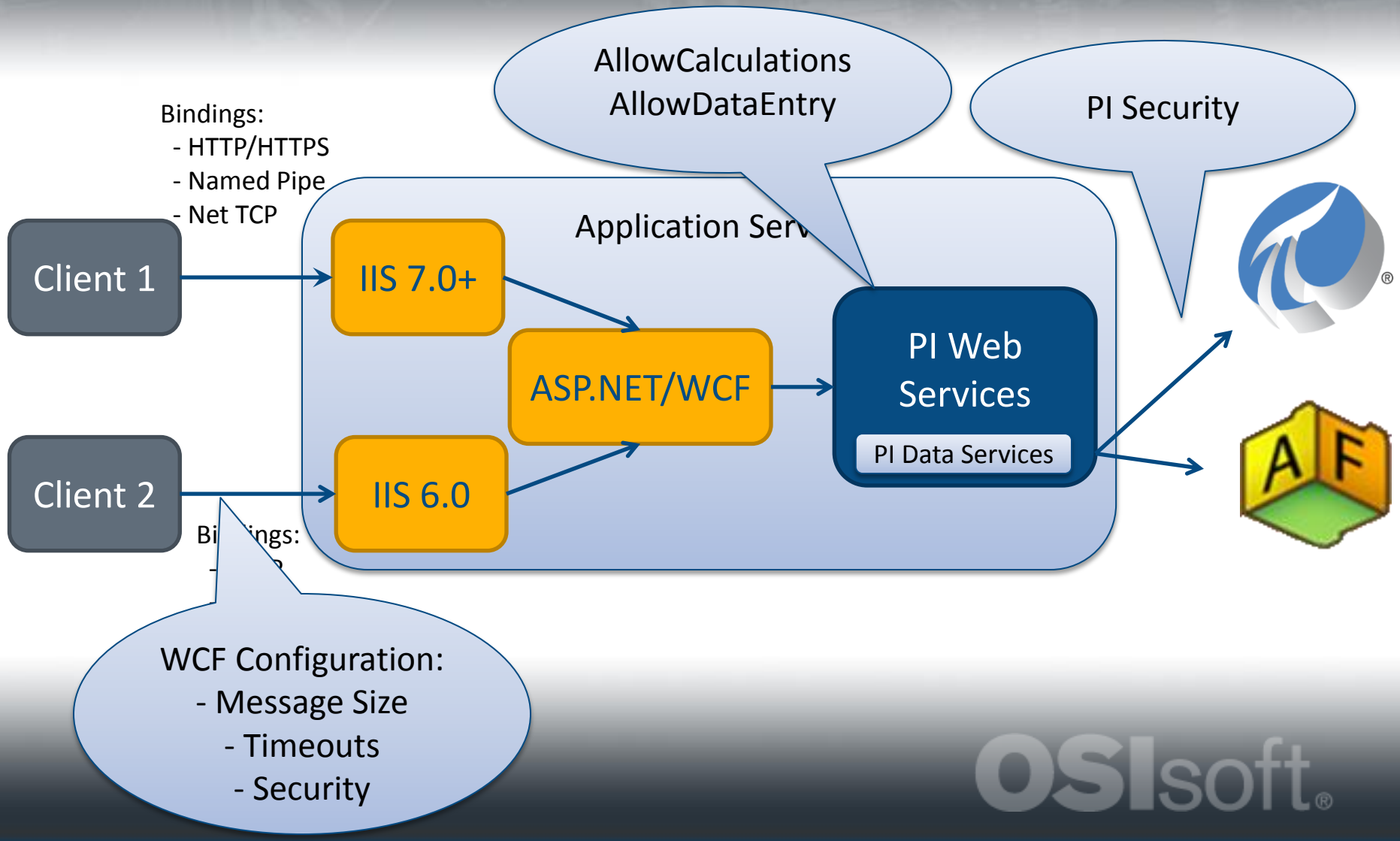
gSOAP Security

- Tested with PI Web Services' basicHttp Binding
- Security requires SSL and Certificates
- gSOAP supports SSL through the OpenSSL Project
  - http://www.openssl.org

Demo – Security

- Windows integrated security
- Configuration, not programming

Other Security Options



Bindings and Standards

Binding Name	Data Format	Description	Application Compatibility
basicHttp	XML	SOAP 1.1 Web services conforming to the WS-I Basic Profile. Best interoperability with non-Windows platforms. Security options: none, HTTPS security, SOAP security, and HTTPS security with SOAP credentials.	BizTalk , InfoPath, .NET Framework, Silverlight, or Java
wsHttp	XML	SOAP 1.2 Web services. WS-Security options: HTTPS, message security, both; username/password, Windows identities, X.509 certificates WS-SecureConversation, WS-Trust, WS-ReliableMessaging, WS-Addressing	BizTalk, .NET Framework, Java
netTCP	Binary	A secure and optimized binding suitable for cross-machine communication on an intranet between a WCF clients and WCF Web servers.	BizTalk, .NET Framework
netNamedPipes	Binary	Secure, fast communication between WCF clients and servers on the same machine. Security options: none, transport	.NET Framework

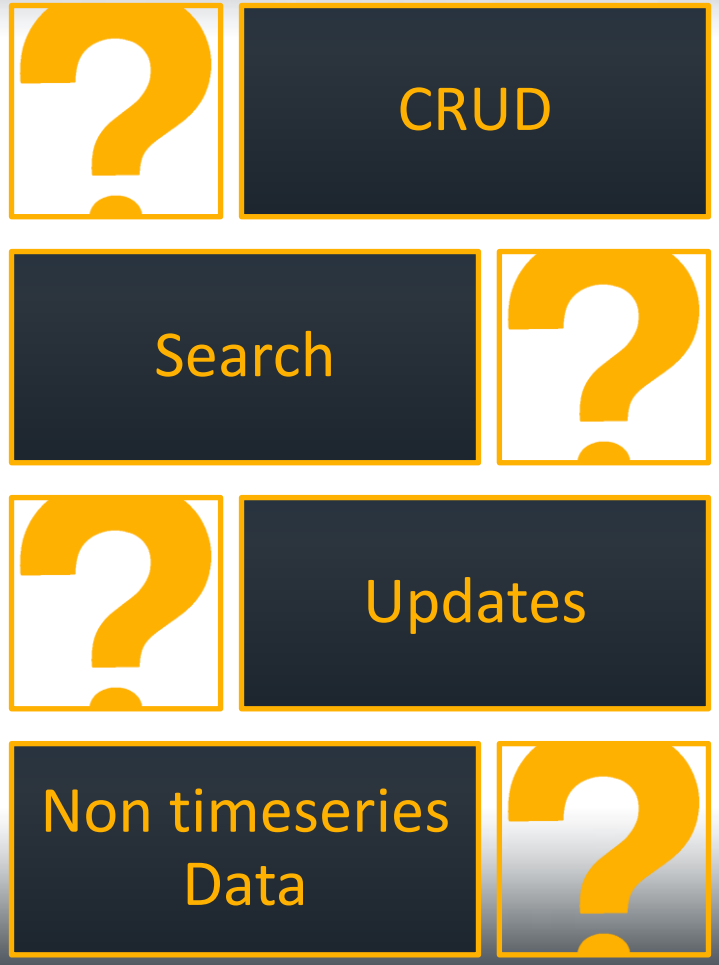
Roadmap

Coming in R2

- `TimeSeries[]` **`GetPISnapshotData`**`(string[] paths)`
  - …because everyone asked for it!

- Expanded support for filters
- Expanded PI AF Data Reference capabilities
- Other features TBD

What do you need for 2011?



Additional Sessions of Interest

- Broadening Web Service Access through REST and OData?
  - Next session, Track 1
- Integration with Enterprise Business Systems using PI Data Access
  - Next session, Track 2
- Using PI Web Services and PI WebParts with Silverlight for creative solutions
  - Tomorrow

Thank You!

OSIsoft®