

---

# **A Practical Guide to Web Services and PI**

**Christopher Russo  
Russo & Associates  
PI T&D Conference  
Sacramento, CA  
17 September 2003**

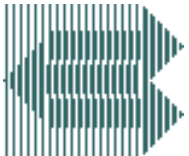
**[www.russoandassociates.com](http://www.russoandassociates.com)**



# Introduction and Overview

---

- **What are Web Services?**
  - “Buzzword Bingo”
  - The impact has probably been overhyped in the short term and underestimated in the long term.
- **Why might I want to use them?**
  - *Interoperability*
  - No more religious wars
- **Some practical examples & pitfalls**
  - Security considerations



# The Three-Minute Guide to Web Services

---

## □ Web Services

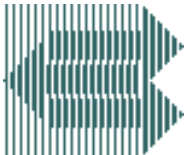
- A better name might be *network* services
- A simple method for exchanging textual data
- Formatted using XML, and transported over multiple protocols, mostly HTTP(S).
- Security can be provided through encryption and whatever authentication method you choose.

## □ SOAP – Simple Object Access Protocol

- A method to structure the XML and to provide a programmatic object-oriented interface to web services.
- Libraries exist for almost all OS's and languages
- XML-RPC is similar in concept, but different in execution

## □ So why is this a big deal now?

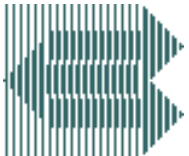
- The support for SOAP and XML is now being built into application servers and languages.



# What does this mean for me?

---

- ❑ **Web services are a common language for computers to speak to each other**
- ❑ **The choice of .NET, J2EE, or something else no longer matters – no fighting religious wars**
- ❑ **Message-bus and middleware technologies interface very nicely with web services**
- ❑ **Centralized maintenance of software components makes changes easier.**
  - **Changing one object on the server is like changing a function in everyone's code**



# PI, .NET, and Web Services

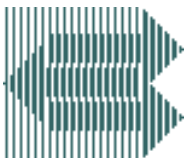
---

## □ How does PI fit into this?

- ICE Portal technology (stay tuned for the next presentation...)
- (Almost) Native .NET libraries for the SDK allow rapid creation of PI web services
- IIS can handle all of the XML and SOAP work.
- Persistent objects can be created on the service

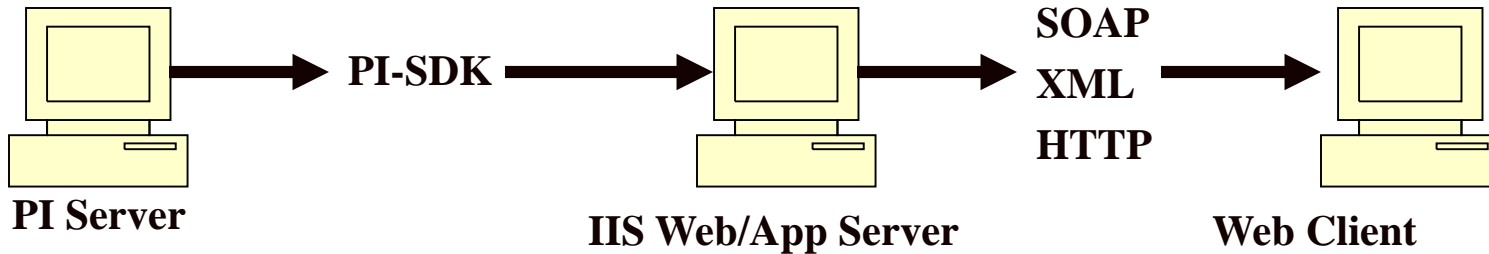
## □ What do I need?

- For the server: IIS, PI-SDK and a knowledge of C# or VB.NET.
- For the client: A SOAP client in a language and platform of your choosing
  - » .NET, SOAP::Lite, Java-SOAP, Apache-SOAP

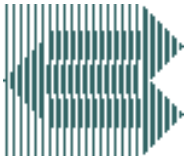


# Data flow for Web Services

---



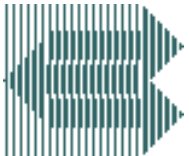
```
POST /pisoap/pi.asmx
HTTP/1.1 Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://localhost/Sinusoid"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Sinusoid xmlns="http://localhost" />
</soap:Body>
</soap:Envelope>
```



# Sample Applications

---

- ❑ **Returning the snapshot of a point**
- ❑ **Returning the property of a module**
- ❑ **Returning information about a user**
- ❑ **Tying a resource database with PI data**
- ❑ **Consuming Web Services**
  - **PJM (Streaming Nodal LMPs)**
  - **NYISO (Zonal LMPs)**
  - **Commodity power prices**
  - **Etc...**
- ❑ **Live Demo**



# A Sample Web Services Client

---

## □ .NET (C#)

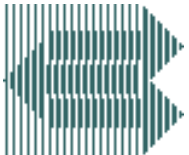
```
// Initialization code here
PISOAP.Service piws = new PISOAP.Service();
Console.WriteLine(piws.Sinusoid());
```

## □ Perl

```
print SOAP::Lite
    -> uri('http://localhost/')
    -> proxy('http://localhost/pisoap/pi.asmx')
    -> Sinusoid();
```

## □ Java

```
// Initialization code here
PISOAP piws = (PISOAP)piws.Sinusoid();
```

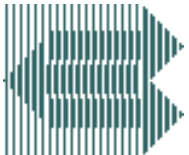




# Security and Authentication

---

- ❑ **Security: Making sure no-one can eavesdrop**
  - Web services can easily be invoked through encrypted protocols (HTTPS)
- ❑ **Authentication: Making sure you are who you say you are**
  - Authentication against Active Directory, LDAP, NIS, etc.
  - Data privacy can be enforced through PI or through the Web service layer



# Links for further information

---

- ❑ **O'Reilly – [www.ondotnet.com](http://www.ondotnet.com) & [www.xml.com](http://www.xml.com)**
- ❑ **Microsoft – [msdn.microsoft.com](http://msdn.microsoft.com)**
- ❑ **OSI Devnet – [devnet.osisoft.com](http://devnet.osisoft.com)**
- ❑ **UDDI – [www.uddi.org](http://www.uddi.org)**
- ❑ **PJM - [www.pjm.com/etools/edatafeed.html](http://www.pjm.com/etools/edatafeed.html)**
  
- ❑ **Sample code will be posted at:**
  - **[www.russoandassociates.com](http://www.russoandassociates.com)**

