

# **XML – Bridging Telecom Applications**

Workshop  
Tuesday April 8, 2008

Martin Taylor  
VP Product Management & Technology Strategy, MetaSwitch

MetaSwitch

# Agenda

- Need for machine-to-machine interfaces
- What came before XML
- Mini-tutorial on XML
- XML for provisioning
- XML for subscriber-facing applications

# Control interfaces to the local exchange switch . . . historically

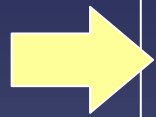


*Switch technician*  
Human-to-machine interface



*Subscriber*  
Human-to-machine interface

# Machine-to-machine interfaces for the local exchange switch



Operations Support Systems



Smart subscriber terminals



Application mashups

# OSS interfaces to Network Element

Billing?

Performance?

Alarms?

Provisioning?



# OSS provisioning interfaces (I)

Early days (1980s): OSS emulates human operator

Proprietary Command Line Interfaces

Later – “standard”: Transaction Language I (TLI)

```
ADD-T1-PORT::468::SLOT-3,PORT-2,ESF;
```

```
System 02-01-11 19:30:43
```

```
M 468 COMPLD
```

# OSS provisioning interfaces (2)

Next phase (1990s): CORBA

**C**ommon **O**bject **R**equest **B**roker **A**rchitecture

Numerous issues:

Complexity – cost – interoperability – lack of built-in OS support etc etc

Still in use – but eclipsed by JAVA and XML in the early 00s

# The rise of eXtensible Markup Language

XML is just a way of packaging data:

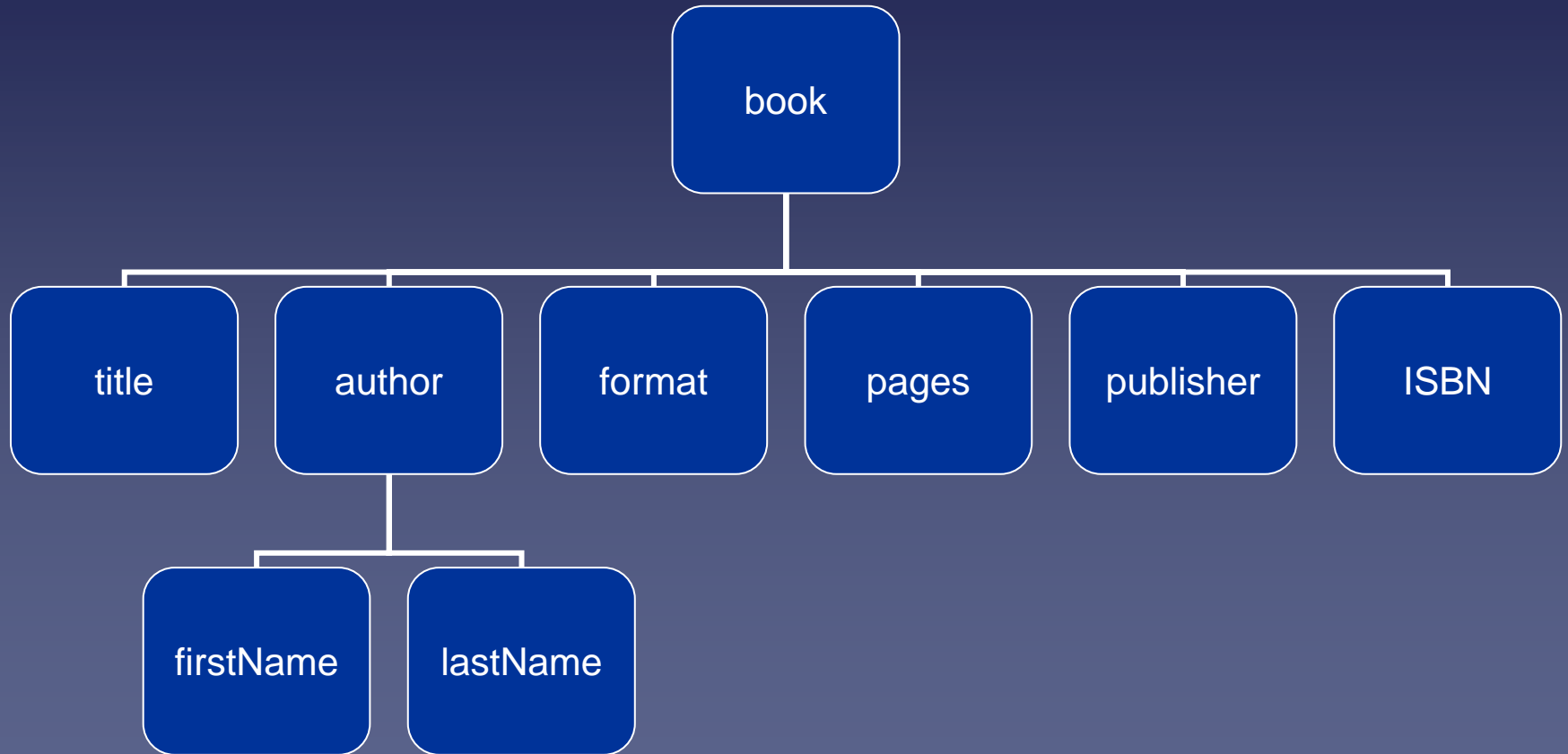
- It's text
- It's self-describing
- It's good for handling complex data content

See <http://www.w3.org/XML/>

# Simple XML example

```
<book>  
  <title>The Right Stuff</title>  
  <author>  
    <firstName>Tom</firstName>  
    <lastName>Wolfe</lastName>  
  </author>  
  <format>Paperback</format>  
  <pages>448</pages>  
  <publisher>Vintage</publisher>  
  <ISBN>978-0099479376</ISBN>  
</book>
```

# XML can represent a data hierarchy



# XML can also handle data in list form

```
<answeredCalls>
  <call>
    <from>5089042190</from>
    <time>2008-04-08 12:41:57</time>
    <duration>00:02:34</duration>
  </call>
  <call>
    <from>2124327684</from>
    <time>2008-04-08 10:22:41</time>
    <duration>00:00:41</duration>
  </call>
</answeredCalls>
```

# Describing XML formats: “XML Schema”

XML Schema provides a formal description of the structure of any given kind of XML-based data

XML Schemas are written in (you’ve guessed it)  
XML

Applications that use XML data need to understand the XML Schema for that data

# What's good about XML for machine-to-machine communications?

- Simple
- Open
- Flexible
- Human readable
- Easy to make backward-compatible
- Natively supported in all common software environments

# Transporting XML data

Most commonly transported over HTTP, HTTPS

- The same protocol as we use for the Web
- Easy to traverse firewalls
- Leverages HTTPS when security needed

# What is SOAP?

- Used to mean “Simple Object Access Protocol”
- Now it doesn’t stand for anything: just SOAP
- XML Schema for request / response exchange

```
<Envelope>
```

```
  <Header>
```

```
    Optional XML stuff about the request
```

```
  </Header>
```

```
  <Body>
```

```
    Body of request in XML format
```

```
  </Body>
```

```
</Envelope>
```



# XML for Provisioning – Example

- MetaSwitch supports a SOAP interface for provisioning
- Modeled on the IMS standard “Sh” interface
- Two operations: “Pull” and “Update”
- Specify User Identity (e.g. Directory Number) and “ServiceIndication”
- ServiceIndication is a name for a particular subset of the provisioned data

# MetaSwitch ServiceIndication Examples

Meta\_Subscriber\_BaseInformation

Meta\_Subscriber\_LineClassCodes

Meta\_Subscriber\_UnconditionalCallForwarding

Meta\_Subscriber\_Find-me-follow-me\_RulesList

Etc.

# Example “Pull” Request Message

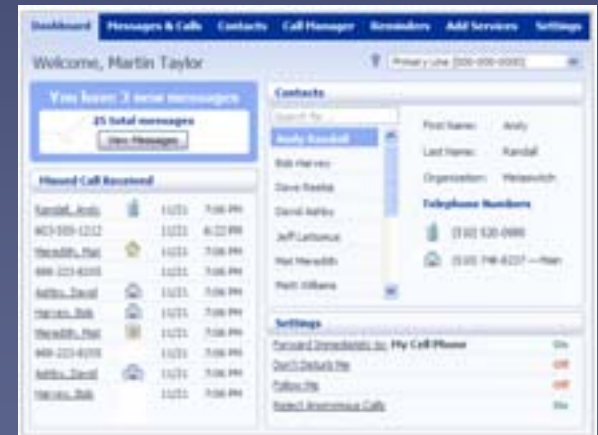
```
<Envelope>  
  <Body>  
    <ShPull>  
      <UserIdentity>5089042190</UserIdentity>  
      <DataReference>0</DataReference>  
      <ServiceIndication>  
        Meta_Subscriber_UnconditionalCallForwarding  
      </ServiceIndication>  
      <OriginHost>oss.mydomain.com</OriginHost>  
    </ShPull>  
  </Body>  
</Envelope>
```

# Web Services Description Language

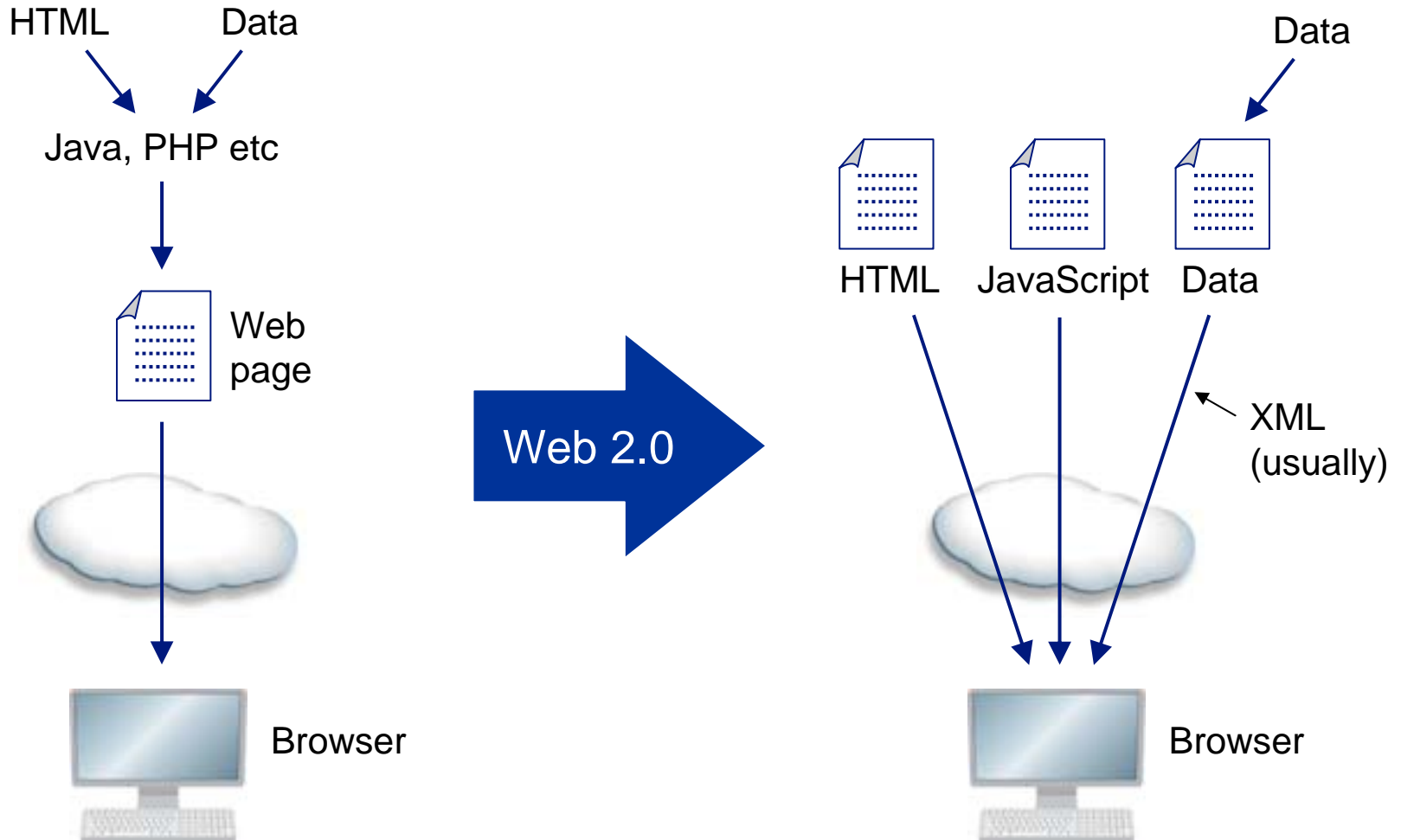
- SOAP, XML over HTTP → “Web Services”
- WSDL provides formal definition of the operations and XML Schemas of the data to be exchanged
- Most software development environments (e.g. JAVA, .NET) include utilities for generating program code directly from WSDL

# “Smart Subscriber Terminals”

- SIP business phone
- Cell phone
- PC with browser
- PC with local application
- Set-top box (cable, IP-TV)
- Home hub



# Web Application Evolution



# Anatomy of a Web 2.0 App

- *HMTL* provides a template for page layout
- *Data* provides dynamic content for page
  - E.g. list of voicemail messages
  - Usually XML
  - Alternative is “JavaScript Object Notation (JSON)”
  - These can be considered interchangeable
- *JavaScript* instructs browser how to
  - insert the data into the template
  - dynamically re-arrange on request

# Benefits of Web 2.0 Approach

- Enables dynamic “application-like” user experience
  - E.g. MetaSwitch CommPortal
- Easy to customize
  - Modify HTML to change layout (alternate skins)
  - Modify JavaScript to change behavior

# Example Web 2.0 App

Dashboard **Messages & Calls** Contacts Call Manager Reminders Add Services Settings

Welcome, Martin Taylor Primary Line (000-000-0000)

**You have 3 new messages**

**25 total messages**

[View Messages](#)

**Missed Call Received**

<a href="#">Randall, Andy</a>		11/21	7:06 PM
603-555-1212		11/21	6:22 PM
<a href="#">Meredith, Mat</a>		11/21	7:06 PM
888-223-8255		11/21	7:06 PM
<a href="#">Ashby, David</a>		11/21	7:06 PM
<a href="#">Harvey, Bob</a>		11/21	7:06 PM
<a href="#">Meredith, Mat</a>		11/21	7:06 PM
888-223-8255		11/21	7:06 PM
<a href="#">Ashby, David</a>		11/21	7:06 PM
<a href="#">Harvey, Bob</a>		11/21	7:06 PM

**Contacts**

Search for...

- Andy Randall**
- Bob Harvey
- Dave Reekie
- David Ashby
- Jeff Lattomus
- Mat Meredith
- Matt Williams

First Name: Andy  
Last Name: Randall  
Organization: Metaswitch

**Telephone Numbers**

- (510) 520-0999
- (510) 748-8237 — Main

**Settings**

- Forward Immediately to: **My Cell Phone** **On**
- Don't Disturb Me **Off**
- Follow Me **Off**
- Reject Anonymous Calls **On**

# Web 2.0 Apps and Common Data APIs

- Data delivered separately from layout and presentation
- Same (XML) data API supports other types of app
- PC desktop productivity apps
  - E.g. MetaSwitch CommPortal Assistant Toolbar
- Apps running on user devices
  - IP-TV or Cable Set Top Box apps
  - “Home hub” applications
  - Cell phone apps
- Application mashups
  - E.g. click-to-dial app embedded in social networking site

# Visual Voicemail on TV



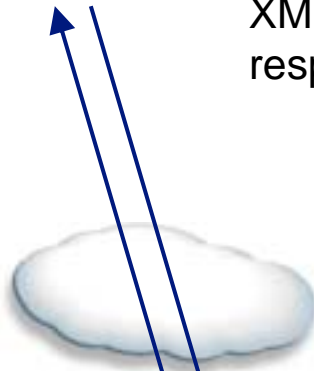
# Application Mashups

Social networking site

Telecom services site



XML request /  
response



Browser

Embedded in your  
personal page:

- List of voice messages from your buddies
- Click-to-dial from your buddy list









