Document Interop from an Open Source perspective

PHPExcel & OPENXML4J

Maarten Balliauw – http://blog.maartenballiauw.be
Julien Chable – http://blogs.developpeur.org/neodante

12-Dec-08

Who are we?

- Maarten Balliauw
  - RealDolmen – http://www.realdolmen.com
  - http://blog.maartenballiauw.be
  - maarten@maartenballiauw.be

- Julien Chable
  - Wygwam – http://www.wygwam.com
  - http://blogs.developpeur.org/neodante
  - julien@wygwam.com

PHPExcel and OPENXML4J

- PHPExcel

- OPENXML4J

PHPExcel

- Provides an in-memory spreadsheet engine
  - Workbook with worksheets
  - Worksheets with cells
  - Formula support + calculation engine
  - Styles
  - Images
  - …

- http://www.phpexcel.net or http://www.codeplex.com/PHPExcel

PHPExcel

PHPExcel

- PHP library (PHP 5.1.x and up)
- Provides an in-memory spreadsheet engine
- De-facto standard in PHP world
- Multi-platform
- Open-source
- Great documentation!
- Currently 14 developers - 4 active
- Average 280 downloads per day!

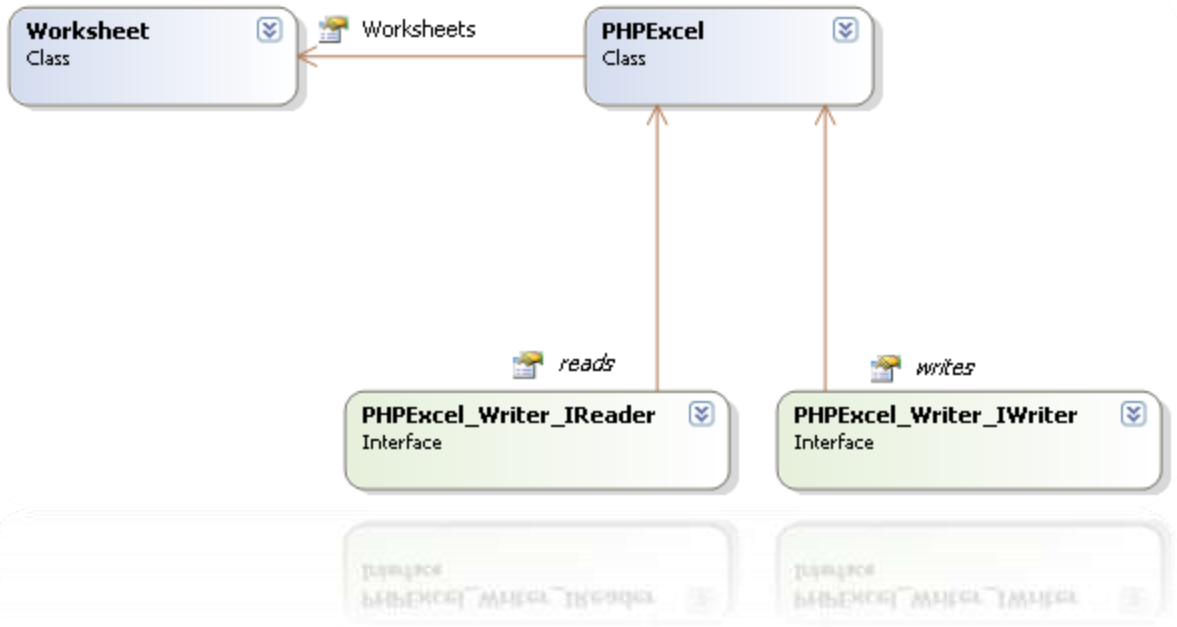
- http://www.phpexcel.net or http://www.codeplex.com/PHPExcel

PHPExcel

PHPExcel – Usage scenarios

- Generating Excel documents on server
- Reporting from PHP applications in different file formats
- Processing existing Excel documents on server
- Importing data from Excel document
- Re-using business logic stored in Excel document (check my blog!)
- …

PHPExcel - Architecture



- Core class library featuring spreadsheet engine
- Supported by IReader and IWriter instances

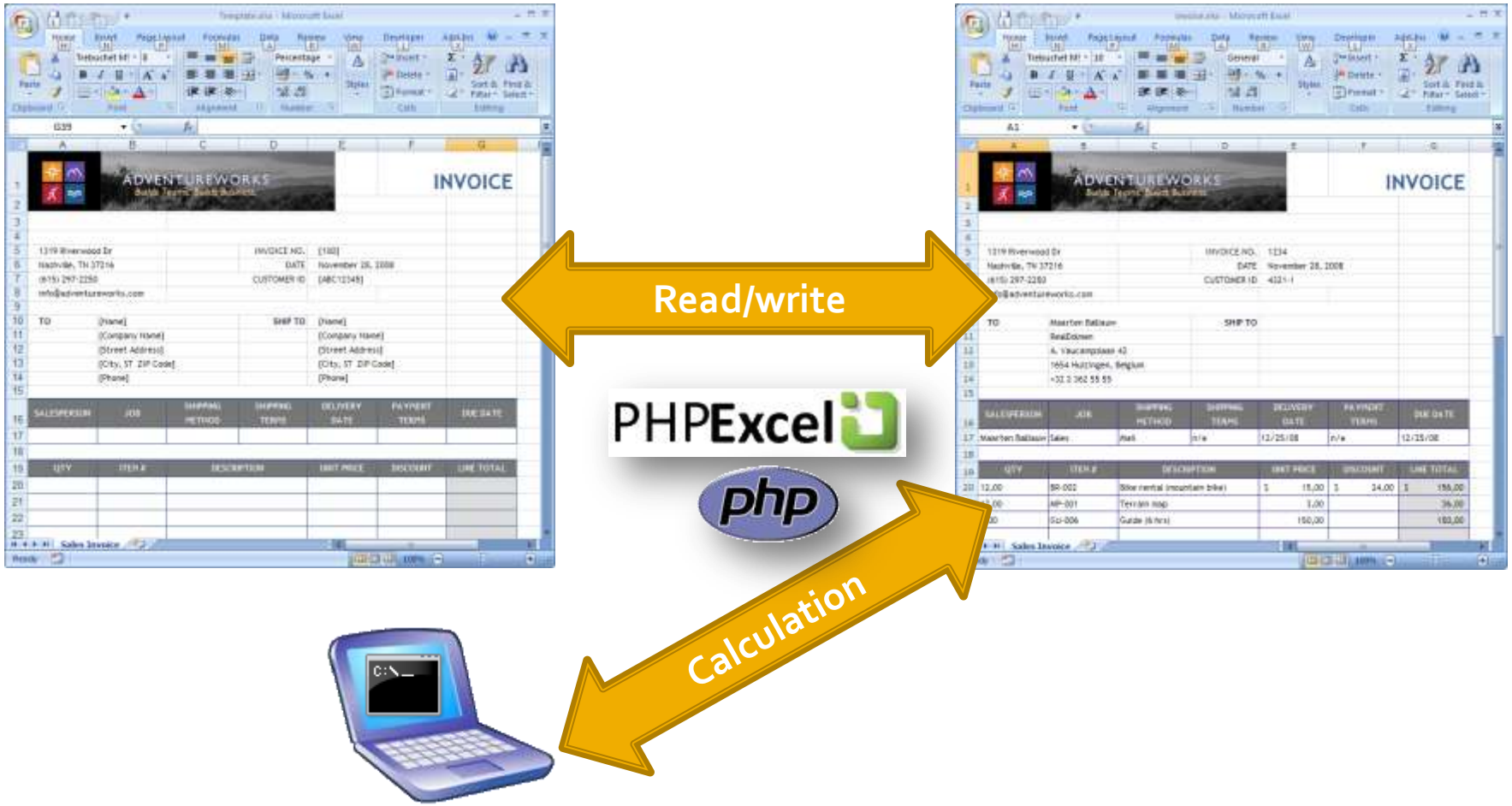
PHPExcel

PHPExcel – Readers and writers

- Readers for Excel2007, Excel5, CSV, Serialized format
- Writers for Excel2007, Excel5, CSV, HTML, PDF and Serialized format

PHPExcel

Interoperability demo



Before we start…

PHP

- $x = 5;
- $x = “hello”;
- $x = “Hello ” . “world”;
- echo “hello !”;

C#

- int x = 5;
- string x = “hello”;
- string x = “Hello ” + “world”;
- Console.Write(“hello!”);

DEMO

PHPExcel and OPENXML4J

- PHPExcel

- OPENXML4J

Open XML for Java

OpenXML4J

- OpenXML4J launch May 30th, 2007

OpenXML4J

- Multi platform API for Java client/J2SE and serveur/J2EE applications
- Open Source Project under Licence BSD (dual license Apache V2)
- Official website : http://www.openxml4j.org/
- Based on ECMA 376 specifications

Why OpenXML4J and why open source ?

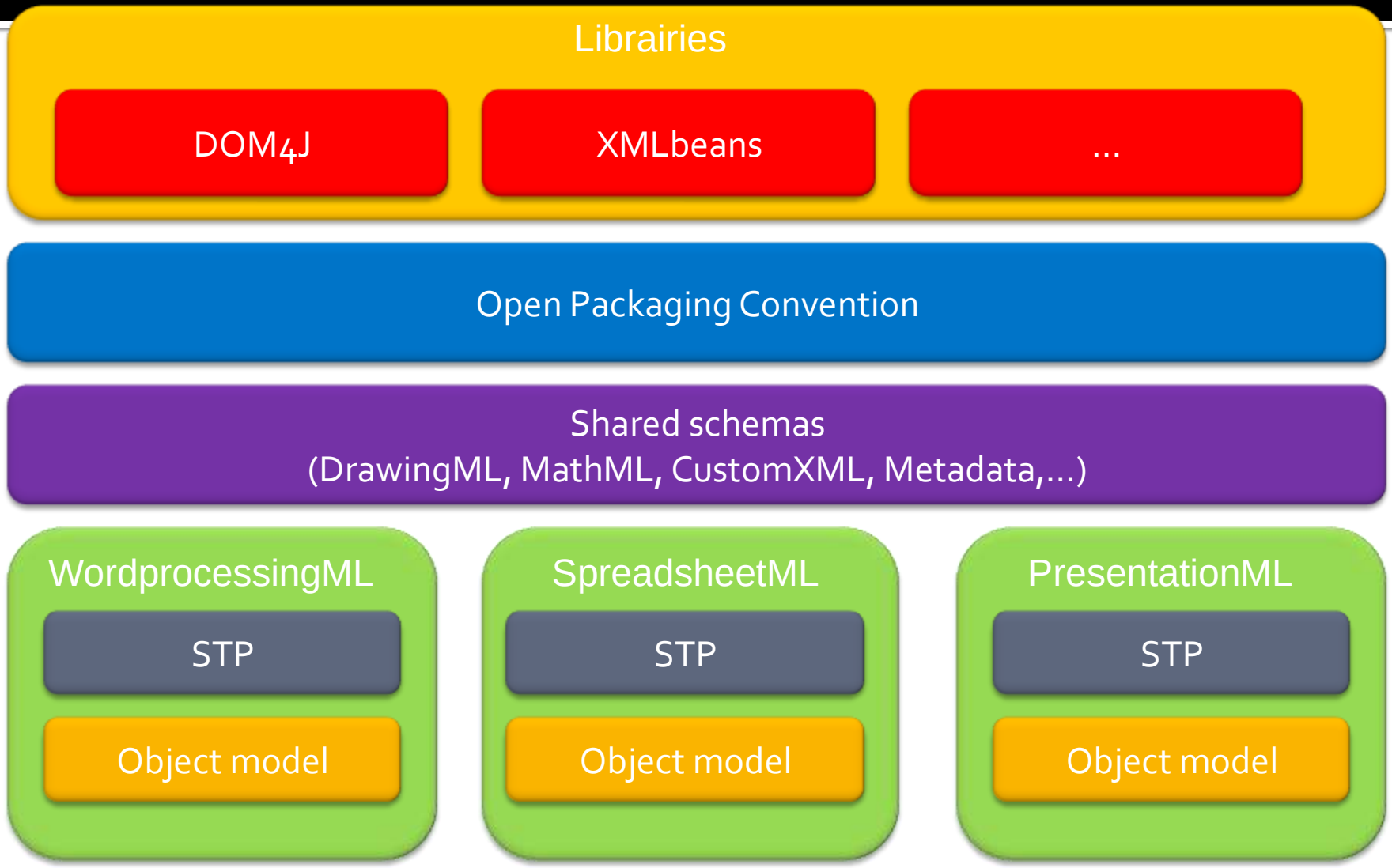
- Unified companies and communities efforts into a concrete and common implementation
- Promote Open XML Interoperability in heterogeneous environments

An open standard, an open implementation !

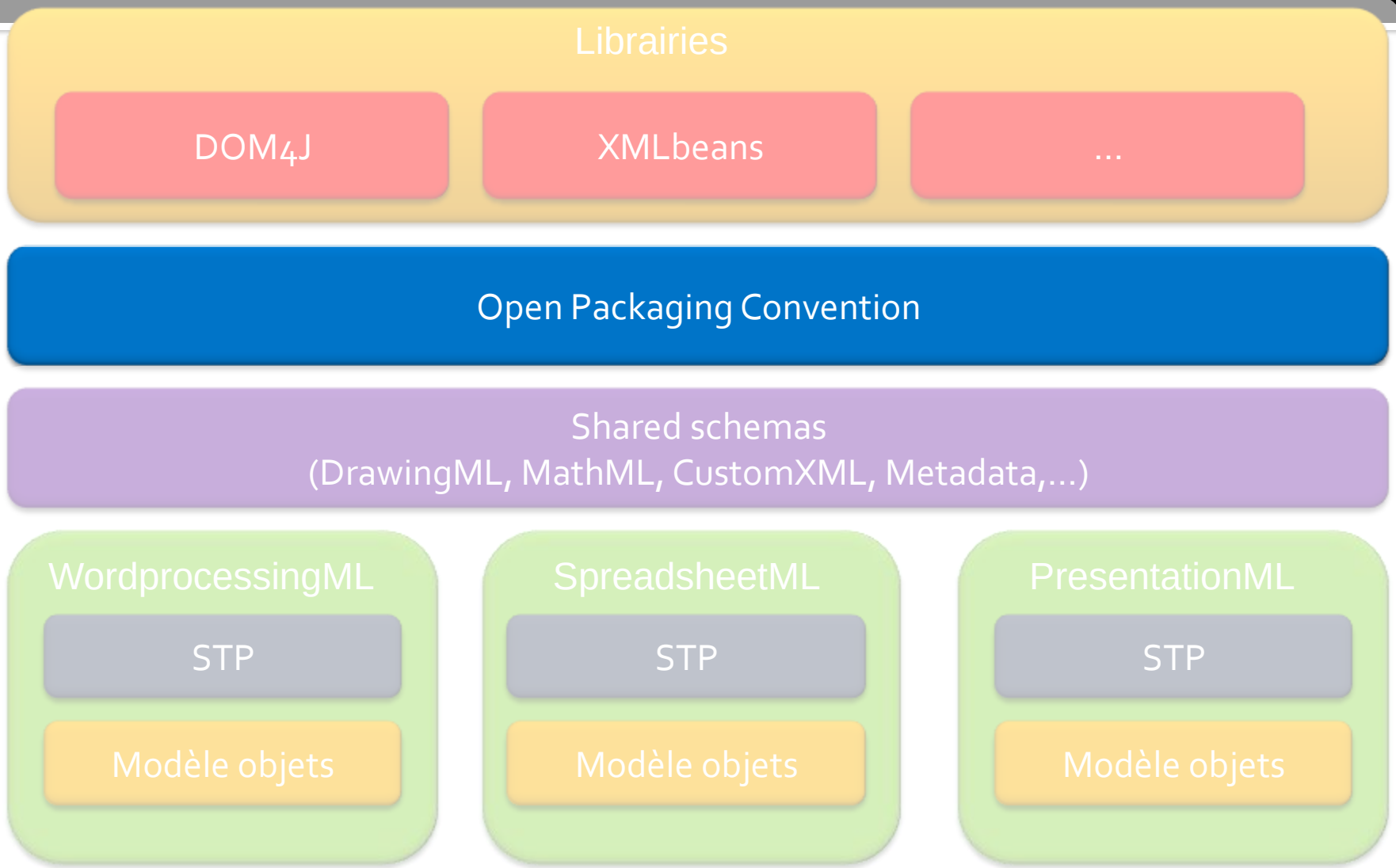
Initial steps overview

- Several steps:
  1. Implementation of *Open Packaging Convention* (part 2)
  2. Implementation of typed parts : *WordprocessingML, SPreadsheetML and PresentationML*
  3. Implementation of an object model for each format
- Now, only *step 1* is implemented. We need contributors and … time !

Overview


Librairies
DOM4J
XMLbeans
...
Open Packaging Convention
Shared schemas
(DrawingML, MathML, CustomXML, Metadata,...)
WordprocessingML
STP
Object model
SpreadsheetML
STP
Object model
PresentationML
STP
Object model

Overview : Open Packaging Convention



A .NET API clone ?

```
import org.openxml4j.opc;
...
Package p = Package.open(filepath,
PackageAccess.READ);
      for (PackagePart part : p.getParts())
            System.out.println(part.getPartName() +
" -> " + part.getContentType());
```

```
using System.IO.Packaging;
...
Package p = Package.Open("",
System.IO.FileMode.Open);
      for (PackagePart part in p.GetParts())
            Console.WriteLine(part.Uri + " -> " +
part.ContentType);
```

A .NET API clone ?

```
Package p = Package.open("c:\....docx",
PackageAccess.READ);
PackageProperties props = p.getPackageProperties();
System.out.println("Title: " +
props.getTitleProperty().getValue());
p.close();
```

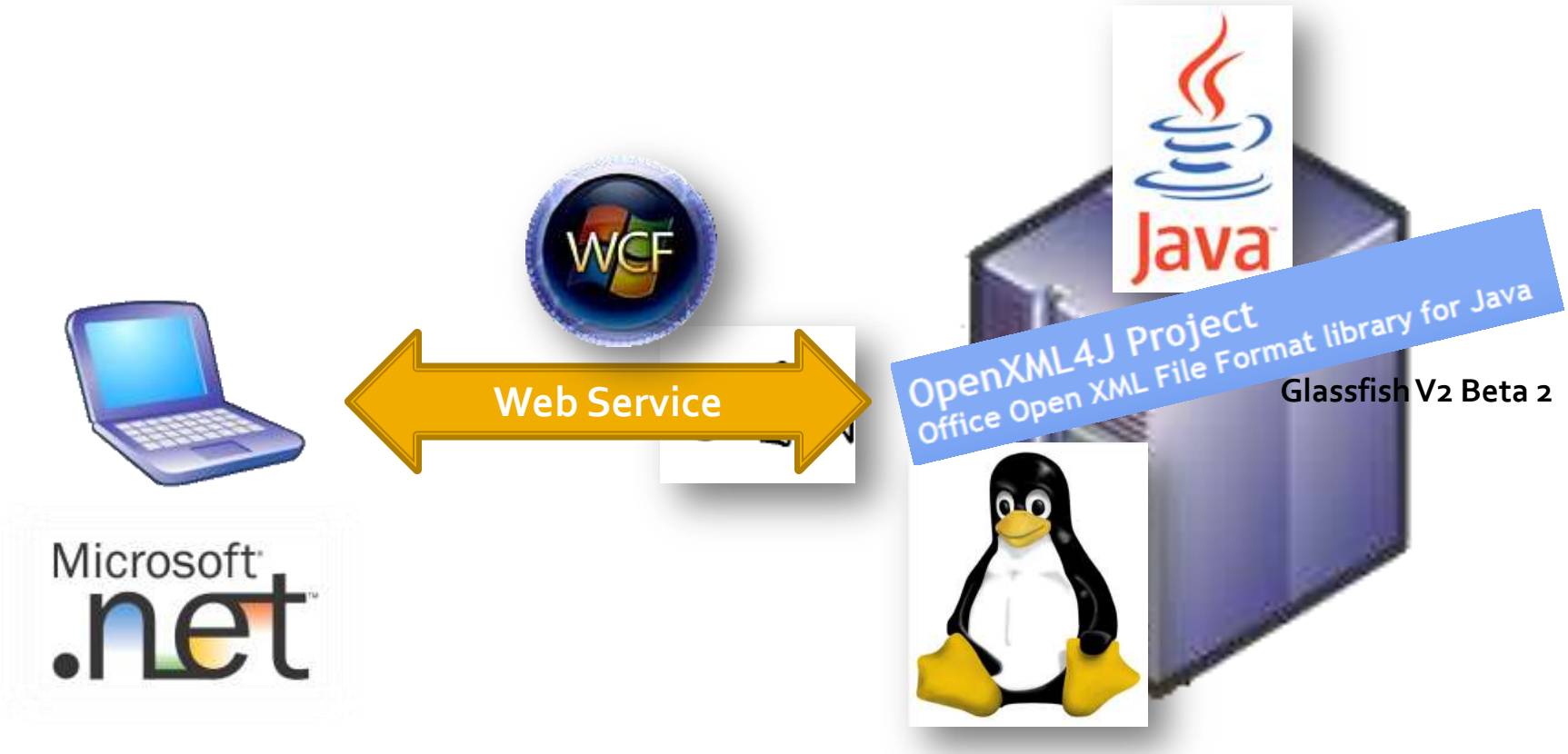
```
Package p = Package.Open("c:\....docx",
FileMode.Open);
PackageProperties props = p.PackageProperties;
Console.WriteLine("Title: " +
props.Title);
p.Close();
```

Roadmap

- Beginning in 2006 :
  - Now uses by several open source projects :
    - POI (http://poi.apache.org)
    - Doc4J (http://dev.plutext.org/trac/docx4j)
  - Several projects in companies (French)
- Today we're joining the Apache foundation and the POI project to offer a single API and point of contact to the community

DEMO

Interop demo

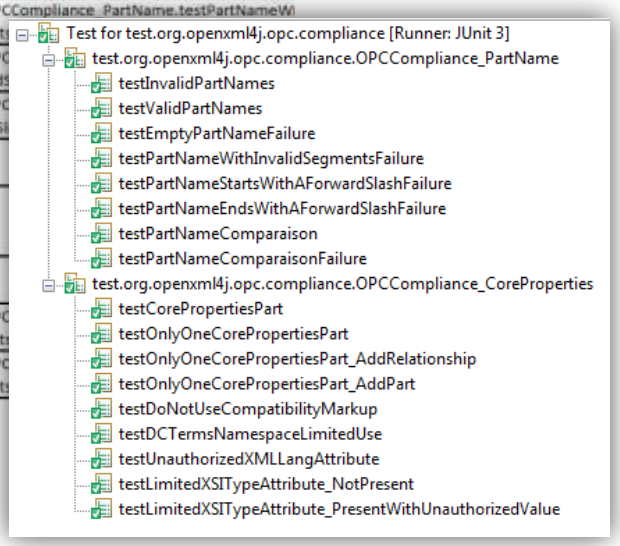


DEMO

Specifications conformity

- Close to the ECMA standard specifications→ Validation tests

ID	Rule	Reference	Package implementer	Format Designer	Format Producer	Format Consumer	Statut	Check location	Associated Test
M1.1	The package implementer shall require a part name.	8.1, 8.1.1	×				OK	PackagePart class and PackagingURIHelper.isValidPartName	OPCCompliance_PartName.testEmptyPartN
M1.2	The package implementer shall require a content type and the format designer shall specify the content type.	8.1	×	×			OK	PackagePart class	NA
M1.3	A part name shall not have empty segments.	0	×				OK	PackagingURIHelper.isValidPartName	OPCCompliance_PartName.testPartNameW ents
M1.4	A part name shall start with a forward slash ("/") character.	0	×				OK	PackagingURIHelper.isValidPartName	OPC ards
M1.5	A part name shall not have a forward slash as the last character.	0	×				OK	PackagingURIHelper.isValidPartName	OPC rdSl
M1.6	A segment shall not hold any characters other than pchar characters.	0	×						
M1.7	A segment shall not contain percent-encoded forward slash ("/"), or backward slash ("\") characters.	0	×						
M1.8	A segment shall not contain percent-encoded unreserved characters.	0	×						
M1.9	A segment shall not end with a dot (".") character.	0	×				OK	PackagingURIHelper.isValidPartName	OPC ents
M1.10	A segment shall include at least one non-dot character	0	×				OK	PackagingURIHelper.isValidPartName	OPC ents
M1.11	A package implementer shall neither create nor recognize a part with a part	8.1.1.1	×						



POI

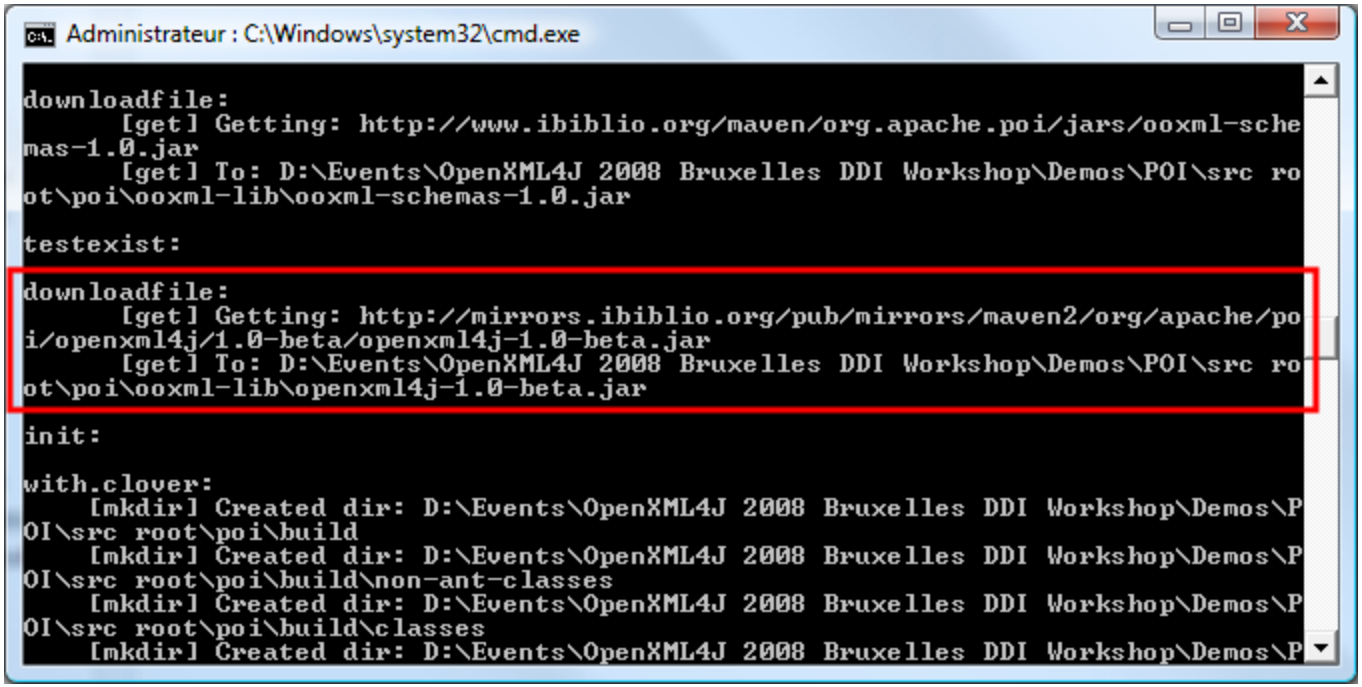
- A set of APIs to read and write Excel files using Java
- Still very active since April 2001: Top Level Project within Apache !
- Also support several MS file formats : Word, PowerPoint, Visio, Publisher
- Version 3.5 will support XLSX and PPTX

POI APIs

- Several APIs for several file formats:
  - **HSSF and XSSF**: Read and write Excel documents (97 – 2003 and 2007)
  - **HWPF**: Read (and partly write) Word 97 documents. Early stages of developement.
  - **HSLF**: Read and write PowerPoint 97-2003 documents.
  - **HPSF**: Read and write OLE 2 property (title, author, etc)
  - **HDGF**: Read at very low level (and simple text extraction) Visio 97-2003 documents
  - **HPBF**: Read at very very low level (and simple text extraction) Publisher 98-2007 documents

POI – OpenXML4J

- POI compilation with Ant :



DEMO

Questions?

Thank you!

- Maarten Balliauw
  - http://blog.maartenballiauw.be
  - maarten@maartenballiauw.be
- Julien Chable
  - http://blogs.developpeur.org/neodante
  - julien@wygwam.com