



# Mazda Advanced web services & PHP Strengths & Flaws



**Olivier Lépine** – CEO Quatrain Technologies



ZEND/PHP CONFERENCE OCTOBER 19-22, 2009

# About Myself

Olivier Lépine (not Lã^pine nor any other charset mess)

Live & work in Avignon, Provence, France, Earth

Graduated in Communication in 1994

Founded Quatrain in 1997 as a Web Agency

Started to play with PHP & MySQL in 1998

Now develop, maintain & host critical PHP web applications

Finishing the first release of t41 (t-4-1) our “4GL RAD Toolkit” over

Zend Framework (t41.quatrain.com)

Apart from geekery, I enjoy Music, Art, Flying, Series & Movies,

Cooking, etc.

# What are we talking about ?

---

The what, the when and the why

# The SIV Project

---

Presentation and challenges

# SIV project in a nutshell (1/2)

SIV stands for “Système d’Immatriculation des Véhicules”  
(Vehicles Registering System : like US DMV)

Nationwide system managed by the Ministry of the Interior

Web Services based on SOAP 1.2 specification

More than 30 different possible actions

Source : <http://www.ants.interieur.gouv.fr/siv/-siv-.html>

# SIV project in a nutshell (2/2)

250 XSD definition files with dependencies

Messages can be either synchronous (immediate response) or asynchronous (delayed response) depending on action type

Messages can contain attachments in MIME format

Communications are made via HTTPS with SSL client certificate authentication (client IP address is also validated)

SAML (postponed to 2010)

# SIV project workflow

Customer wants to buy a car



Dealer enters data & requests registering

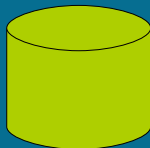


Controls are performed  
Request is sent to PHP



mazda

The Mazda Implementation



Car registration document is directly sent to customer

Ministry transmits Data to print doc

Request is sent by PHP  
Over the internet with HTTPS

Response is sent, containing a PDF document if successful

Government Printing Office



Ministry of the Interior

# Project Context

---

# France Automobile Industry

87 Billion € Turnover (2006) - roughly \$130 Billions

More than 3,000,000 new vehicles registrations per year

35 different makes available in France

Chosen languages for SIV interfaces development :

- Java
  - One packaged application
  - custom development
- .Net
- PHP

# Mazda Automobiles France

258 M € Turnover (2008) - roughly \$385M

16,000 new vehicles registrations per year

160 car dealers

## Information System

IBM i Servers (525 w/ OS/400 V5R4)

Applications are ERP, CRM, DMS...



**mazda**

# SIV Specifications

---

And how to understand them

# A definition of SOAP (1/3)

W3C states :

*“SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP can potentially be used in combination with a variety of other protocols; however, the only bindings defined in this document describe how to use SOAP in combination with HTTP and HTTP Extension Framework.”*

Source : <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

# A definition of SOAP (2/3)

Wikipedia states :

*“SOAP is a protocol specification for exchanging structured information [...] in computer networks. It relies on Extensible Markup Language (XML) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission.”*

... and adds :

*“Although SOAP is an open standard, not all languages offer appropriate support. Java, Delphi, .NET and Flex offer excellent SOAP integration and/or IDE support. Python and PHP support is much weaker”*

Source : <http://en.wikipedia.org/wiki/SOAP>

# A definition of SOAP (3/3)

*“**HTTP-based** Client/Server Communication based on Messages composed of an **XML-formatted** array of **structured data**”*

*Obviously partial but defines the constraints to be met.*

# SOAP With Attachments

*It is not that complicated, but it's really ugly*

```
-----=_Part_a0342cd4096cdc8773a622
Content-Type: text/xml; charset=UTF-8
Content-Transfer-Encoding: binary
Content-Id: <456F6BA913B27CAA3301C0B9363B202A>

<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope
xmlns:ns="http://siv.mi.fr/DefinitionsServices/2007-06"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Body><depot_notif
_immat_sur_vehicule_neuf_marque xmlns="http://siv.mi.fr/DefinitionsServices/2007-06"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://siv.mi.fr/DefinitionsServices/2007-06
conf/siv/xsd/ImmatSivService/asynNotifImmatSurVehiculeNeufMarque/onBody/brp_asynNoti
fImmatSurVehiculeNeufMarque_Dep.xsd">[...]<xmlMessageFichier
href="cid:0621F2E74AFB4B486F0C62D77E9FAD43" xsi:type="ns1:octet-stream"
xmlns="http://siv.mi.fr/DefinitionsService/2007-06"
xmlns:ns1="http://xml.apache.org/xml-soap"/></soapenv:Body></soapenv:Envelope>
-----=_Part_a0342cd4096cdc8773a622
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Id: <0621F2E74AFB4B486F0C62D77E9FAD43>
[...]
-----=_Part_a0342cd4096cdc8773a622--
```

# The Application

---

Architecture, Integration, etc.

# Application Definition

An interface between Mazda Registering Application and the Ministry of the Interior Web Services

No GUI, just a front controller called from CLI

Capable of calling SOAP WS and handling attachments

Capable of fetching data from db2 and update some of them

Capable of calling RPG programs

Capable of handling SSL client certificate

Capable of working through a proxy

# Application principles (1/2)

- ✓ Based on Zend Products
- ✓ Zend Core for i5/OS
- ✓ Use i5 Toolkit Functions (i5\_\*)
- ✓ Zend Framework
  - ✓ Zend\_Db, Zend\_Db\_Select
  - ✓ Zend\_Config
  - ✓ Zend\_Http\_Client
  - ✓ etc.

# Application principles (2/2)

- ✓ Best Practices

- ✓ Abstractize & Extend

- ✓ Pluggable at all ends (all components use adapters)

- ✓ Design Patterns

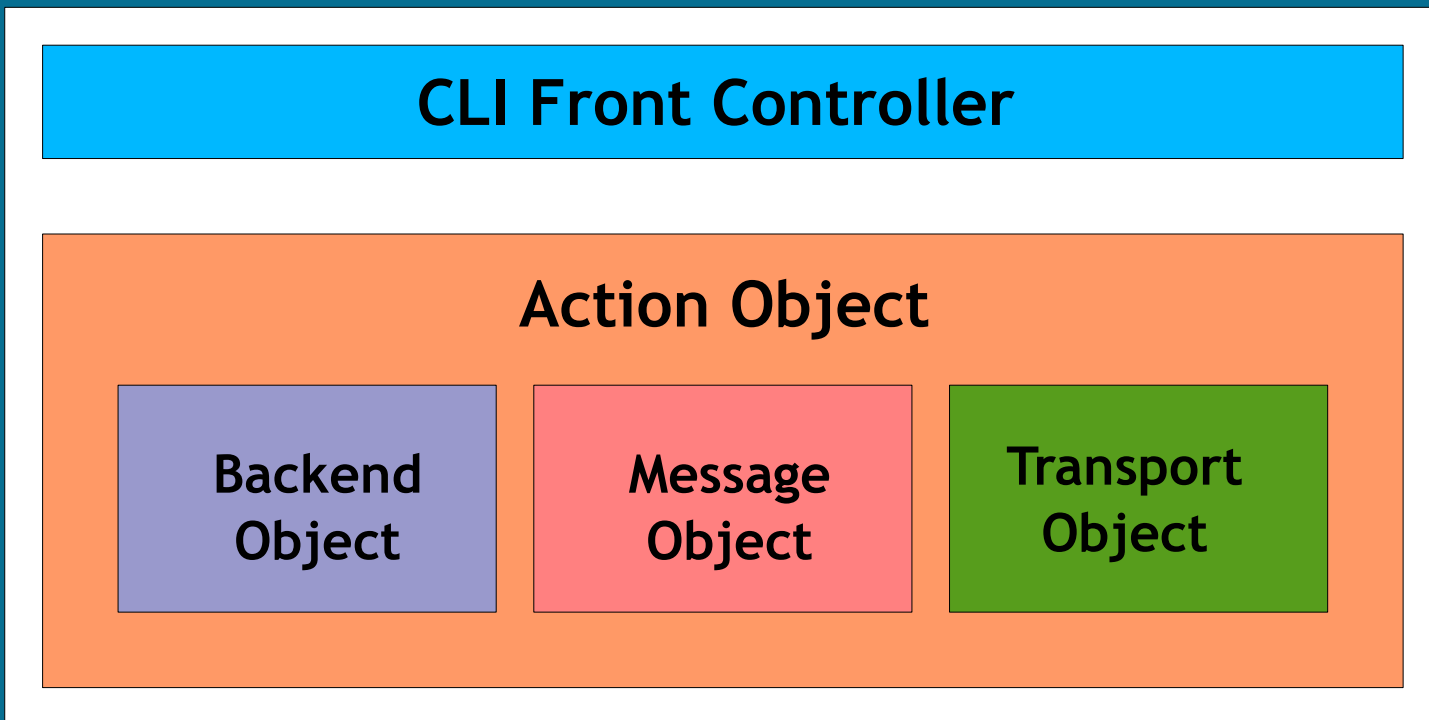
- ✓ Factory

- ✓ Singleton

- ✓ Adapter

- ✓ Dependency Injection

# Application Architecture



## Front Controller :

- Reads parameters from command line
- Instanciates and injects all needed objects
- Triggers Action object
- Handles exceptions

# How the front controller works (1/2)

- Gets parameters from command line (id, action, transport)
- Instanciates all needed objects
  - Backend Adapter (Mazda adapter)
  - Message Adapter (DomDocument-based adapter)
  - Transport Adapter (Soap or Zend Http Client Adapters)
  - Action Adapter selected from context :
    - Synchronous
    - Asynchronous call (demand)
    - Asynchronous callback (response)
- Inject all objects in Action object

# How the front controller works (2/2)

- Triggers Action object prepare() then execute() methods
  - Data are acquired from backend
  - Message is composed, building sections recursively
  - Message is validated against WSDL & XSD Definitions
  - Message is injected in Transport and sent
  - Response is parsed and data sent back to Backend

# Available options for Transport

## High Level solutions (SOAP)

- ✓ ~~WSO2-Framework~~ (not available for IBM i)
- ✓ PHP Built-in SOAP Client
- ✓ NuSoap

## Low Level solutions (HTTP)

- ✓ Curl
- ✓ Zend Http Client

# SOAP With Attachments

*Some headers sent before data are very important :*

```
POST /axis/services/ImmatSivService HTTP/1.1
Host: siv-vabf-part.interieur.gouv.fr
Accept-encoding: gzip, deflate
SOAPAction: "ImmatSivService:depReqNotifImmatSurVehiculeNeufMarque"
Accept: application/soap+xml, multipart/related, text/*
Content-Type: multipart/related; type="text/xml";
start="<456F6BA913B27CAA3301C0B9363B202A>"; boundary="----
=_Part_a0342cd4096cdc8773a622"
Content-length: 2860
```

*You need to keep full control over them*

# Selected components for Transport

## High Level solutions

- ✓ WSO2-Framework (not available for IBM i)
- ✓ **PHP Built-in SOAP Client (using `__doRequest()` method)**
- ✓ NuSoap

## Low level solutions (HTTP Clients)

- ✓ Curl : lacking controls of important headers
- ✓ **Zend Http Client : just the level of control you need**

# Even the Java guys were in hell

Translated excerpts of message from Java developers to Ministry developers :  
“We are using JAX-WS 2.1.1 which indeed doesn't generate a finale CRLF after the last boundary. **We have no option to simply add one CRLF** at the very end of the framework flow.”

[...] To try and solve this problem, we will try to take a newer version of the framework (JAX-WS 2.1.5) to see if we can find more available options to generate a conformant flow for your framework.

If it doesn't work, we will try to patch the framework source.

Ultimately, we will recode all the whole frontend part with another framework !

# Conclusions (Part I)

---

# The Good Part

## Zend Framework

Dom extension (as a XML document builder & validator)

SimpleXml extension (as a fast XML document decoder)

# The not-so-good Part

Various SOAP specifications and implementations. Sometimes shaky.

Problems in SOAP extension that should have been addressed long before (WSDL parsing...)

Lack of native support for WS-\* extensions and attachments for IBM i

# Using PHP for SOAP Pros & Cons

## Pros

- ✓ Versatility : Many possible ways to call WS with PHP keeping compliance with W3C recommandations
- ✓ Hacking : if the current components don't fit you or are buggy, write your own in (almost) no time
- ✓ PHP : Runs on every server, even very small, with no hidden costs

## Cons

- ✓ Support : so far, feedback from community on advanced WS is rare
- ✓ Can' t see any more but I am probably biased

# Conclusions (Part II)

---

# SIV project as Today

Started 15 April 2009 (phase 1 - new cars only)

1,984,750 registered vehicles of all makes since this date,

16,487 registered vehicles in one day (October 8<sup>th</sup>)

Phase 2 (used cars plates upgrade) started on 15 October.

Source : <http://www.ants.interieur.gouv.fr/siv/-siv-.html>

# Mazda Opinion

*“Compared to similar developments made with Java, .Net or Websphere, PHP development cost was 30 to 70% cheaper.”*

The same project was developed using different languages during the same period of time, allowing to compare development and integration costs.

# The Unexpected Reward



On October 15, Mazda got the very first plate : AA-001-AA

# Questions ?

Further questions :

Email : [olivier@quatrain.com](mailto:olivier@quatrain.com)

Skype : [crapougnax](https://www.skype.com/people/crapougnax)

Twitter : [twitter.com/crapougnax](https://twitter.com/crapougnax)

# Thanks to them...

@ Mazda Automobiles France  
Pascal Matruye, SIV Project Leader

@ Zend France  
Christian Durel, General Manager  
Alban Hanry, Professional Services Manager

Thank you !