

# Business Oriented Development with XPE

Dr. Hao He, Chief Architect

"Things should be made as simple as possible, but no simpler."

-- Albert Einstein

# Outlines

- Introduction
- The need for business-oriented project deliveries
- SOA and XPE
- Implementing SOA with XPE
- Case study

# Questions for you

- Which creature on earth do you think is most amazing?
- Would you still use Kung-Fu to fight a war if guns are available?
- Why ERPs have caused some companies to die?
- Which one is better, J2EE or .Net?
- As a developer, which technology should you bet your future on?

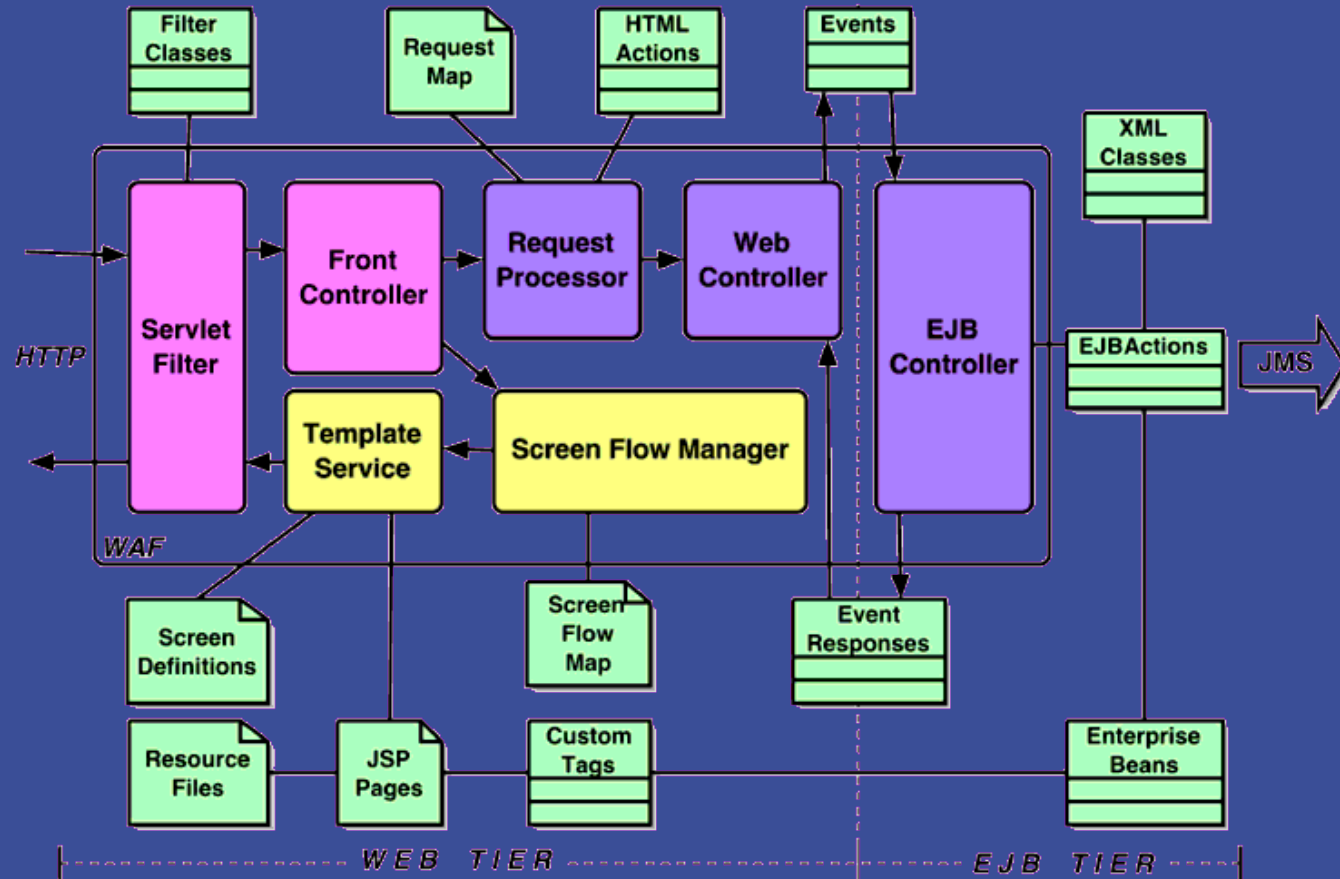
# A Typical Experience of Software Development



Image copied from the Internet

# The MVC (PetStore)

Source: [http://java.sun.com/blueprints/guidelines/designing\\_enterprise\\_applications\\_2e/sample-app/images/sample-app1.3.13.gif](http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/sample-app/images/sample-app1.3.13.gif)



# What is wrong with MVC?

# What is wrong with MVC?

- The partition is not real. It is irrelevant to the business problem you are trying to solve!
- Most customers do not normally care about which technologies you use. What do they care most is the end results, and the ROI.

# SOA Essences

- A service causes a valuable change to its consumers.
- In normal situation, a service is invoked by a consumer only if the total cost of consuming the service is lower than the value the service provides.
- SOA is all about making a service cost-effective to its consumers by leveraging economy of scale.

# Service

- Value – centralised intelligence, or resources
- Cost of consuming – standard interface, and messaging, availability
- Longevity – extensible messaging, and agility for longer life-span
- Lifelike, the fittest survives the best

# SOA impacts on projects

- Projects become business oriented as opposed to technology driven
- Business analysis is no longer just requirements gathering but also more on business value-adding
- Architecture becomes even more important
- A project is more like building service providers
- Developers need to improve their business awareness

# XPE in a nutshell

- It does web, it does XML and it is SOA from ground up.
- XPE – a SOA framework combining the best of the Web and the best of XML.

# The XPE approach to SOA

- Each service is modeled as a value-chain
- Each service is implemented as a pipe composed of filters (value-adder)
- Using HTTP and XML for lowest possible service consuming cost and maximum number of potential consumers
- Leveraging XML for extensibility
- All nasty technical things are taken care by XPE (concurrency, caching, transaction, XML processing, database connection, deployment)

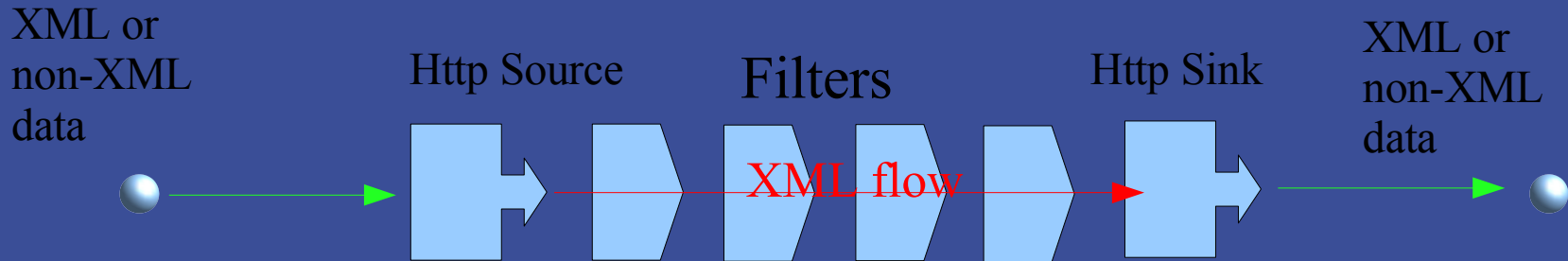


# XPE principles (similar to those of the Web, and lately Microformats)

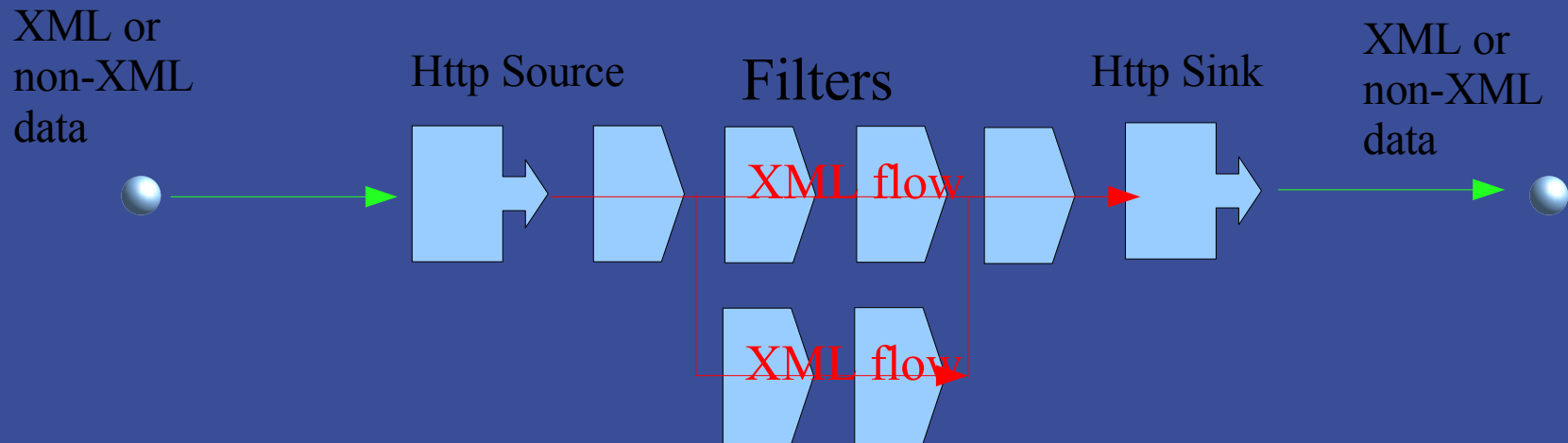
- Solve a specific business problem
- Evolutionary improvements
- As simple as possible
- Directly model the business process
- Reuse rather than reinvent
- Modularity / embeddability
- Decentralized development, content, services

# Service design

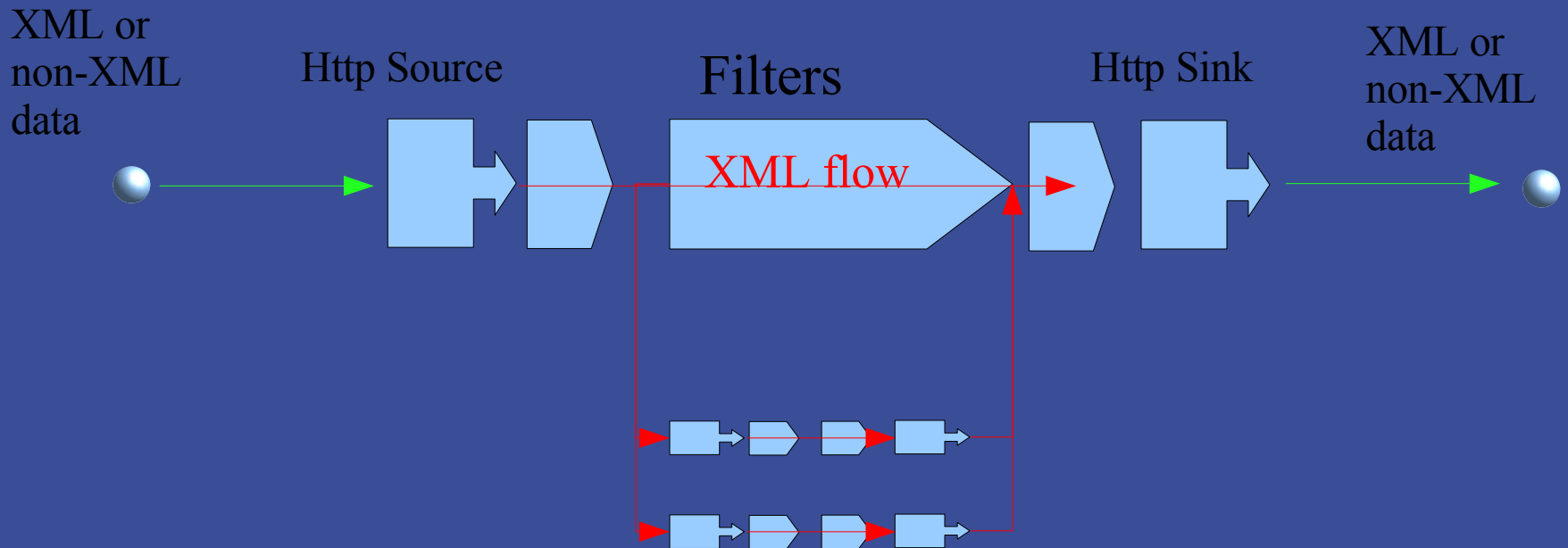
- Why people want to use my service?
- Divide a service into stages
- The pipe model:



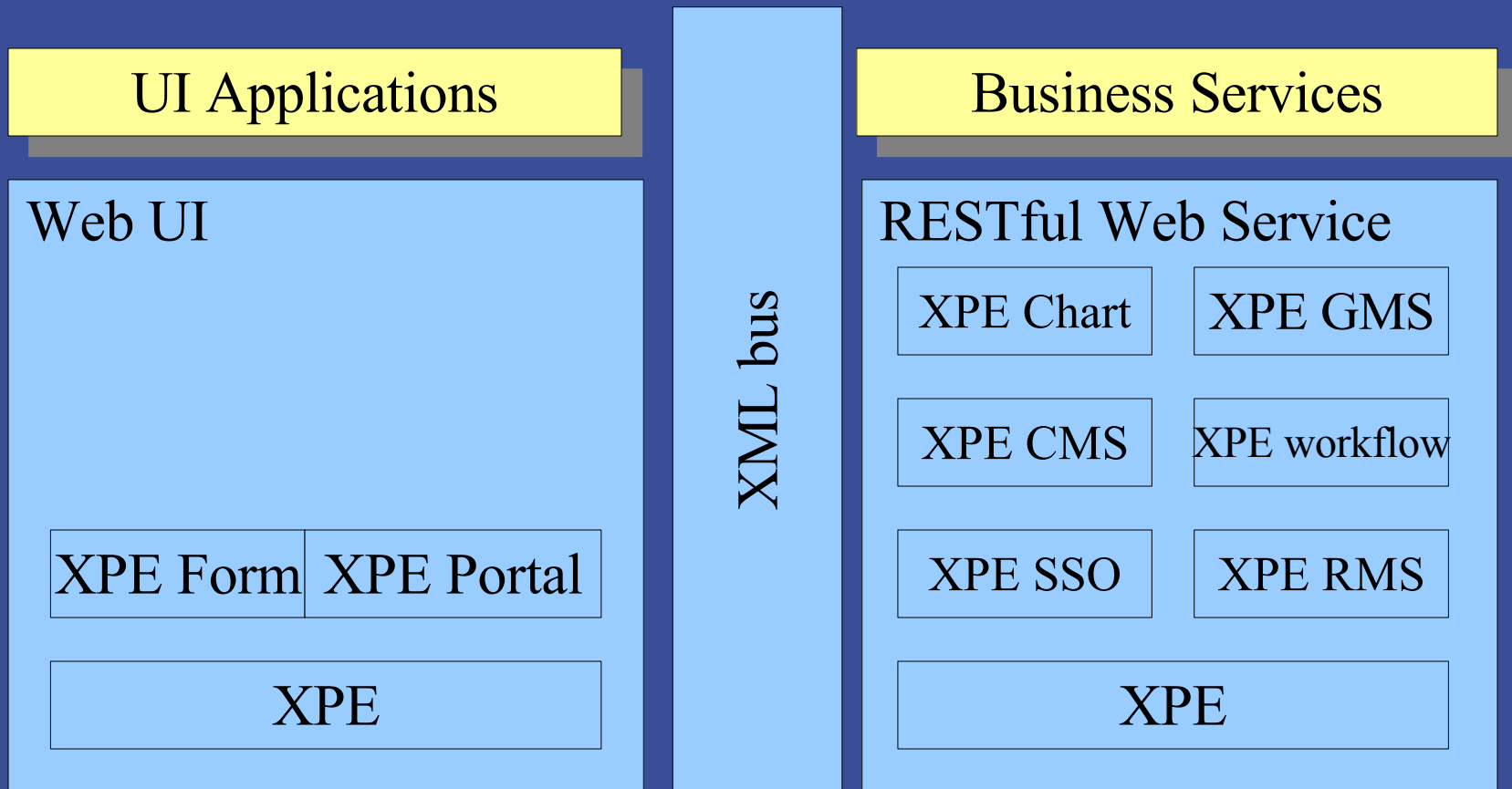
# Service pattern



# Service pattern



# XPE technology stack



# Extending Xinclude

## ■ Nice thing about Xinclude

```
<xi:include href="{ $cms }/cmcore/resource?uuid={ @node }&" >  
<xi:fallback>  
<xi:include href="{ $backupcms }/cmcore/resource?uuid={ @node }&" >  
  <xi:fallback>  
    <error>CM is not available</error>  
  </xi:fallback>  
</xi:include>  
</xi:fallback>  
</xi:include>
```

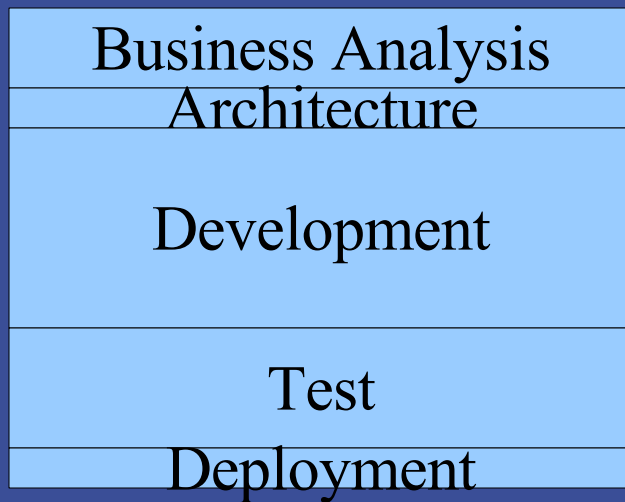
# Extending Xinclude

## ■ Support POST

```
<xi:include href="{ $cms }/cmcore/cmcore/resource?uuid={@node}" >
  <xi:fallback>
    <xi:include href="{ $backupcms }/cmcore/errorReport" method="POST">
      <xi:content>
        <error>Could not access <xsl:value-of select="$cms" /></error>
      </xi:content>
    <xi:fallback>
      <error>CM is not available</error>
    </xi:fallback>
  </xi:include>
</xi:fallback>
</xi:include>
```

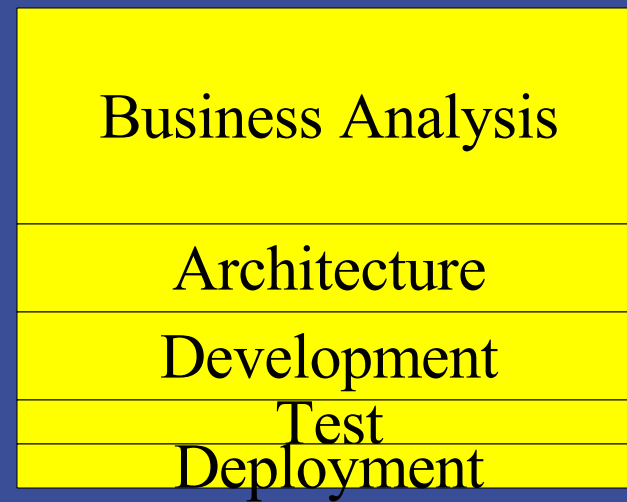
# XPE impact on a project

Relative proportion of major project activities



Business value

Before XPE



After XPE

# XPE mind shift to a developer

## Before XPE

- A programmer
- Programing business rules into programs
- Spending most time solving technical problems
- Active
- Rigid

## After XPE

- A value adder
- Defining business rules as templates
- Spend most time understanding and solving business problems
- Passive
- Flexible

# XPE Productivity

- New graduates can become productive in one month
- Experienced developers can make transitions in 2 weeks
- Experienced developers can finish a project in less than 20 percent of the time it took them using other technologies

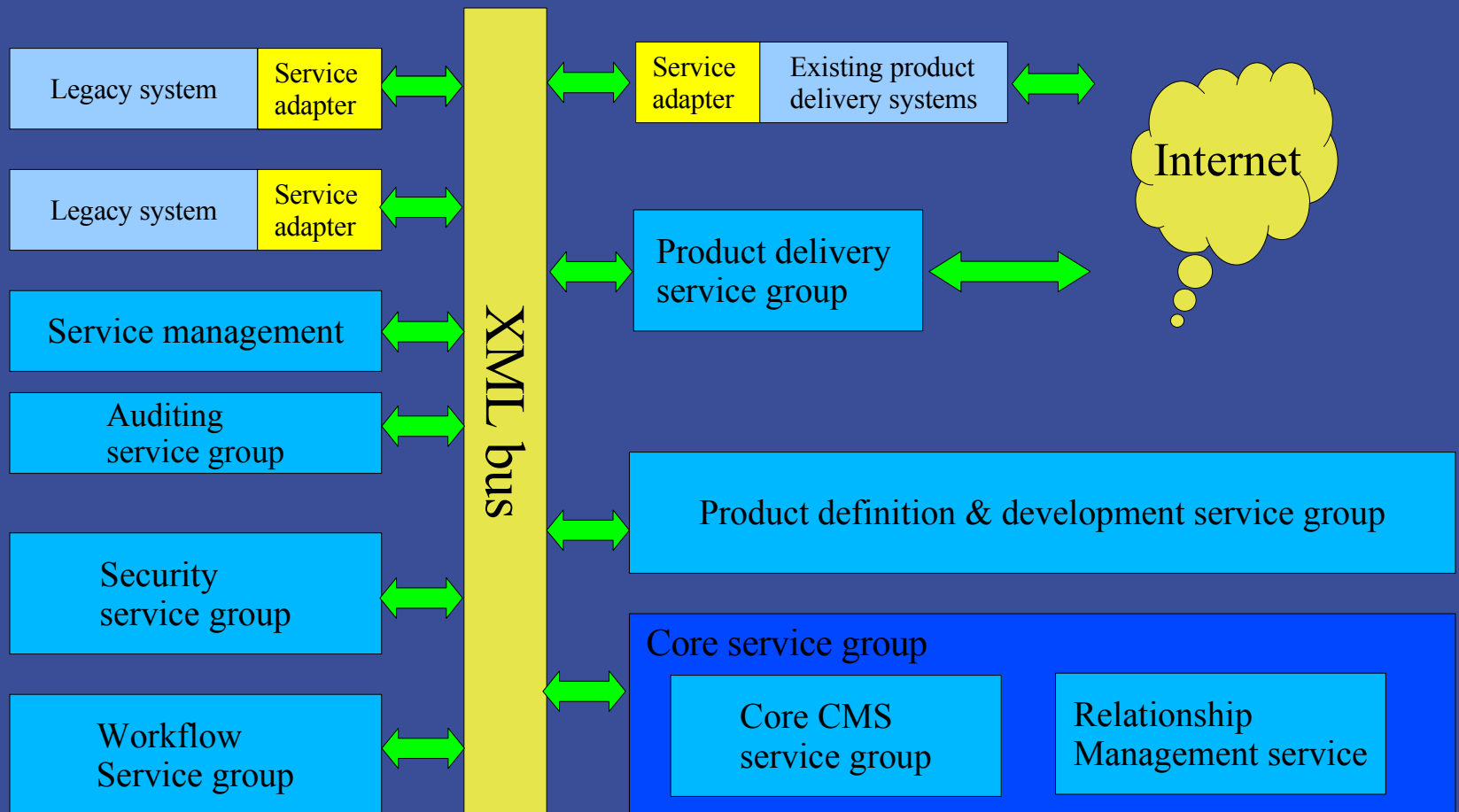
# XPE productivity - example

- An on-line shop can be decomposed into around 40 pipes
- On average, a pipe can be constructed in 1 hour.
- The more one builds, the more services can be reused.

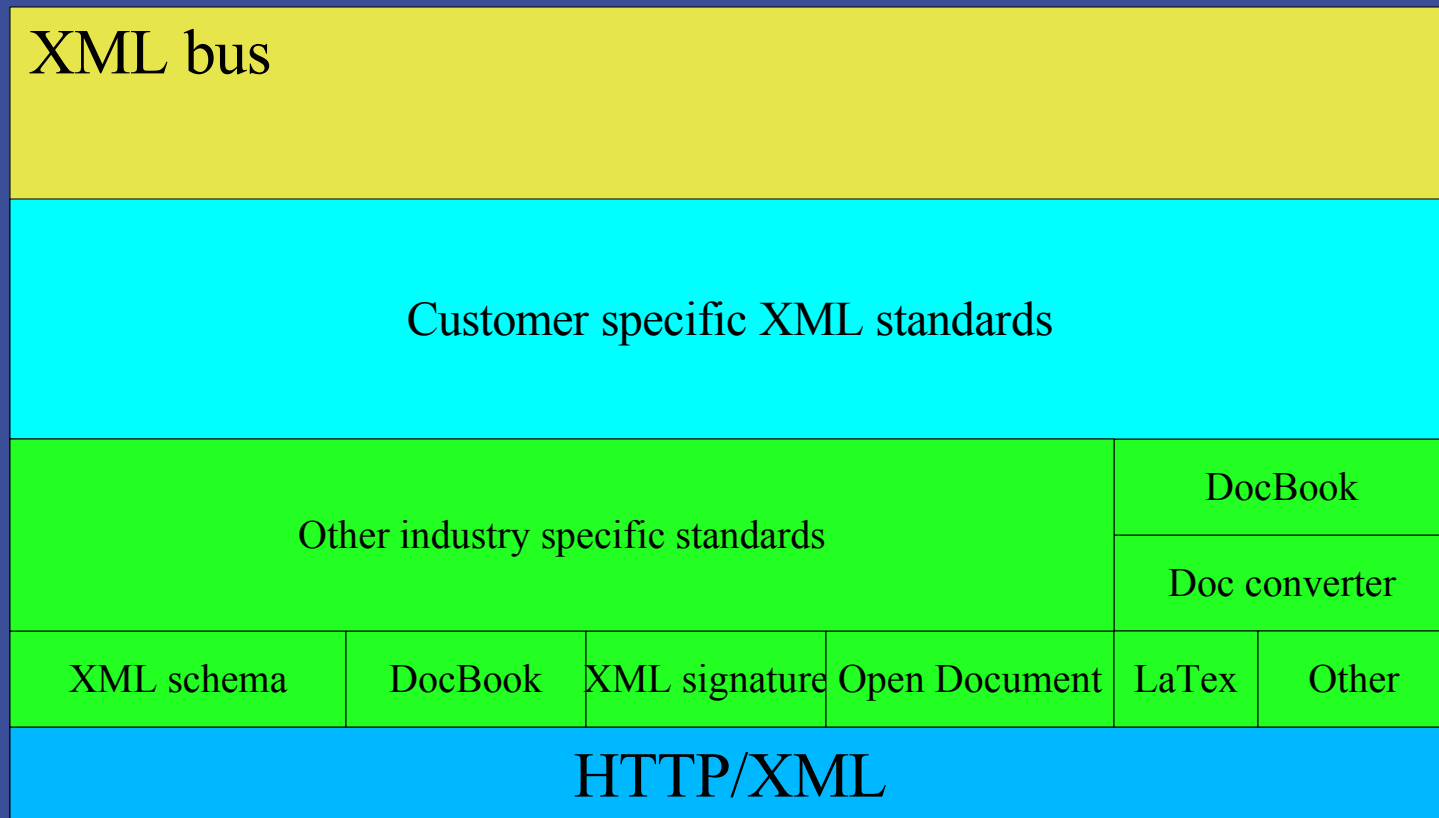
# Case study

- A CMS project

# High-level architecture



# What is inside the bus?



# Lesson learned

- A business oriented approach is essential. Always ask “what value am I contributing?” Business leads technology.
- Educate team about the architecture early on
- Flexible services were impressive to customers. Nobody can get it right first time but having the ability to get it right second time is important.
- A developer needs to understand the business.

# What others have been saying about XPE?

- ... from the our short engagement with XPE it seems that your paradigm deals directly with the complexity of the data, whereas the paradigm we are familiar with tries to manage the complexity by writing java programs. Also, your engine seems to be more efficient and robust than J2EE, at least within the intranet."  
-- Vagelis, CIO

# How long does it take to train a XPE developer?

- New graduates fresh from university: one month.
- Experienced developer with a willing mind to change: two weeks.
- Experienced developers with good XML skills: one week.

# Questions?

# XPE – life like computing