

# AlternativeFrameworks

## Objectives

Cocoon has started as a simple XML/XSLT processing framework and has grown over the years to become much more than that. This page should be sort of a technological overview of what is Cocoon and how it integrates into nowadays server-side development background.

### Caution

*This content is inevitably biased by the personal experience and point of view of people who write it. It's meant to be as objective as possible and shouldn't become a competition argument between different tools (whether Open Source or not). So if you feel this content doesn't represent reality so well, feel free to edit this page and add your own user experience.*

## What is Cocoon ?

Let's position ourselves in the context of Java server-side development. Java because .NET has its own tools, most of them are commercial and very specialized products and when you come to develop your first server-side application, you are likely to be forced to make a choice between those two worlds (or even more exotic ones like Python/Zope for example). And server-side tools that are compatible with both environments are quite rare. And server-side because desktop or system development is completely out of the scope of Cocoon.

That said, there are many technologies involved in Java server-side development (or J2EE). For instance, you might be familiar with the following concepts (FIXME : it would be great to associate some links with those different concepts):

- web applications
- application server
- servlets, JSP's
- databases/persistence
- transactions
- security
- J2EE design patterns (MVC, SoC, etc.)
- 3-tier development
- etc. (to be extended...)

And sometimes it can be somewhat tricky to understand how those concepts fit together and especially what tools implement which concepts. There are so many development environments and frameworks in that field that you can get a few headaches while trying to make the link between your specific needs, your solution's architecture and which tools to choose to help you implement that solution.

So here we are : let's see things from a newbie point of view. I'm a developer, I know many general concepts about Java server-side development and I have to start a new project which implies certain features and certain needs. So according to this, how can Cocoon be part of (or all of 😊) my solution and what are the other tools that could be useful to consider (even if obviously Cocoon is the best one ;-P).

This question can be sort of a guideline to organize and structure this technological overview. And for each aspect there should be a first short paragraph explaining what technologies are used in Cocoon to play that role, followed by one short paragraph for each alternative tool in that domain, ideally extracted from project's website but possibly completed by experience feedback and personal opinions.

⚠ There is already a Wiki page, [SimilarServers](#), which gathers information about alternative tools, mainly XML-oriented. This content is very interesting as soon as you have determined that XML will be an imperative need for you. Maybe it could eventually be merged with this page...

As far as I see it, Cocoon can be viewed as a...

## XML processing framework

Cocoon : SAX implementation, Xalan integration, etc.

Other tools : PHP5/libxslt ...

## XML Publication server

Cocoon : sitemap, pipelines, etc.

Other tools : Orbeon [PresentationServer](#), XMLRAD [?](#)

## MVC framework

Cocoon : M = database logicsheets/SQL transformer, V = XSLT/CForms, C = sitemap/flowscript

Other tools : Jakarta Struts, Spring, [WebWork...](#)

## Web application abstraction

Cocoon : in fact Cocoon is itself a web application which provides a supplementary abstraction layer above J2EE to manipulate XML files instead of JSP's or servlets

Other tools : ...

## Typical configurations

It could be a good idea to list a few typical architectures where Cocoon has been successfully used like:

- Cocoon as a presentation layer : [Cocoon/Hibernate/Spring](#) :
- Cocoon as a core publication server : [Lenya](#), [Daisy](#), [Myotis](#)
- etc. (to be extended)

## Future extensions

Cocoon has grown quite fast. It's only in version 2.1 and it has already covered much more than its initial purposes. And with blocks it's going even further. For example...

- With a better integration of Axis block into Cocoon, it can become a real Web Service producer/consumer (see [WebServiceServer](#))
- With recent Sylvain's work on AJAX implementation with Cocoon forms, it could become an interesting interactive GUI development framework
- etc. (to be extended)

Of course for each of those possible evolutions, it could be interesting to see which alternative tools will be concerned.

## Conclusion

Feel free to give your own personal experience in order to enrich the collective one and help newbies choose their development tools.