
Apache Forrest presents

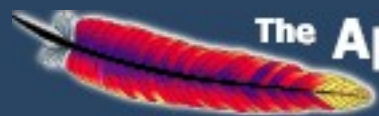
The dispatcher

-

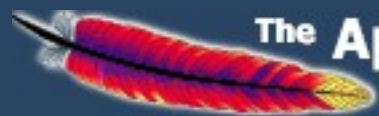
**Advanced separation of
concerns in web-development**

Thorsten Scherler
thorsten@apache.org

- University of Paderborn, Germany
- Master in business administration (2001)
- Master in business computing (2003)
- Consultant, Alcántara sistemas de información (2003)
- COO, Wyona Spain (2005)
- First open source project - Wyona CMS (2001)
- Apache Lenya committer and PMC member (2002)
- Apache Forrest committer and PMC member (2003)



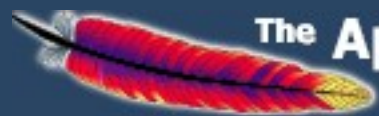
- 1979 Sendmail (mail transfer agent [MTA]) – Eric Allmann
- 1983 GNU (free software operating system) – Richard Stallmann
- 1989 HTML and HTTPD - Sir Berners-Lee
- 1991 Linux (Linus' Minix - kernel) - Linus Torvalds
- 1995 Apache Web Server release by the Apache Group
- 1998 Mozilla Organization founded by Netscape
- 1998 XML W3C recommendation and first version of Cocoon
- 1999 Apache Software Foundation (ASF) founded



Apache Forrest is a publishing framework that transforms input from various sources into a unified presentation in one or more output formats.

The modular and extensible plugin architecture is based on Apache Cocoon and relevant standards, which separates presentation from content.

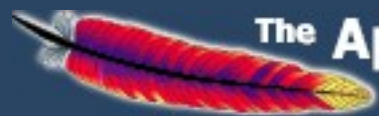
Forrest can generate static documents, or be used as a dynamic server, or be deployed by its automated facility.



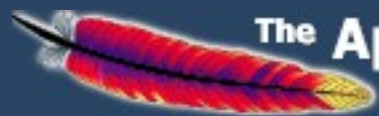
Apache Forrest was started in January 2002, with the major use-case being to create a consistent and efficient xml.apache.org group of project websites with a uniform, lightweight, easy-to-navigate layout and structure.

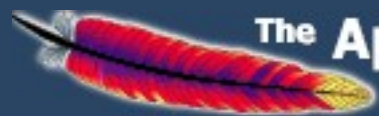
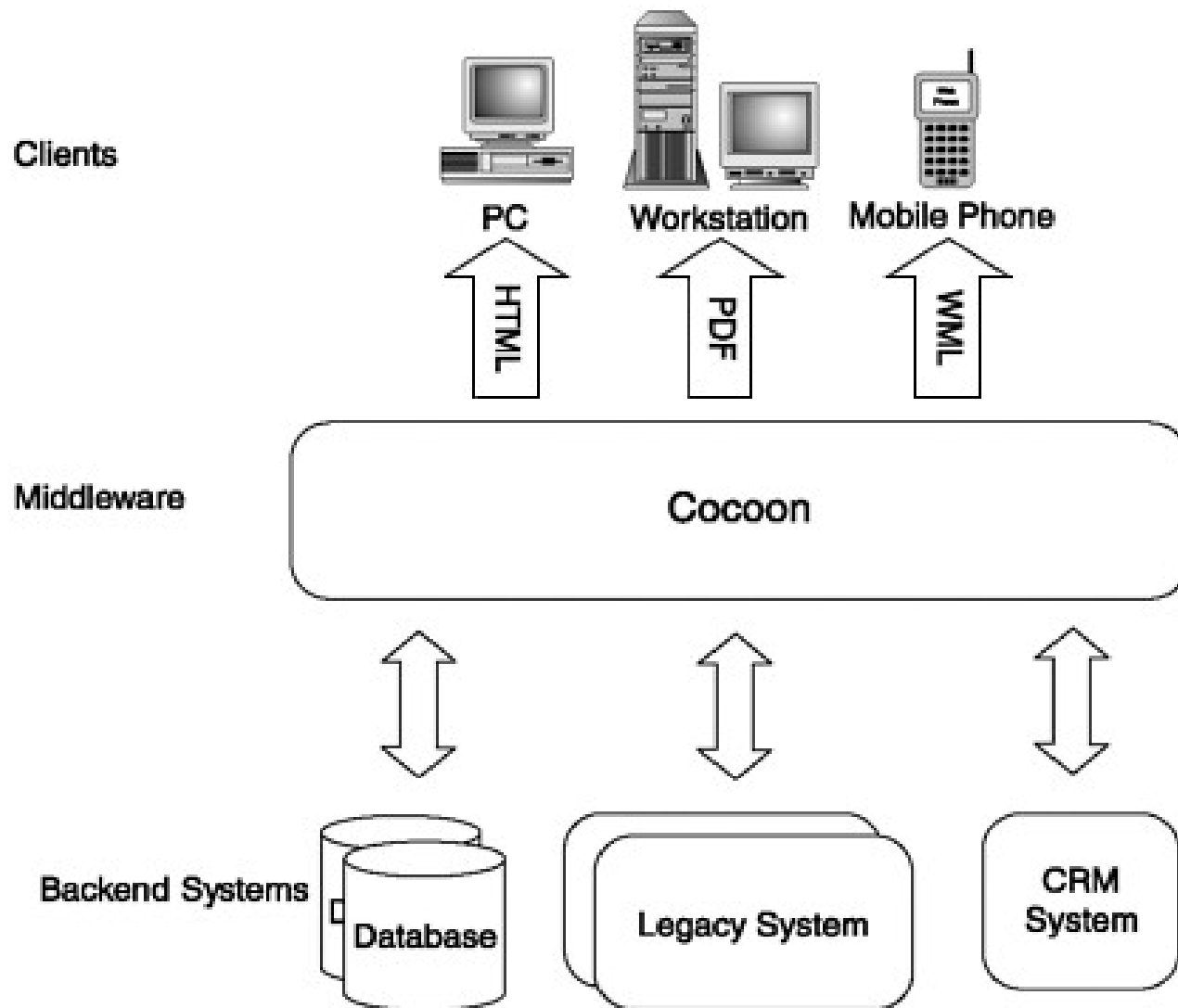
In addition, Forrest was to be a Sourceforge-like project management tool, complementing Apache Gump.

The Forrest vision was articulated by Stefano Mazzocchi and Sam Ruby and was created after a headstart by Nicola Ken Barozzi.



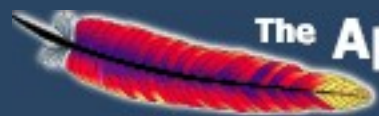
- Started by the Italian student Stefano Mazzocchi in 1998/99 under the java.apache.org umbrella.
- Problems of HTML to mix style and content, he decided to use the new standards XSL and XML.
- Apache Cocoon 1.0 was a servlet that used XML4J (Apache Xerces) and LotusXSL (Apache Xalan) to transform a XML file with a XSL stylesheet.
- Underlying base product for many other projects such as Apache Forrest and Apache Lenya.

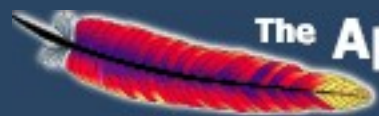
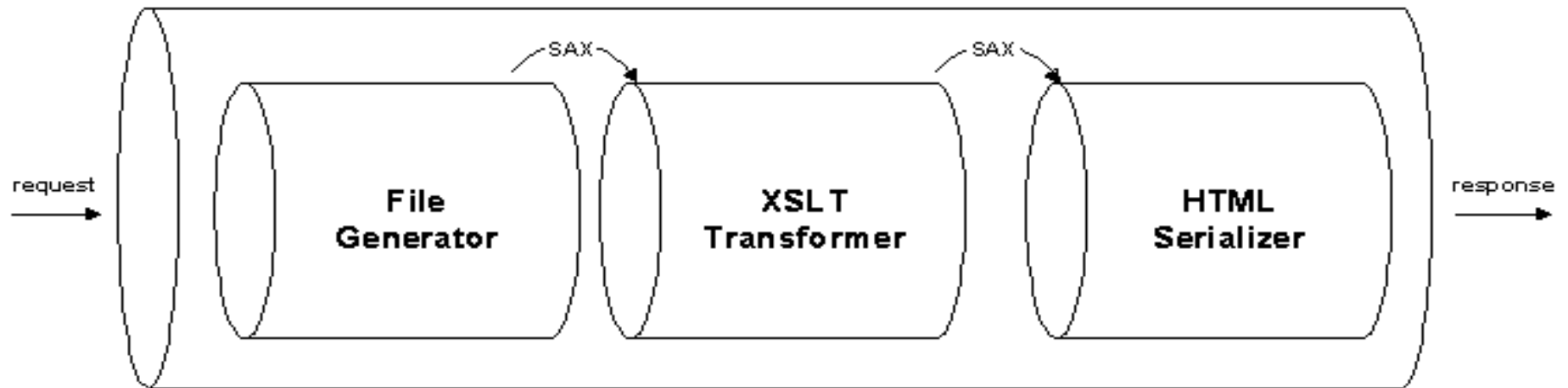




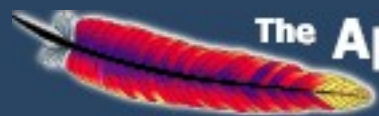
- the pipeline concept of cocoon is coming from the unix world (pipe '|').
- it is being used to pass the output of one program as input of another one.
- the idea is to re-use small programs instead of writing one big application.

```
svn st|grep ?|awk '{print "svn add "$2}'|sh
```





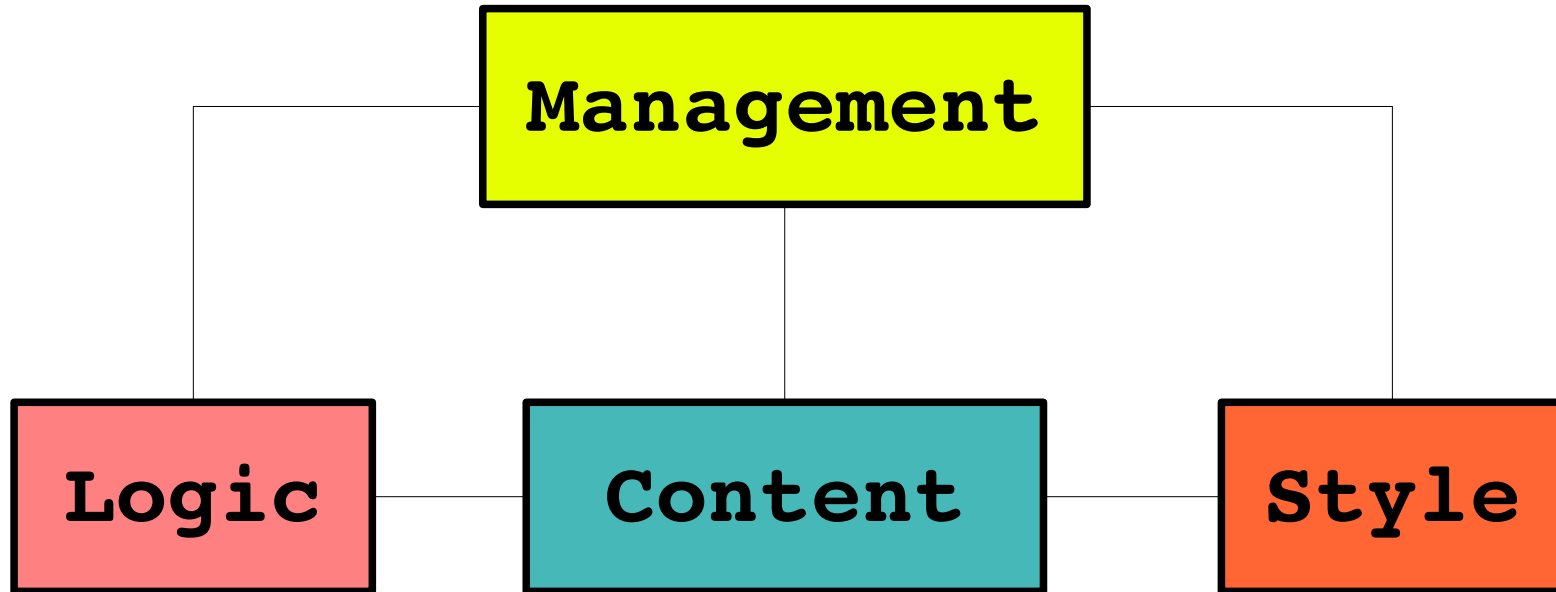
- Dispatching based on Matchers.
- Generation of XML documents (from content, logic, Relation DB, objects or any combination) through Generators
- Transformation (to another XML, objects or any combination) of XML documents through Transformers
- Aggregation of XML documents through Aggregators
- Rendering XML through Serializers



It can be observed that separating people with common skills in different working groups increases productivity and reduces management costs, but only if the groups do not overlap and have clear "contracts" that define their operability and their concerns.

For a web publishing system, cocoon uses what we call the pyramid of contracts which outlines four major concern areas and five contracts between them.



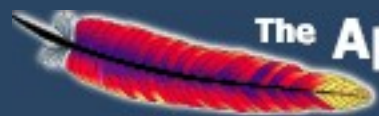


1. Install a subversion client and configure it properly.
2. At a command prompt, enter

```
svn co http://svn.apache.org/repos/asf/forrest/trunk forrest
```
3. Set the environment variable `FORREST_HOME` point to above dir and add `$FORREST_HOME/bin` to your `$PATH`
4.

```
cd $FORREST_HOME/main; ./build.sh
```
5.

```
cd $FORREST_HOME/site-author; forrest run
```
6. Browse to <http://localhost:8888/>

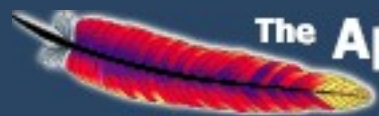


Forrest supplies a collection of default skins which are configurable and so should meet the needs of most projects. The aim is to provide many capabilities so that extra skins are not needed.

Most Forrest skins can be customized through a single XML file (`src/documentation/skinconf.xml`). This file is used to customize the location and appearance of the skin features.

Let us see it action:

```
mkdir skins; cd skins; forrest seed; forrest run
```



The problem with the forrest skins so far has been that even if "only" the design changed (html-skeleton), we still had to write a completely new skin and implement all functionality. Another problem was that the functionality was not easily extensible by a user or they "just" needed extra content in some pages.

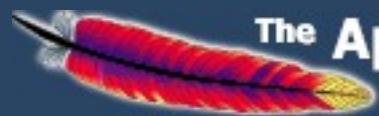
The aim of the "dispatcher" concept is to provide a flexible framework for creating site and page specific layout for different formats using different data sources through an **advanced** separation of concerns.

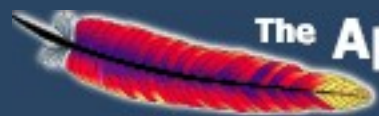
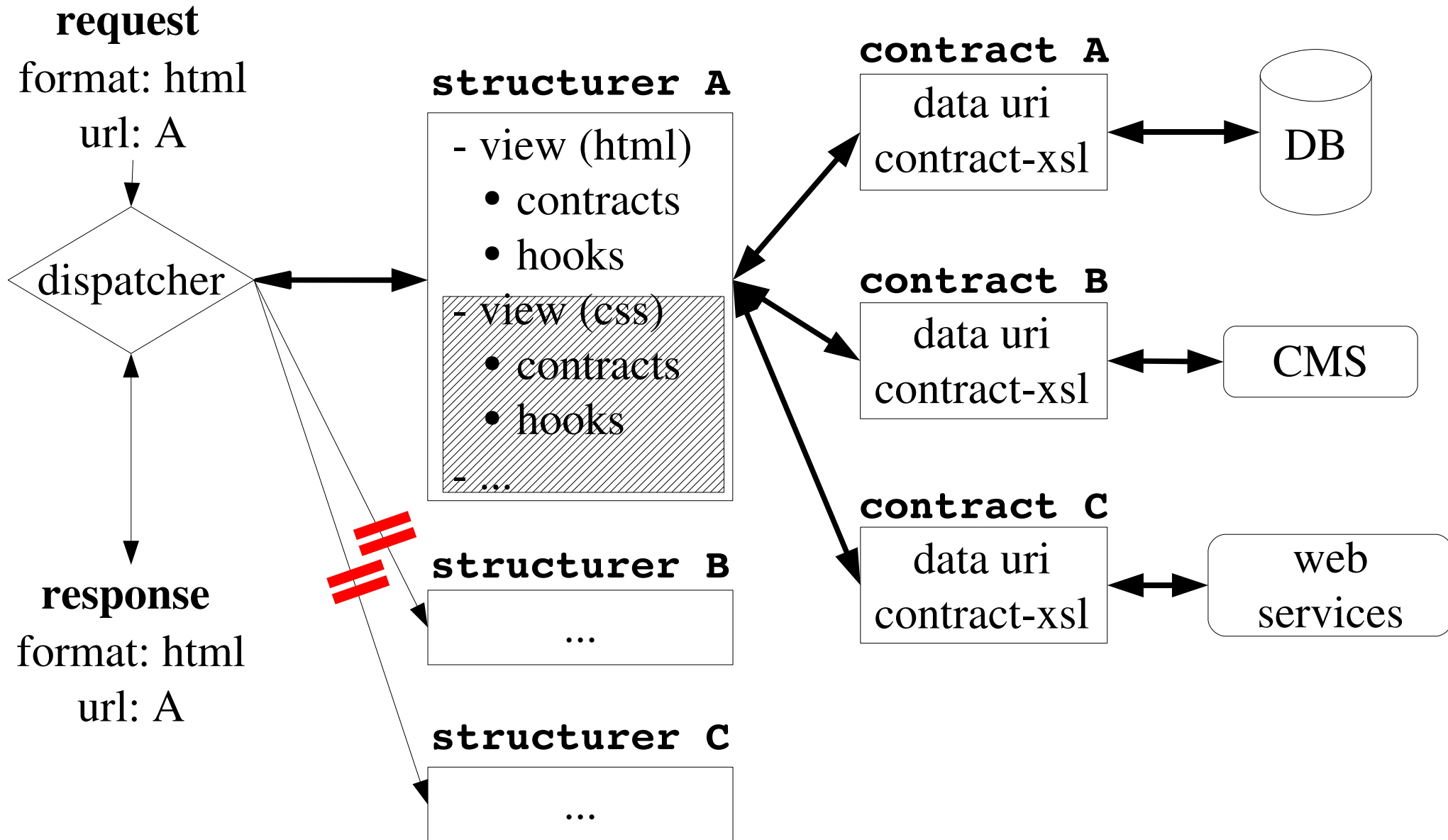


The dispatcher is a filter that limits the data model to a minimum by only requesting what the structurer (the layout definition) needs.

This leads to a different URL handling focus - away from document centric. A document can (but do not have to) be behind a certain URL. Like said a structurer can request any given data as input not only a document and the forrest core contracts (like navigation).

It lets you easily extend the default data models provided by forrest.

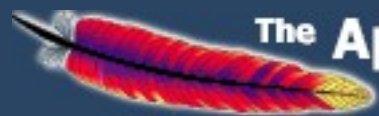




We developed the structurer to let the user decide where to place elements and to give the designer the full control over the output (dedicated to web designer).

The work started with grouping elements (the ones from skinconf) and used css-contracts that we added as @attributes (`<div id="content-main"/>`).

Around this contracts we developed a configuration concept - called the structurer. The structurer allows us to define the order in which forrest:contracts appear, and also to group them using forrest:hooks.



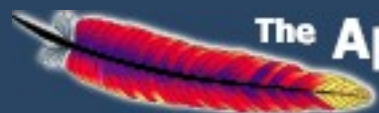
forrest:hooks are the styling side of the structurer. We can imitate arbitrary html skeletons with their help.

forrest:hooks are containers that are used for layout and structuring reasons. They do not add any content nor functionality to the output. They add **only** layout and structure information to the output.

```
<forrest:hook name="layoutId"/>
```

will be transformed for html into

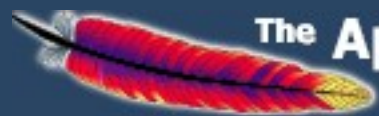
```
<div id="layoutId"/>
```



forrest:contracts are functionality or extra content that themes can use to display the request. Sometimes contracts delivers format-specific markup, other times it delivers a format-independent string.

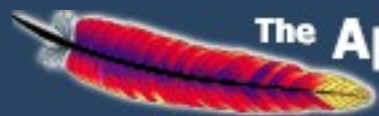
Different kind of contracts:

- static - will return always the same result
- semi static - offer configuration parameter used by the structurer
- dynamic contracts - offer semi-static configuration and/or contacting business services



Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can **help your neighbor** (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the **whole community benefits** (freedom 3). Access to the source code is a precondition for this.



- give user feedback to user@forrest.apache.org
- give enhancement and discuss code on dev@forrest.apache.org
- report issues <http://forrest.apache.org/issues.html>

Please go beyond **using**

→ give something back;)

