

## What's new in Cocoon?

**Carsten Ziegeler**

[chiegeler@apache.org](mailto:chiegeler@apache.org)

Competence Center Open Source  
S&N AG, Germany



## About

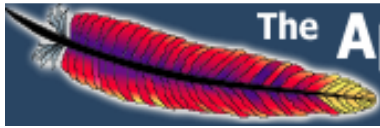
- Member of the Apache Software Foundation
- Committer in some Apache Projects
  - Cocoon, Excalibur, Pluto, WSRP4J, Incubator
  - PMC: Cocoon, Incubator, Portals
- Chief Architect of the Competence Center Open Source, S&N AG, Germany
- Article/Book Author, Technical Reviewer
- Member of the JSR 286 spec group (Portlet API 2.0)

## Agenda

- What is Cocoon?
- Why 2.2?
- The Vision
- The Current State
- Outlook
- Discussion



ApacheCon



The Apache Cocoon Project

<http://cocoon.apache.org/>

COCOOON

Present



US 2006

# What is Apache Cocoon? ☺

*Apache Cocoon is a web development framework built around the concepts of separation of concerns and component-based web development.*

*Cocoon implements these concepts around the notion of 'component pipelines', each component on the pipeline specializing on a particular operation. This makes it possible to use a Lego(tm)-like approach in building web solutions, hooking together components into pipelines without any required programming.*

*Cocoon is "web glue for your web application development needs". It is a glue that keeps concerns separate and allows parallel evolution of all aspects of a web application, improving development pace and reducing the chance of conflicts.*

## What is Cocoon?

- Cocoon is a powerful web application framework
  - It is **not** just an XML web publishing platform
- Serious web applications can still be fun
- Write code only when you need to
- The magical trio: pipelines, flow and forms
- Cocoon is aimed for larger projects/teams!



## Apache Cocoon

- Top-Level Apache Open Source project
  - <http://cocoon.apache.org>
- Started in 1999 (XML Publishing Framework)
- Today
  - One of the most important Apache projects
  - Incorporates technologies from various project
  - Used/Supported by several minor and major companies



# Extensible Architecture

- Component Based Architecture
- Focus on *Composing* rather than *Programming*
  - “*We figure out the hard parts, you get to do the fun stuff.*”
- Core + Blocks
  - Core = pipelines, sitemap, flow, forms, template
  - Blocks: Portal, Cron, FOP, Axis, POI, Batik, Databases, Authentication, OJB, SAP, WebDAV ...
  - Built-time application assembly configuration
  - (Hot-deployment and -reconfiguration is coming)

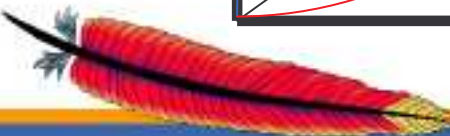
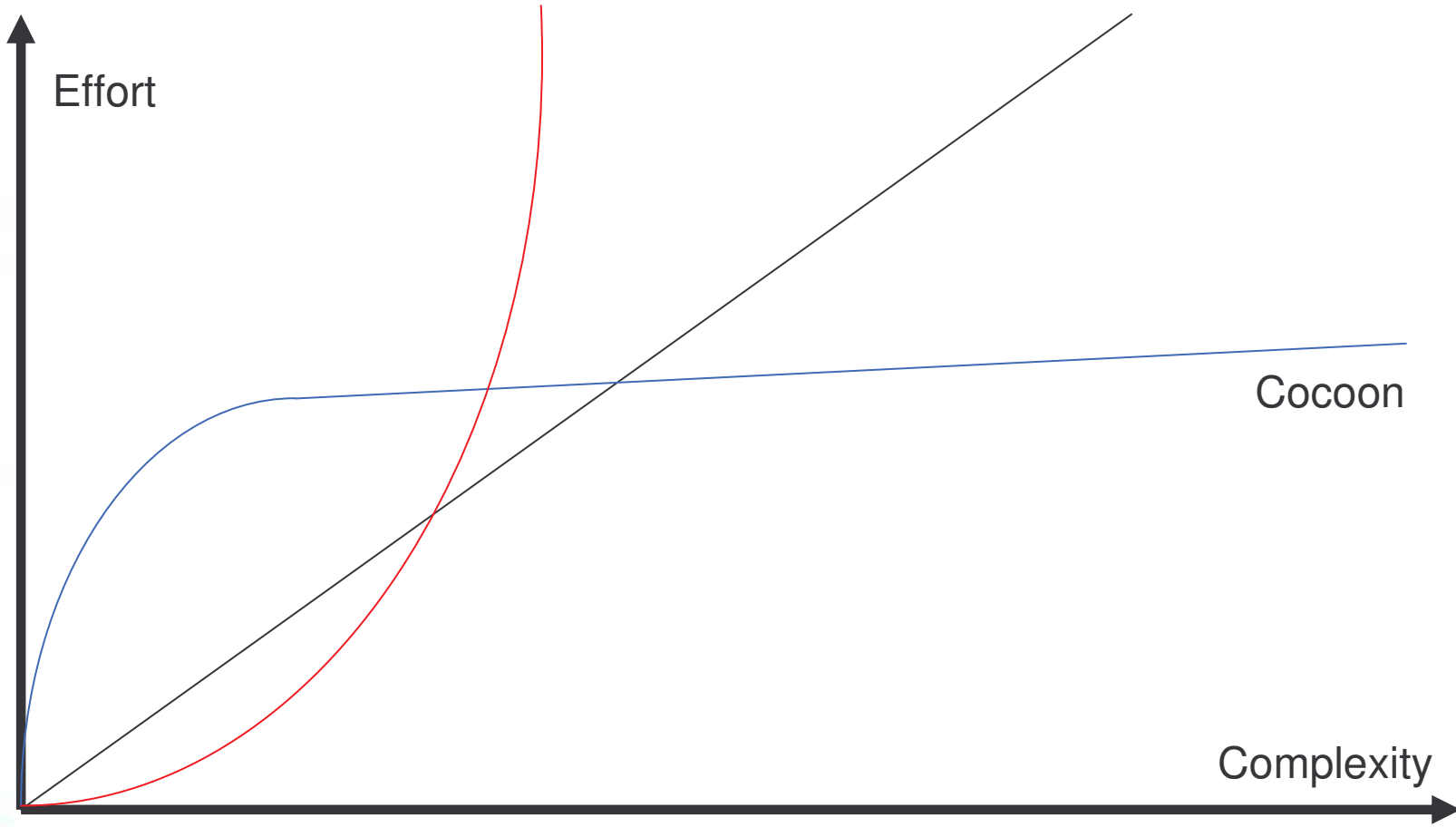


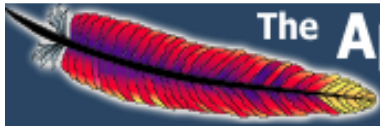


## Motivation for 2.2

- Learning curve can be steep at the beginning
  - New technologies: XML, XSL, SAX
  - New architecture: Sitemap, Pipelines, Flow
  - Lots of "features"
    - What do I really need?
  - "Could be better" documentation
    - Books are available / Wiki
- Build Time Configuration
- Getting at the top again 😊

# Learning Curves





## Cocoon in a Nutshell



## Scenarios

- Dynamic Document Generation
  - Based on XML/XSLT (but not limited to)
- Used for various application scenarios
  - Web Sites, Content Publishing
  - (XML) Portals/Workplaces
  - Processing Systems, Business Integration
  - CMS, Reporting, Administration Tools



## Some References

- VNU
- Vodafone
- BASF
- ÖAMTC
- Major (German) banks
- Insurance Companies

And more (we are not allowed to mention) ☺



## Real World Applications

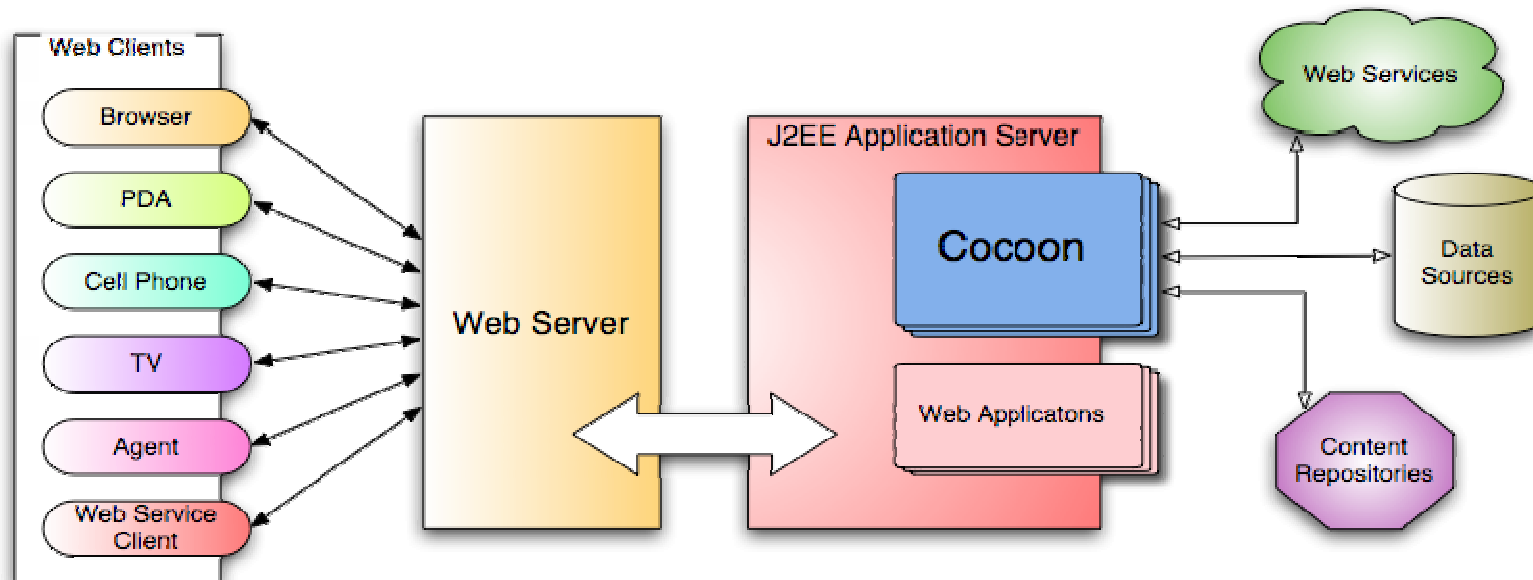
The screenshot shows a web browser displaying the Rhion Versicherungen website. The browser's address bar contains the URL: `https://www.systest.rhion.de/cocoon/epos/portal?cocoon-portal-action=0&cocoon-portal-event=27&cocoon-portal-event=28`. The website layout includes a top navigation bar with links for 'Home', 'Kontakt', 'Sitemap', 'Impressum', and 'Datenschutz'. A main banner features a cheetah in motion with the text 'Volle Leistung für deutlich mehr Profit.' Below this, there are several content sections: a 'Login' form, a search for agents ('Ihr Maklerberater vor Ort'), a sidebar menu with categories like 'Philosophie', 'volle Leistung', and '...optimale Prozesse', and a main content area with a heading 'Volle Leistung für deutlich mehr Profit' and sub-sections such as 'Rhion - der neue Maklerversicherer innerhalb der RheinLand Versicherungsgruppe' and 'Mehr Profit für Maklerprofis Klaus hat'. The browser's status bar at the bottom shows 'Fertig' and 'Internet'.

## Separation of Concerns

- UI: Templating and XSLT
  - Available components (Cocoon Forms)
  - XHTML, CSS, AJAX
- Controller/Application Flow
  - Cocoon Flow (JavaScript), Spring WebFlow
- Business Layer
  - Cocoon Forms
  - JavaScript/Java
  - 3rd party frameworks



## Apache Cocoon





## Web Publishing with Cocoon

- Flexible Data Integration / Aggregation
  - XML files, XML over HTTP
  - Databases, LDAP, SAP
  - ...
- Flexible Publishing (using XSLT)
  - HTML, WML, XML
  - PDF, SVG, PS, Office Documents
  - ...



## Web Publishing with Cocoon

- Dynamic Document Generation
  - Separation of Content and Layout
- Powerful Processing Description (Sitemap)
- Flexible Pipeline Concept
  - Many available components
- Conditional Processing (Matches/Selects)
- Caching

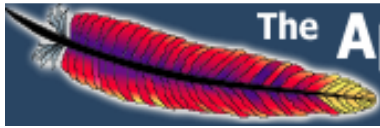


## Building Web Applications

- The Easy Way
  - Use Cocoon with Flow, Forms and Template ☺
  - Add your own business layer (Spring, Hibernate)
    - Don't use flow script for everything
  - Benefit from separation of concerns
    - Develop component oriented
- But
  - The first steps might be hard!
  - Don't give up – the effort pays back



ApacheCon



The Apache Cocoon Project

<http://cocoon.apache.org/>

COCOOON

# The Vision



US 2006

## The Vision: Real Blocks™

- **Blocks** are reusable functional modules at the cocoon web application level.
- **Blocks** were introduced in Cocoon to allow users to:
  - easily deploy their content on Cocoon without requiring operation downtime
  - package functionality /services in modules that can be reused as-is
  - easily extend existing modules
  - create complex web applications by high-level composing of these functional modules
  - depend on module behaviors, allowing for polymorphic module composition



## The Vision: Real Blocks™

- Hot-Deployment
- Resolving of dependencies
  - Maven-style repository downloads
  - Versioning
  - Classloading
- Reconfiguration
- Make blocks independent (own versioning)



## The Gordian Knot

- Plans for 2.2 are **very** old (nearly 4 years)
- Last year: „quiet period“ for development
  - Rest on our laurels
  - Hope that Avalon improves/helps
- Other component frameworks evolved
  - Especially Hivemind and Spring
- Rapid application frameworks
  - Ruby on Rails
- Noone dared to tackle the issues
  - Way too complex!



## The Gordian Knot has been cut!

- The Legend Of Avalon...has ended
  - Avalon is closed but maintained at Excalibur ☺
- Avalon was a core-dependency for Cocoon
  - Cocoon not runnable without it!
  - No evolution possible in Cocoon without Avalon
- Cocoon wanted to
  - have an independent core but
  - reuse efforts of other projects
  - without reinventing the wheel





## The New Core – 2005

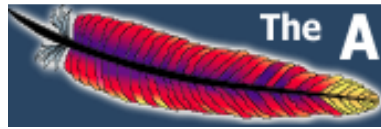
- Own Component Container
  - Based on Avalon Code
  - Provides Avalon Compatibility
  - Adds own functionality
- Separation between core and application!
  - Core uses own container
  - Applications are encouraged to use what they need! (Avalon, Spring etc.)



## The New Core – Final

- Spring Framework 2.0
  - Added Avalon Compatibility
  - Added own functionality
- **Slowly** turning away from Avalon Interfaces
  - Only for Cocoon itself!
- Separation between core and application!
  - Core uses Spring
  - Applications are encouraged to use what they need! (Spring, Hivemind etc.)





## The current State of Cocoon 2.2

*„Nothing is carved in stone yet!“*



## Goals for 2.2

- Flatten the learning curve
- Consolidation
- Support rapid prototyping/development
- Make building and configuring easier
- Better documentation
  - Make writing documentation easier
- Enabler for Real Blocks™
- BUT: Be as compatible as possible!



## Configuration

- Central Configuration (cocoona.xconf) is split
  - Includes per block configurations
  - Dependencies are handled by Spring
  - Removing/adding a block by removing/adding the appropriate configuration
  - Own configuration can be put into own files
- Avalon based configuration is merged with Spring application context configuration



# Configuration

- Using Cocoon is easy through Spring namespace authoring:

```
<beans>
  <!-- Load all the properties for Cocoon -->
  <cocoon:settings/>

  <!-- Load Avalon configurations -->
  <avalon:avalon location="/WEB-INF/cocoon/cocoon.xconf"
                loggingConfiguration="/WEB-INF/cocoon/log4j.xconf"/>

  <!-- Add your own beans here -->
  ...
</beans>
```



# Configuration - Properties

- Dynamic properties can be used
  - In all component configurations (cocoon.xconf etc)
  - In all log4j configurations
  - In sitemap
- Support of running mode:
  - dev,prod, test
- Ant-style usage:
  - `<dburl>${myapplication.dburl}</dburl>`
  - `<user>${myapplication.dbuser}</user>`
  - `<password>${myapplication.dbpasswd}</password>`



## Configuration - Properties

- Property resolving
  - WEB-INF/cocoon/properties/\*.properties
  - WEB-INF/cocoon/properties/{running.mode}/\*.properties
  - Properties specified on startup
  - Own property provider (e.g. from database)
  - System properties





## Per Sitemap Configuration

- Per Sitemap Configuration Possible
  - only available to sitemap and their sub sitemaps
  - Components (Avalon/Spring)
  - Properties
  - Especially interesting with classloading...



## Sitemap Classloading

- Per Sitemap classloading possible
  - Use different versions of libraries
  - Hot-Reloading when sitemap is reloaded
  - Faster development
  - Just drop your webapp including Java code as a sub sitemap into Cocoon



## Sitemap Listeners

- Configure per sitemap listeners
  - Invoked when a request enters a sitemap
  - Invoked when a request leaves the sitemap
- Events contain information about request
- Enabled initialization of request values
  - Can be done with actions or flow as well
- Enabled cleanup after request is processed!



## Consolidation in 2.2

- The Burden of Compatibility ☹
- Remove deprecated blocks
- Streamline parallel development
- Cocoon Forms
- Templates (JXTG)



## Benefits of Cocoon 2.2

- Rapid Development
- Spring 2.0
- Modular Configuration
  - Avalon (Compatibility)
  - Spring
  - Property Replacements
- Running Modes
- Hierarchical Configuration



## The Current State of 2.2

- Requires JDK 1.4 / Servlet 2.3
- New Build System
  - Maven 2
  - Archetypes
- First "Milestone Releases" out!
- Check it out and give feedback!

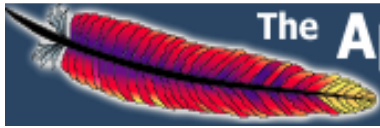


## Distribution/Release

- 2.2 will be released when it's finished 😊
  - Hopefully released in 4th quarter of 2006
  - Many features already implemented/prototyped
  - Nothing is carved in stone yet!
- Blocks will be made available separately
- Binary Distribution
- Different Flavours (Core, Demo, Full)?



ApacheCon



The Apache Cocoon Project

<http://cocoon.apache.org/>

COCOOON

# Summary



US 2006



## What You Should Do ☺

- Checkout/Download Cocoon 2.2
- Try out the Demos
  - Forms and Flow
  - AJAX
  - Portal



## Summary

- Stable platform (3 years+)
- Large community
  - Including large corporations
  - "Awareness" in the public is growing fast
- Separation of Concerns
  - Team Development
- **Driven by user/developer needs**
  - Not a Research Project!



## Benefits

- No real alternative
  - That offers everything available in Cocoon
- XML driven architecture
  - Extensible with own components
- Flexible data integration and publishing
  - Often: No programming needed
- Large code-base
  - Many components provided
  - Most of the hard work is done already

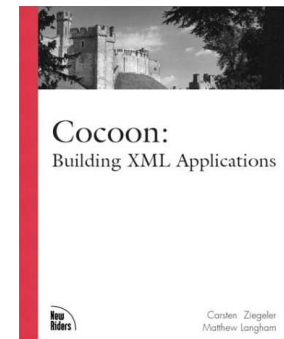
## Your Benefits

- Solid Framework (more than 5 years of experience)
- Separation of Concerns
- Real Team Development
- Rich Set of Functionality
- "Modern" techniques and frameworks



## Further Information

- Apache Cocoon Project
  - <http://cocoon.apache.org>
  - Downloads, Mailing-Lists, Links
- Cocoon Documentation Wiki
  - <http://wiki.apache.org/cocoon>
- Apache Forrest, Apache Lenya and others!
- Books available
- Competence Center Open Source ☺
  - <http://www.ophelos.de>



Thanks for your attention!

Questions?

