

# CocoonDocumentationSystemSUMMARY

!!! NOTE: This document is a summary of [CocoonDocumentationSystem](#) and maintained by [ReinhardPoetz](#). Please add your comments there !!!

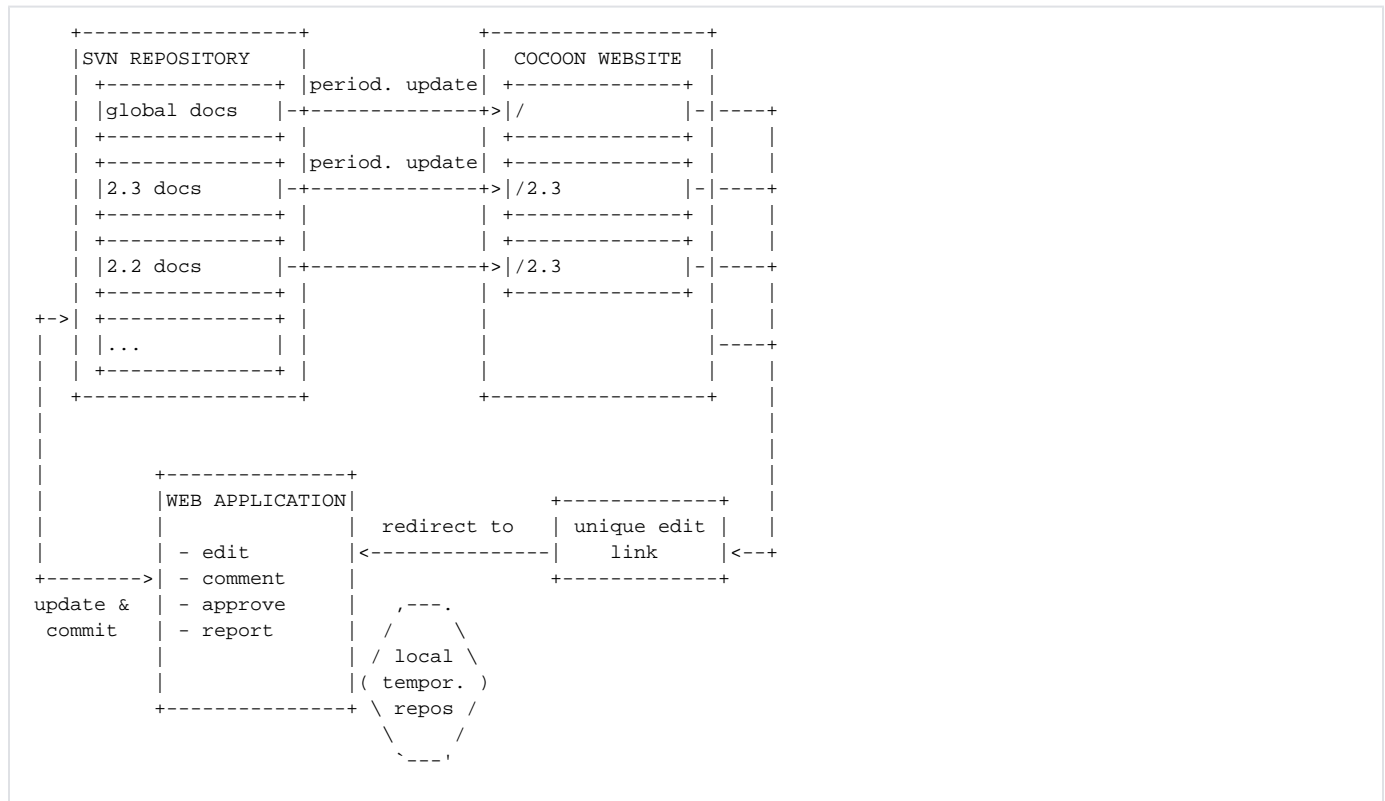
## Motivation (or problems I want to solve)

- make writing docs as simple as possible
- separation between global docs (docs about the project) and user docs
- evaluate the existing docs (no automatic doc import into new repositories)
- enable comments for doc
- have a modular architecture (having a repository and webspace should be enough to run, other features should be "pluggable")

I don't think that a technical solution will directly lead to better docs. I just think that making the doc authoring process easier will invite much more people to help out.

## Architecture

### Overview



The center of the new architecture is SVN. SVN contains all Forrest repositories. For the first, these are[1]:

- 2.2 docs
- global docs

These repositories are periodically published using some kind of automatism (cron, forrestbot)

Let's look at the structure of the website:

```
http://cocoon.apache.org ..... contains the global docs
http://cocoon.apache.org/2.2/..... contains the periodically (e.g. every 6 hours)
                                   published docs
```

The "living docs" docs that are periodically published out of the current repositories contain an "edit" and a "comment" link. This link is redirected to a web application where authors can write new docs or comments, e.g.

```
http://cocoon.apache.org/edit/2.2/23.html --> http://someapacheserver.apache.org/webedit/edit/cocoon/2.2/23.html

http://cocoon.apache.org/comment/2.2/23.html --> http://someapacheserver.apache.org/webedit/comment/cocoon/2.2/23.html
```

The web application at <http://someapacheserver.apache.org/webedit/> has following features:

- edit a document e.g. document 2.2/23.html (everybody)
- save the modified document in a local repository (everybody)
- commit the document to SVN (committers only)
- get a list of all locally available versions and the latest SVN version of a document so that the author can select from which version he wants to start (everybody)
- add a comment to a document (everybody)
- commit the comment to SVN (committers only)
- send mails to docs@cocoon.apache.org when a new doc or comment has been written
- make site.xml editable using a nested list of <ul> (committers only)
- user management (committers only)
- report: list of all modified/new docs and comments available

The implementation of this mini CMS could be done based on

- Daisy
- Hippo CMS (if it is OS licensed in the meantime)
- Apache Lenya
- Forrest
- Gianugo's little [CMS](#)
- write my own

## Document format

I propose the format that the default configuration of the HTMLCleaningConvertor generates. What is good enough for Daisy should work as well for us 😊 Alternatively we can use plain HTML4. This would have the advantage that the document can be edited using any HTML editor.

In the repository, each document consists of a directory, which can contain following files:

```
./content_en.xml ..... content as cleaned XML
./content_en.html ..... the content as HTML (only content_en.xml OR content_en.html are
allowed)
./content_de.xml ..... content in German
./meta.xml ..... contains meta information (date, authors, status, target audience,
keywords)
./comments_en.xml ..... contains user comments in English
./comments_de.xml ..... contains the user ccomments on the German version
./files/ ..... contains all images and attachements
```

Note: Having the language postfix gives the chance to provide translations. For the first we concentrate on English documents.

## Document identifiers

- use of Wiki-style names
- flat structure (the structure of the docs is not the concern of the repository)

## Forrest repositories

See <http://svn.apache.org/repos/asf/cocoon/whiteboard/doc-repos/> for two examples that work with the latest SVN version of Forrest 0.6.

## Published docs

[22NewDocumentsGeneration](#) shows where the current version of the docs is available.

## What do we have to do?

## Step by Step

The good news is that we don't need all the features at once. We can go the path to better docs and a better documentation system step by step:

- Step1: Setup Forrest Repositories (Reinhard: DONE)
- Step2: Initial content
  - find a new structure (Upayavira and Reinhard: DONE)
  - evaluate good docs and move them into the new repository !!! more volunteers needed !!!
  - integrate auto-generated docs (n.n.)
- Step2: enable the periodically running jobs that publish repositories
- Step3: work on the web application and have a demo running on brutus.apache.org
  - component that reads from and commits to SVN (infos provided by Jeremy)
  - do the SSL setup of Apache (n.n.)
- Step4: move the demo application over to Apache infrastructure