

## Apache Lenya: Architektur

In Lenya 2.0 wurde die starre monolithische Architektur durch eine modulare Architektur abgelöst. Die Vorteile einer solchen Konzeption liegen auf der Hand:

- bessere Wiederverwendbarkeit von Komponenten, die eine definierte Funktionalität über eine standardisierte Schnittstelle anderen Service-Komponenten zu Verfügung stellen
- strukturierte Gesamtsysteme, die in ihrer Komplexität verringert werden und somit eine schnellere und vor allem leichtere Problemanalyse bieten
- Schnelligkeit bei der Implementierung neuer Services ohne kostspielige Aufwände

Die Architektur einer mit Apache Lenya implementierten Applikation umfasst folgende Schichten:

- Client-Schicht (Front-Ends): Content Acquisition Devices, Content Presentation Devices
- Publikationsschicht: Publikationen (Websites)
- Applikationsschicht: Standard-Module, Module von Drittanbietern, eigene Module
- Content-Management-Schicht: Lenya-Core-Module
- Storage-Schicht: Content Repository, Suchindex, weitere Daten

### Client-Schicht

Zur Client-Schicht zählen Geräte und Applikationen, die zum Einpflegen von Inhalten in das CMS (Content Acquisition) und zum Konsumieren von Inhalten (Content Presentation) dienen.

Für Web-Publishing-Projekte bietet sich die Bearbeitung der Website im Browser an. Standardmässig unterstützt werden die WYSIWYG-Editoren Bitflux, Kupu, FCKEditor und TinyMCE sowie formularbasiertes Editieren der XML-Dokumente. Individuelle Rich-Clients können implementiert werden.

Über die Java-API und die WebDAV-Schnittstelle können projektspezifische Content-Acquisition-Mechanismen angebunden werden.

Beispiele für Content-Acquisition-Front-Ends:

- Browser-basierte WYSIWYG- und XML-Editoren (Out of the Box)
- Browser-basierte Applikationen (individuelle Implementation für spezifische Dokumenttypen)
- Technologien: JavaScript, AJAX, XUL, XAML, Adobe Flex, OpenLaszlo
- WebDAV + externer XML-Editor
- alternative Schnittstellen (Shell etc.)
- Content Syndication: Feed-Integration, Page Scraping für Legacy-Content
- automatisierter Content-Import (z.B. über Cron-Jobs, Java-API)

Die Content-Presentation-Front-Ends ermöglichen es dem Nutzer, die publizierten Inhalte zu konsumieren.

Beispiele für Content-Presentation-Front-Ends:

- Web-Browser, Flash-PlugIn
- PDF-Viewer
- Print-Produktions-Systeme
- mobile Endgeräte (Browser, SMS-Empfang)
- Informationssysteme in Firmen und öffentlichen Gebäuden (Anzeigetafeln etc.)

## Publikationsschicht

Die vielfältigen Verarbeitungs- und Ausgabe-Komponenten von Apache Cocoon erlauben es, Inhalte automatisiert in verschiedenen Formaten zu publizieren (Cross-Media-Publishing). Zur internen Verarbeitung werden auf offenen Standards basierende XML-Formate wie SVG und XSL-FO verwendet. Vor der Auslieferung werden diese Formate mittels integrierter Open-Source-Komponenten in Web-Standardformate wie PNG, JPG und PDF transformiert. Die flexiblen Rendering-Mechanismen erlauben es, Inhalte automatisiert mediengerecht aufzubereiten. So können z.B. Bilder, die als Vektorgrafiken (SVG) ins CMS eingepflegt werden, in PDF-Dokumenten als Vektorgrafiken und im Web als Bitmaps dargestellt werden.

Beispiele für Ausgabeformate:

- HTML, XHTML
- Bilder, Videos
- generierte Bilder über SVG
- PDF über XSL-FO

## Applikationsschicht

Die Applikationsschicht wird in Form von Modulen implementiert. Es kann auf diverse Standard-Module von Lenya zurückgegriffen werden. Eigene Module und Module von Drittanbietern können beliebig hinzugefügt werden. Dabei steht dem Entwickler die volle Funktionalität von Cocoon zur Verfügung.

Beispiele:

- domänenspezifische Erweiterung der CMS-Funktionalität
- individuelle Funktionalität
- Zugriff auf domänenspezifische externe Dienste
- individuelle Import/Export-Schnittstellen
- Controller für formularbasierte Front-Ends (leicht zu implementieren mit dem Usecase-Framework von Lenya Content-Management-Schicht)

Die Content-Management-Schicht bietet Standard-CMS-Funktionalität, auf die über eine Java-API zugegriffen werden kann. Alle Komponenten können konfiguriert und durch individuelle Implementationen ersetzt werden. Die Content-Management-Schicht bietet u.a. folgende Funktionalitäten:

- Dokumentenverwaltung
- Workflow
- Zugriffsschutz
- Benachrichtigungen
- Versionierung
- Transaktionen
- Link-Management

## Repository-Schicht

Die Repository-Schicht unterstützt die performante und sichere Verwaltung von Dokumenten. Die Dokumente werden im Dateisystem abgelegt, um eine flexible Konfiguration der Speicherung zu ermöglichen. Ein Suchindex gewährleistet den schnellen Zugriff auf Metadaten und Dokumentinhalte.

In 2.0 wurde der Repository-Zugriff entscheidend verbessert. Dokumente werden über UUID (Universally Unique Identifiers) referenziert. Bilder und andere Binärdaten werden als Dokumente abgelegt und können daher über Workflows und Metadaten verfügen. Die erweiterte Repository-API und neue Protokolle machen es wesentlich einfacher, Dokumente zu adressieren und zu verarbeiten.

