

# Der Wiki-Dienst des KIM

Sorin Gheorghiu

## Wiki-Hosting

Das KIM<sup>1</sup> der zentrale Dienstleister der Universität für IT- und Bibliotheksdienste bietet an, Wiki-Instanzen basierend auf MediaWiki<sup>2</sup> zu hosten. Jedes Wiki hat seine eigene URL und die URL kann einer anderen Domain zugeordnet sein. Jedes Wiki ist unabhängig von den anderen und erscheint als eine separate Webseite mit seiner eigenen Konfiguration, Erweiterungen und Designs (Skins)<sup>3</sup>. MediaWiki ist eine bestimmte Wiki-Engine, die für Wikipedia<sup>4</sup> und andere Wikimedia-Projekte entwickelt und von diesen benutzt wird. Es ist in der Skriptsprache PHP<sup>5</sup> geschrieben und nutzt eine relationale Datenbank (wie MySQL<sup>6</sup> oder MariaDB<sup>7</sup>), um die Inhalte zu speichern. Webseiten auf Wiki-Basis sind besonders dann hilfreich, wenn Erfahrung und Wissen gemeinschaftlich gesammelt und dokumentiert werden soll.

## Dockerisierung

Die technische Implementierung basiert auf Docker<sup>8</sup>. Das Ziel von Docker ist, die Vorteile virtueller Maschinen unter Linux ohne den sonst üblichen hohen Ressourcenverbrauch zu nutzen. Ein Container wird in einer eigenen isolierten Laufzeitumgebung mit Prozessbereich und einem Netzwerkstapel ausgeführt. Dies erleichtert die Behandlung von Sicherheitsproblemen. Ein Vorteil der Isolation ist die Fähigkeit, ohne Abhängigkeiten und großen Aufwand Tests durchzuführen. Durch das Containerisieren wird die Bereitstellung nicht nur schneller, sondern auch viel performanter. Docker ermöglicht eine effiziente Nutzung der Systemressourcen. Die Instanzen von containerisierten Anwendungen verbrauchen viel weniger Hauptspeicher als virtuelle Maschinen. Sie lassen sich schneller starten und stoppen und sie können viel dichter auf eine Host-Hardware gepackt werden. Das bedeutet eine hohe und effiziente Auslastung der Ressourcen und damit weniger Kosten auf der IT-Seite.

Ein anderer Vorteil der Containerisierung ist, modulare Systeme zu erlauben. Statt eine komplette, komplexe Applikation in einem Container zu betreiben, lässt sich diese in Module aufteilen (zum Beispiel Database, Front End, etc.). Das ist der sogenannte Microservices-Ansatz. Applikationen, die nach diesem Schema aufgebaut sind, sind leichter zu managen, weil die einzelnen Module überschaubar bleiben und Änderungen auf Modulebene jederzeit möglich sind - ohne dazu die ganze Applikation neu schreiben zu müssen.

Die Basis für Container bilden sogenannte Images, eine einfache Datei, durch die die Installation und das Updaten einer Software wegfällt. Images beinhalten alle Komponenten, um eine Anwendung plattformunabhängig auszuführen. Somit kann ein Image lediglich durch einen einfachen Kopiervorgang auf ein anderes System übertragen werden. Durch Verwendung von offiziellen Images von dem bekanntesten öffentlichen Registry, Docker Hub<sup>9</sup>, wird das Sicherheitsrisiko minimiert.

Docker Hub ist ein Online-Dienst, der eine Registry für Docker-Images und Repositorien beinhaltet und teilt sich in einen öffentlichen und einen privaten Teil auf. In dem öffentlichen Bereich gesellen sich immer mehr offizielle Images von Linux-Distributoren.

Docker bietet eine eingebaute Versionsverwaltung. Diese erlaubt es, den aktuellen Stand des Containers in einem Image zu sichern, dieses auf das Docker Hub zu laden und die Unterschiede zwischen dem aktuellen Zustand des Containers und dem ursprünglichen Image darzustellen. Das offizielle MediaWiki Image<sup>10</sup> läuft in der Regel unter der LTS (Langzeitsupport) Version. Eine LTS Version wird alle zwei Jahre veröffentlicht<sup>11</sup>.

Aus Sicht eines Admins spricht einiges für Docker: Docker-Container lassen sich mit reduzierten Berechtigungen betreiben und sorgen so für eine Barriere zwischen einem Dienst und dem darunterliegenden Betriebssystem. Sie sind einfach zu handhaben und lassen sich mühelos ersetzen. Möchte der Admin ein neu-

es Mediawiki Image ausrollen, lädt er es sich in den entsprechenden Container herunter und startet ihn, dann werden alle Mediawiki Container einmal aktualisiert.

Die MediaWiki Erweiterungsdateien und der spezifische Dienstserver (wie der Parsoid-Server für den Visual Editor<sup>12</sup>) werden als eigene Docker-Container gestartet und gemeinsam mit den Wiki-Instanzen geteilt.

## Anmeldungsmethoden

Mit einer Anmeldung legt man sich einen festen Benutzerzugang an. In Wikipedia muss man sich nicht anmelden, um die Inhalte in einem öffentlich zugänglichen Wiki lesen zu können. Auf den privaten Wikis ist dies konfigurierbar, je nach Bedarf.

Der Wiki-Server erkennt den Anwender anhand eines Cookies, das mit Hilfe des Browsers auf dem Gerät des Benutzers verwaltet wird. Das Cookie enthält eine Benutzernummer, mit welcher die Wiki Software am Server die Einstellungen der Benutzer auseinanderhält. Cookies müssen darum aktiviert sein, sonst kann das Wiki den Benutzer nicht erkennen.

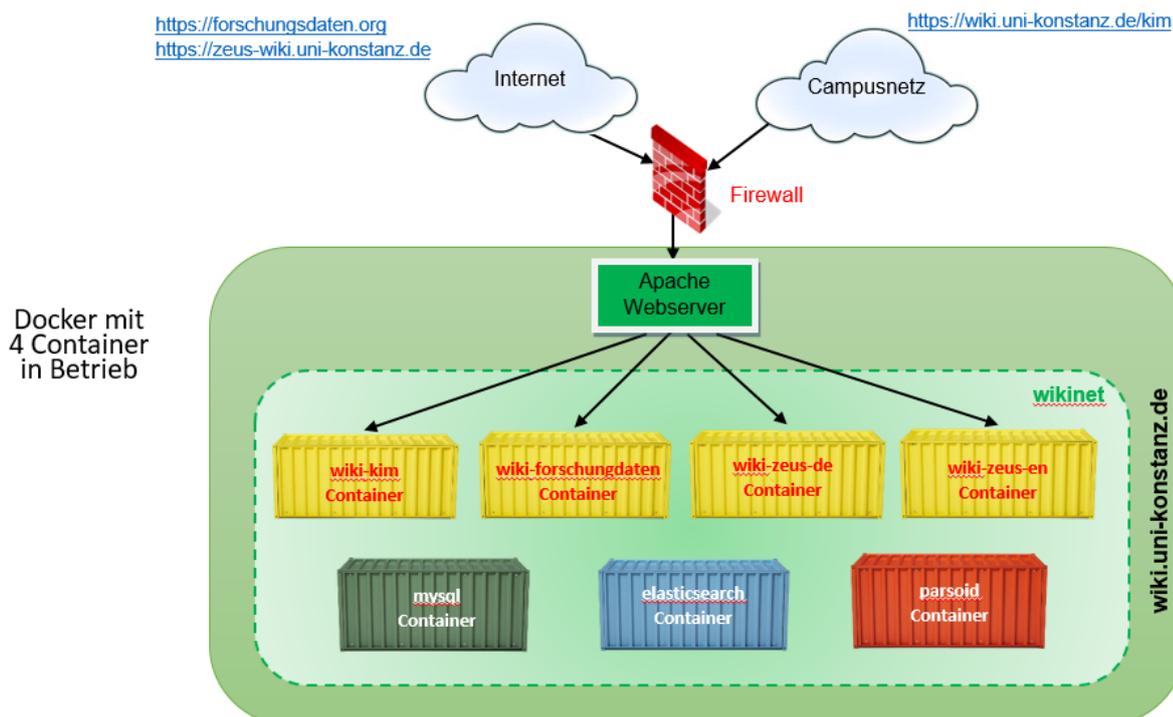
Die Anmeldung kann man über zwei Methoden einrichten: Basisauthentifizierung (Standardmethode über einzelnen Benutzernamen und Passwort) oder einen zentralen Authentifizierungsserver (über LDAP<sup>13</sup> oder Shibboleth<sup>14</sup>).

## Zentralisierte Authentifizierung und Autorisierung mit SAML

Das KIM der Universität Konstanz ist seit einiger Zeit in der Lage, SAML-basierte Service Provider (SP) aus dem Campusnetz der Uni in eine lokale Föderation, welche in Zusammenarbeit von KIM mit der DFN-AAI verwaltet wird, zu integrieren. Somit ist es möglich, einzelne Mediawiki-Instanzen als Service Provider an den zentralen Shibboleth Identity Provider (IDP) der Uni anzuschließen.

Mit dem SAML-Protokoll (Security Markup Language)<sup>15</sup> steht eine verteilte Authentifizierungs- und Autorisierungsinfrastruktur (AAI) zur Verfügung, mit deren Hilfe der Identity-Provider im ersten Schritt die Authentifizierung eines Benutzers (mit Namen und Passwort) für einen Service Provider übernimmt und im zweiten Schritt die erforderlichen Autorisierungs-Attribute an den Service-Provider zurückschickt. Für jede Wiki-Instanz muss ein SP (Service Provider) konfiguriert und registriert sein. Auf dem Mediawiki Container wird SimpleSAMLphp<sup>16</sup> für die Authentifizierung verwendet. SimpleSAMLphp ist eine Anwendung, geschrieben in nativem PHP und eine grundlegende Implementierung des SAML-Authentifizierungsprotokolls.

Eine XML-Datei enthält die Metadaten der lokalen Föderation und enthält die Namen, SAML-Endpoints, Zertifikate aller IDP und SP



einer Föderation. Zudem kann man in den Metadaten eines SP die Autorisierungsattribute angeben, welche vom SP zur Autorisierung von Benutzern benötigt werden. Die Datei von IDP und SP wird zur Laufzeit eingebunden und regelmäßig gelesen und aktualisiert. Die Datei muss bei allen SP und IDP stets aktuell und synchron sein, um die Sicherheit zu gewährleisten und Interoperabilitätsprobleme zu vermeiden.

Das Modul Autorisieren des SimpleSAMLphp<sup>17</sup> bietet einen Benutzerautorisierungsfiler, basierend auf der Attributübereinstimmung für diejenigen Anwendungen, die die Authentifizierung nicht sauber von der Autorisierung trennen und einige Standardberechtigungen für authentifizierte Benutzer festlegen. Dieses Modul kann so konfiguriert werden, dass Benutzer basierend auf dem Autorisierungsattribut „business-Category“ (differenziert zwischen „Studierende“ oder „Bedienstete“) autorisiert werden, auf bestimmte Dienste zuzugreifen – so können z.B. alle Studierende und Mitarbeiter der Uni Konstanz auf das ZEuS-Wiki zugreifen. Es kann jedoch Wikis geben, die so „autorisiert“ werden, dass ein eingeschränkterer Benutzerkreis, wie eine Abteilung (durch die Organisations-ID), oder auch eine einzelne Person Zugriff hat.

Wenn ein Benutzer erstmals den Dienst eines Anbieters nutzt, werden ihm die zu übertragenden Attribute angezeigt und er könnte der Übertragung für diesen Vorgang oder für alle nachfolgenden Anmeldungen zustimmen. Wenn er der Übertragung der Attribute nicht zustimmt, wird er den Dienst in der Regel nicht nutzen können.

Die PluggableAuth<sup>18</sup> und SimpleSAMLphp<sup>19</sup> Erweiterungen sind nötig, um die Mediawiki Konfiguration abzuschließen. PluggableAuth stellt den gemeinsam genutzten Code bereit, der zum Implementieren dieser Erweiterungen erforderlich ist. PluggableAuth wird im Folgenden als Authentifizierungs- und Autorisierungs-Plugins bezeichnet. Durch die PluggableAuth lassen sich notwendige SimpleSAMLphp-Erweiterungen zur besseren Authentifizierung einrichten.

## Wiki Berechtigungen

MediaWiki ist ein auf Berechtigungen basierendes Wiki-System. Das bedeutet, dass Benutzer nur das machen bzw. verändern können, was sie dürfen.

Im ersten Schritt wird der Benutzer authentifiziert, das heißt, Benutzername und Passwort werden abgefragt und es wird überprüft, ob die Eingaben stimmen. Wenn ja, erhält der Benutzer Zugang zum Wiki. Danach erfolgt der zweite Schritt in Form der Autorisierung: Wiki schaut, welche Attribute der Nutzer hat, und daraus werden zum Beispiel die Gruppen und zusätzliche Attribute einbezogen mit denen Benutzerrechte verwaltet werden.

Benutzerrechte erteilen dem Benutzer die Berechtigung, bestimmte Aufgaben auszuführen. Die Benutzerrechte sind in Benutzergruppen zusammengefasst, die den verschiedenen Aufgaben im Wiki entsprechen (z.B. Benutzer, Administratoren und Bots).

## Zentraler Webserver

Apache HTTP-Server wird als Webserver verwendet, weil er durch den freien Quellcode an den individuellen Bedarf angepasst werden kann. Damit steuert man die Zugriffe, so dass die Wiki-Webseiten innerhalb oder ausserhalb des Campus erreichbar werden.

Als zentraler Webserver wird zur verschlüsselten Kommunikation auch ein SSL-Zertifikat benötigt, weitere Domains werden über SAN (Subject Alternative Name) erweitert. Standardmäßig sind die Wiki-Instanzen über die URL `wiki.uni-konstanz.de/<wikiname>` erreichbar, aber es ist möglich auch interne (z.B. `zeus-wiki.uni-konstanz.de`) oder externe Domains (z.B. `forschungsdaten.org`) einzurichten.

## Die Suchfunktion

Der schnellste Weg, etwas in Wikimedia-Projekten zu finden, ist, gezielt danach zu suchen. Auf jeder Seite befindet sich dazu ein Suchfeld mit einem Lupensymbol. Der Artikelinhalt wird in allen Wiki Seiten im Rohtext (Wikitext) durchsucht und es werden nur Seiten gefunden, die alle eingegebenen Suchbegriffe enthalten.

Um verbesserte Suchfunktionalitäten (Suchvorschläge, Volltextsuche) bereitzustellen, kann man die CirrusSearch Erweiterung<sup>20</sup> verwenden. CirrusSearch implementiert die Suche in MediaWiki mit Elasticsearch. Eine Volltextsuche ist eine indexbasierte Suche. Alle Seiten werden in der Wiki-Datenbank (z.B. MySQL) gespeichert und alle darin enthaltenen Wörter werden in der Suche-Datenbank (z.B. Elasticsearch) gespeichert, die ein Index zum Volltext

des Wikis ist. Der Suchindex wird nahezu in Echtzeit aktualisiert.  
Derzeit ist CirrusSearch in der Lage, mit Hilfe der PdfHandler Erweiterung<sup>21</sup>, Text nur in DjVu und PDF-Dokumenten zu suchen.

## Wikifarm

Eine WikiFarm<sup>22</sup> ist eine Sammlung von zwei oder mehrere Wikis, die auf einem Webserver installiert sind und einige Komponenten gemeinsam nutzen, wie zentrale Konfiguration der einzelnen Wikis oder geteilten Datenbanktabellen für zentralisierte Autorisierung. Der Begriff ist von ‚Server Farm‘ abgeleitet. Für den Nutzer der verschiedenen Wikis ist jedes Wiki der Farm absolut unabhängig von

den anderen und erscheint als eine separate Webseite. Jedes Wiki in einer Farm hat seine eigene URL und die URL kann einer anderen Domain zugeordnet sein.  
Die Umsetzung von Mehrsprachigkeit mit einem einzelnen Wiki ist so nicht vorgesehen. Als Lösung gibt es zwar verschiedene Konzepte, der Einsatz separater Wikis pro Sprache ist jedoch die eleganteste und sauberste Variante davon (siehe Wikipedia). Für jede Sprache wird ein eigenes Wiki verwendet, dessen Artikel mit den korrespondierenden Artikeln durch Interwiki links<sup>23</sup> verknüpft sind. Mit dem Sprachmenü kann man direkt zur Übersetzung navigieren. Für das ZEuS-Wiki gibt es zwei Interwikis, für Deutsch<sup>24</sup> und Englisch<sup>25</sup>.

## Fußnoten:

- 1 <https://www.kim.uni-konstanz.de>
- 2 <https://www.mediawiki.org>
- 3 [https://de.wikipedia.org/wiki/Skin\\_\(Computer\)](https://de.wikipedia.org/wiki/Skin_(Computer))
- 4 <https://www.wikipedia.org>
- 5 <https://de.wikipedia.org/wiki/PHP>
- 6 <https://de.wikipedia.org/wiki/MySQL>
- 7 <https://de.wikipedia.org/wiki/MariaDB>
- 8 <https://www.docker.com>
- 9 <https://hub.docker.com>
- 10 [https://hub.docker.com/\\_/mediawiki](https://hub.docker.com/_/mediawiki)
- 11 [https://www.mediawiki.org/wiki/Version\\_lifecycle/de](https://www.mediawiki.org/wiki/Version_lifecycle/de)
- 12 <https://www.mediawiki.org/wiki/Extension:VisualEditor>
- 13 [https://de.wikipedia.org/wiki/Lightweight\\_Directory\\_Access\\_Protocol](https://de.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol)
- 14 <https://de.wikipedia.org/wiki/Shibboleth>
- 15 [https://de.wikipedia.org/wiki/Security\\_Assertion\\_Markup\\_Language](https://de.wikipedia.org/wiki/Security_Assertion_Markup_Language)
- 16 <https://simplesamlphp.org>
- 17 <https://simplesamlphp.org/docs/stable/authorize:authorize>
- 18 <https://www.mediawiki.org/wiki/Extension:PluggableAuth>
- 19 <https://www.mediawiki.org/wiki/Extension:SimpleSAMLphp>
- 20 <https://www.mediawiki.org/wiki/Extension:CirrusSearch>
- 21 <https://www.mediawiki.org/wiki/Extension:PdfHandler>
- 22 <https://de.wikipedia.org/wiki/Wikifarm>
- 23 [https://en.wikipedia.org/wiki/Interwiki\\_links](https://en.wikipedia.org/wiki/Interwiki_links)
- 24 <https://de.zeus-wiki.uni-konstanz.de>
- 25 <https://en.zeus-wiki.uni-konstanz.de>