

Ekkehard Gumbel

# TYPO3 Single Sign-On

## Integration von Web-Anwendungen in ein TYPO3-Portal

Wer heute eine Website aufbaut, findet sich immer wieder in der Situation, dass eine Funktionalität benötigt wird, die so für TYPO3 nicht verfügbar ist. In den allermeisten Fällen ist dann die Entwicklung oder Erweiterung einer Extension für diesen Zweck der richtige Weg.

Manchmal jedoch kann es sinnvoll sein, über die Integration einer Fremdanwendung nachzudenken, statt diese in TYPO3 „nachzubauen“. Etwa wenn es sich um eine sehr komplexe Lösung handelt, wenn der Aufwand gering gehalten werden muss, wenn es sich um eine Einzelfalllösung ohne Aussicht auf Wiederverwendung handelt oder wenn die Fremdanwendung bereits in Benutzung ist und die Umstellung vom Auftraggeber (aus welchen Gründen auch immer) nicht gewünscht wird.

In seltenen Fällen könnte man nun eine TYPO3-Extension entwickeln, die im Prinzip eine Oberfläche für die Fremdanwendung bereitstellt. Die einfachste, aber am wenigsten elegante Lösung wäre es ansonsten, aus der TYPO3-Navigation (oder aus dem Content) auf die Fremdanwendung zu verlinken. Nachteil: Der Benutzer muss sich an der Fremdanwendung erneut anmelden, Kennwörter an verschiedenen Stellen ändern etc.

Einen Lösungsansatz bringt „TYPO3 Single Sign-On“, genauer gesagt das unter GPL verfügbare „Single Sign-On Framework“ [1]: Ein am TYPO3 angemeldeter Benutzer kann durch diese Lösung in eine angebundene Web-Anwendung wechseln, ohne erneut seine Anmeldedaten eingeben zu müssen. Deutlich wird dies an folgendem Beispiel – hier der Extranet-Bereich eines Anbieters:

- Nach Frontend-Anmeldung (z. B. mit Benutzer und Kennwort) erscheint der geschützte Bereich „Extranet“.
- Hinter einigen Menüpunkten verbirgt sich nun – ohne dass der Benutzer dies erkennen kann – eine Fremdanwendung (möglicherweise auf einem anderen Server, so z. B. im Menüpunkt „Trouble Tickets“.
- Wird ein solcher Menüpunkt angewählt, öffnet sich die Fremdanwendung (z. B. im neuen Fenster, im bestehenden Fenster, im Frame oder iFrame) und der Benutzer ist dort bereits angemeldet.



Fremdanwendungen im TYPO3-Portal per Navigation integriert.

Um ein solches Single Sign-On (SSO) einzurichten, ist neben der TYPO3-Erweiterung auch ein Zusatz zur Fremdanwendung nötig. Beides zusammen bildet das so genannte „Single Sign-On Framework“, welches genauer betrachtet aus

- SSO-Server (hier: die TYPO3-Extension)
- SSO-Agent (generische Software auf Seiten der Fremdanwendung)
- SSO-Adapter (anwendungsspezifische Software auf Seiten der Fremdanwendung)

sowie einem Public/Private Schlüsselpaar besteht. Dieser Schlüssel ist der Kern des Sicherheitsmechanismus: Mit seiner digitalen

Signatur bestätigt der Server die Identität des Benutzers. Verwendet wird hierfür OpenSSL, was manchmal zu Verwirrung führt, da der Mechanismus nichts mit SSL zu tun hat. Es wird lediglich die „RSA“-Funktion benötigt!

### SSO einrichten

Die Installation sollte am besten mit der Erzeugung des Schlüsselpaars beginnen. Diese ist in der Dokumentation gut beschrieben und könnte zum Beispiel wie folgt auf dem TYPO3-System erfolgen:

```
openssl genrsa -out /usr/local/sigsso/etc/sigsso_private.key 2048
openssl rsa -pubout -in /usr/local/sigsso/etc/sigsso_private.key -out
/usr/local/sigsso/etc/sigsso_public.key
```

Listing 1

Die generierte Datei „sigsso\_public.key“ wird später auf Seiten der Fremdanwendung benötigt.

Nun kann das Zielsystem eingerichtet werden. Wie bereits festgestellt, sind dafür „SSO-Agent“ und „SSO-Adapter“ erforderlich. Der oft verwendete PHP-basierte SSO-Agent besteht aus einer einzigen Datei, die so abgelegt wird, dass sie per Browser erreichbar ist („SSO-URL“, im Beispiel <https://secure.naw.de/sso/sigsso.php>). Hinzu kommt die Konfigurationsdatei, die sich standardmäßig in „/usr/local/sigsso/etc/sigsso.conf“ befindet.

```
[global]
loglevel: 5
public_ssl_key:
  /usr/local/sigsso/sigsso_public.key
tokensfile:
  session.txt
logfile:
  sigsso.log
externalOpenssl: 1
tmp_signature_dir: /tmp
tmp_signature_prefix: sign_

[errorcodes]
# no special ones configured here...
# see docs for options

[main]
mantis: php:///www/secure.naw.de/mantis/index_sso.php

--url=https://secure.naw.de/mantis/main_page.php
openwebmail:
  cmd:///www/secure.naw.de/cgi-bin/openwebmail/openwebmail_sso.pl
--remote_addr=%remote%
--agent=%agent%
```

```
--url=http://secure.naw.de/cgi-bin/openwebmail/openwebmail-main.pl
--user=%user%
```

Listing 2

In der Konfiguration ist auch der Pfad zum SSO-Adapter sowie der Browseraufruf für die eigentliche Fremdanwendung erkennbar. Die Installation des SSO-Adapters beschränkt sich meist auf das Entpacken eines Archivs. Adapterspezifische Hinweise sowie Beispiele für die „sigssso.conf“ werden beim jeweiligen Adapter mitgeliefert. Abschließend muss noch der „public Key“ des zuvor erzeugten Schlüsselpaars (siehe oben) auf das Zielsystem (an die in sigssso.conf definierte Stelle) übertragen werden.

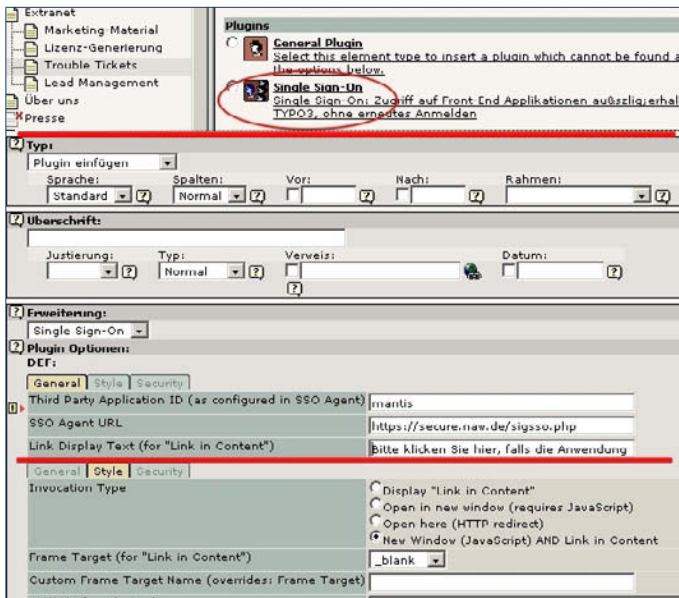
Die Einrichtung der TYPO3-Extension geschieht in wenigen einfachen Schritten. Zunächst können im Installations-Dialog des Extension Managers einige Optionen gesetzt werden – vor allem jedoch der Pfad zum „private Key“.



Installation der TYPO3-Extension.

Nun können an den gewünschten Stellen im Seitenbaum die SSO-Verlinkungen zu den Fremdsystemen als Plugin eingefügt und konfiguriert werden. Dabei ist unter „SSO Agent URL“ die Adresse einzugeben, unter der der SSO-Agent zu erreichen ist (Beispiel: „https://secure.naw.de/sigssso.php“). Die „Third Party Application ID“ hingegen ist ein beliebiger Bezeichner, der lediglich mit der ID übereinstimmen muss, die in der sigssso.conf unter „[globals]“ verwendet wurde (Beispiel: „mantis“).

Schließlich muss das Verhalten bestimmt werden: Soll lediglich ein Link angezeigt werden, und bei Klick auf diesen Link öffnet sich die Fremdanwendung („Link in Content“)? Oder soll sich ein neues Fenster öffnen, sobald diese Seite aufgerufen wird („open in new window“)? Diese und andere Varianten finden sich unter dem Reiter „Style“.



Die wichtigsten Konfigurations-Optionen des Plugins.

Was bleibt ist Testen. Bei Problemen hilft neben der umfangreichen Dokumentation auch das SSO-Forum [2] ) und manuelles „Feintuning“. Ans Herz gelegt sei vor allem der Abschnitt „Security Configuration Tasks“ in der Extension-Dokumentation.

## Einsatzfelder und erweiterte Möglichkeiten

Mit der beschriebenen Technik lässt sich prinzipiell jede Anwendung einbinden – wenn einige Grundvoraussetzungen erfüllt sind:

Nicht erforderlich
gemeinsame/gleiche Benutzer/Kennworte (LDAP etc.)
Installation auf gleichem Server
Server-zu-Server Kommunikation
Erforderlich
Unterstützt wird nur Browser-Zugriff.
Die Fremdanwendung muss ein eigenes Benutzer-Handling haben (sich also z.B. nicht nur auf „htaccess“ des Webservers verlassen).
Zu jeder Fremdanwendung wird ein passender Adapter benötigt!

Eine Vielzahl von Adaptern findet sich unter [3], es ist jedoch möglich und gewünscht, bei Bedarf selbst Adapter zu entwickeln. Da hierzu Einblick in die (Session-) Arbeitsweise der Fremdapplikation erforderlich ist, ist die Adapterentwicklung bei „closed source“-Anwendungen unter Umständen sehr aufwendig – bei Quellcode offenen Systemen hingegen typischerweise an einem Tag zu bewältigen. Je weitergehender man die Integration realisieren möchte, desto mehr stößt man auch an die Grenzen der gegenwärtigen Implementierung: Nicht Bestandteil ist zum Beispiel die optische Integration – dies kann im einfachsten Fall „manuell“, also durch Verwendung gleicher CSS-Informationen etc., geschehen.

Ein Detail, welches im Rahmen des applikationsspezifischen Adapters implementiert werden muss, ist das Verbergen der Benutzer-/Kennwortverwaltung in der Fremdapplikation (denn ein „Kennwort ändern“ in der Zielapplikation wäre irreführend, da das dortige Kennwort nicht mehr verwendet wird).

Die Forderung, dass ein Benutzer in der Zielapplikation hinterlegt sein muss, bevor diese per SSO verwendet werden kann, wird mit der neuen „Version 2“ des Single Sign-On Frameworks aufgehoben: Nunmehr kann die Anbindung so konfiguriert werden, dass ein Benutzer in der Ziellanwendung automatisch angelegt bzw. aktualisiert wird, wenn seine Daten im TYPO3 geändert wurden – eine große Verbesserung insbesondere in Verbindung mit Benutzerselbstregistrierung.

Weitere fortgeschrittene Optionen, wie z. B. das „User Mapping“, finden sich in der Dokumentation wieder. Das künftige Entwicklungstempo wird, wie so oft, vom praktischen Einsatz abhängen – Ideen gibt es jedenfalls genug.

## Links und Literatur

- [1] Single Sign-On für TYPO3: <http://www.single-signon.com>
- [2] SSO-Forum: <http://www.single-signon.com/en/support/forum.html>
- [3] SingleSignOn-Website: <http://www.single-signon.com/>

### DER AUTOR



Ekkehard Gümbel ist seit 1997 Geschäftsführer der net&works GmbH (www.naw.de) in Hannover und leitet dort TYPO3-Kundenprojekte. Seit elf Jahren liegt ein Tätigkeits-Schwerpunkt auf Hochleistungs-/Hochverfügbarkeits-Systemen. Im TYPO3-Projekt ist er für Sicherheit zuständig und zudem Mitbegründer der TYPO3 User Group Hannover.