

## WEB UI



BrixWire ist über Webbrowser (bevorzugt unterstützt werden Chrome und FireFox) aufrufbar. Die Web UI kann durch eine kundenspezifische Entwicklung ersetzt werden, da sie auf einer vollständigen REST Api beruht.

## BETRIEBSMODI

BrixWire wird sowohl interaktiv als auch automatisiert genutzt.



*Interaktiv:*

ein Benutzer entscheidet, ob und wo eine Story veröffentlicht werden soll

*Automatisiert:*

Über Regeln in BrixWire wird festgelegt, dass eine eingehende Story automatisiert publiziert werden soll. Die Regeln bestimmen, welches Ressort, Status und für Printartikel, welches Issue gewählt werden sollen.

Interaktiv wird normalerweise eingesetzt, wenn nur ein Teil der Stories publiziert werden soll (beispielsweise Agenturmeldungen). Automatisiertes Publizieren wird dann eingesetzt, wenn die Mehrzahl der eingehenden Stories sofort auf die jeweiligen Outputkanäle durchpubliziert werden sollen und die Nachbearbeitung auf diesen Channels stattfindet

## PLUGIN ARCHITEKTUR (INPUT / OUTPUT)

BrixWire arbeitet in fast allen Bereichen plugin-orientiert. Hierdurch können kundenspezifische Umgebungen einfach adaptiert oder konfiguriert werden



### ALLZWECK-PLUGINS

- Desknet
- File System
- FTP
- WoodWing
- DCX
- RSS
- Mail
- Manual Input



### SOCIAL MEDIA PLUGINS

- Twitter
- Facebook
- Whatsapp
- LinkedIn

## DAM PLUGINS

- DMS Light
- WoodWing Assets (Elvis)

## CMS PLUGINS

- Drupal
- Joomla3!
- Wordpress
- LivingDocs
- Typo3

## EINGABE-FORMATEN



### *XML:*

Wird konfigurierbar in das Zielformat übersetzt (XSL).

### *JSON:*

Wird derzeit via XML Import konvertiert.

### *PlainText / Regex:*

Aus PlainText wird via RegEx ein XML erzeugt, das über die XML Konvertierungen dann weiter verarbeitet wird.

### *Office Formats:*

Word, Excel, PDF

### *Other Formats:*

Werden via DAM Plugin zur Story gespeichert.



## AUSGABE-FORMATEN

Generell: dieselben Formate wie für Inputs werden unterstützt.

## PLUGINS FÜR DIE BENUTZERVERWALTUNG

Die Plugins der Benutzerverwaltung führen die Authentifizierung und Autorisierung durch.

### AUTHENTIFIZIERUNG

#### *Built in provider:*

Proprietärer Provider für kleine Installationen ohne spezielle Anforderungen an Authentifizierung. Kennwörter werden nicht im Klartext gespeichert.

#### *LDAP / Active Directory:*

Wird gerne für eine OnPremise Installation verwendet und bindet das Kundensystem für Authentifizierung an BrixWire..

#### *OAuth2:*

Wird im Rahmen von Cloud basiertem Zugriff oder Erreichbarkeit der Installation ausserhalb des Firmennetzes eingesetzt.

## AUTHORISIERUNG

Derzeit werden folgende Berechtigungsstrukturen über ein Gruppenkonzept unterstützt:

- Zuordnung von Eingang Kanälen und
- Publikationen (Ausgangskanäle) zu Benutzergruppen.

Das Gruppenkonzept wird mit dem Authorization Plugin des Endkunden verbunden.



## WEB API PLUGINS

Web Apis können kundenspezifisch eingerichtet werden. Sie dienen als Frontend der Integration für Push orientierte Integrationen und stehen neben der API von BrixWire.



## MONITORING PLUGINS

Monitoring erfolgt in BrixWire in einem logunterstützten Format, BrixWire dient als Datenquelle für Monitoring Daten. Die Monitoring Daten können über externe Monitoring Systeme, wie Nagios historisiert, verdichtet und interpretiert werden.

## CUSTOMIZIERBARKEIT

Findet auf vielen Ebenen statt:

### *API:*

BrixWire stellt eine REST basierte API zur Verfügung, über die es von außen gesteuert werden kann

### *UI:*

Eine eigenständige oder vollständig customisierbare UI ist über die REST Api möglich

### *Plugins:*

Sämtliche Plugins arbeiten gegen Interfaces, das beinhaltet Input / Output / DAM Plugins

### *PIM Plugins für die Konvertierung:*

Diese sind konfigurierbar (teilweise komplexe XSL / JScript)

### *AutAut:*

BrixWire verfügt über ein Plugin Modell, das die Integration unterschiedlicher AutAut Technologien erlaubt

### *Custom API:*

Custom APIs werden über einen Plugin Mechanismus für API Komponenten unterstützt.

### *In Arbeit:*

V8 engine basiertes Event Modell, über das via JScript Content oder Metas angereichert / verändert werden können.

*Hinweis:* bis die API final ist, besteht das Risiko der API Änderung, d.h. eigenständiges Customizing wird nicht bevorzugt angeboten.





## I18N

Internationalisierungsmodule erlauben die Übersetzung von BrixWire in beliebige Sprachen. Zusätzlich sind sie zuständig, für die Ortsbestimmung einer Story (Land, Region, Stadt) und ermöglichen somit eine teilautomatische Attributierung.



## TECHNICAL ENVIRONMENT

BrixWire läuft auf Windows und Linux, kann von verschiedenen Webservern (Apache / IIS) gehostet werden und mit einer Datenbank (MS SQL, PostgreSQL) verbunden werden. Anschließend wird die Dockerisierung der Anwendung in Angriff genommen. Wir laufen auch in der Cloud und unterstützen Load Balancing.



## DATENBANK

BrixWire kann in existierende SQL Server oder PostgreSQL (und Cluster) mit eingebunden werden. Alternativ kann eine lizenzfreie DB Umgebung mit eingerichtet werden.

## SAAS

Wird derzeit noch nicht installiert, da die Redaktionssysteme hierzu einen Zugang in die Cloud anbieten müssten. Architekturell ist BrixWire mandantenfähig (im Sinne von Lizenznehmer).



## ON PREMISE INSTALLATION

BrixWire wird auf einem Server des Endkunden installiert und durch uns gewartet. Da hierdurch die Redaktionssysteme nicht speziell geschützt werden müssen, ist das die derzeitige Hauptinstallationsart.



## UPDATEZYKLEN

Updates werden alle 3-6 Monate von uns installiert. Die Produktionsunterbrechung für ein Update beträgt in der Regel 5 Minuten.