

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

FILE SYSTEM PLUGIN (HAUPTSÄCHLICH INPUT)

Das Filesystem Plugin überwacht einen Eingangskanal, liest bisher nicht verarbeitete Dateien und transformiert diese über das PIM in das zentrale Format.

Für Dateibasierte Exporte kann das File System Plugin als Output Plugin genutzt werden.

FTP INPUT PLUGIN

Analog dem Filesystem Plugin wird ein FTP Folder und optional dessen Unterverzeichnisse verarbeitet.

Derzeit ist kein FTP basierter Export (Outputplugin) in Planung

WOODWING PLUGIN



Das Woodwing Plugin exportiert die BrixWire Daten nach Woodwing. Hierzu werden kundenspezifische Transformationen und Metadaten erzeugt. Das Woodwing Plugin ist in der Lage, auch als Input Plugin zu arbeiten. Auf diese Weise kann auch ein Print First Workflow umgesetzt werden.

DCX PLUGIN



Das DCX Plugin exportiert die BrixWire Daten nach DCX. Hierzu werden kundenspezifische Transformationen und Metadaten erzeugt. Das DCX Plugin ist in der Lage, auch als Input Plugin zu arbeiten. Auf diese Weise kann ein DCX First Workflow umgesetzt werden.

RSS PLUGIN

Das RSS Plugin erlaubt die Synchronisation einer beliebig langen Liste an RSS Inputs. Es ist derzeit als Input Plugin implementiert. RSS Output ist möglich, hat jedoch aktuell keine Priorität

TWITTER PLUGIN



Liest konfigurierbare Tweets ab und stellt diese in einem Channel zur Verfügung. Wird derzeit ergänzt über eine Plausibilitätsrecherche zur Überprüfung / Recherche anderer Stories. Das Plugin ist als Input Channel implementiert, eine Output Funktion „publish to Twitter“ ist aktuell nicht vorgesehen.

TYPO3 PLUGIN

Neben Printsystemen können auch CMS angebunden werden. Aktuell implementiert sind Plugins für Typo3 und für Drupal.

DESKNET PLUGIN



Importiert geplante Stories nach BrixWire. Wird normalerweise mit dem AutoPublish kombiniert so dass der geplante Storycontent ohne manuelle Interaktion im Redaktionssystem oder im CMS zur Verfügung steht. Die Zusammensetzung der Stories aus Desknet Content ist umfangreich konfigurierbar.

INPUT



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 Formate

Sind teilweise für Excel / Word vorhanden

Andere Formate

Werden via DAM Plugin zur Story gespeichert

OUTPUT



Generell: dieselben Formate wie für Inputs werden unterstützt, speziell:

- WoodWing, Adobe Formate (WCML, IDML etc) als XML basierte Formate
- JSON, AtomPub etc.

KONVERTIERUNG



Textbasierter Input wird einer Konvertierung unterworfen, um für das Multi Channel Publishing zur Verfügung zu stehen.

ASSET MANAGEMENT PLUGINS



Über den Plugingedanken können analog DAMs eingebunden werden, so dass einer Integration eines kundenspezifischen DAMs nichts im Wege steht.

DMS Light

Intern verwendetes, DB gestütztes filebasiertes Asset Management. Wird auch für einen schnellen Cache der Previews verwendet

Elvis

Anbindung des Elvis DAM mit standardisiertem Funktionsumfang.

PLUGINS FÜR DIE BENUTZERVERWALTUNG



Über die Plugins zur Benutzerverwaltung wird die Authentifizierung und die Authorisierung durchgeführt.

Authentifizierung

Built in provider

Proprietärer Provider für kleine Installationen ohne spezielle Anforderungen an Authentifizierung. Passwörter 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 Eingangskanälen 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.

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.

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 customizierbare 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.

TECHNISCHE UMGEBUNG



BrixWire wird unter einem IIS installiert und an eine Datenbank (MS SQL, PostgreSQL) angeschlossen. BrixWire ist derzeit auf einen Windows Server (2012+) und einen SQL Server (2012+) angewiesen. Die Migration zu Linux wird derzeit durchgeführt, d.h. das Hosting wird zukünftig auch über Non Windows Maschinen erfolgen. Im Anschluss wird die Dockerisierung der Anwendung angegangen.

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 im 3-6 Monatsrhythmus installiert. Produktionsunterbrechung für ein Update beträgt im Minimum 3 Minuten, Regelfall für ein Update incl. Kurztest und Vorbereitung: 45 Minuten.