WEB UI

BrixWire can be accessed via web browser (Chrome and FireFox are preferred). The Web UI can be replaced with a custom development because it relies on a full REST Api.

MODES OF OPERATION

BrixWire is used both interactively and automatically.

Interactive:
A user decides if and where a story should be published.

Automated:
Rules in BrixWire specify that an incoming story should be published automatically.

Interactive is usually used if only a part of the stories is to be published (Agency Use Case). Automated is used when most incoming stories are published immediately on the respective output channels and the post-processing on these channels takes place.

PLUGIN ARCHITECTURE (INPUT / OUTPUT)

BrixWire works in almost all areas plugin-oriented. This allows customer-specific environments to be easily adapted or configured.

GENERAL PURPOSE 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)

Contact: D.Schwertfuehrer@BrixWare.com

CMS PLUGINS

- Drupal
- Joomla3!
- Wordpress
- LivingDocs
- Typo3

INPUT FORMATS

XML:
Configurable translated into the target format (XSL).

JSON:
Currently being converted via XML import.

PlainText / Regex:
An XML is generated from PlainText via RegEx, which is then further processed via the XML conversions.

Office Formats:
Word, Excel, PDF

Other Formats:
Stored via DAM plugin to the story.

OUTPUT FORMATS

Generally: the same formats as for inputs are supported

PLUGINS FOR USER MANAGEMENT

The user management plugins perform authentication and authorization.

AUTHENTICATION

Built in provider:
Proprietary provider for small installations without special authentication requirements. Passwords are not stored in plain text.

LDAP / Active Directory:
Willingly used for an on-premise installation and binds the customer authentication system to BrixWire.

OAuth2:
Is used as part of cloud-based access or accessibility of the installation outside the company network.

Contact: D.Schwertfuehrer@BrixWare.com

AUTHORIZATION

Currently, the following authorization structures are supported via a group concept:
- Assignment of input channels and
- Publications (output channels) to user groups.

The group concept is connected to the end user's authorization plugin.

WEB API PLUGINS

Web Apis can be customized. They serve as a front-end to integration for push-oriented integrations and stand alongside the BrixWire API.

MONITORING PLUGINS

Monitoring takes place in BrixWire in a proprietary format. This format communicates with monitoring plug-ins, which implement the implementation on the respective cockpit technology.

CUSTOMIZABILITY

Takes place on many levels:

API:
BrixWire provides a REST based API that allows it to be externally controlled.

UI:
An independent or completely customizable UI is possible via the REST Api.

Plugins:
All plugins work against interfaces, this includes input / output / DAM plugins

PIM plugins for conversion:
These are configurable (sometimes complex XSL / JScript).

AutAut
Externally hardly feasible, internally a plugin model is set up.

Custom API:
Custom APIs are supported via a plugin mechanism for API components.

Work in progress:
V8 engine-based event model, via which JScript content or Metas can be enriched / changed.

Note: until the API is final, there is a risk of the API change, i. independent customizing is not preferred.

Contact: D.Schwertfuehrer@BrixWare.com

BRIXWIRE
News Hub

I18N

Internationalization modules allow you to translate BrixWire into any language. In addition, they are responsible for the location of a story (country, region, city) and thus enable a semi-automatic attribution.

TECHNICAL ENVIRONMENT

BrixWire runs on Windows and Linux, can be hosted by various webservers (Apache / IIS) and connected to a database (MS SQL, PostgreSQL). Subsequently, the dockerization of the application will be tackled. We are running in the cloud as well and support load balancing.

DATABASE

BrixWire can be integrated into existing SQL Server or PostgreSQL (and clusters). Alternatively, a license free DB environment can be set up.

SAAS

Currently not yet installed, as the editorial systems would have to provide access to the cloud. Architecturally, BrixWire is multi-client capable (in the sense of licensee).

ON PREMISE INSTALLATION

BrixWire is installed on a server of the end customer and maintained by us. Since this does not require the editorial systems to be specifically protected, this is the current main installation type.

UPDATE CYCLES

Updates are installed every 3-6 months by us. Production interruption for an update is commonly 5 minutes.

Contact: D.Schwertfuehrer@BrixWare.com