WEB UI

BrixWire can be accessed via a web browser (Chrome is 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 story 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
- etc.

SOCIAL MEDIA PLUGINS

- X (Twitter)
- Telegram
- Facebook

DAM PLUGINS

- DMS Light
- WoodWing Assets (Elvis)

CMS PLUGINS

- Drupal
- Wordpress
- LivingDocs

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.

SAML2:
With OKTA and Azure ADFS

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. Supported platforms include for example Grafana and Prometheus.

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).

Authorization / Authentication:
Externally hardly feasible, internally a plugin model is set up.

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

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

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 web servers (Apache / IIS), and can be connected to a database (MS SQL, PostgreSQL). Additionally, it can run within a Docker. In the first step, it is a hosting alternative for our customers. In a further step, the deployment of Docker is used for automated updates of released versions.

Advantages include a more stable environment, as Docker is customized and released by us, as well as less dependency on the environment and its installation, and often more favorable hosting conditions. 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.

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.

DEVELOPER PACK

This package is designed for large projects or for customers who have requirements that go beyond the standard installation. The idea behind it is to involve you in further development and provide you with innovative functionality independent of the general BrixWire Schedule.

Includes:
- Possibility to bring own ideas and improvements
- Integration into the BrixWire communication system (chat & backlog)
- Participation in our planning and reviews, every two weeks
- Possibility to discuss as a partner

More information here: Link

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