FirstSpirit™
Unlock Your Content

FirstSpirit
Roadmap 2013 - 2017

Version	**1.01**
State	**RELEASED**
Date	**2014-02-13**
Department	FS-Core
Copyright	2014 e-Spirit AG
File name	ROAD51EN_FirstSpirit_Roadmap

e-Spirit AG
Barcelonaweg 14
44269 Dortmund | Germany

T +49 231 . 477 77-0
F +49 231 . 477 77-499

info@e-Spirit.com
www.e-Spirit.com

e-Spirit

Table of contents

1 Document contents

The "FirstSpirit Roadmap 2013 - 2017" document introduces the current project status with regard to the further development of FirstSpirit™ and describes the strategic direction of the product. Chapter 2 (starting on page 4) focuses first on FirstSpirit release management and maintenance periods. Chapter 3 (starting on page 9) outlines the history of FirstSpirit development since it was first released. The featured functions deployed in FirstSpirit version 5.0 are presented briefly in chapter 4 (starting on page 12).

Chapter 5 (starting on page 14) provides a look at the strategic considerations made with regard to the further development of FirstSpirit version 5.x. Application integration is extremely essential to the product's development. Section 5.2 will cover this subject in detail. Additional information specifically with regard to the technical aspects of application integration plans are also available in the *FirstSpirit 2014 Whitepaper*.

Chapters 6, 7 and 8 provide an overview of the upcoming versions 5.1, 5.2 and 6.

This document is intended for readers who have a technical background in a relevant field and ideally are already familiar with FirstSpirit™.

The FirstSpirit applications have been renamed in FirstSpirit version 5.1. These applications are now known as ContentCreator and SiteArchitect (refer to section 5.1, starting on page 18 for more information). These names will be used consistently in future sections (starting with Chapter 6).

2 FirstSpirit release management

FirstSpirit has been on the market for over ten years. The company has followed a consistent product release policy from the start of product development. FirstSpirit is a highly innovative product established in a market segment that is dependent on extremely long-term maintenance periods and that demands high quality software, which presents special challenges. FirstSpirit meets these high demands for quality, long-term maintenance periods and technical innovation through systematic release management. This gives customers and partners the ability to make accurate and reliable upgrade plans; migration expenses – even in the case of updates and major version releases – are known to be minimal.

- **Major versions** are identified by the first digit in the version number (e.g. version **5**.0) and represent a significant change in the software's functional performance. Major versions of FirstSpirit are released every four to five years and usually include two to three minor versions (see below). Due to extensive structural changes to the product, implementing a new major version is usually associated with migration expenses for existing customers. However, special product functions (e.g. migration wizards) are provided to help with the migration process.

- The **maintenance support period for major versions** is set up for an extremely long period of time and is valid until the release of the major version that follows the next major version (Long Term Support, abbreviated LTS). For instance, the 3.x major version maintenance support period extended from the first quarter of 2005 (version 3.0 release) to September of 2012 (release of version 5.0), which is over seven years (see Figure 2-1). Customers can basically count on long term support, allowing them to systematically control the financing and scheduling of updates and the migration of existing projects.

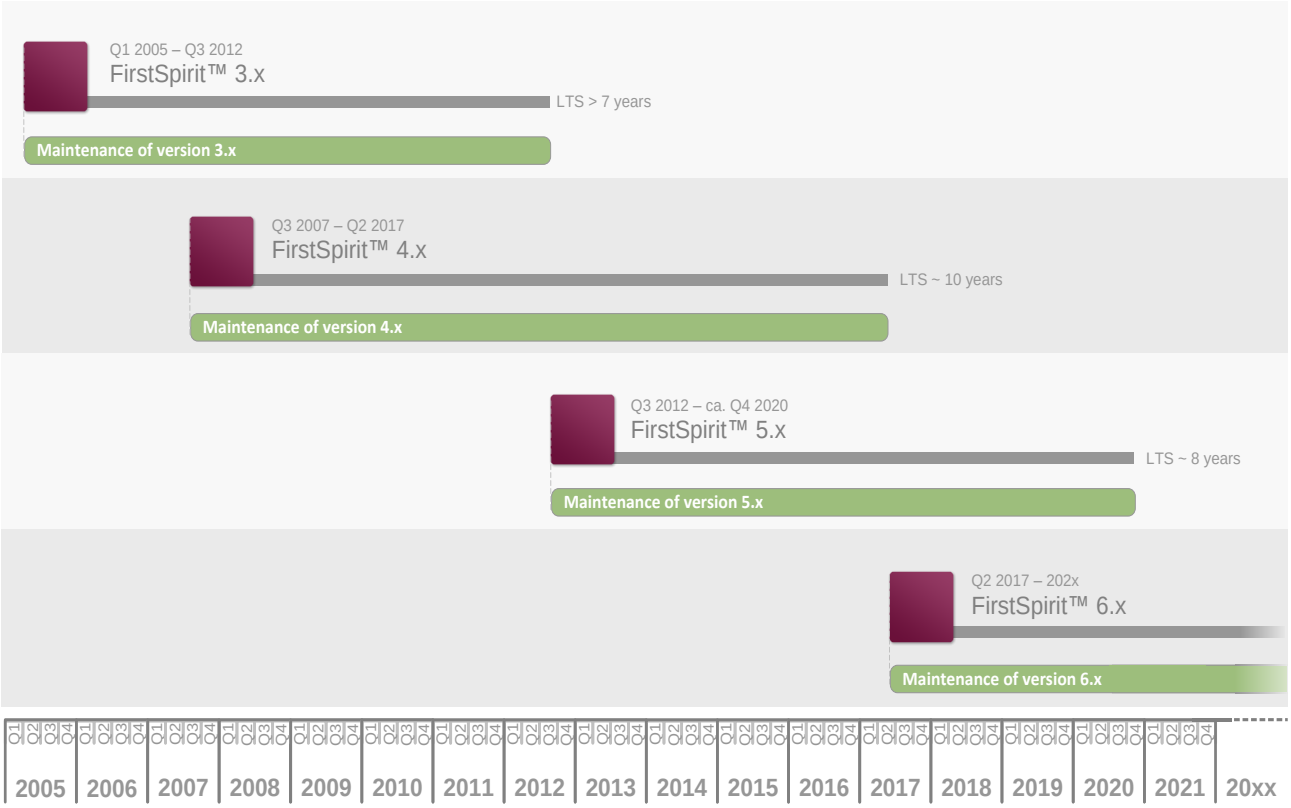


Figure 2-1: Long Term Support (LTS)

- **Minor versions** represent the development cycles within a major version and, as is generally typical, are indicated by the digit following the period in the version number (e.g. version 4.**2**). Minor versions appear much more frequently than major versions (approximately every 12 to 16 months) and allow for the quick integration of new, innovative functions into the product (for example: "integrated preview" and "AppCenter" in version 4.2). Due to the shorter cycles, minor versions are usually designed in such a way that existing customers do not incur any project-side migration costs when upgrading to the subsequent minor version. Exceptions to this may occur occasionally and are announced far in advance as part of "deprecation management" (see page 8).
  However, if the new minor version functions integrated in the product are to be implemented at the project level, (depending on the type of functional enhancement) implementations may be required in a particular project. These adaptations are optional, however, since new functions do not have to be implemented. After updating to a new minor version, a project can also remain unchanged and can continue to be used with the original functions.

- The **maintenance period for minor versions** (Medium Term Support, abbreviated: MTS) extends until the release date of the subsequent minor version and the 12 months thereafter. A minor version can thus be used by a customer for approximately 24 months with full software support. This means, for instance, that support for version 4.1 continues for another 12 months after the release of version 4.2 in order to provide customers with a smooth transition.

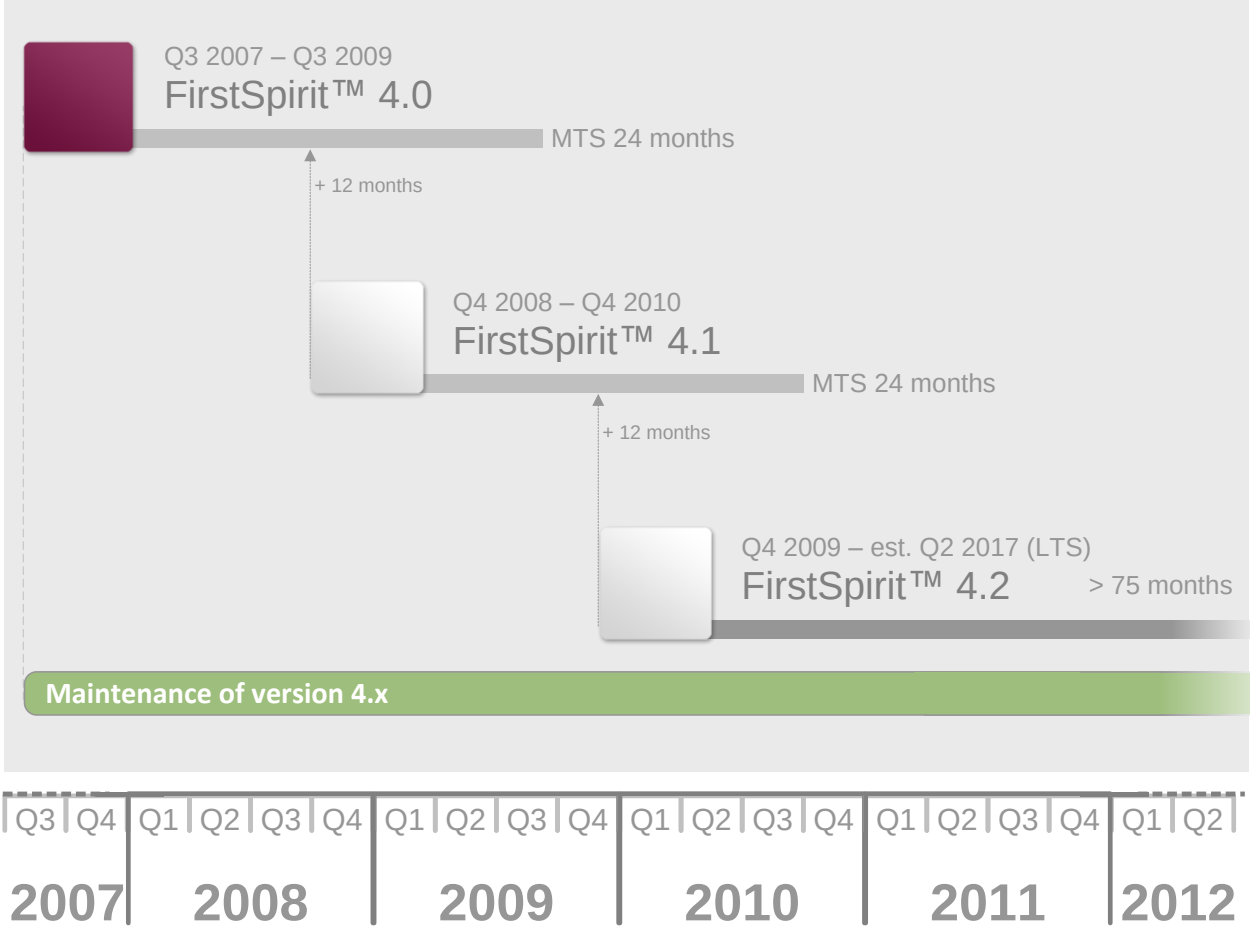
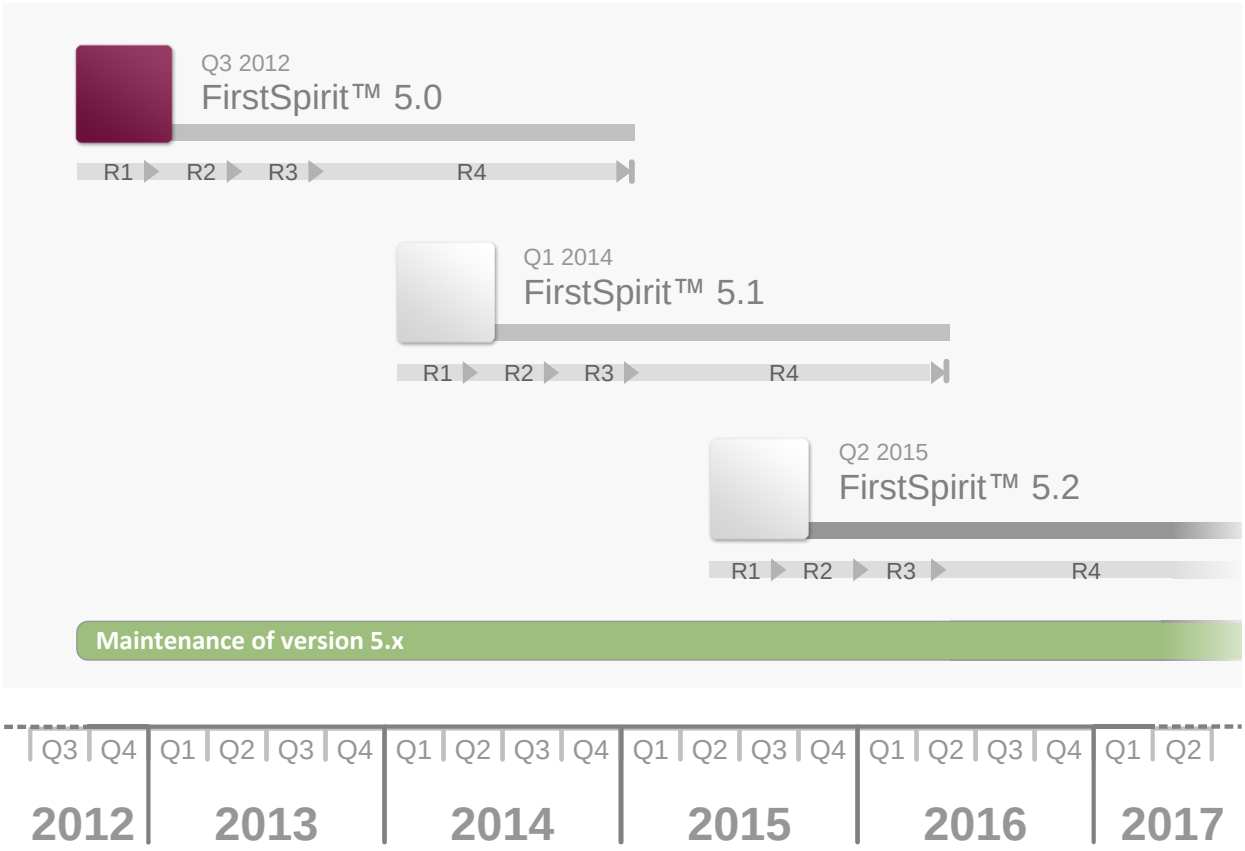


Figure 2-2: Medium Term Support (MTS)

This time period is usually sufficient due to the low (or non-existing) project-side migration expenses and the ease of upgradeability of FirstSpirit. After 12 months, support for the previous minor version is discontinued. At this point, the conversion to the new (by then already over a year old) version should have been completed in order to be able to continue to receive software support. An exception to this is the last minor version of a major version (in Figure 2-2: version 4.2). For this version, the LTS applies automatically and extends to the two major versions released thereafter (in the case of version 4.2 that would be version 6.0, for instance).

- **Release versions** are viewed as standalone software versions within a series of minor versions and are indicated by the addition of "Release" and a release number, such as "V5.0 Release 4". The numbering of release versions within a minor version started in version 5 with "1", but the "1" in particular is frequently omitted, since the minor version name, such as "V5.0" for "V5.0R1", is sufficient for the first version of a series of minor releases. Release versions are primarily used for deployment of bug fixes as part of software support and are deployed only to a limited degree as new function enhancements that are needed quickly (delivery of new functions is reserved for major and minor versions). Release versions starting with FirstSpirit V5.0 appear quickly at about 3 to 4 month intervals. They are designed so that the software update does not involve any migration costs on the project side. All bug fixes falling under the maintenance agreement are carried out exclusively on the current state of a version series. Consequently, customers who want to implement these bug fixes must update their software to the current release version.

- **Release version maintenance periods**: Due to the very short deployment periods (3 to 4 months), release versions do not have a maintenance period, which means that bug fixes are only made and released in the respective current release version.



- **Hotfix versions:** These contain only critical bug fixes that cannot wait until the release of the next release version. Hotfix versions are deployed to respond very quickly to these rare occurrences and are usually available within a few days or weeks. Due to the limited time available to perform quality control, hotfixes are provided only to the customers who are directly affected by the particular issues. Hotfixes are officially released and thus generally available to all once the next release version is released (see above).

- **Deprecation management**: one goal of FirstSpirit release management is to make the upgrade to a new version of FirstSpirit as easy as possible. This of course applies to quick bug fixes via release or hotfix versions, but even minor versions are usually designed in such a way as to avoid any project-side migration costs to existing customers when upgrading. Therefore (specifically in the case of programmer interfaces), new, enhanced functions are frequently provided and old functions are still included initially for the purpose of compatibility. However, to ensure continued, quality software support, these legacy functions are removed at particular time intervals. FirstSpirit deprecation management informs FirstSpirit users who use these legacy functions that they will no longer be available in the foreseeable future. For this purpose, the particular function is officially announced first in the release notes, but in this case can continue to be used at least until the customer upgrades to the next minor version. The next step is to generate warnings that refer to the use of a deprecated function. The deprecated functions are only removed in the third and final step. At this point, project-side conversion is required or else it will not be possible to upgrade to the next version of FirstSpirit. Deprecation management is actually separate from minor and major versions. However, deprecations usually only occur with major versions or for LTS versions.

3 A look back at FirstSpirit development from 1999 to 2011

After taking up the development of a Java-based content management system (CMS) in 1999, the year the company was founded, the first FirstSpirit build (**version 0.9**) was produced in June of 2000. The first official version of FirstSpirit (**version 1.0**) was released and deployed at the beginning of 2001. Version 1.0 already contained the fundamental concept of the strict separation of structure and content as well as broad multilingual support, which is still missing in many other content management products.

The introduction of FirstSpirit **version 2.0** (09/2001) added the Data Store to the existing "store" concept that separates the layout, content (page) and structure (site) in FirstSpirit. This was a key feature for editors and gave them the ability to enter and manage database-supported, structured content in a table format. It is still one of FirstSpirit's unique selling points today.

The first version of the "FirstSpirit CorporateContent", deployed in FirstSpirit **version 2.1** (09/2002), made it possible to efficiently reuse a project's objects in other projects.

FirstSpirit **version 2.2** (01/2003) introduced a mechanism as part of FirstSpirit metadata that could be used to define project-specific data structures which permit a hierarchical description of the data in the Page, Site and Data Stores.

FirstSpirit **version 2.3** (05/2003) included the first official version of WebClient ("ContentCreator" in FirstSpirit version 5.1 and higher, see also section 5.1, starting on page 18). Since then, FirstSpirit has consistently pursued this dual-client strategy.

FirstSpirit **version 3.0** (10/2004) development included for the first time a graphical editor for database schemata and a graphical user interface for designing workflows. These editors could be used by template developers to model and control workflows on their own without outside support, such from as their IT department. In addition, the first version of the "SAP Business Package for FirstSpirit" was released, which enabled the integration of navigation structures and content from FirstSpirit into the SAP Enterprise Portal.

Instead of implementation work, FirstSpirit **version 3.1** (08/2005) focused on support of high availability environments ("FirstSpirit HighAvailability").

FirstSpirit **version 4.0** (08/2007) was developed over several years in parallel to the further enhancement of the FirstSpirit version 3.1 product line. In this case, the core of FirstSpirit Server was fundamentally overhauled with the intention of optimizing FirstSpirit for scalability and high performance.

FirstSpirit **version 4.1** (09/2008) concentrated on the further reworking of the JavaClient GUI (SiteArchitect) as well as on optimizing editorial processes with new functions. This gave JavaClient (SiteArchitect) a completely fresh and modern look and feel. A new search function and the integrated image cropping feature were among the key new functions in this version.

Since FirstSpirit **version 4.2** (11/2009), the product has included the integrated preview and Content Highlighting in addition to existing workspaces ("multi-tabbing") in order to provide a completely new user experience in JavaClient (SiteArchitect). These new functions offer all users the ability to customize their own workspace environment based on their individual needs and work processes. User guidance in WebClient (ContentCreator) was improved by the introduction of "Easy Edit", which gave editors an even greater sense of control by allowing them to change content directly in the website.

FirstSpirit **version 4.2 release 2** (abbreviated: 4.2R2, 03/2010) included the initial steps in the introduction of application integrations. The first application integrated in this version was Microsoft Internet Explorer.

The release of FirstSpirit **version 4.2 release 4** introduced the highly innovative application integration ("FirstSpirit AppCenter") technology at the beginning of 2011. Integration solutions and examples based on this powerful and fresh, one-of-a-kind AppCenter technology include, among other things:

- **Fully integrated use of office suite products** in the editing system (word processing, spreadsheets and presentations, with the option of using Microsoft Office or Oracle OpenOffice) ("FirstSpirit OfficeIntegration")".
- **Embedding of Microsoft Media Player** for seamless visualization of media objects (images/audio/video).
- **Integrated image editing** with the choice of using a Java application or the online version based on Google Picnik (http://picnik.com/, known from Picasa and Google+) or Autodesk Pixlr (http://pixlr.com/).

- **Integration of various online web services: geolocation** in Google Maps (http://maps.google.de) and Google Earth (http://earth.google.de), **image banks** such as Fotolia (http://fotolia.de) or Pixelio (http://pixelio.de), **translation services** such as the Google Translator Toolkit (http://translate.google.com/toolkit).

For information on "Application integration", also refer to the supplemental section 5.2, page 23.

The featured functions deployed in **version 5.0** are presented briefly in the following chapter.

4 FirstSpirit Version 5.0 Highlights (09/2012 release)

The major release of FirstSpirit version 5.0 featured an extensive range of innovations covering several areas. The most important, core revisions involved the redesign of **WebClient** (called "ContentCreator" in FirstSpirit version 5.1 and later; also refer to section 5.1, starting on page 18) based on a changed user interface and command concept, with new technology and a new code base. This is consistent with our mission to provide an easy and intuitive user interface. To achieve this objective, the availability of preview-based work has been increased and operation has been more streamlined by reducing tasks to functions and content that are actually relevant to editing work performed on a daily basis. On the other hand, compared to older versions of WebClient (ContentCreator), version 5.0 features new functions for streamlining complex work steps and several expansion and customization options to respond more flexibly to customer requirements. Functions that are not included as standard but are desired for a project can be introduced as a module and can be integrated into the new WebClient (ContentCreator) by the customer.

JavaClient (known as "SiteArchitect" in FirstSpirit version 5.1 and later; also refer to section 5.1, starting on page 18) has also been revised for improved visual and functional performance. One major objective was to provide the user with additional assistance in retrieving and reusing existing content, particularly in the case of large projects. The **integrated search** feature was updated with new, powerful functions to help retrieve content faster. The search function now permits direct filtering of search results based on a wide variety of criteria, thus reducing the number of hits down to the most relevant without having to restart the search. Another key feature is the ability to search for FirstSpirit objects or inserted values using drag-and-drop. The user is able to quickly view additional areas where these objects or values are used in the project. For an even more convenient, project-wide reuse of existing FirstSpirit content, the functionality of **FirstSpirit CorporateContent** has been revamped and now includes a completely redesigned interface optimized for usability.

The new **dynamic forms** provide increased interaction with the editor in WebClient (ContentCreator) and JavaClient (SiteArchitect). These forms can be used to check the content and logic behind entries made by editors. Depending on what is required, it is possible to output messages adapted to the particular project that prompt the user to make a change to content, for instance, or even to prevent the user from saving or releasing content. Likewise, it is possible to create logical relationships between input components or their content so that depending on the input in a field, for instance, other

fields that include the same form as that field are hidden or appear and function differently.

In addition, JavaClient (SiteArchitect) can now be tailored even more to the individual needs and work styles of individual editors. For this purpose, the new **Organize** area helps make it possible for each user to personalize his or her own workspace environment. Frequently used FirstSpirit objects can be saved as **bookmarks**, for instance, and can be used in the modeling system as a template for the creation of new objects. Moreover, all content in the **clipboard** can be managed more easily, since it can now be viewed directly in JavaClient (SiteArchitect). The new **Bird's-Eye View** provides a better overview of the content currently being edited and faster navigation between the different content. It now also allows JavaClient (SiteArchitect) to compare content even when using the AppCenter.

In addition, the **FirstSpirit API** has been significantly enhanced in many areas to allow for the implementation of individual solutions and to adapt FirstSpirit even more to customer-specific requirements. For instance, an API interface has been developed that can be used to create a custom "URL generator", which can generate URLs on demand. This way it is possible to undertake search machine optimization (SEO), such as by using "talking" URLs that are easier for users to understand and are easier for search machines to analyze. The "Advanced URL Creator" referencing implementation makes it possible to develop a URL generation strategy based on the display names of FirstSpirit objects instead of on the reference names in FirstSpirit's standard URL generation feature.

A key aspect in the development of FirstSpirit has always been continuous quality assurance with the aim of detecting and fixing bugs as early as possible in the software development process. The company therefore invested heavily in the bug detection infrastructure as early as FirstSpirit version 4.2R4. This infrastructure makes it possible to easily consolidate and compress bug reports at the press of a button with all the necessary software and hardware configuration information as well as the required section from the log file into a ZIP archive. FirstSpirit 5 takes **automated error reporting one step further**: all Java exceptions that occur during operation can be collected and regularly transmitted automatically from the system to e-Spirit.

5 Strategic direction of FirstSpirit V5.x

The primary focus of FirstSpirit's development and strategic direction in terms of its functional scope and release management features (see section 2) is on meeting the demands of corporate customers. To provide these customers with a mature, stable and yet highly innovative product conducive to planning while fitting their budget, the FirstSpirit product development team focuses on the "core CMS" (see Figure 5-5). Particular attention is given to developing an efficient, intuitive product that is easy to operate and is capable of automating frequently repeated processes. As a content integration platform, the purpose of FirstSpirit is not only to capture newly created content, but also to integrate existing sources of company information so that this data can also be used in the content production process.

Furthermore, to meet the considerable market demands to the greatest extent possible and to integrate new trends and technical features quickly and flexibly, the company is pursuing a "best of breed" strategy to integrate top-quality, market-leading products. The development strategy has always consisted of investing as few resources as possible into the development of software features that have already been successfully implemented in other software products and instead integrates existing product solutions from the market leaders. The innovative FirstSpirit application integration technology provides seamless and above all smooth interaction between the individual product components, not only on the back end (e.g. in terms of database integration), but also in the editing system (office or web application integration).

FirstSpirit is a sophisticated, stable core product that features standard functions and integration options and provides individual innovative, modular components that can be integrated via FirstSpirit framework interfaces. The advantages are clear: new functions can be flexibly integrated or replaced, and customers and partners can also develop their own modular solutions beyond the standard functionality for integration into FirstSpirit. This modular building-block system prevails in all areas of FirstSpirit — from back end to front end.

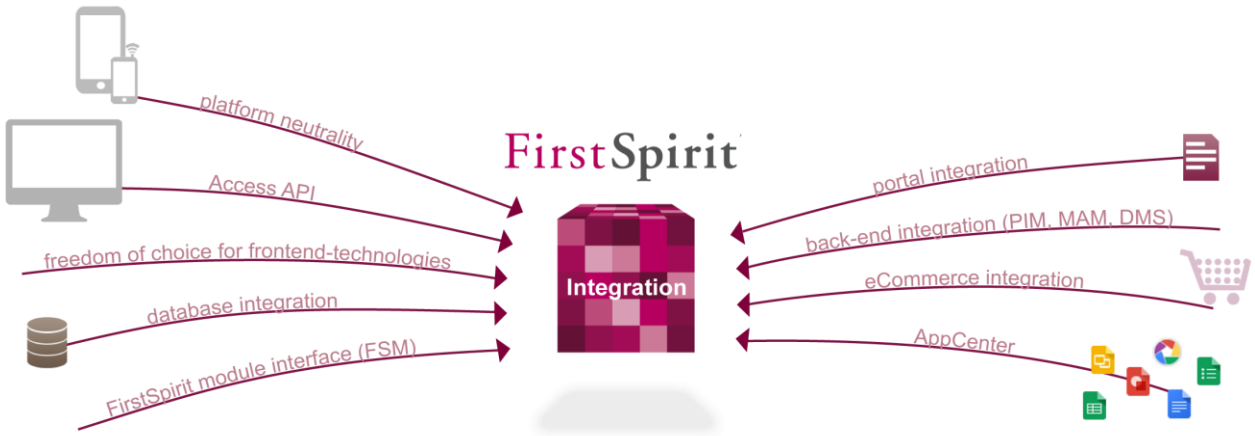


Figure 5-1: Integration options in FirstSpirit

The "best of breed" strategy gives the customer the freedom to choose ("freedom of choice") from a wide range of options without having to worry about compatibility issues or becoming dependent on a particular vendor ("vendor lock-in"). The ability to replace individual components is part of the FirstSpirit product strategy.

The example in section 5.2.2 should serve to demonstrate these abstract ideas. In this example, the web applications Google Maps (for determining geo-data) and Pixelio (for selecting images based on the geo-data) are integrated via the FirstSpirit AppCenter. These web applications can of course also be replaced, as can many other FirstSpirit product components. Thus, Bing (instead of Google Maps) and the Fotolia image bank (instead of Pixelio) can be integrated by making minor changes in JavaClient (Site Architect) (see Figure 5-2 and Figure 5-3).

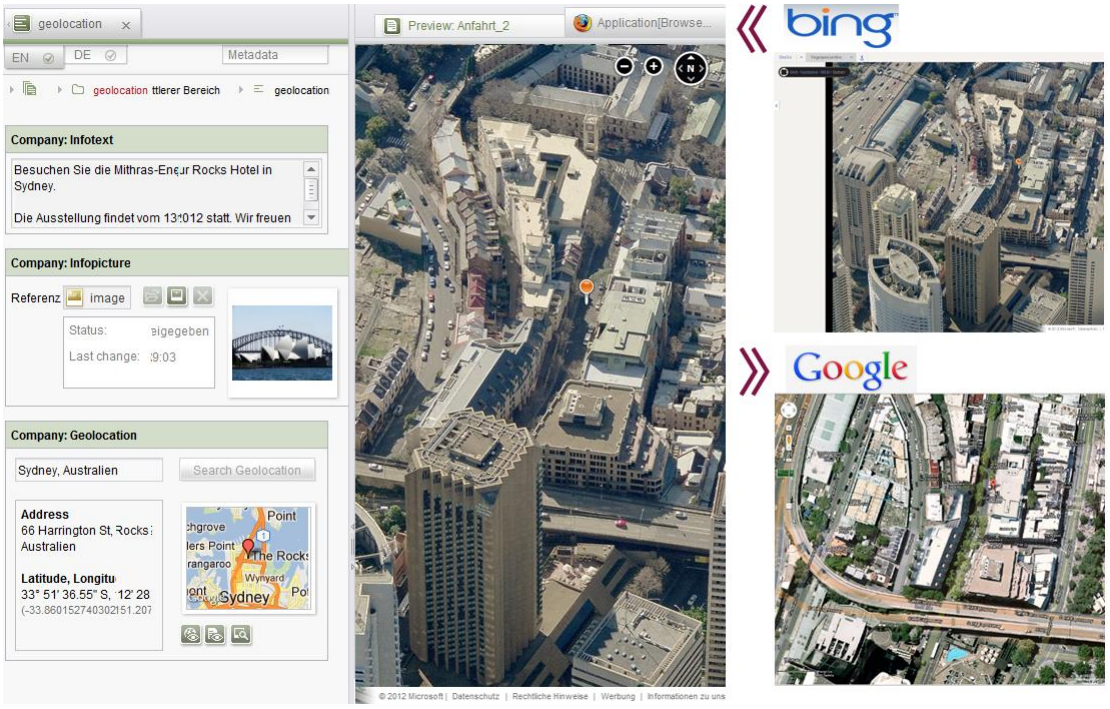


Figure 5-2: Google Maps web application replaced with Bing Maps

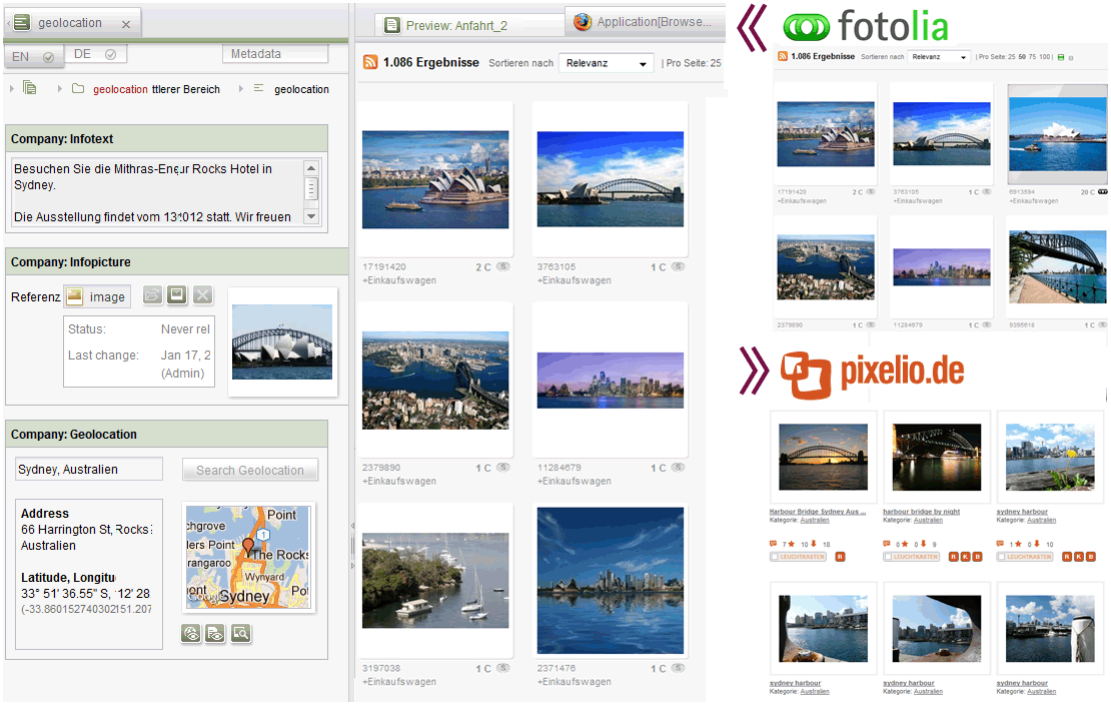


Figure 5-3: Pixelio image bank replaced with Fotolia

Another crucial aspect for an efficient enterprise content management solution is reuse. For over 10 years, FirstSpirit has consistently focused on the concept of separating structure, content and presentation material and is therefore capable of supporting multi-channel reuse (alternative structure for PDF documents, for instance) as well as content reuse in several corporate branded solutions which are completely different visually. One key in development is in optimally supporting the complete process from the development of a new product idea to the final multi-channel presence.

Another key to the success of FirstSpirit product development is in the direct communication with customers and partners. Specific partnership programs and a comprehensive communication strategy are an essential part of the e-Spirit AG business model and a key aspect in the strategic planning and further development of FirstSpirit. Partners and customers share their technical expertise and industry-specific knowledge, which adds a considerable amount of value to the implementation of customer-specific requirements and the development of practical software solutions based on FirstSpirit. In addition to purposeful management of partnerships, e-Spirit offers a variety of options related to information (website, press releases), knowledge transfer (partner collaboration programs, training programs, workshops) and the exchange of knowledge (partner networking, partner and customer event days). FirstSpirit Community, which comprises e-Spirit employees, partners and customers, is also a key component. It acts as a platform for exchanging experiences and for actively contributing ideas on ways to improve FirstSpirit. In this context, it is also important to mention our regularly held e-Spirit customer events. FirstSpirit projects used in the field are presented at these customer events. The experiences and suggestions resulting from the events also flow into the product's development.

Summary of FirstSpirit product development strategies:

- Best of breed strategy: The company integrates leading (software) systems for each area instead of developing its own product
- Extensive automation of processes for creating a new product or company presence (e.g. connecting to a domain registrar)
- Design model: Even if the online product design is to be as unique as the product to be communicated, (basic) elements of the design can be copied from a "model" and can be reused ("Corporate Design")
- Content model: Particularly in corporations and large companies, content is developed and managed centrally. The availability of this content to all parts of the company or corporate website should be fully automated
- Application model: Easily develop and manage custom apps for a job exchange, news, etc.

- Language model: Efficiently design translation processes
- Involve customers and partners to ensure continued improvement in processes

5.1 Global User Experience – user-group-specific product development

The previous section described the FirstSpirit product development strategy from the company and market perspective. This section illuminates the perspective of different user groups and their CMS requirements:

One of the key success factors for any business activity is a high level of customer satisfaction. In the case of FirstSpirit, the "customer" is the end user of a website that has been created with FirstSpirit. The early FirstSpirit versions up to about 2007 focused primarily on this user group: The aim has always been to present the website user with a modern site that supports multiple devices and is both intuitive and easy to use (**Customer Experience**). FirstSpirit redoubled its focus on state-of-the-art technology such as Ajax and Web 2.0 components and most notably was able to come up with one of the first highly integrated, multi-vendor portal integration products. Additional key enhancements to the customer experience included the introduction of personalization functions, the seamless integration of the best of breed shop, and support for mobile and print channels.

e-Spirit realized very early on that other key factors can come into play when developing and optimizing FirstSpirit besides meeting the needs of and satisfying the end user. Usability and the user experience from the perspective of other user groups also represent other important aspects of CMS development. In this context, "integration" has always been central, as a high level of integration capability is absolutely fundamental to ensuring a best of breed strategy that works (see also the introduction to the section titled "best of breed" starting on page 5). This is because no CMS that is designed for ease of use can offer all the functions desired by the various user groups. Rather, a CMS should be extendable and feature sufficient interfaces to allow the connection of external systems. That is why FirstSpirit started positioning itself as an "integration platform" right from FirstSpirit version 3.1 (about 2005). During each development phase, special attention has been paid to the issue of integration capability and to providing FirstSpirit developers at customer and partner companies with suitable interfaces (together with documentation) plus code and implementation examples (**Integrator Experience**).

Figure 5-4: FirstSpirit Global User Experience

FirstSpirit releases 4.2 and 5.0 (2010 and 2012 respectively) consistently pursued the trend toward usability optimization and focused on the key issue of **Content Creator Experience**, and in turn, on improved usability for FirstSpirit editors. During this development phase, FirstSpirit set a trend of its own by introducing the integrated browser preview with the intuitive Content Highlighting navigation concept and the AppCenter for the completely seamless visual integration of best of breed applications (Office programs, web applications, etc.) into the editing environment. In addition, the user interface of the browser-based WebClient (ContentCreator) was redeveloped from the ground up and was tailored to the needs of editors. FirstSpirit is increasingly relying on simplified controls such as drag-and-drop, reuse of elements for efficient content creation and a search-centered approach for faster orientation within a project. A special product feature now allows editors to use the AppCenter technology from within FirstSpirit to connect to almost all web applications. It consistently implements the cloud computing concept and provides decentralized website content collaboration within a team using seamlessly integrated web applications and cloud storage services.

For a long time, site users (front end) and editors (back end) have taken center stage within the context of FirstSpirit's ongoing development. Now the time has come to make life as easy as possible for FirstSpirit developers as well. In particular, version 5.1 focuses on activities connected with every aspect of template development (**Developer Experience**). The overall aim of the planned 5.1 release in 2014 is to optimize the project development cycle in order to improve time-to-market and the cost-effectiveness of FirstSpirit projects, including by integrating applications that are relevant to and useful in the development of FirstSpirit projects.

A key component of this will be "Template Highlighting", which provides a completely new type of client user guidance from the developer perspective (also refer to section 6.3.1, starting on page 38 for more information). The aim is to provide intuitive navigation within the project structure for better analysis of technical interrelationships. A tool known as the "Template Inspector" will be available to do this, which is visually oriented toward Content Highlighting for editors. This tool allows the developer to navigate interactively through the associated FirstSpirit templates starting from the integrated WYSIWYG preview. Within the templates, the fully integrated debugger can be used to fix bugs with the help of single-step debugging and the context interpreter (also refer to section 6.3.2, starting on page 39 for more information). Moreover, the software will integrate external development environments (IDEs) and version control tools (refer to section 6.4, starting on page 41 for more information).

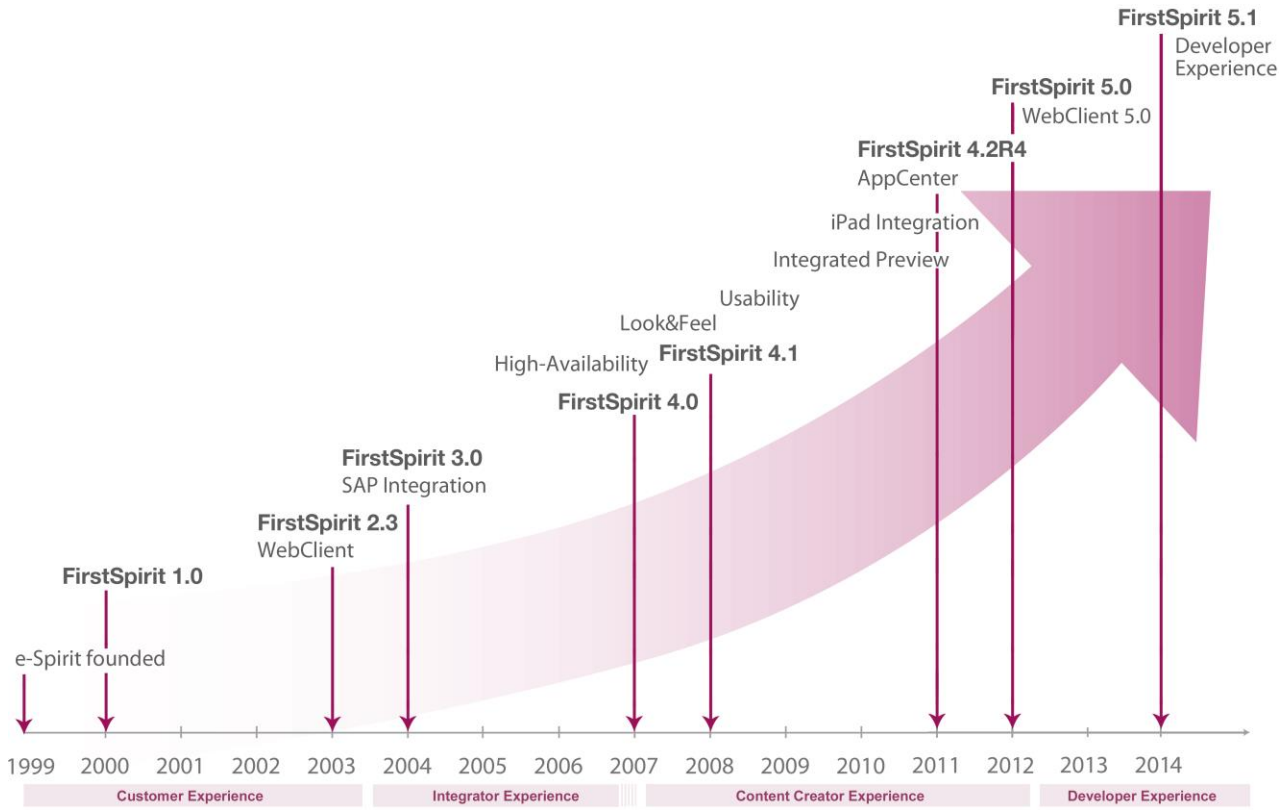
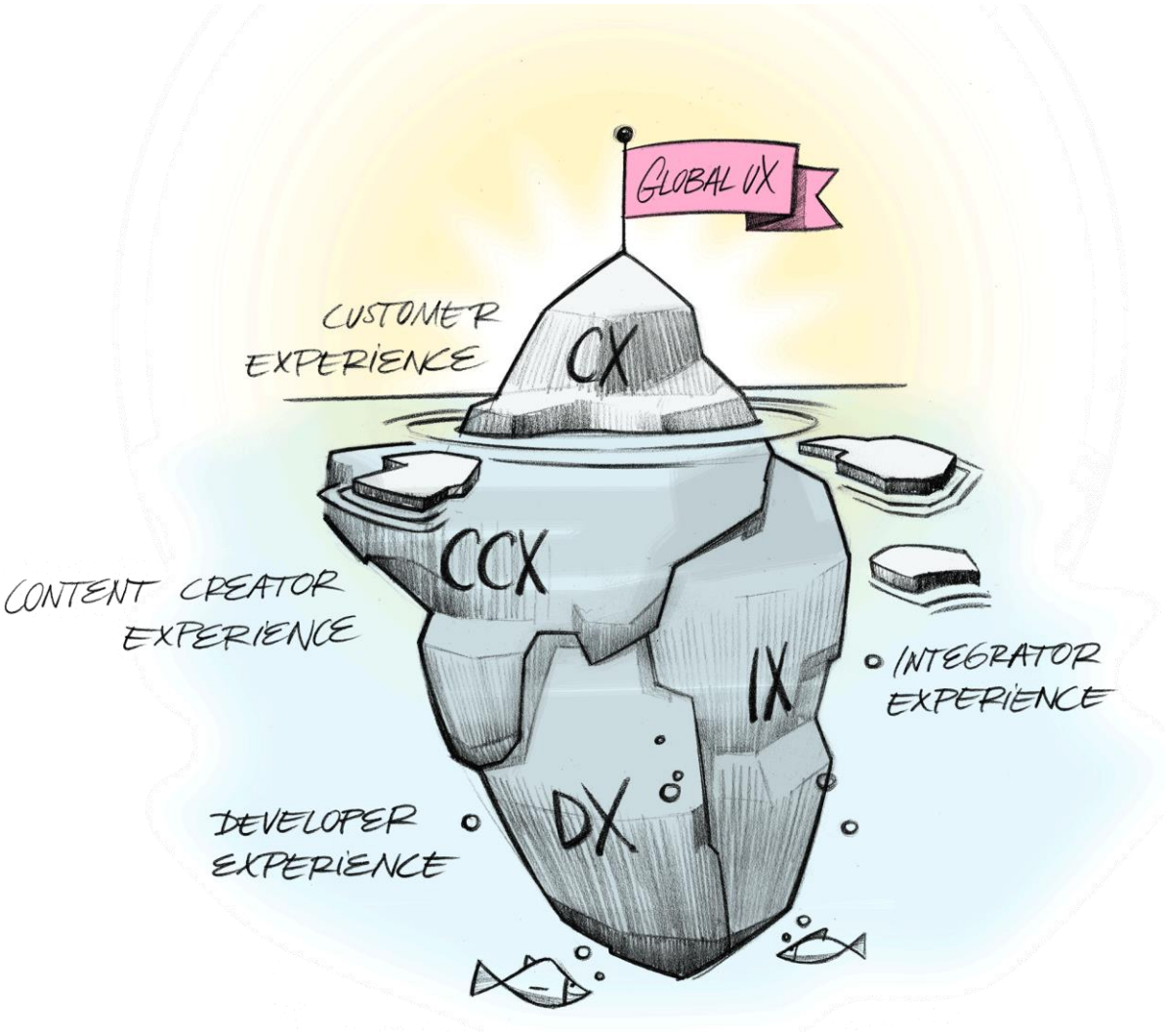


Figure 5-5: FirstSpirit product development from 1999 to 2014

Previous versions have demonstrated that the established term "user experience" is desperately in need of differentiation, since the "experience" for different user roles ("website visitors", "editors" and "developers") can assume completely different characteristics. Ultimately, the customer experience is the product of the content creator, developer, and integrator experiences: If FirstSpirit is not enjoyable to work with and fails to provide the right experience at this early stage (back end), the result for website users at the front end will likewise be less than optimum. That is why the path that e-Spirit has been treading for several years while developing FirstSpirit has been consistently guided by these role-specific requirements of a CMS. It also helps explain why it is now bringing all these activities together under the overarching term **Global User Experience**.



The differentiation in software development based on the different target groups has also flowed into FirstSpirit version 5.1 in a corresponding adjustment to the nomenclature used in FirstSpirit applications: While **FirstSpirit ContentCreator** is the primary platform for all editing activities (specifically the input and arrangement of text, images and videos, and the definition of structure and navigation; previously known as "WebClient"), activities more related to development are carried out in **FirstSpirit SiteArchitect** (specifically the development of templates, workflows and schedules; previously known as "JavaClient"). The configuration of servers and projects (server and project properties) is handled in **FirstSpirit ServerManager**, and monitoring operations continues to be handled by **FirstSpirit ServerMonitoring** (also refer to section 6.10, particularly Figure 6-1 for more information).

In the following FirstSpirit versions, the functions provided by the individual FirstSpirit applications will focus even more on this target group differentiation so that some

functions may be removed from a particular FirstSpirit application or functions may be moved from one application to another.

Additional detailed information on the "Global User Experience" can be found in "FirstSpirit Whitepaper 2014".

5.2 The role of application integration in FirstSpirit

As part of the strategic further development of the FirstSpirit product, "application integration" has taken on such an essential role that an extensive subsection has been dedicated to its placement on the roadmap.

5.2.1 Classification

FirstSpirit was designed and developed as an integration platform from the very start. The development strategy has always consisted of investing as few resources as possible into the development of software features that have already been successfully implemented in other software products and instead integrating existing product solutions as seamlessly as possible from the leading products on the market (see section 5).

On the server side, this development is for the most part complete. The key back end systems for databases and portal/application servers have been successfully connected. Now there is a universal infrastructure for integrating special customer-specific systems.

Since version 4.2, FirstSpirit has been focused on a new aspect of integration that had not yet appeared in the market in this form: the seamless, client-side integration of application software and web applications of leading providers (see "Best of breed" strategy).

What does this mean exactly? FirstSpirit is an enterprise content management system that places a very high value on providing a comfortable and convenient editing environment. The editor should be optimally supported in his work. This also means that the editor should not have to switch as frequently between different program packages and can instead take care of daily editing work primarily using FirstSpirit Client.

The editing work involves developing and entering content. In addition to completely redeveloped content, information that is already available and used just as frequently within the company can be used and modified. The content may be text by nature and

may come from office suite documents or e-mail, for instance, but now it is starting to come from online services with more frequency. Moreover, editing work also includes the selection of suitable images, cropping or other image editing steps, and finally control over whether the prepared information also appears correctly in the different presentation formats (e.g. HTML, PDF or for mobile end devices).

In conventional content management systems, different applications, such as office suites and image editing programs for editing as well as web browsers for checking the work, are required for the various formats, and the editor must manually switch between them.

FirstSpirit application integration reduces the need to switch between the individual applications by using FirstSpirit JavaClient (as of FirstSpirit Version 5.1: "SiteArchitect"; also refer to section 5.1, starting on page 18), since the applications are fully integrated seamlessly in the client.

FirstSpirit version 4.2 already fully integrated the Mozilla, Firefox and Microsoft Internet Explorer browsers, into which the preview area was added in JavaClient (SiteArchitect). The preview area displays a native (non-Java-based technology) instance of the web browser. The integration of the web browser as a preview application is not only completely seamless, but also linked very closely to the program, which makes it possible to control JavaClient (SiteArchitect) in the web browser (and vice versa) by navigating interactively within the preview. This ensures that the user has fully intuitive control over content within FirstSpirit and is able to change it directly as needed. Even the developer is supported in his work, since he can check the results of his HTML programming directly in the client.

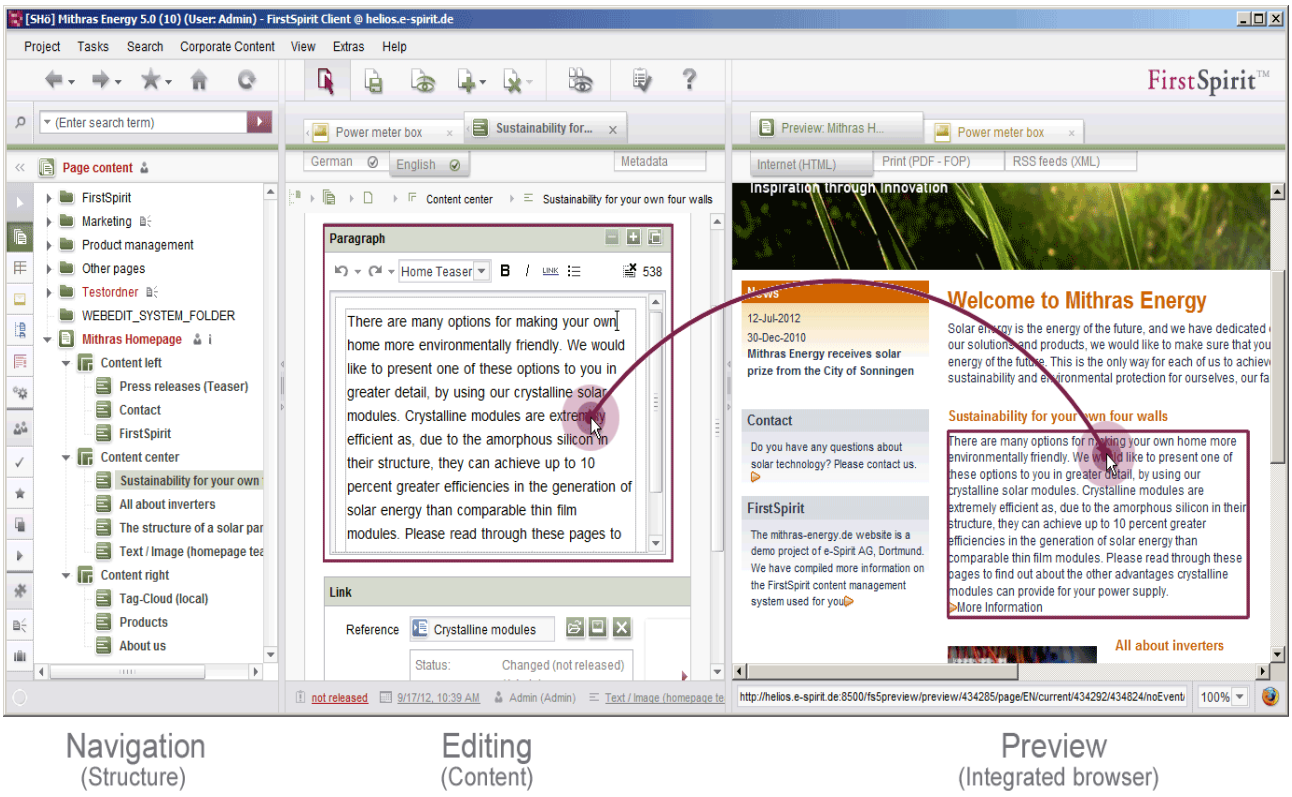


Figure 5-6: Interactive navigation in JavaClient (Site Architect)

After successfully integrating web browsers, office suite products have now been incorporated as well (FirstSpirit OfficeIntegration"). The integration has been carried out with two things in mind:

1) Making the integration as seamless and tight-knit as possible: the user should not notice that he is not working in FirstSpirit.

2) Best of breed integration: market-leading products have been integrated while making sure that FirstSpirit remains a platform-neutral system that also supports Linux and Mac OS.

Many areas have clear market leaders: Microsoft Office, for instance, used to be the leader in the office suite application market. But there are also competitors, such as the free OpenOffice, which is rising in popularity. As for "FirstSpirit OfficeIntegration", the Microsoft Office and Oracle OpenOffice products were therefore also integrated (similarly to the web browser integration). The integration of OpenOffice even allows it to run under Linux (however, there are currently still limitations to this).

Even in this case the integration is native, which means it is based on the program packages installed (and configured) on the workstation system. Microsoft Office products like Word, Excel and Powerpoint (and their counterparts in the OpenOffice environment) have become such an integral part of FirstSpirit that under ideal circumstances the editor does not even notice that he is using a Microsoft program to edit a Word document.

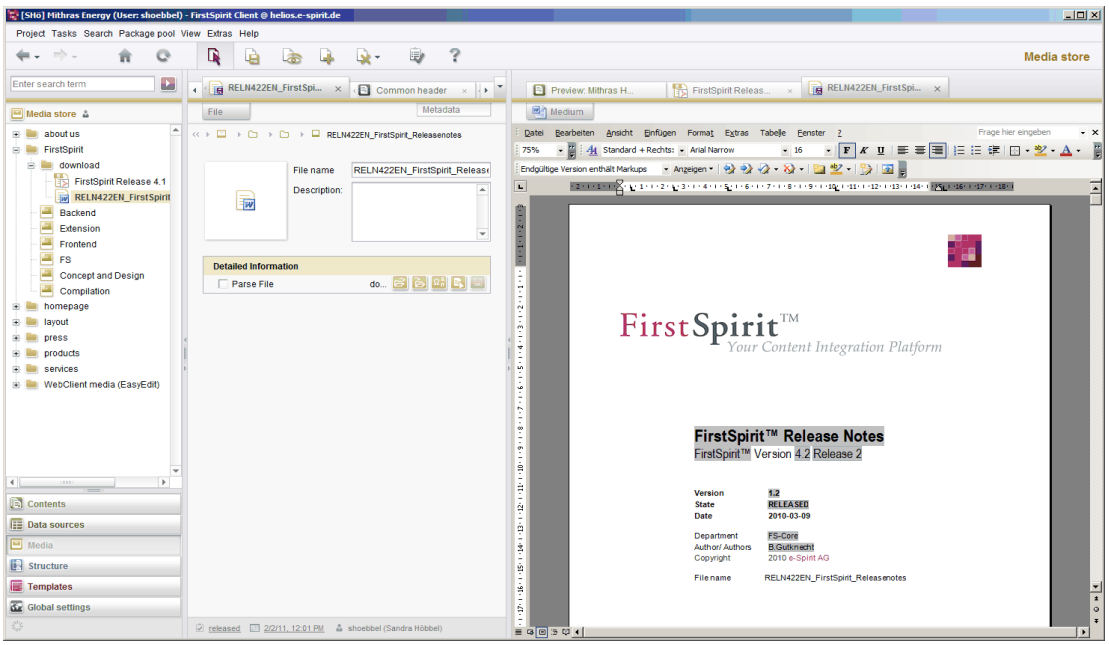


Figure 5-7: FirstSpirit Office Integration (in this case: Microsoft Word)

Here, the FirstSpirit strategy is to give the customer the freedom to choose (even if the choice isn't completely free). Specifically, this means that not just one solution will be integrated, but an infrastructure has been developed that can generally be used to integrate a particular application class (office suite applications in this case) in FirstSpirit. In addition to desktop applications, web applications (online services such as image editing, online video databases, etc.) can also be integrated in the software.

At the technical level, an application integration framework was first developed to integrate Microsoft Office applications. To validate the general applicability of the framework, a second application in the same application category (OpenOffice in this case) was immediately added as proof of concept for the integration framework once Microsoft integration was completed. The upshot is that there are now two integrated solutions from which the customer is free to choose and at the same time (not visible to the customer) a valid framework that facilitates the ability to integrate a potential third solution using the framework at a (hopefully) reduced cost.

Overview of applications for the editor's workstation:

Application		Function
Type	Product	
Web browser	Mozilla Firefox	HTML content preview
Web browser	Microsoft Internet Explorer	HTML content preview
Office application	Microsoft Office and Oracle Open Office	Displaying and editing Word, Excel and Powerpoint documents
Multimedia	Windows Media Player	Audio and video file playback
Image editing	Google Picmonkey	Basic image editing
Image editing	Autodesk Pixlr	Semi-professional image editing
Geolocation	Google Maps, Google Earth	Visualizing and searching for geolocations
Image bank	Fotolia, Pixelio	Searching for and buying professional images
Translation	Google Translator Toolkit	Semi-professional translation solution with translation memory

The application integration concept was taken a step further in FirstSpirit version 5.0 with the implementation of WebClient (ContentCreator) to allow for custom enhancements and adjustments there as well.

Application integration in FirstSpirit version 5.1 and higher will concentrate on the improvement of development processes that are performed when carrying out complex FirstSpirit projects. The focus on the development of new functions will therefore shift from the FirstSpirit editor to the FirstSpirit developer (see section 6, page 32).

5.2.2 Potential

Different web browsers (Mozilla Firefox and Microsoft Internet Explorer), office suites (OpenOffice, Microsoft Office, Google Docs) and online image editing applications based on Flash applications are currently available to the end user in JavaClient (SiteArchitect).

FirstSpirit therefore already offers editors a high degree of extra value when it comes to daily tasks. To understand the actual potential of application integration, you have to see with your own eyes that these functions are only some of the use cases for applications capable of being integrated which have made e-Spirit particularly attractive and thus

have been implemented as part of the core product.

In addition, an interface has been created in the background (not visible to the end user) that has made it possible for partners to develop fully self-contained individual application integrations for FirstSpirit. Access is available to an application integration API, which is the culmination of practical experiences related to integrating the core product and thus has already been proven in practice.

In connection with the expansion interface (FirstSpirit modules) already long established in FirstSpirit, product enhancements can be developed that are seamlessly incorporated the FirstSpirit application integration infrastructure (see Figure 5-1).

Additional detailed information on the technical aspects of application integration plans are also available in the "FirstSpirit Whitepaper 2014".

The following example is provided for better clarification:
In practically all areas related to corporate communication, the use of high-quality images has become standard, which frequently now involves the use of professional image banks (such as Getty Images or iStockphoto) that offer attractive research functions (e.g. based on geolocation data) and also support online payment functions.

If on the occasion of an event (such as a customer presentation) an online announcement is developed with the professional image of an adjacent landmark attached (see Figure 5-10), the first step is to find the geolocation of the event. This can be done directly in FirstSpirit JavaClient (Site Architect) using an input component implemented by the customer for this purpose:

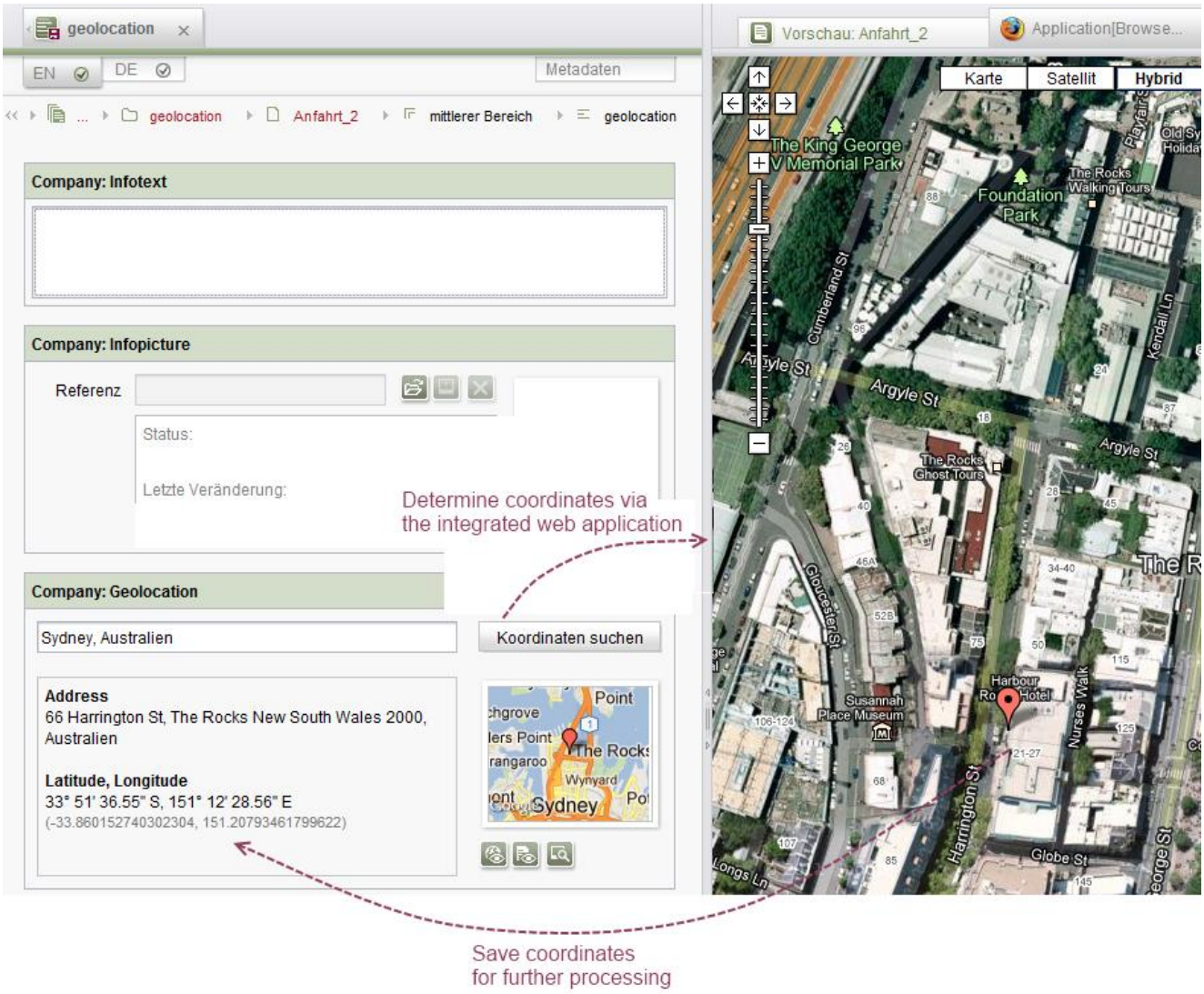


Figure 5-8: Finding a geolocation using an integrated web application

The component includes a button used to display or modify the exact geographic location and a button is used to initiate a search based on the address string. For both actions, the integrated web application Google Maps is opened in the application area of JavaClient (Site Architect) ("seamless application integration"). The geographic location data determined using the web application is saved for further processing (see Figure 5-8)[1].

[1] For more information about this example, see the FirstSpirit AppCenter documentation.

This information can be used for (online) route planning with Google Maps (see Figure 5-10), but can also be used as a point of entry into the geodata-based search for an online stock photo agency. The input component includes another button for this purpose, which opens the integrated web application Pixelio in the JavaClient (Site Architect) application area:

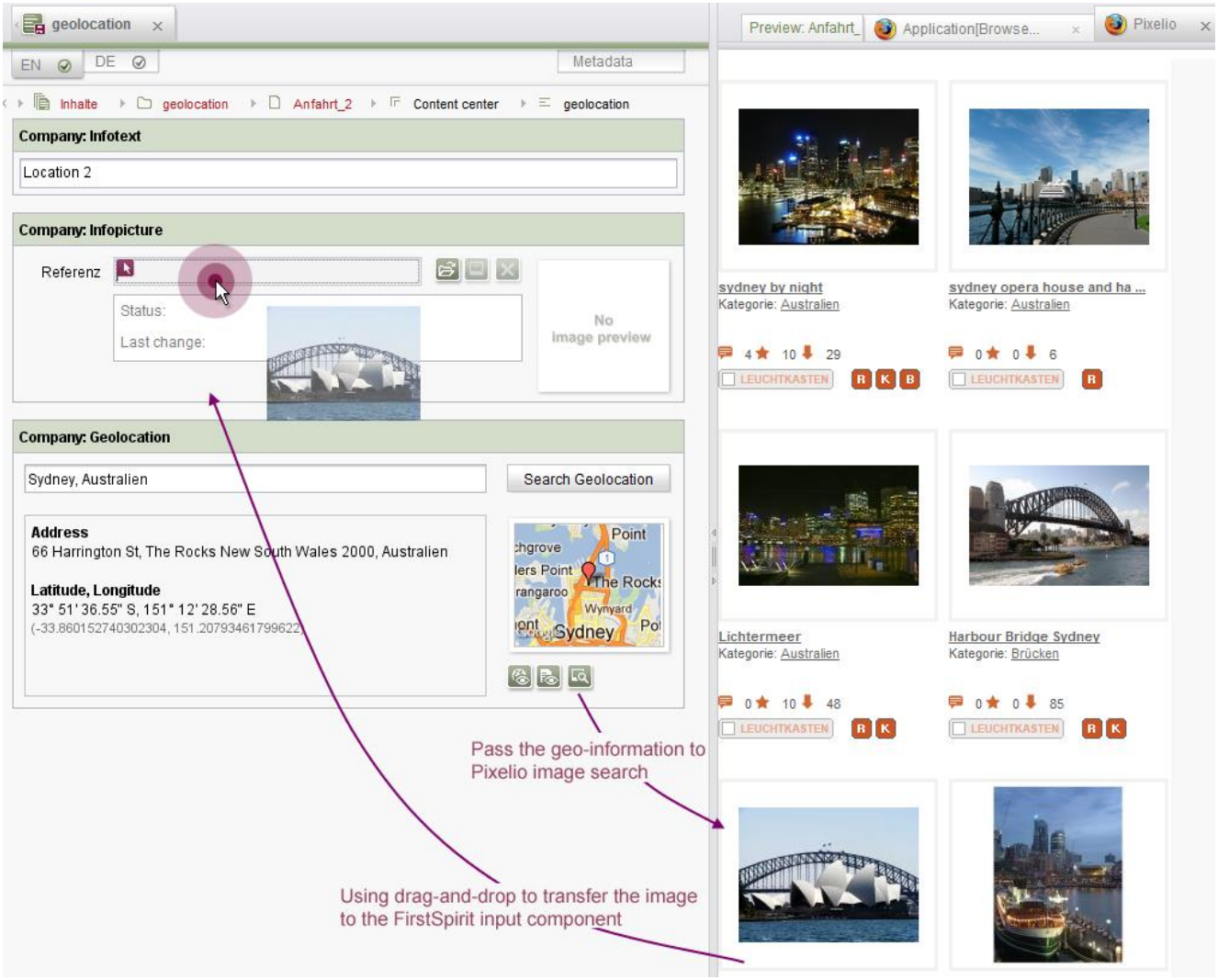


Figure 5-9: Opening Pixelio image search in FirstSpirit JavaClient (Site Architect)

The user can select the appropriate image from images related to the event location and use drag-and-drop to copy it directly from the online image bank to FirstSpirit. The underlying technical processes are hidden from the editor (handover of drag-and-drop processes from the web browser to JavaClient (SiteArchitect), checking for duplicates in the Media Store, carrying out the payment process and downloading the high-resolution

images including applying metadata).

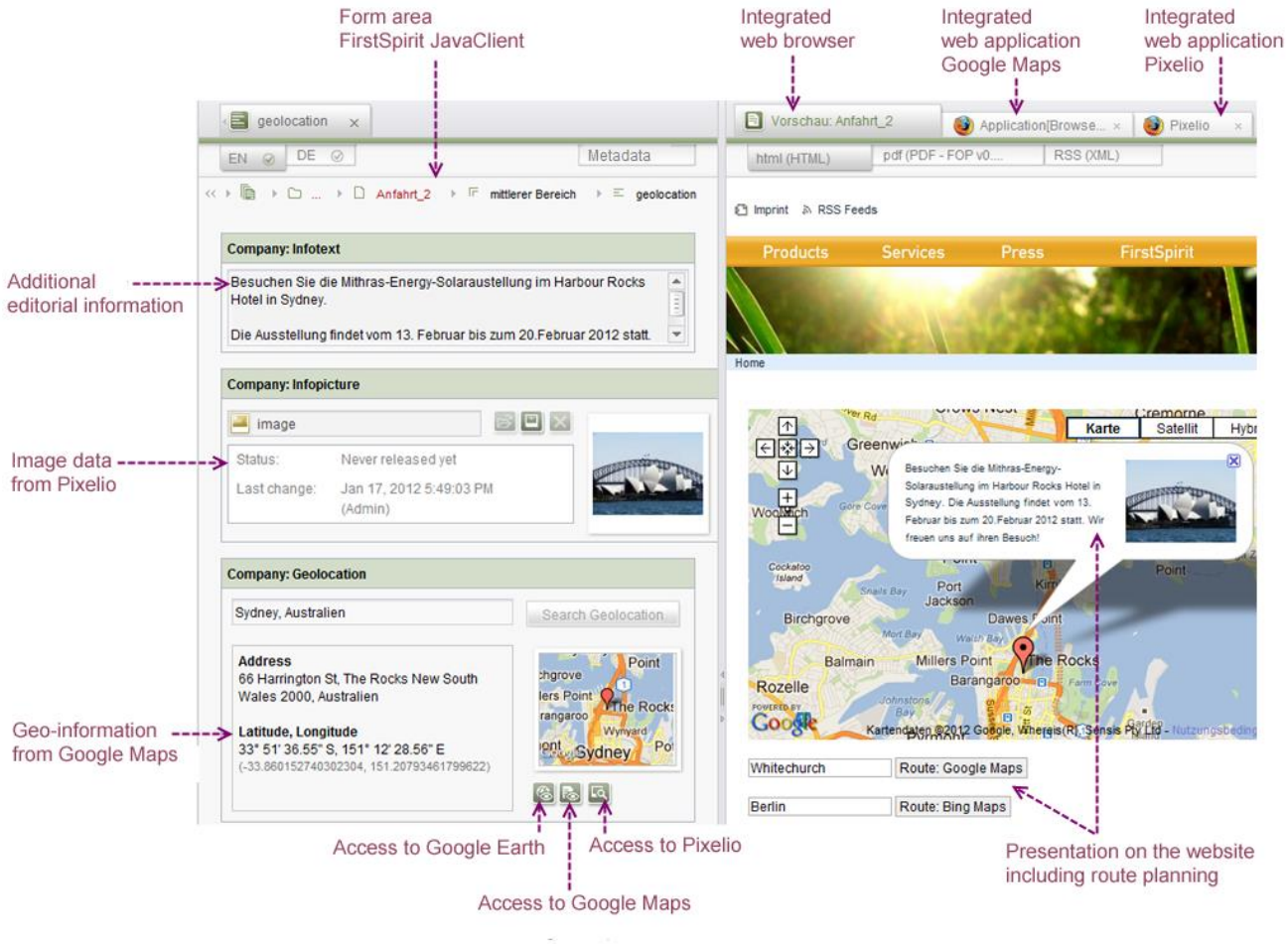


Figure 5-10: Result of work steps

The actual potential of application integration therefore lies in the FirstSpirit development partners in particular who are given a powerful tool through the application integration API for developing specialized integration solutions in their specific areas of expertise.

An equivalent integration solution is also possible in WebClient (ContentCreator).

The following sections describe the functional enhancements presently planned for FirstSpirit versions 5.1 and 5.2.

6 FirstSpirit Version 5.1 (planned release: 1st quarter of 2014)

As already described in the section covering our strategy, FirstSpirit version 5.1 development is primarily focused on improved **support for development processes** that take place when carrying out complex FirstSpirit projects (see "developer experience", section 6.3, starting on page 38). But numerous enhancements and improvements are being made even with regard to the user experience.

6.1 FirstSpirit application nomenclature

The renaming of FirstSpirit applications is the result of the Global Experience ideas presented in section 5.1, starting on page 18.

The following terms are now being used:

- FirstSpirit ContentCreator
- FirstSpirit SiteArchitect
- FirstSpirit ServerManager
- FirstSpirit ServerMonitoring

From the point of view of usability, **ContentCreator** offers an optimized editing environment for efficient content management and so is aimed directly at editors. The relevant button, which can be used to start the individual FirstSpirit applications, is located in a prominent position on the FirstSpirit start page. This emphasizes, in a visual way, the importance of this application for the primary FirstSpirit user group (= editors).

SiteArchitect is generally used by a smaller number of users. It has been designed for the configuration of projects and the development of templates and so is primarily aimed at template and project developers. However, it can also be used to carry out more complex editing tasks ("power user").

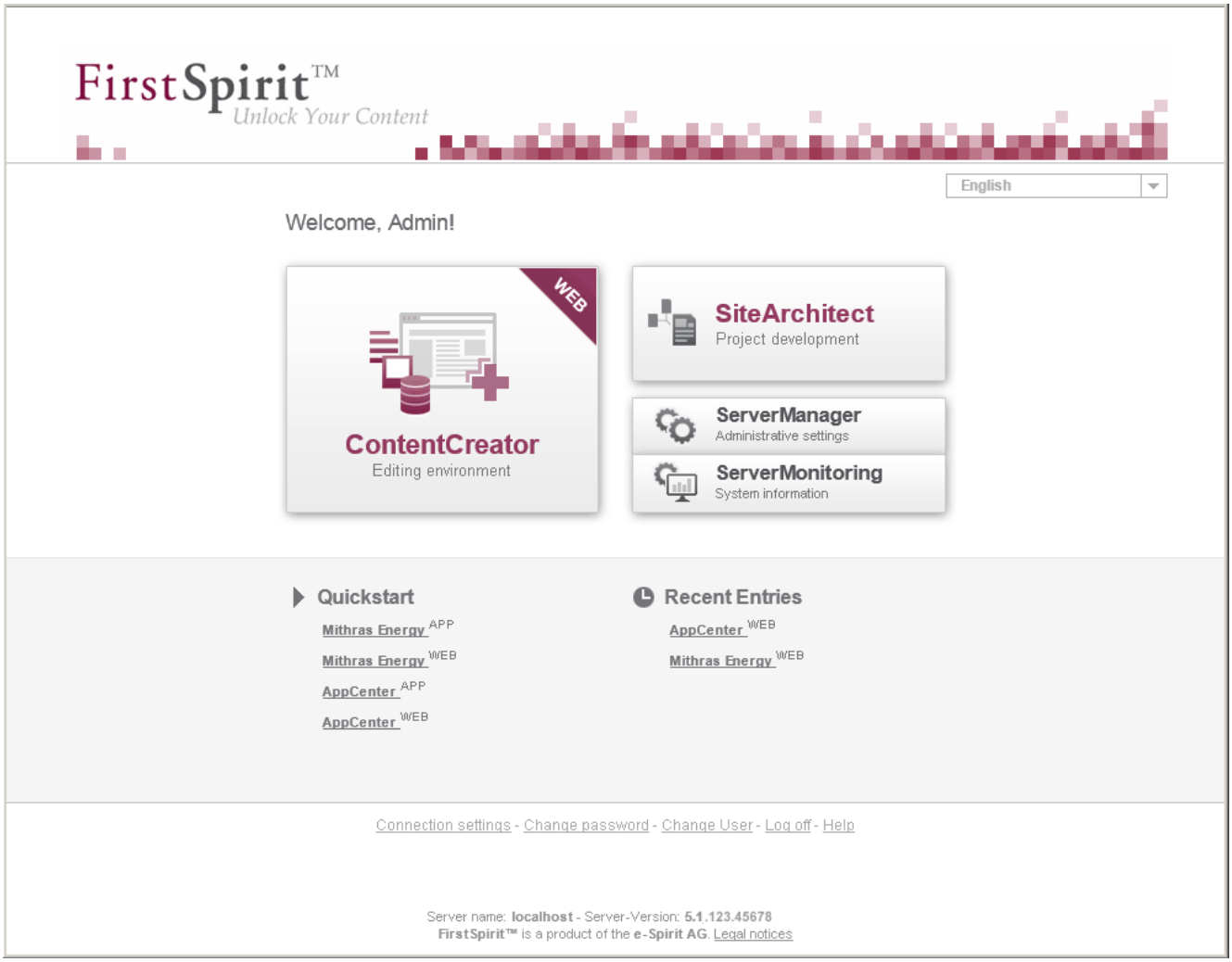


Figure 6-1: FirstSpirit 5.1 start page with new nomenclature

6.2 Multi-perspective preview: content in all dimensions

The ubiquity of Internet-capable smartphones and tablets has caused many companies to change their thinking in recent years. To ensure the successful implementation of web projects, it is no longer enough to have high-quality content that is precisely tailored to the respective target group; rather, it is becoming increasingly important to ensure that the content is properly formatted for a variety of output devices with different resolutions and display sizes. In light of this, FirstSpirit is putting its faith in sustainable website concepts such as "responsive design" and "mobile first" so that the quality of all web content remains consistent in the long term. "Responsive web design" is a technical design approach for creating sites with the ability to adapt. The graphical structure of a responsive web page is highly flexible and it takes account of the requirements of the device used to access the page. The web page sends what are known as "media queries" to the device that is being used and automatically loads the appropriate layout for the display size and resolution concerned ("fluid grid") together with appropriately adapted type and images ("fluid images"). The result: A web page that can adapt itself perfectly and automatically to any mobile end device.

The contemporary understanding of the "mobile first" approach is that a web page geared toward mobile requirements is no longer just a welcome side effect, but is the primary format that must guide the process of designing a website: A web page is designed first and foremost for mobile output devices where there is not much room for text and navigation elements due to the small screen size. Small displays and the resulting lack of space force the author to eliminate all but the most important content. Only once this has been done are additional layouts and items of content gradually added for larger displays. Thus, "mobile first" is not just a sensible design concept, but also a strategic approach that disciplines companies to place their key statements and content in the foreground.

First of all, FirstSpirit meets the most stringent demands of responsive web design (front end) and second of all it helps editors to implement and maintain the "responsive web design" concept from an editor's perspective (back end). Consequently, version 5.1 offers a convenient way of checking display and navigation in the ContentCreator preview with a variety of display sizes. It also allows content, layouts, and images to be perfectly adapted to suit the output device concerned. The aim here is to minimize the amount of effort involved in developing responsive layouts and maintaining the pages implemented on this basis. In version 5.1, ContentCreator will still be at the heart of the optimization work for the "responsive web design" approach; in version 5.2 these optimizations are to involve SiteArchitect as well (see section 7.3 page 64).

The relevant views ("viewports") can be predefined in the project and fully tailored to the needs of editors. As far as they are concerned, it is not of primary importance that the simulated view of the devices is as close to reality as possible. They are more concerned about whether the selection of desired perspectives (e.g. Desktop (large)", "Desktop (small)", "Tablet", "Mobile Phone (large)", "Mobile Phone (small)") is easy and intuitive. The display changes when the editor switches from the normal view to the selected view. The editor then sees the current page (or a subsection of the page) in a simulated view that matches the required display size.

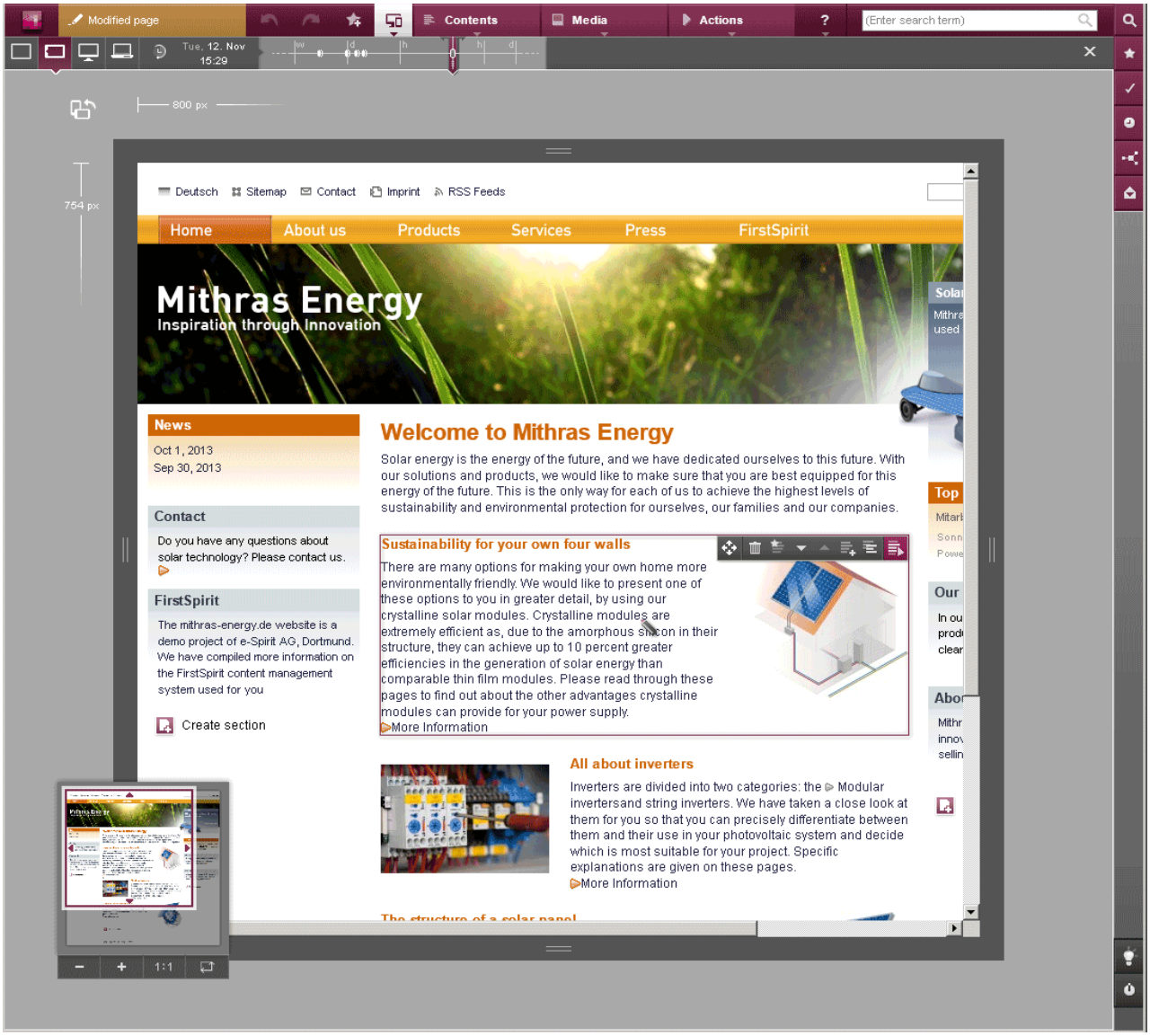


Figure 6-2: Mobile content preview (ContentCreator)

In this view, editors can manipulate page elements in the usual way (e.g. edit images and individual sections, show or hide sections, or define resolution-specific settings for images). This allows them to adapt the content directly to suit the modified display size.

However, the Multi-perspective preview concept goes even further than this: In addition to the display size of the various output devices, the editor may also be interested in the chronological development of a page or (for example) what the page looks like to user-specific roles.

Chronological perspective: To cater specifically for scenarios that involve controlling marketing campaigns or the activation of time-limited offers in the online shop, FirstSpirit 5.1 has a convenient preview feature that shows what each page will look like at a given time. Likewise, the past development of the page can also be tracked. For this purpose, a time bar function has been implemented. The page's development over time is displayed continuously in the preview and you can use a slider to select the required point in time. Releases and other events are marked on the time bar and serve as navigation and orientation points. Figure 6-3 shows (by way of an example) how a teaser (3) is to be displayed at a particular point in the future:

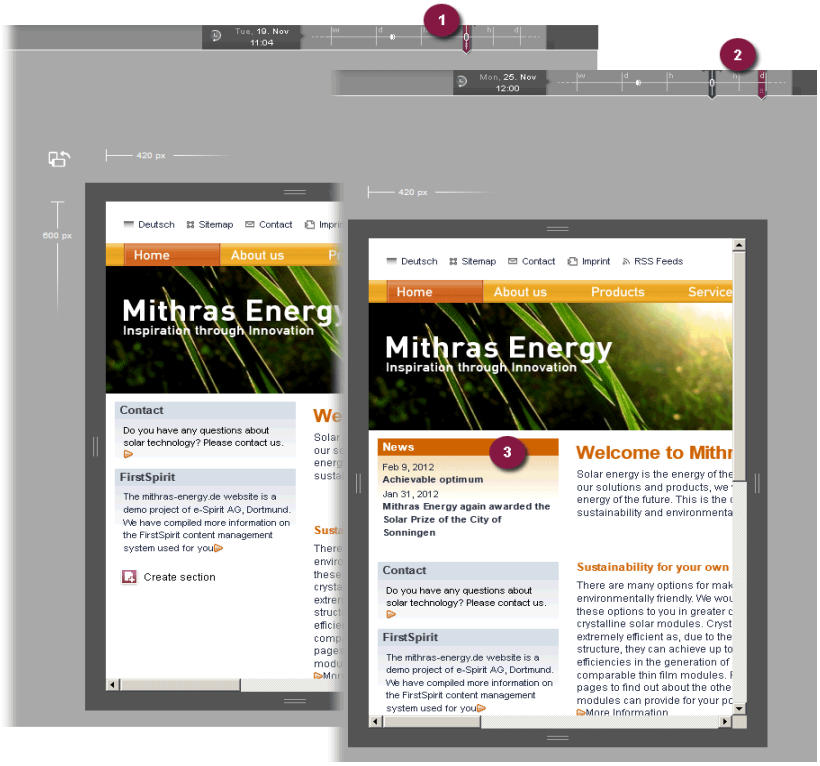


Figure 6-3: Time-dependent change to the content of a project

Project-specific perspectives: Other aspects can be simulated in the preview for each project, meaning that the views can be geared even more closely to the needs of the editor and the project. Content can, for example, be easily displayed and checked from the perspective of different visitor and user roles (including in connection with the "FirstSpirit DynamicPersonalization" module). The desired user or group can be selected simply by clicking.

Figure 6-4: Project-specific configuration

The page content is then displayed according to what it would look like to a private customer, partner, or business customer. In addition, different variants of a page can be placed side by side (e.g. for A/B testing) or extra information can be shown on the page (e.g. from external sources such as Google Analytics, etracker, etc.). There is virtually no end to the configuration options. All the perspectives can be combined with each other as well, e.g. the web page can be viewed based on what it would look to someone with the "male visitor" role accessing it in two weeks' time on a tablet.

For more information about the "Multi-perspective preview" refer also to "FirstSpirit Whitepaper 2014".

6.3 Project implementation support: visual debugging

The development process cycle in FirstSpirit (as well as in many other development environments) consists of "changing, testing and correcting", which only comes to a (preliminary) end after a whole series of iterations with the introduction of a new development status in the version control system. Frequently, several developers are employed to work to some extent at different locations, particularly in the case of large projects and large companies.

A series of enhancements were made in version 5.1 to support FirstSpirit developers the best way possible in implementing their projects quickly and cost-effectively.

6.3.1 Know where it's coming from: the Template Inspector

The first steps in supporting template developers in developing HTML output in Site Architect have already been taken, including the highlighting of code text, code completion and validation. The new "Template Inspection" function was introduced in version 5.1 to provide better orientation in existing projects and to retrieve existing code faster. Thus, Content Highlighting for editors ("Where can I change the content displayed on my web page?") is the equivalent to the "Template Inspection" function for template developers: "Where can I change the template that generates the content displayed on the web page?" This allows the user to display the tag structure of the HTML code for the current page within the integrated preview. From here, the associated FirstSpirit templates can be opened in the workspace:

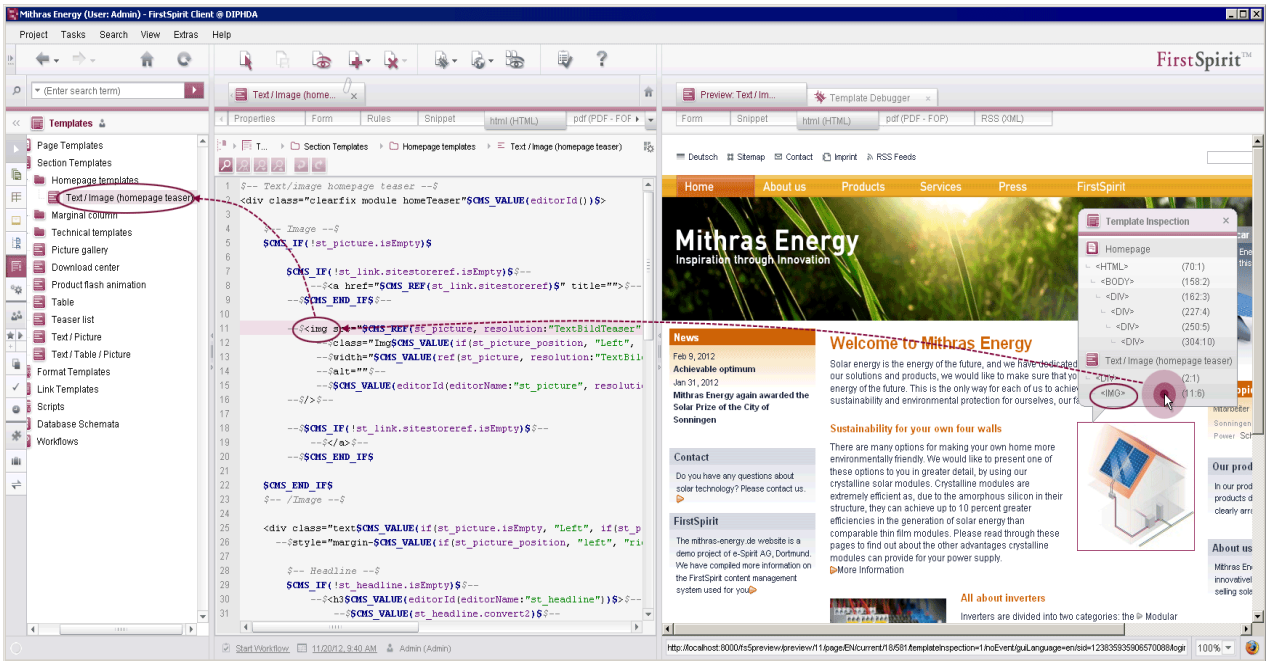


Figure 6-5: Template Inspector

The affected lines are also highlighted directly (Template Highlighting). This allows for faster retrieval and editing of locations of code in the HTML channel.

6.3.2 Troubleshooting and template development using the Template Debugger

The Template Debugger not only provides software-assisted troubleshooting in HTML code, but also assists the developer in the (further) development of templates. Using the debugger, the execution paths of the template generation can be completed in detail based on the particular HTML page currently displayed or generated in the preview. Based on the underlying page template, all dependent templates (e.g. section templates of sections from the current page, link templates, format templates, etc.) are also included.

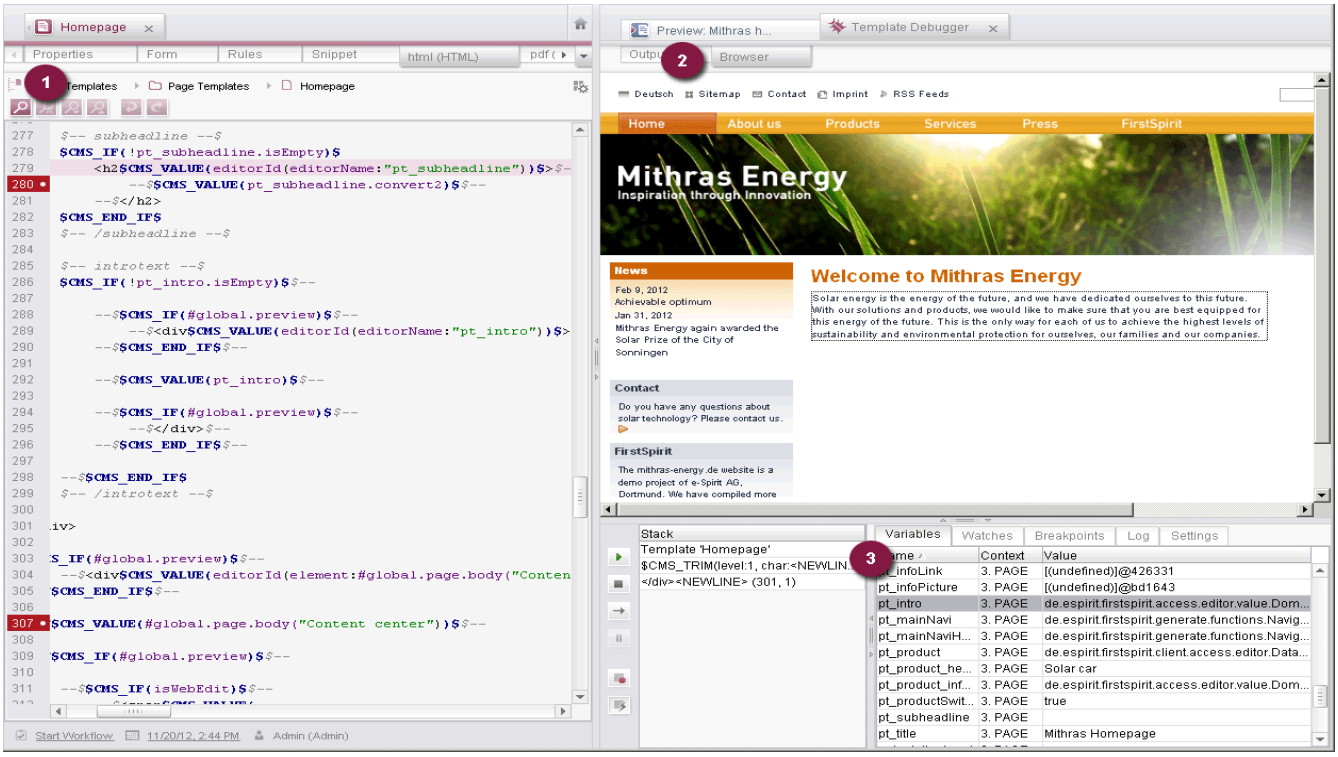


Figure 6-6: Template Debugger

In the process, a single step corresponds to an instruction in the software code. The HTML page is constructed further with each step, and the developer can then check the (partial) result either as source text or as generated HTML ((2) in Figure 6-6). Clicking in the generated output then takes the developer straight to the corresponding code location in the HTML channel of the affected template ((1) in Figure 6-6). This procedure presents valuable insight into the inner workings of FirstSpirit syntax and the interaction between instructions, functions, variables and their output in FirstSpirit—particularly in the case of users who are new to FirstSpirit template development.

As is familiar from other debuggers, breakpoints are also defined in FirstSpirit template code. The debugger stops at these points so that the code being run through can be analyzed, thus making it possible to isolate code for troubleshooting. The integrated log output also makes troubleshooting and error analysis easier without the need to open detailed external log files first.

A key aspect of FirstSpirit template development is the use of variables: it is primarily variables that facilitate intelligent reuse as well as making content dynamic. Particularly in large projects in which many variables are used and on which many developers may be working, it is quickly clear which variables exist, which apply in the current context,

which value they actually have and where they were initialized in the project. With the Template Debugger, each developer immediately has all of the information on variables for the currently generated page or the page displayed in the preview with the current value at his disposal ((3) in Figure 6-6) and thereby has easier access to existing variables and the ability to edit them. The current value of a variable can also be checked directly on the HTML tab where the value is displayed via the variable's tool tip. In addition, the specific methods related to the variables can be shown (including brief documentation) that can be used with the variable. The Template Debugger also works without limitation for other output media such as PDFs, etc.

Thanks to the Template Inspector and Template Debugger, not only is it possible to create forms with just a few mouse clicks, but the most complex and extensive HTML output in the FirstSpirit Template Store can be create this way as well. In addition, it provides the developer intuitive navigation through the project and offers a convenient entry into a sound analysis of technical relationships within the project.

6.4 Collaborative development process

Assembly and deployment on a FirstSpirit server is key in the development of FirstSpirit templates, modules and application integrations. To check the results, usually an instance of SiteArchitect also has to be started. This expensive process can be reduced significantly by providing a development environment of FirstSpirit components in which all of the necessary components of both the FirstSpirit server and SiteArchitect (including specific test functions) are brought together. The goal is that the developer can simply call up a build function in the trusted IDE directly (without a restart) and test the current results in the FirstSpirit execution environment.

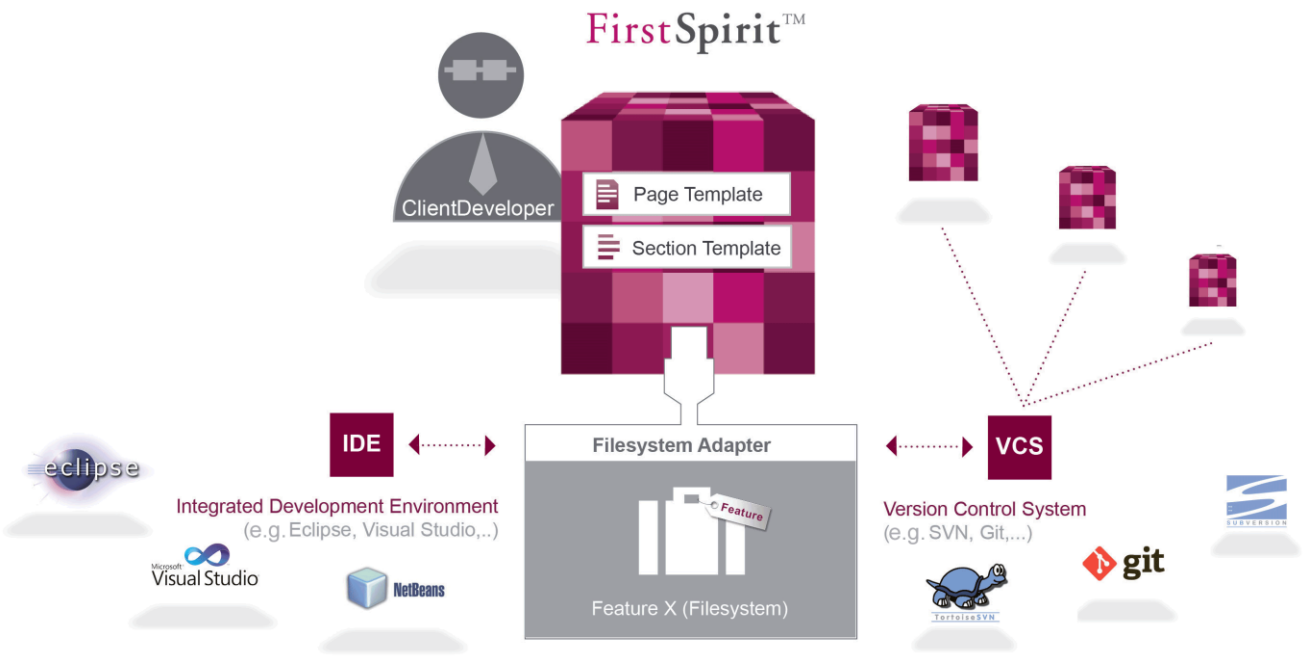


Figure 6-7: FirstSpirit file system adapter

In FirstSpirit version 5.1, the first steps have been taken to this effect, e.g. substructures from FirstSpirit (templates, store subtrees, etc.) can be exported to a file system structure with a folder hierarchy – in a format that is "legible to humans" and "can be interpreted by IDEs":

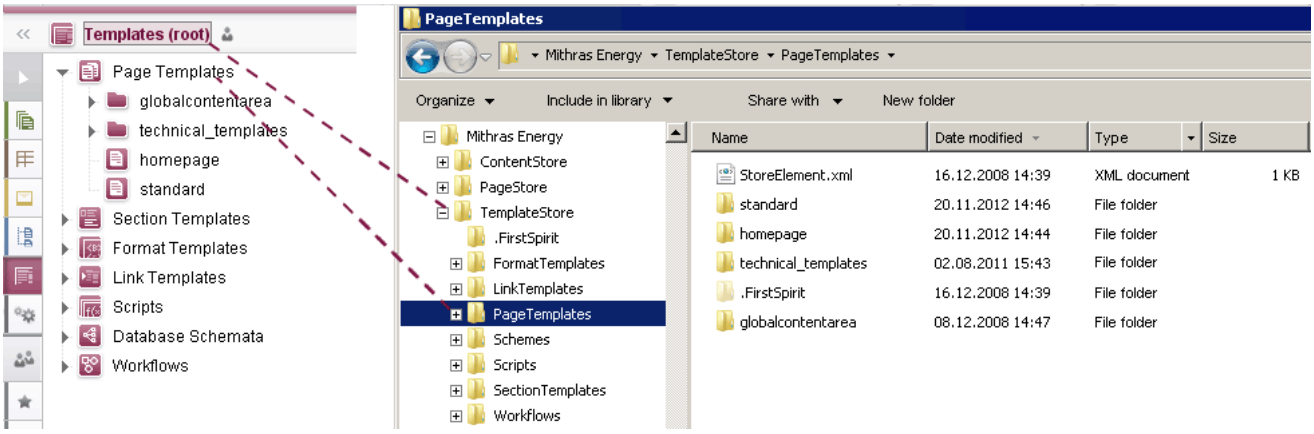


Figure 6-8: Exporting from the Template Store

There (that is, outside of FirstSpirit), the exported files can be processed, which means that they can be saved using a version control system, for instance, and then resynchronized with FirstSpirit. The aim is to synchronize changes in the file system with changes in the FirstSpirit project using a process that is largely automatic. As a result,

any changes made to a template in the IDE are immediately reflected in the local (!) FirstSpirit project belonging to the developer.

This feature allows comparisons with older template versions and means that versions can be restored; it also allows the parallel development of templates by multiple developers in a team.

In principle, the data format used for this special export function when FirstSpirit Version 5.1 is initially released is to be retained in subsequent (minor or release) versions. However, given the highly complex structure involved, the first step is to gather empirical data regarding this new function from real projects (including from customers and partners). On the basis of this empirical data, later versions could potentially see a change in the data format and the introduction of minor changes that may be incompatible. Therefore, feedback from customers is expressly requested.

6.5 Application integration enhancements

As already mentioned in section 5.2, starting on page 23, e-Spirit is consistently pursuing a best of breed strategy to integrate third-party software into FirstSpirit. This allows customers to implement ideally suited (offline and online) products with FirstSpirit (office suite and image editing software, media databases, web analysis and search engine optimization tools, translation programs, references, etc.) to meet the requirements of the particular company and the variety of use cases. Additional expenses for developing new software and training in this area can thus be avoided. At the same time, customers do not have to worry about being forced to rely on any particular supplier (no "vendor lock-in") or about any compatibility issues because the ability to swap out individual components is part of the FirstSpirit product strategy and will continue to be developed even further in the future as well.

Despite the completely different basic technologies (Java vs. web application) used in the integration of applications, e-Spirit has succeeded in providing ways to integrate applications into the ContentCreator and SiteArchitect clients. Slightly different integration solutions were implemented for the two clients for technical reasons as well as due to different user prompting and guidance concepts. Newer customers will not notice the difference between the two environments despite the way they were originally developed. For this purpose, a distinction can be made between solutions in which either the files (e.g. from Google Drive, "data integration") or functions (e.g. geolocation tool from Google Maps, "function integration") can be made accessible by third-party

systems.

The following types of integrations are basically available in ContentCreator as well as SiteArchitect in FirstSpirit 5.1 and higher:

1. **Input application (input components)**
   Expansion of the FirstSpirit input forms to include additional form types.
   Example: a Google Maps geolocation component that visually appears to be smoothly integrated into the respective editing environment.
   (Also refer to section 6.6, page 44 for more information.)
2. **AppCenter applications**
   Seamless integration of external applications into the interface of the respective FirstSpirit client. In SiteArchitect, this is through a permanently visible workspace (called the "AppCenter"), and in ContentCreator it is through a temporarily displayed window.
   (Also refer to section 6.6, page 44 and section 6.7, page 47 for more information).
3. **Report application (reports)**
   Displays data from third-party systems in the form of a special list view ("Report") that the user can modify and that is shown in the form of a search results list in the respective editing environment.
   (Also refer to section 6.8, page 50 for more information.)
4. **Drag-and-drop**
   Exchange of any type of file using intuitive drag-and-drop gestures, e.g. between input components and the AppCenter area, but also from one report or from the (Windows) desktop to selected input components.
   (Also refer to section 6.6, page 44 for more information.)

6.6 Application integration in ContentCreator

The basic redevelopment of ContentCreator in FirstSpirit version 5.0 placed a higher value on expansion and adaptation options. This means that as early as version 5.0 the infrastructure had been laid down to allow for the integration of applications in ContentCreator as well. In version 5.1, application integration in ContentCreator has been pushed even further to the fore.

Compared to SiteArchitect, ContentCreator presents a particular technical challenge, since the browser follows significantly stricter security system guidelines with markedly reduced variance than the runtime environment. Separating the individual web applications from each other only allows for black box integration, i.e. the application

must provide suitable interfaces and cannot be integrated through manipulation or by expanding on its HTML code.

The following figure is an illustration of the most important integration options in FirstSpirit ContentCreator:

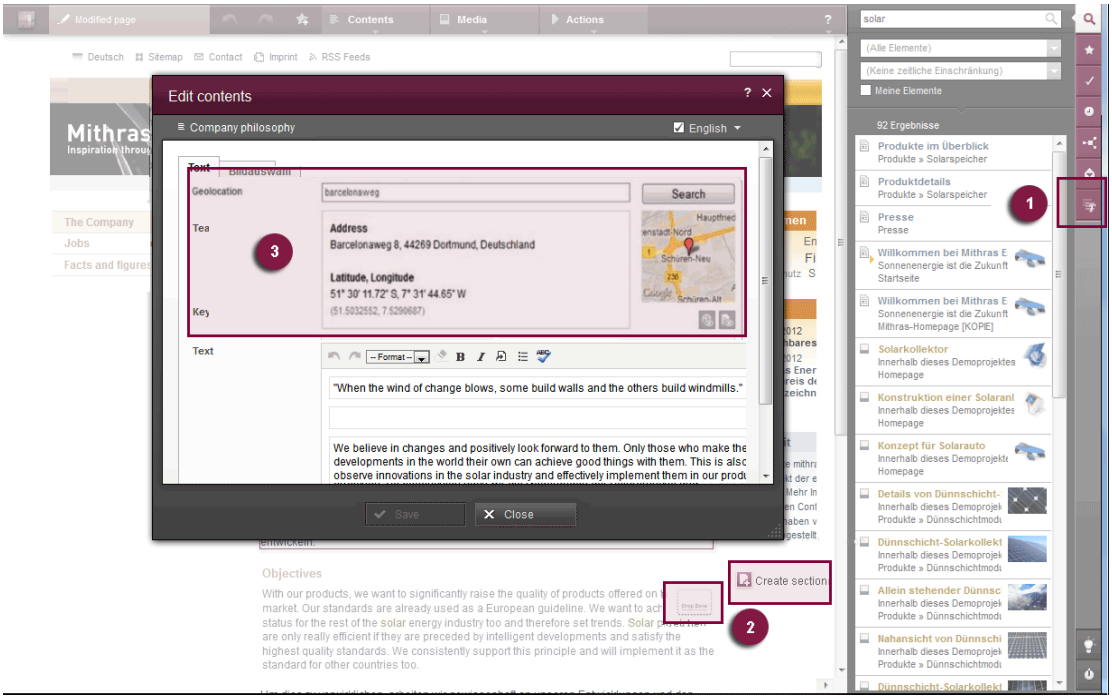


Figure 6-9: Integration options in FirstSpirit ContentCreator

1. **Report application (reports)**
   Reports are integrated in ContentCreator in the intended report area and are primarily used for data integration. Data can be queried from any source, displayed in the reports area and from there conveniently used in the project. In addition to the standard reports provided (search, bookmarks, tasks, project history, etc.), version 5.0 also provides for the ability to use custom reports.

2. **Drag-and-drop**
   Version 5.0 already allows the editor to use drag-and-drop to import files from the desktop by dropping them onto the locations where they will be used. This system is continually being further developed in order to integrate custom reports and thus third-party systems to provide the same level of very simple and intuitive functionality for the editor. The drop zones and drop actions that are to be executed can be defined very flexibly in order to determine, for instance, whether to import or to reference data. For instance, drag-and-drop can be used to drag more complete data from a report (e.g. geo information) to a geolocation input component.

Additional application integration options are available using the ContentCreator JavaScript interface. This interface features functions to control ContentCreator, such as are necessary for automating processes. But the embedded display of external Web content is also supported in order to provide native application integration methods.

3. **Input application (input components)**
   The new component model for developing custom input components is used mainly for functional integration. An example of this is the entry and storage of geo-positions using Maps and Address Search:

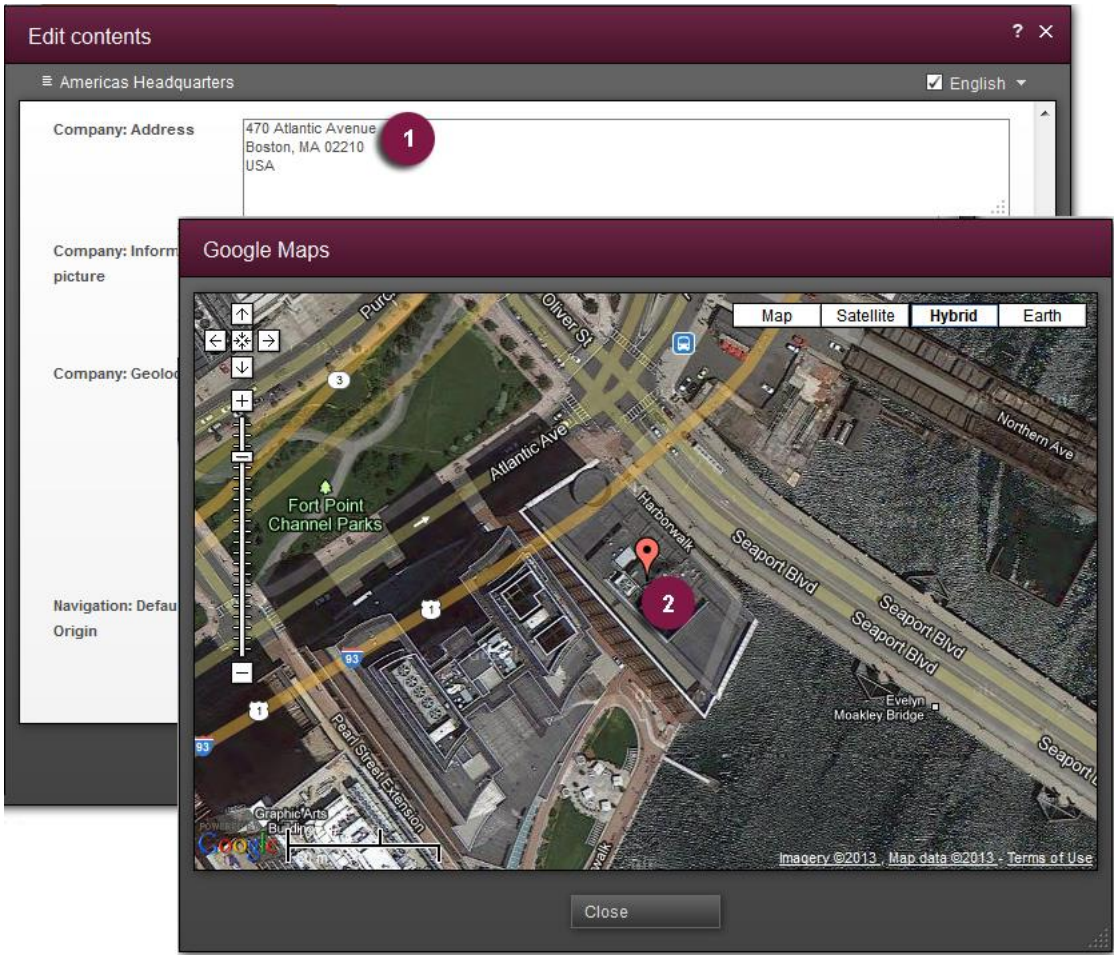


Figure 6-10: Integrated Google Maps application in FirstSpirit ContentCreator

While the actual selection of the geo-position is made using the integrated Google Maps application and the JavaScript API (2, for this purpose it is an **AppCenter application**), the controlling geolocation input component (1) is responsible for the entry and storage of the geographic coordinates as well as for the display of the complete address information (street, city, country) associated with these

coordinates, determination of the actual geographic longitude and latitude, and communication with the ContentCreator form. For convenient, unrestricted exchanges between reports and forms (provided with the software or custom), forms are displayed in non-modal windows so that even drag-and-drop from Reports is also possible in forms.

Communication with Google is handled by the Google Maps API. This API provides the necessary JavaScript tools to display maps and retrieve addresses. The combination of JavaScript API and input component provides for a seamless integration of existing Google Maps functionality in ContentCreator. This means that the editor cannot tell by appearance or operation whether the functionality is standard or whether it is project-specific, integrated functionality.

6.7 Connecting the Cloud – Google integration in FirstSpirit

The cloud services provided by Google fit perfectly in the FirstSpirit best of breed and freedom of choice strategy. The focus particularly on the integration of many individual services within a Google account and on the open architecture, which allows for the expansion and integration of Google services through well-defined interfaces, practically force a connection between FirstSpirit and Google. Google offers a diverse and extensive scope of integration options ranging from Google Search, Google Maps and Earth to Google Office applications, the Chrome web browser, the Google Android mobile operating system and paid content (games, music, videos) available from Google Play.

Earlier FirstSpirit versions demonstrated that the geolocation input components such as Google Maps and Google Earth could already be seamlessly integrated into SiteArchitect, and GPS coordinates, such as those for the geolocation of an editorial article or for the integration of route planning, could be set there easily by selecting a map. With the release of FirstSpirit version 5.1, this is now also possible in ContentCreator.

However, the Google integration options in FirstSpirit go even further. For instance, office suite documents that are stored either in FirstSpirit or in Google Drive can be edited in SiteArchitect as well as ContentCreator thanks to the integration of Google Drive. The use of cloud services to save FirstSpirit documents has made it possible to access these documents online (e.g. mobile) from anywhere without having to be logged on to FirstSpirit. The documents can also be edited in tandem by multiple employees within a team (collaboration).

In a specific implementation scenario, a document can be transferred to Google Drive from FirstSpirit. The editing process is thus started seamlessly without the user having to perform any manual steps in advance, such as logging on to a Google account. All functions offered by Google Docs can then be used for editing. After editing is finished, the changed document is returned to FirstSpirit:

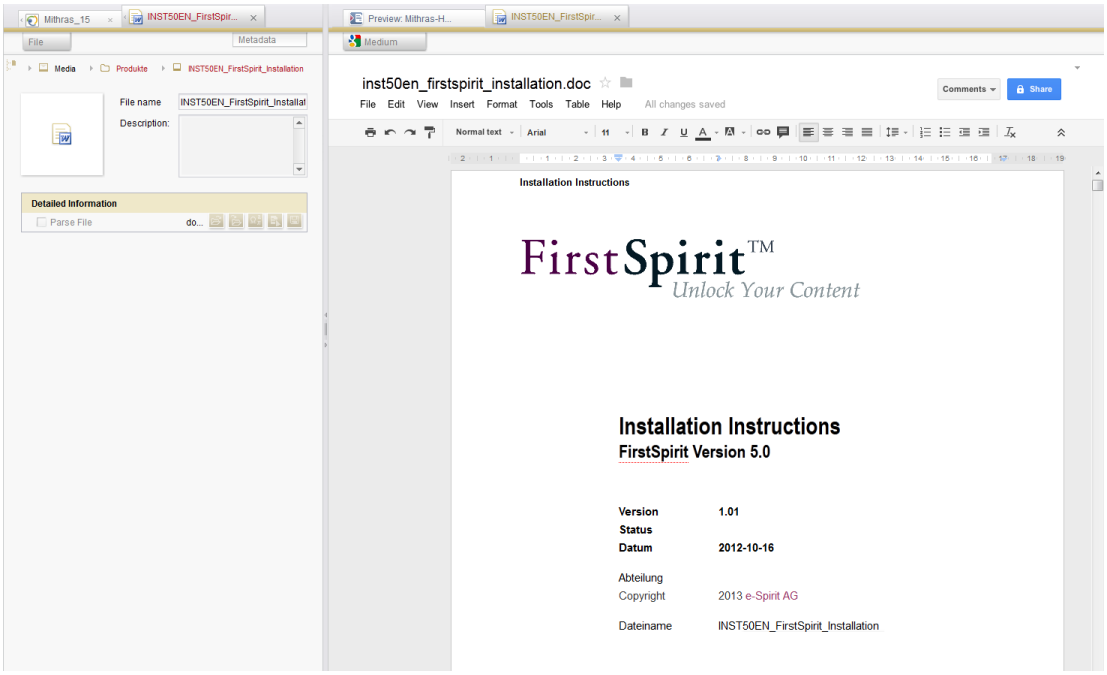


Figure 6-11: Editing a Word document using Google Docs in Site Architect

The particular document is no longer edited using Microsoft Office on the editor's local desktop, but is instead edited directly in the web browser using Google Cloud services (e.g. Google Docs), which are fully and seamlessly integrated in FirstSpirit. During integration, it is possible to choose whether documents are to be transferred to the Google Cloud infrastructure just for the editing process (i.e. temporarily) or whether the documents (and thus also change ownership) are to be pushed completely to the cloud.

The implementation in Content Creator is similar. Convenient and secure: Simply closing the Google Docs browser tab is all that is necessary to transfer the changes in the document made in Google Docs to FirstSpirit.

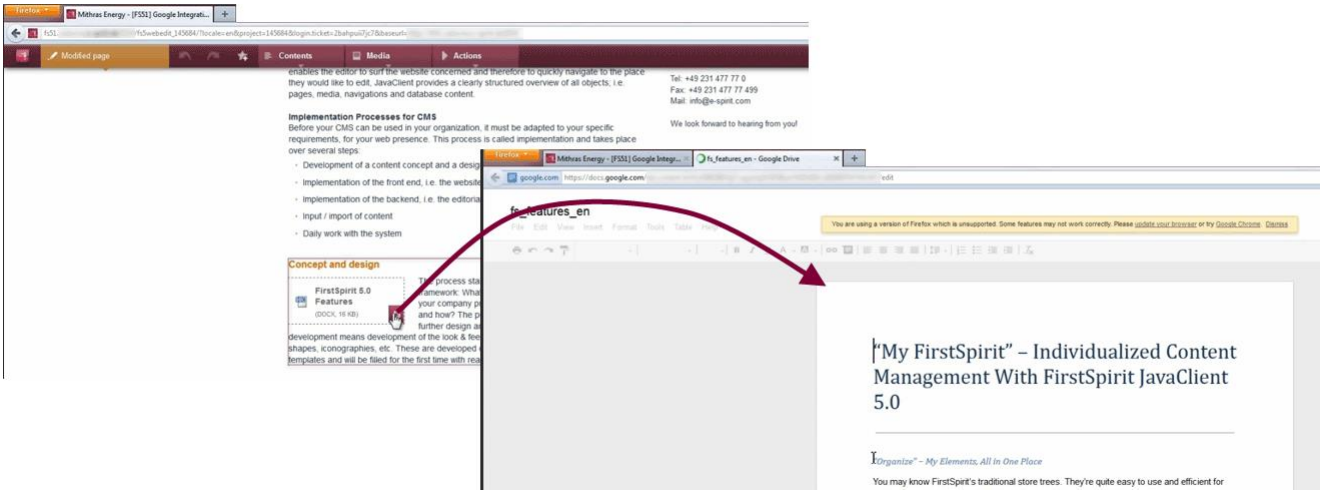


Figure 6-12: Editing a Word document using Google Docs in ContentCreator

To make the website content available to users to download regardless of the application and platform, office suite documents are automatically converted to the PDF format on demand when they are added to a website. However, any editing is done only to the office document.

Not only office documents can be edited this way in Google and saved on demand. Images and video clips that are already in Google Drive or other web services can also be edited using FirstSpirit and can be used for content creation. Or, approaching this from the other side, images in FirstSpirit can be edited semi-professionally with Google Drive apps such as Pixlr without the editor noticing the switch between the applications used. For an easy way to create links, web pages that are found on the client, e.g. via the integrated Google Search function, can simply be added to the FirstSpirit preview via drag-and-drop, where they are automatically converted to an external link on this page.

In addition, integrating Google Drive also opens access to other services and applications that do not come from Google directly but can be very useful in everyday editing. These include applications for creating MindMap graphics as well as calendar apps, project management tools, etc.

Google integration is not an "out-of-the-box" solution, but rather an example of something that has to be implemented for a specific project. The implementation depends on if and how Google apps are already used in the company.

More information about the integration of Google in FirstSpirit, including specific sample scenarios, can be found in the sections of the FirstSpirit Whitepaper starting with "Connecting the Cloud".

6.8 Including external services with the help of reports

Reports are considered a special type of integration. These were first introduced in version 5.0 only in ContentCreator as a central starting point for implementation in order to provide current project information (e.g. Which pages in the project were last worked on? Which pages do I need to release?) and (e.g. via the search function) to navigate to the desired pages. With the release of FirstSpirit version 5.1, they are now also available in SiteArchitect.

In both clients, reports now offer a wide variety of options for displaying data in almost any format from practically any source and using them in the project. The data are displayed in the form of a list, which the user can adjust. For instance, files can be filtered and sorted based on relevant criteria, which is particularly useful if there are a lot of them. Usually the files are displayed with their name and a brief preview. The editor can then choose the desired elements from this list preview, display the detailed information and/or use drag-and-drop to use them in a project. Depending on the data type, additional functions are also possible, such as starting scripts and workflows in ContentCreator.

In addition to the standard ContentCreator reports (1), Figure 6-13 shows three additional reports: "Google Drive Search" (2), "Google Web Search" (3) and "Fotolia" (4). They make external files available.

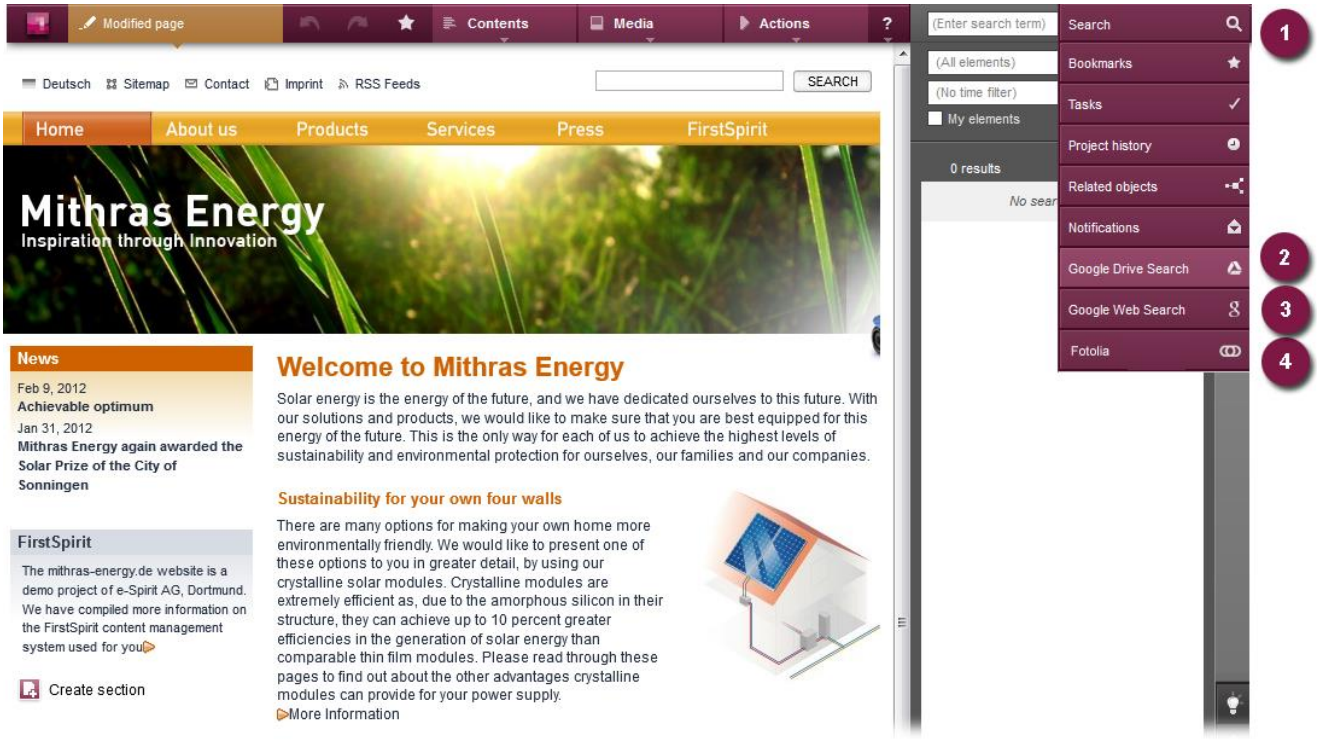


Figure 6-13: Integration of additional search options in FirstSpirit ContentCreator

The reports are able to accept a high volume of input parameters set by the user (in Figure 6-14, e.g. search terms via a search field (2) and a filter option to limit searches to particular file types (3) – in this case web content, images and videos) and thus analyze a sufficient number of results. Access to Google Search is available primarily for research assistance during the editorial process and is based on the Google Search API. This means that the search can be configured for a specific project so the user can limit browsing to only specific areas of the Internet, for instance. In the example presented here, the Google Search API is configured to perform a global Internet search. The ContentCreator then handles the display of the results in the form of a uniform, seamless list integrated in the UI (see Figure 6-14 (4)). This displayed list features, among other things, a thumbnail preview and continuous scrolling so that the report developer does not have to manage it himself.

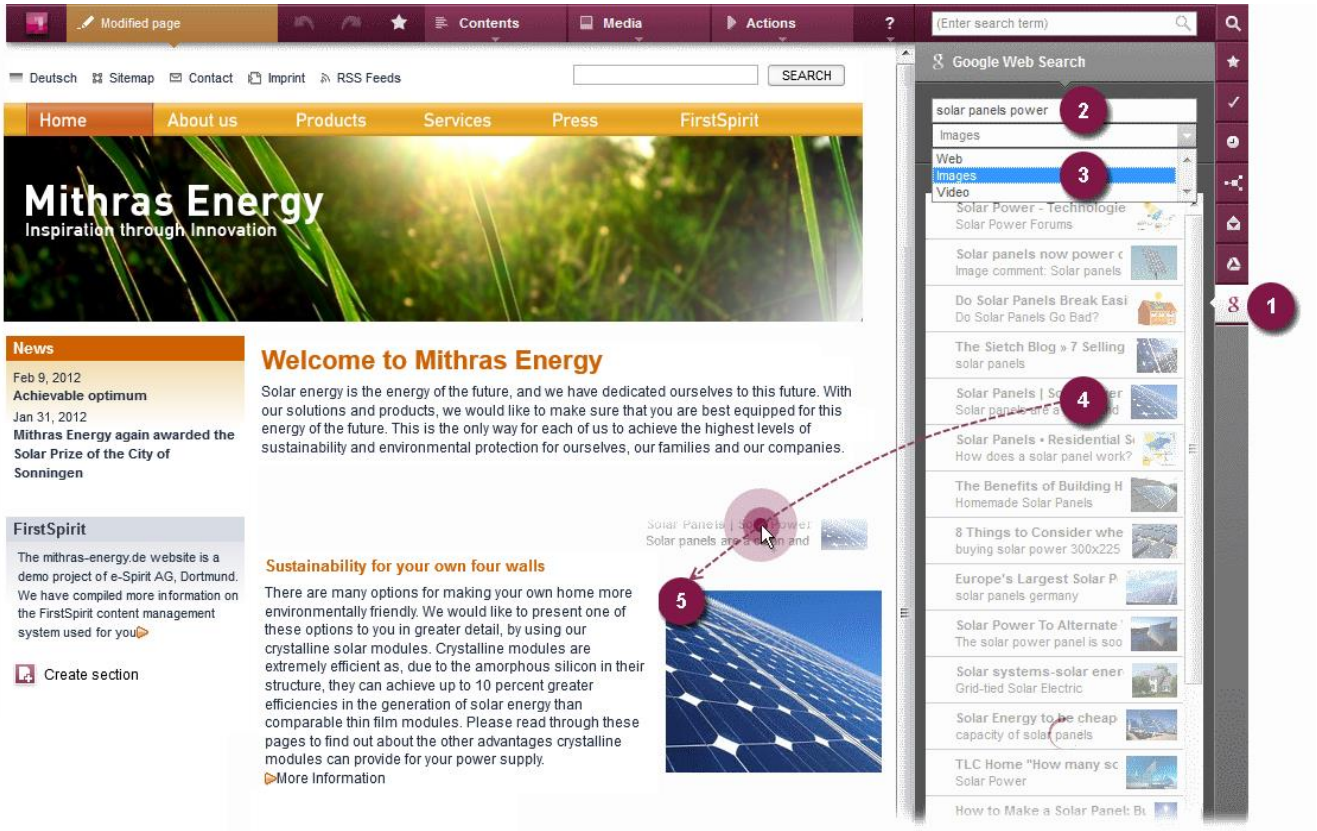


Figure 6-14: Drag-and-drop from the integrated Google Images search (ContentCreator)

The report elements can be copied to the ContentCreator preview using drag-and-drop, and input components can be used to upload the images (e.g. from a Google photo search) or other data and reference them on a page. The upload process is hidden from the editor, making it look to him as though all he has to do is click a button to add a wide variety of content (text, images, videos) to his page using drag-and-drop. Areas highlighted in color on the page reveal exactly where objects from a report may be placed. Figure 6-14 shows an example of a Google Images search with the subsequent drag-and-drop of a search hit (4) from the report onto an image input component (5). Since FirstSpirit Version 5.1, data from reports can also be used in forms that are open (also refer to Chapter 6.6, page 44).

Figure 6-15 shows the integration of a report with a similar function in SiteArchitect. In SiteArchitect, reports are placed in the Organize column on the left-hand side. In addition to the project tree structure, this column also inherently contains meta information and navigation options. The editor can enter search terms in the search box and limit the number of hits to specific file types (1). The search is again handled by the Google Search API and can be configured for a specific project. The search results are

then displayed in the form of a uniform, seamless list (2) integrated in the UI. Even here the results can be copied from the report to the editing area of SiteArchitect via drag-and-drop (3):

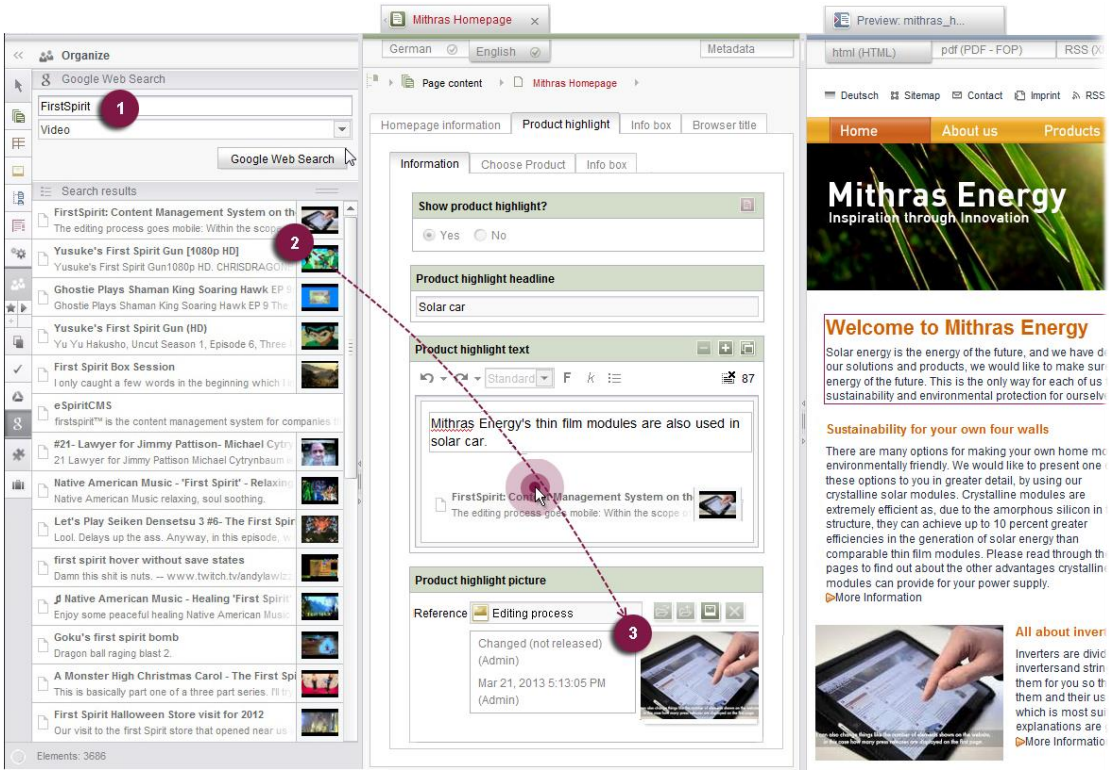


Figure 6-15: Drag-and-drop from the integrated Google Web search (SiteArchitect)

As in ContentCreator, depending on the requirements of the project and editors, the data source can be hidden from the user here as well so that during content development it is almost impossible to tell whether the data is coming from the FirstSpirit Organize column or from an external source, perfecting the seamless integration of external services. On the other hand, integrated services in the AppCenter area of SiteArchitect (e.g. online photo services) usually offer more functions than reports due to the greater amount of available space, but they also include the UI of the integrated service (also refer to Figure 5-9).

In SiteArchitect, reports can also be called from the project entry page, which is also a new feature implemented in version 5.1. The project entry page serves as a central, personalized dashboard that offers among other things quick access to different reports (also refer to section 6.10, page 56).

6.9 ContentCreator: new and enhanced features

A major feature included in the redevelopment and enhancement of ContentCreator in FirstSpirit version 5 is support for the intuitive, time-saving **drag-and-drop** control option. The infrastructure in version 5.1 is intended to allow for the ability to use drag-and-drop to move more complex data, such as data from customized reports. Drag-and-drop can therefore be used to move geolocation information from a report to a geolocation component in the preview (see Figure 6-9).

Drag-and-drop can also be used on menu items so that the user can easily change the menu structure in ContentCreator by moving the menu items in the preview with the mouse – including within projects with a deep hierarchical structure. As in other areas, the editor has a direct view of possible drop targets.

Thanks to the **image map** input component, link-sensitive graphics can now be easily created and managed in ContentCreator as well:

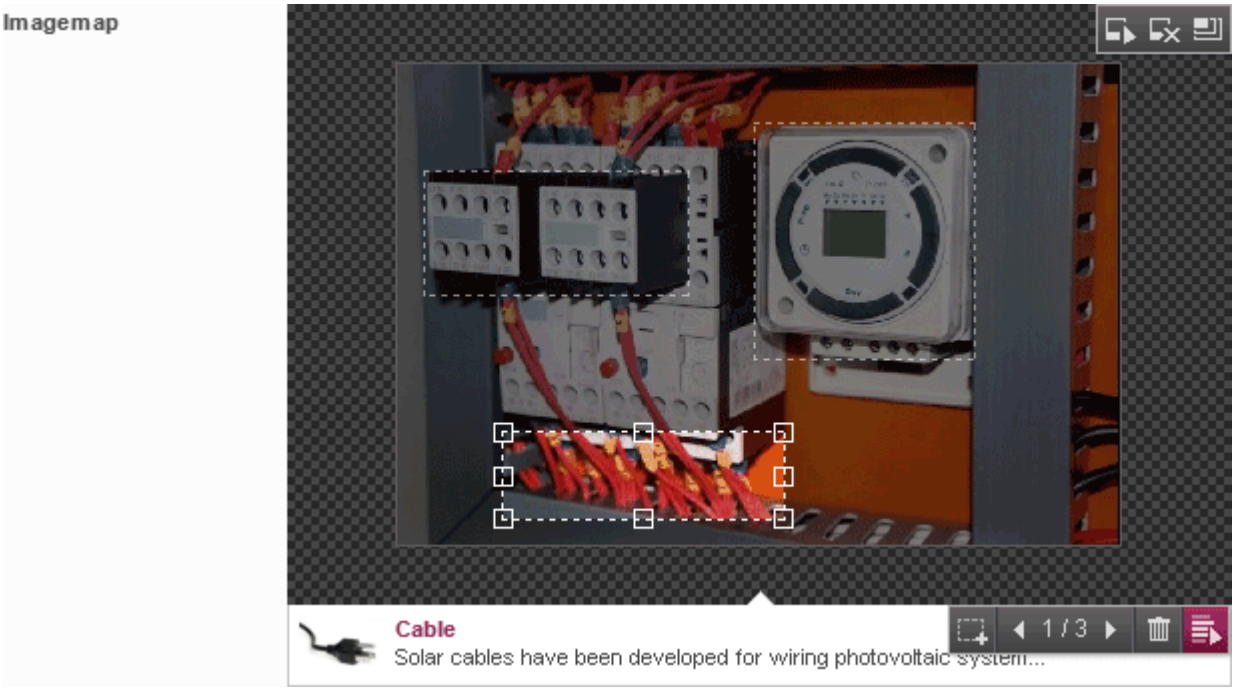


Figure 6-16: Input component for image maps in ContentCreator

To make work in a team more convenient, version 5.1 introduces the ability for editors to **communicate** directly in ContentCreator : For instance, if editor A is working on the content of a page, editor B can send him a message with a request to release the page for editing. When editor A is finished editing the page, editor B receives a message to that effect.

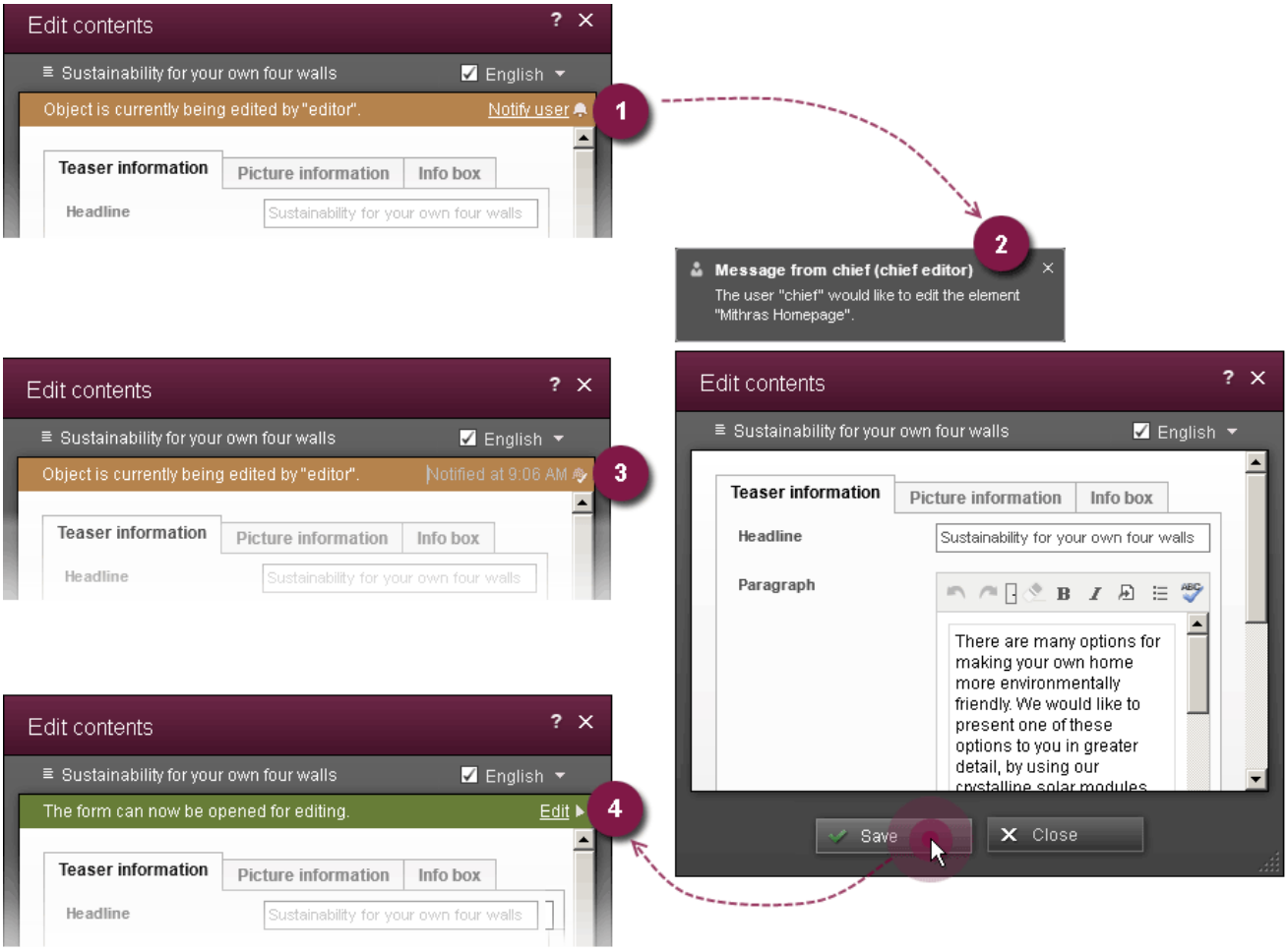


Figure 6-17: Messaging function in ContentCreator

In addition, support for content entry by the editor has been further enhanced by applying the **Snippet** concept to additional areas of ContentCreator so that, for instance, links can be added in a rich text editor (CMS_INPUT_DOM).

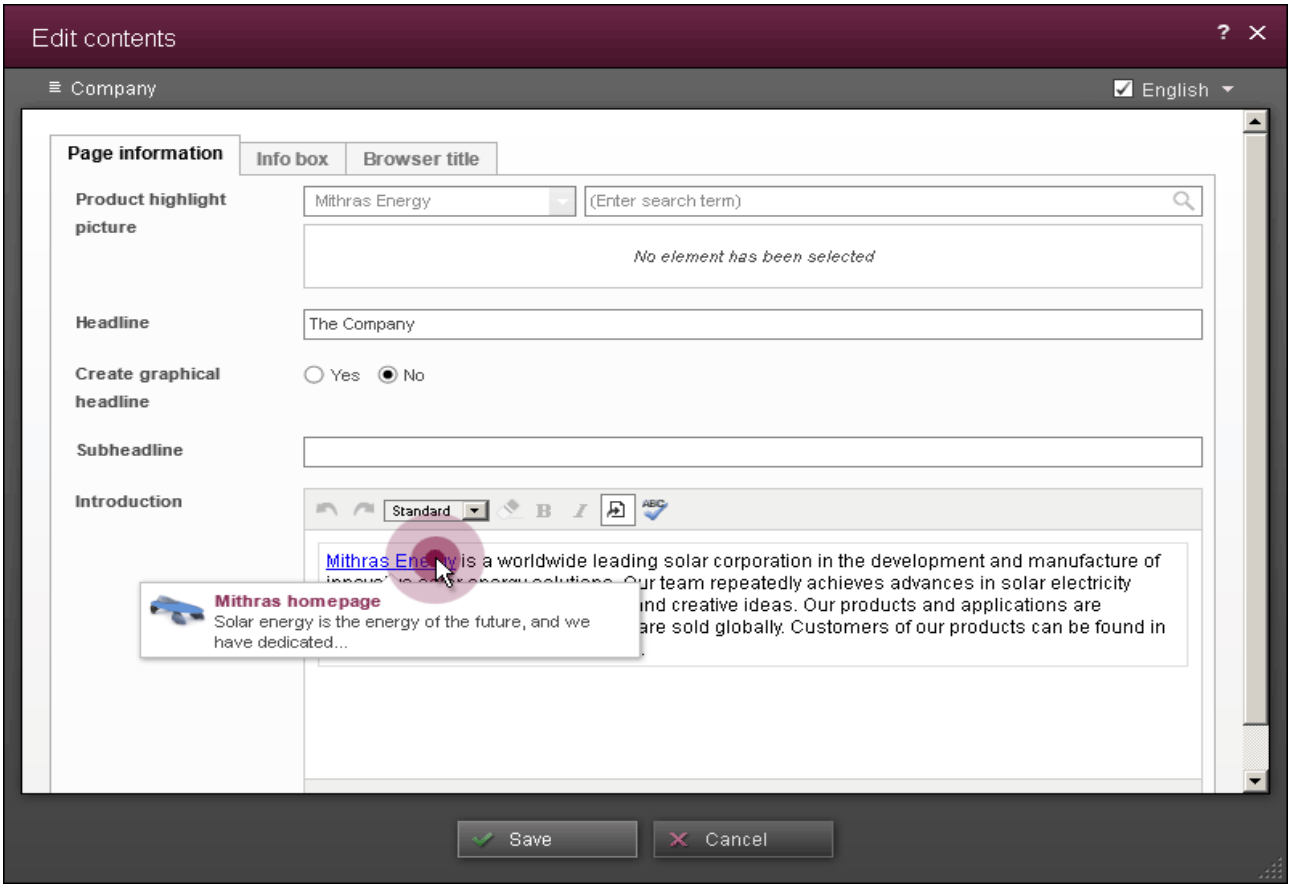


Figure 6-18: Snippet link in ContentCreator

General improvements have been made to the visual and functional design of ContentCreator.

6.10 SiteArchitect: new and enhanced features

FirstSpirit version 5.1 adds new features to SiteArchitect to provide an even better user experience. The SiteArchitect **software interface** will not only have a fresh, more streamlined look and feel, but will also feature improved performance thanks to refactoring "under the hood."

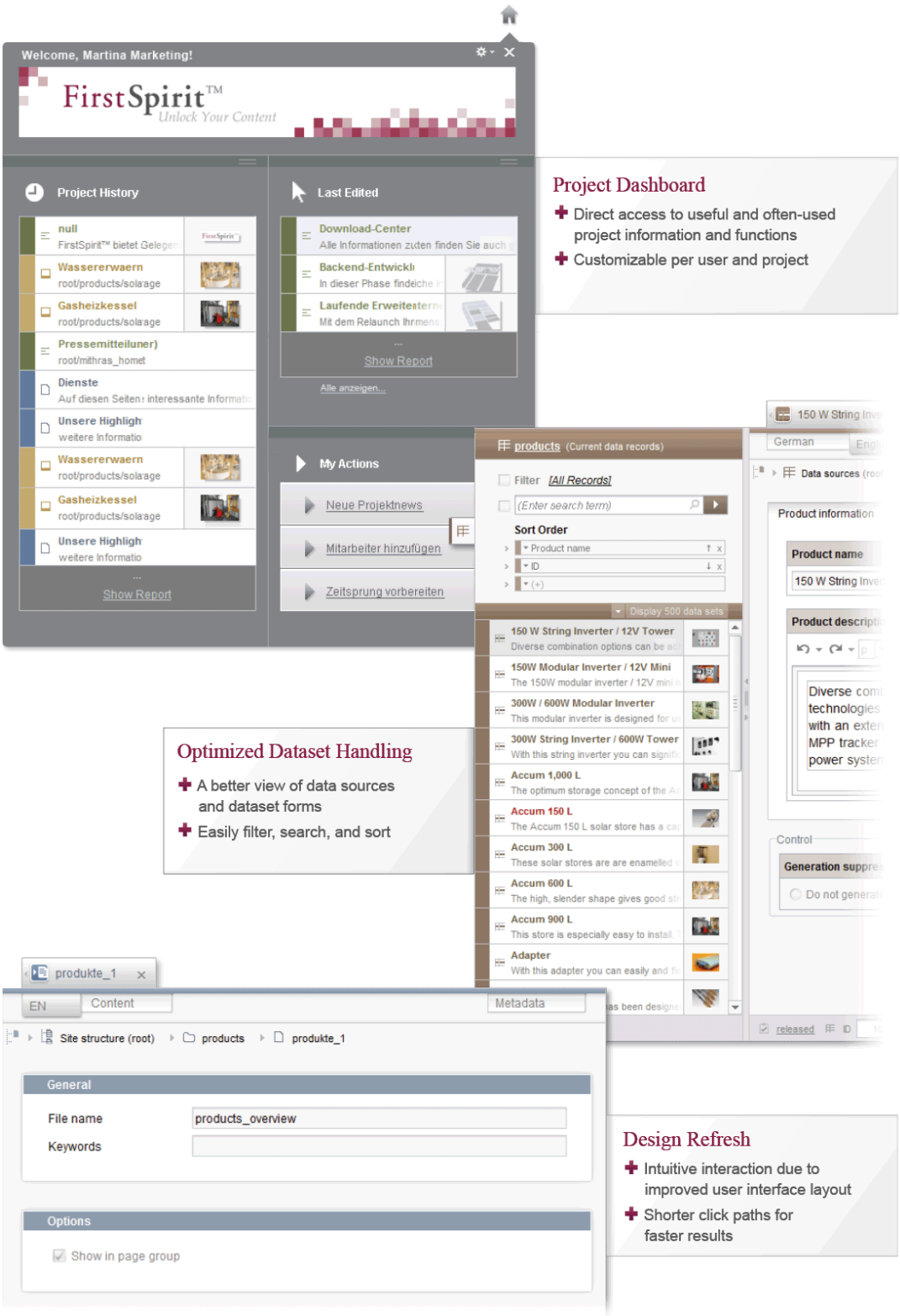


Figure 6-19: Improvements to the interface design

One key feature will be the ability to customize the software even more using the project entry page, for instance, which is now playing an even more important role as a central hub and starting point for all work in SiteArchitect.

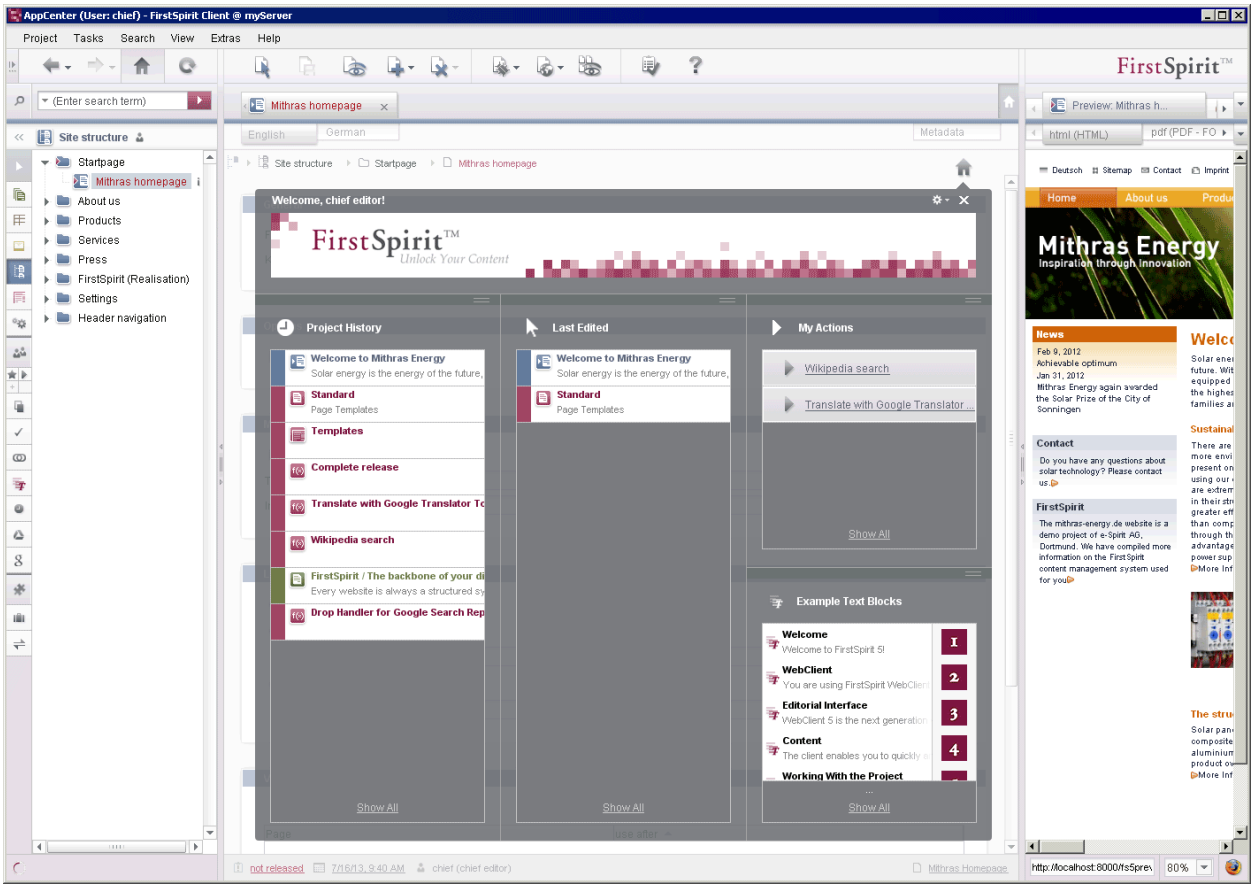


Figure 6-20: Project entry page

In addition to displaying content from reports (also refer to section 6.8, page 50) and other frequently required content, users will also be able to view FirstSpirit objects that were most recently edited. This will help the user access work from the last session faster while also allowing the user to view which objects were last edited by colleagues, which is very useful when collaborating as a team.

To reduce the number of clicks to a destination, object contents will now no longer be divided on multiple tabs in the individual stores, but will instead be clearly displayed on a single level. Editors for model workflows and database content storage will feature a new look with improved user control.

6.11 Further enhancement of the "FirstSpirit Content Transport" module

FirstSpirit data can be grouped together into a package using the "FirstSpirit Content Transport" module function. This allows the data to be imported into other projects and onto other servers. FirstSpirit version 5.1 is to implement a new and intelligent storage option so that projects can be easily synchronized in a traceable manner – even across server boundaries. This is necessary in the context of DQP scenarios, for example, where it is a question of transferring new functions from a development system (D) to a quality assurance system (Q) so that the function can be tested there. After being tested successfully, the function is transferred further to the production system (P) starting from D.

In addition to the local file system, other file systems can also be configured for the purpose of exchanging the content transport packages. Web-based file storage systems (such as Dropbox) are particularly interesting and pioneering in this regard. Using what is known as a "cloud connector" (also refer to section 8.2 page 74, Figure 8-1) FirstSpirit Servers are able to store and modify data in Dropbox, and then retrieve it again from there. This means that different FirstSpirit Servers can exchange data, including in cases where a firewall would normally prevent them from collaborating in this way.

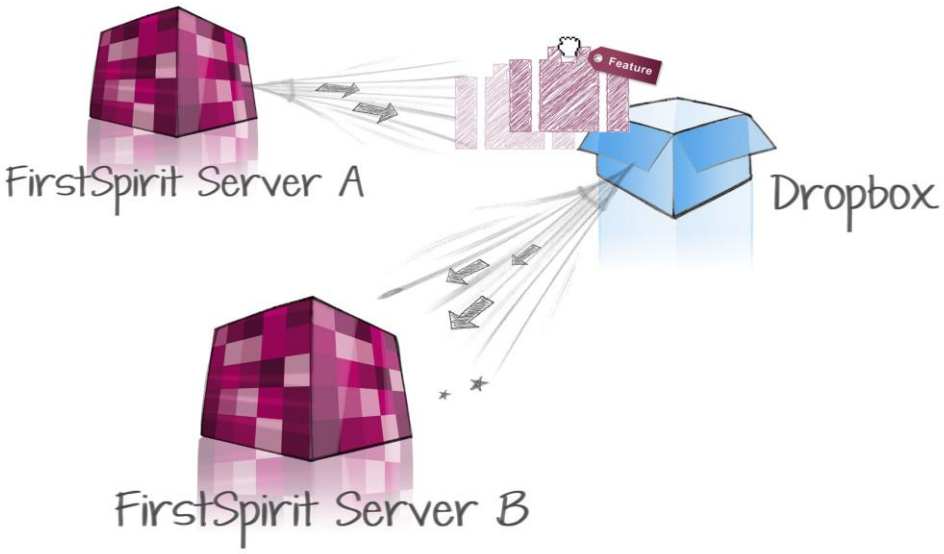


Figure 6-21: FirstSpirit Server exchange via Dropbox

Metadata is stored in addition to the actual content data. This makes it possible to trace, for example, which project is using which content package and on which server.

Content transport content can be automatically updated at predefined times using the FirstSpirit schedule entry planning feature. Workflows enable a systematic response to changes and new objects within the updated projects. When importing content from a source project to a target project, differences between the state of the content of the features to be imported and that of the target project can now be checked.

6.12 Additional features

Version 5.1 brings **IPv6** connectivity to FirstSpirit, making it ready for the future of the Internet. The new Internet protocol will affect features such as FirstSpirit Server configuration, license validation and even external software components such as the integrated Eclipse Jetty web server, but will not be noticeable to users. IPv4 will of course continue to be supported. Users will have the option to choose whether to use IPv6, IPv4 or both at the same time ("dual stack").

In addition to continuing to offer the reliable 3.x version of Oracle Berkeley DB as a **repository** for saving content data, FirstSpirit version 5.1 will also include version 5.x in order to provide the latest performance advantages over the previous version, to remain at the forefront of technical development, and to stay on top of future development trends. For reasons of compatibility, both versions can be used in tandem on the same server within different projects. Migration from one version to the other is as easy as pressing a button.

FirstSpirit is a commercial off-the-shelf product designed for a wide variety of uses. Due to this basic universal structure, it is not always possible to implement specific, detailed project requirements using the standard functions. FirstSpirit therefore offers the ability to customize and expand its capabilities in many areas using the **FirstSpirit API**, for instance. FirstSpirit 5.1 has added a wide range of improvements to the API environment. For example, tasks such as installing and updating modules and starting and switching workflows via the API are now possible or have been made easier. In addition, the horizontal menu bar and the context menu in SiteArchitect can be easily supplemented. Functional enhancements can be implemented at the same time in SiteArchitect and ContentCreator in the form of modules.

7 FirstSpirit Version 5.2 (planned release: 2nd quarter of 2015)

Version 5.2 development will also focus on the FirstSpirit **developer experience** and will continue to improve on support for development processes. See sections 7.1 through 7.3, starting on page 61 for more information.

Another key focus has been on making FirstSpirit **cloud-ready**. This involves improved communication and collaboration between FirstSpirit users on the one hand and collaboration between FirstSpirit and various cloud services on the other. See section 7.5, page 65 for more information.

Eventually the Global Experience concept will also be applied to the Administrators user group (**Admin Experience**). Usually this user group is only peripheral to FirstSpirit and has more to do with the internal IT systems of the company that implements FirstSpirit. Administrators therefore quickly need to be able to find FirstSpirit-related tasks that require urgent attention in the available FirstSpirit applications. The configuration options and interface should closely resemble those of other software applications with which they are familiar. See section 7.6, page 65 for more information.

7.1 Transformation of source material in FirstSpirit templates

The product-side support of development processes starts the moment the material provided by the agency arrives and has to be prepared for use within FirstSpirit. The time and effort required for this step can vary greatly depending on how extensive the agency's expertise is with regard to content management in general and FirstSpirit in particular. However, the task essentially consists of transferring a certain number of sample documents to FirstSpirit (usually HTML click dummies, but sometimes XML documents) that describe the project goal to be worked out with the customer. This first involves transferring sample documents and any media used in them to the relevant FirstSpirit stores. The second step is to "split up", or deconstruct, the sample material into FirstSpirit modules (page, section and format templates, etc.). The next step involves removing the core content. This means that usually areas filled with text and media samples are removed from the source text and are replaced by FirstSpirit editable forms and related output instructions.

The support goal on the product-side is to relieve the developer of routine activities. The aim is for FirstSpirit to provide support for approximately 80% of routine tasks that involve adapting the output material. Full automation is virtually impossible due to the

wide variety of output documents.

In addition to the fully automatic transformation process in the implementation, emphasis is primarily placed on the interactive (semi-automatic) deconstruction of the documents.

Fully automatic transformations are based on the idea that rules can recognize certain structures in raw materials, making it possible to transform them directly. The semi-automatic transformation solution on the other hand is controlled by the developer and is somewhat comparable to "refactoring" support in well-known development environments. For instance, selected areas from the raw material can be saved out to a new FirstSpirit section template using a simple instruction. Form elements can then be added to the template interactively. FirstSpirit forms are configured using a graphical user interface that allows even fairly new FirstSpirit developers to develop template forms quickly without having to consult the documentation.

One area that poses a bit of a problem for new users in particular due to its complex nature is the template-side description of the navigation structure (key words: multilevel navigation, expansion strategy, etc.). Even in this area, efforts have been made to provide support for the tools.

The cyclical development process of design development, implementation in HTML and implementation in FirstSpirit syntax with the inclusion of external service providers is supported by the fact that differences between different design states and HTML states are recognized by FirstSpirit automatically and deviations can be transferred to the FirstSpirit syntax interactively. FirstSpirit thus not only makes collaboration with external companies easier, but also facilitates a practically simultaneous development and implementation of the design.

 *The purpose of this FirstSpirit enhancement is to improve on the creation processes and not to migrate from existing, potentially extensive data pools, such as those from legacy systems. Established partner solutions have been around for quite some time and have been implemented precisely for this purpose.*

7.2 Application integration for an optimized development cycle

The development process consists of the "changing, testing, and correcting" cycle, which only comes to a (preliminary) end after a whole series of iterations when a new development status is imported into a version control system (see section 6.3, page 38).

In order to support this development process cycle more efficiently, the following functions have been introduced in FirstSpirit version 5.2:

- **Expanded local execution environment:** FirstSpirit version 5.1 was the first version to provide developers with a local execution environment that offered them the entire FirstSpirit infrastructure (FirstSpirit applications, FirstSpirit server, application server). As part of version 5.2, this execution environment will become even more closely connected to the local software development environment (e.g. IDEA IntelliJ or Eclipse). In addition, it will include integration in established version control systems (such as SVN or GIT). As per the FirstSpirit best of breed strategy, the required interfaces will be provided for the integration of other products as well.

- **Connection to established development tools**: Up to this point, the development of FirstSpirit templates has mainly taken place in the FirstSpirit environment. Version 5.2 will introduce interfaces to enable an efficient connection from FirstSpirit to established external development environments within the company as well as to any existing version control systems. As per the FirstSpirit best of breed philosophy, some popular tools will be supported directly (examples include Eclipse and SVN), while at the same time the API interfaces required for connecting to alternative systems will be developed to support the additional infrastructure.

7.3 Multi-perspective preview in SiteArchitect

One of the many strengths of FirstSpirit is that entered content can be viewed directly in the preview and can be checked in its presented form. Users can already work directly in the ContentCreator preview. SiteArchitect offers the integrated preview for this process. This allows for a close connection between the editor's entry of content and the result output in HTML format controlled by the developer.

The fact that even during the FirstSpirit template development process the integrated preview is not just a simulation, but that the output can be checked live as part of cross-browser testing in a real browser environment using multiple browser engines (currently Microsoft Internet Explorer / Mozilla Firefox (version 3.6) / Mozilla Firefox (version 15)) is an advantage. For the developer, the preview of content should be as accurate as possible so that both the appearance and functionality of the website can be tested the best way possible, since variances between browsers can be quite obvious. For quite some time it has been possible to store even locally installed browser applications in SiteArchitect, which can then be used at the click of a button for the external preview, including even in the more "exotic" browsers.

In order to expand the range of integrated testing options, support for **Google Chrome** will be included in FirstSpirit version 5.2. This browser has been gaining in popularity for some time and therefore has become a true competitor of the current market leaders Microsoft and Mozilla, making it an ideal addition to the best of breed portfolio of FirstSpirit integrations.

Once the **Multi-perspective preview** concept has been introduced in ContentCreator as part of FirstSpirit Version 5.1 (see section 6.2 page 34), in version 5.2 this concept will be applied to SiteArchitect as well. As far as developers are concerned, it is important to have as realistic a simulation as possible so that the layout can be optimally adapted for the various display sizes. For this reason, an existing professional solution is to be integrated. Features such as the option to pay for data with costs involved and data transfer will be included as a result of this implementation.

7.4 LiveEdit: editing web pages directly

To overcome the challenges in editing websites, it will be possible to use a new FirstSpirit control tool called "LiveEdit", for instance, to change the ranking of the importance of content input by colleagues in an intranet or online community page versus customer content.

To do this, buttons will be provided in the live page in the areas to be edited, such as a title, paragraph or image, so that the particular data can be edited directly. The forms for adding or changing content will not be opened in a new browser window, but will instead be displayed in a discreet HTML layer, without requiring the individual to log in.

Function parameters can also be set for the buttons so that functions like the release workflow can be carried out directly after saving.

7.5 Improved collaboration in ContentCreator

ContentCreator will be expanded further for use in cloud environments, among other things. To achieve this, the multi-user capability and team collaboration features will be improved in version 5.2. For instance, it will be possible to edit content by multiple editors simultaneously. If an object is being edited by another user, this will be displayed in ContentCreator (also refer to the messaging function in version 5.1, section 6.9) and a chat session can be started to communicate with the co-editor. Another communication function could be the implementation of interactive memos. Not only could these be used for delayed communication with other editors, but they could also be used for personal task lists. These development features will be added to SiteArchitect in subsequent versions of FirstSpirit.

7.6 Another pillar in the Global Experience: Admin Experience

With FirstSpirit version 5.1, a future-oriented focus on user groups has been established/initiated against the background of the Global Experience approach for the existing clients (JavaClient (SiteArchitect) and WebClient (ContentCreator)) (see section 5.1, starting on page 18). Consequently, this also included moving functions from one FirstSpirit application to another.

In FirstSpirit version 5.2, the administrative tasks will also be grouped together and restructured ("Admin Experience"). The strictly web-based Server Monitoring function

will be expanded to include a tool for the central IT administration and for the interface to the company's internal IT systems. This is where monitoring of operations as well as the connection to the company-wide user directory and the basic security configurations for individual projects will take place. New projects will then also be added to Server Monitoring. In addition to functional enhancements, Server Monitoring will also be updated technically and visually from the ground up.

The technical configuration of a FirstSpirit project, e.g. adding assignments for languages and channels that previously took place in the Server and Project Configuration application (also called "AdminClient" or "Admin Console"), is not an administrative task and is usually also not handled by central IT administration. These rather (project) developer related configuration tasks will therefore be moved to SiteArchitect. Since functions will be more strongly delineated according to user group specific tasks, it will be possible to minimize switching between the individual FirstSpirit applications.

Very similar to the copy templates in ContentCreator and SiteArchitect, mechanisms will be created at the project level which will make it possible to generate "default projects" with particular project settings and pre-populated templates at the press of a button. It will be possible to use a wizard to retrieve specific project parameters, such as those for the basic layout, company logo, etc., during the creation process.

7.7 Further enhancement of the "FirstSpirit ContentTransport" module

The "FirstSpirit ContentTransport" module is specifically intended for use in large companies with lots of projects with a view to simplifying work that involves standardized company-wide data and to ensuring efficient reuse of existing content. ContentTransport makes it possible to combine objects from a project, such as pages, including all links, into what are called "features" and to prepare them for import into different target projects. The advantage of this feature is that objects can be managed in a central location (in the master project).

Extensive revision of this module was already started in FirstSpirit version 5.0 and continued in version 5.1. It will continue to be an important feature in version 5.2 as well. For instance, in addition to project content, which is entered in ContentCreator and in SiteArchitect, project properties that are currently defined in the Server and Project Configuration can also be transferred over. The following are just a few of these configuration options:

- Language settings (What languages can be used for content in a project?)
- User and group definitions (Which users and groups can access the project?)
- Adding modules
- Resolutions (In what resolutions can images be saved and output?)

Usually several individual (i.e. deviating from the initial default configuration of a project) settings can be configured in the different project configuration areas. If these setting configurations are also used in other projects (e.g. in a test project copy), they can easily be transferred using ContentTransport – even across server boundaries.

In addition, differentiations are possible when importing FirstSpirit objects into a target project, allowing only changed objects to be added to the target project. A type of "blacklist" is used to allocate the objects that are to be removed from an update in the target project – even if they are included in the feature.

7.8 Technical documentation and translation processes

FirstSpirit basically also provides authoring tools for the creation of user manuals and online help. To make the content creation and entry process as easy as possible, some enhancements to the existing editing functions are desirable. These enhancements will be implemented in the form of a technical documentation module.

A key focus will be the **reusability of content**. The module will support working with text modules that can easily be added to the continuous text as needed. Some topics will be modularized and can be reused as boilerplates. It will be possible to search through all the editorial content. For this purpose, a **server-wide search option** will be required for searching across multiple projects. This will make it possible for editors to search other projects for content that already describes the particular subject matter. It will be possible to easily copy the content of this search occurrence into the current project or to use it as a cross-reference. It is conceivable that a search occurrence could be highlighted and applied as an internal link to the current continuous text just by clicking on a button.

To ensure a uniform style of language and a consistent look across the project (even when different editors are involved), the **inclusion of a (cross-project) glossary** and a **style guide** will be supported. In addition, it should be possible to display messages when certain word markers are entered.

In this context, it would also be helpful to include the ability to **search and replace** individual terms beyond the boundaries of the overall project. It will have to be possible to find and replace the search terms within the entire editorial content (beyond pages and sections). To do this, the editor will have to be able to specify limitations, such as the exact case (upper or lower) parameter.

The ability to **use reference tools** within the editing environment should also be included. If a word within the continuous text is selected, it should be possible to use a context menu or button to display a **thesaurus** with suggested synonyms and other subject-related terms (acronyms, abbreviations, translations, antonyms).

In addition to conventional spell checking, which is provided using the "FirstSpirit Spell Service" module, an option should be included for user-specific configurable **auto correction** of content. For instance, simple typos could be corrected automatically during entry. These functions would also allow the technical editor to define abbreviations that are replaced automatically by the frequently used longer terms during

typing. It would also be nice to **display alternative suggestions** when entering text.

The editor should also have the ability to add **metadata** to text fragments. This would make it possible to **classify editorial content** by version numbers, for instance. It is possible to imagine a scenario in which different states of a software are documented. If all editorial content is assigned a classification, it would be very easy to create version-specific documentation in which only the general descriptions and the content associated with a particular version are generated. The editors could then enter all the content in a project and during generation they could decide whether a manual should be created for version A or version B. The metadata would also be used to select text that is outdated and needs to be revised or text that has changed and needs to be retranslated. The metadata could also be used to display **comments** in the continuous text. Using the comments, internal questions related to particular text fragments or other editor remarks could be saved. The comments should only be displayed when doing editing work (e.g. as a tool tip) and not when generating the content.

Tag-based change tracking would also be possible as part of the "Mobile Tagging" functionality. Mobile Tagging can be used to enter barcodes to which editorial content would be added. These barcodes could be read by a camera on a mobile end device and then decoded. The functionality could be used to compare different versions of a document. Here is an example: A customer prints out a technical document (e.g. a technical data sheet). This is labeled with a tag (barcode). After a year, the customer wants to find out whether a newer version of the document is available. To do this, he simply uses the barcode on the existing document, which he can read into his mobile phone via the phone's camera, and then obtains a change history with all changes that have been made to the document since the version he has currently.

7.9 Support for translation memories

FirstSpirit already has a proven translation strategy for language-dependent project content. Using the enhancements described in the next section for editorial management of text modules, it is possible to make additional improvements to translation processes.

First, the knowledge from translation processes already completed should be saved and reused for new translations in order to minimize translation costs. In the process, the editor could be supported by the **translation memory** functionality:

- Translation memory is used to save content already translated as paired text fragments (sentences, parts of sentences or individual words) **in the source and target language within a database**. The translated text (individual words and whole text fragments) is automatically imported into the database (in the source language and one or more target languages). Formatting remains intact. Manual revision of the content can be done as needed.
- To achieve optimum results for the translation process, recurring content should be modularized as text modules (see also "Reusability of content" in section 7.8 (page 68)). Using these types of text modules reduces the cost during editing (content can be reused) and considerably reduces the cost of subsequent translation work. If, for instance, the phrase "Please read and comply with the safety information" is already contained in the translation memory, the same sentence does not need to be retranslated for each new translation. However, a variant of the same content – "Compliance with safety information is required" – would require a new translation.
- If a new translation process is started because content has been added or changed, the editor can first compare the text to be translated with the content stored in the translation memory and then have the translation memory display **suggested translations**. During this process, in addition to a strictly "**full text search**", a search for similar terms and for the same meanings would also be conceivable ("**concordance search**"). For instance, the search for "house" would also return "houses" as a result, or the search for "Please read and comply with the safety information" would also return "Compliance with safety information is required". The deviations between the current sentence and the search occurrence from the translation memory entry are highlighted for the editor. The editor can then decide whether to apply the suggested translation or whether a new translation is required for the text. If a search occurrence matches the text searched, the translation can be applied automatically.
- It is useful to save and display **context information** related to a translation memory entry. Translations can mean different things depending on the context. For instance, the German term "beenden" can be translated as "terminate", "exit" or "shut down" in English. Depending on the context, a different term should be used for the translation (e.g. "beenden" in the menu item "Server" – "Beenden" would be translated as "Shut down", but in the menu item "Projekt" – "Beenden" it would be translated as "Exit").
- Analogous to the user of dictionaries in the "Spell Service" module, it should be possible to add and manage a **language-specific glossary** with frequently used expressions and/or standard expressions within the company via the "Translation Engine Connector" module. The glossary could then be used for internal translation

processes via the translation memory and could be sent to a translation agency as a guideline. Terms contained in the glossary should be highlighted in the text.

- To aid in translation, an interface is also planned for connecting to additional references.

In the second step, the "Translation Engine Connection" module will offer a better overview of changed or new, but not yet translated text and GUI support for editing these lists of changes and sending to the translation agency:

- This module will determine at the input component level whether the content has changed since the last translation and will **automatically generate a list of changes**. Potential changes can also be recorded which are not to be incorporated in the translation process, such as the correction of a typo in a language or the modification of a (language-independent) sample configuration. These changes can be highlighted manually as "not relevant for translation" by the project manager.
- The list of changes to be translated is presented so that it is easy to see the content that needs to be translated and the language into which it will be translated. The **change list can be filtered by certain criteria** so that it is possible to exclude particular input components or to include only referenced pages for the export.
- After creating the change lists, the project manager can press a button to create an **export file for the desired languages** and send it to the translation agency.

The "Translation Engine Connector" module supports editors and project managers when changing and translating language-dependent project content. Convenient filter functions and the consistent reuse of existing text modules can effectively minimize the scope and costs related to translation processes in the project. The module is closely connected to the "FirstSpirit Technical Documentation" module (section 7.8 page 68).

8 FirstSpirit Version 6: Cloud Edition (planned release: 2nd quarter of 2017)

After version 5.x, the focus of development will switch from the developer experience to the issue of **cloud integration** in version 6.x. This will still relate to the client side ("AppCenter") initially, but will then increasingly start to move toward the server side (with FirstSpirit running in the cloud) and will lead to a **FirstSpirit Cloud Edition**.

Given that there is no large-scale acceptance of cloud solutions at this stage, particularly within the enterprise segment, the "Cloud Edition" will be developed as a proprietary product line. Initially, this will be aimed at innovative companies that have already entered the cloud. FirstSpirit will continue to exist as an on-premise solution and will still be actively developed, but every effort will be made to ensure that the on-premise and cloud structures are, insofar as possible, based on the same architecture.

In order to remain faithful to its successful best of breed strategy (see, for example, section 5) e-Spirit is determined to rely on external – and to a large extent specialized – service providers in the areas of memory utilization and data management as well. Where current cloud solutions from competitors presently allow the entire software to run in a cloud, FirstSpirit 6.x will pursue the aim of outsourcing individual FirstSpirit components (such as data management). While the FirstSpirit cloud connectors that are currently in existence allow FirstSpirit to connect to cloud services (see section 8.2 page 74), the "true" FirstSpirit cloud solution will involve fundamental changes to the underlying code (see section 8.3 page 78).

8.1 Cloud computing models: Definitions and differences

Cloud computing is always a question of outsourcing resources or services to specialized providers to achieve cost and performance advantages along with faster and higher availability rates plus maximum flexibility. There are various different forms of operation:

On-premise
On-premise is the traditional operating model for software applications: Users acquire the software and run it on their own hardware. It is not usually enough to budget for one-off purchase costs, as there are follow-up costs for support and maintenance (updates) if the software is to continue running efficiently in the long term. The users themselves

are responsible for installing, configuring, operating, and maintaining the software.

Hosting
Hosting providers make hardware resources available to users, which equals computing performance and storage space. In the most straightforward scenario, the software (i.e. FirstSpirit) is operated and maintained by the customer itself. This form of hosting is very similar to the on-premise variant except for the fact that the software runs on an external server instead of an internal one. Specialized hosters go one step further by regularly updating FirstSpirit in addition to providing operation and maintenance services. With conventional hosting, capacities, runtime, and costs are defined in advance.

IaaS ("Infrastructure as a Service")
The IaaS model is similar to hosting in that a specialized provider makes basic IT resources available, such as computing performance and storage/network capacity. The customers themselves are responsible for managing the operating system and software applications, but the amount of capacity can be increased or reduced at short notice in a flexible manner (according to requirements), often via a self-service portal (scalability). Furthermore, customers benefit from usage-based billing, which means they only pay for what they actually use.

Examples: Amazon Elastic Compute Cloud (Amazon EC2), Google Compute Engine, and HP Cloud.

PaaS ("Platform as a Service")
PaaS solutions provide runtime and development environments that allow developers to develop and run various types of application, including web applications. The necessary hardware and software (e.g. operating system, database, web server) are also included. However, users are not allowed to access these basic resources; only programming interfaces, which they access via the cloud. Storage capacity and computing performance are usually scaled automatically, which means they adapt flexibly to the customer's specific needs. Billing is usage-based and relies on straightforward and standardized models.

Examples: Google App Engine and Windows Azure.

SaaS ("Software as a Service")
The SaaS model allows customers to use application software that runs and is maintained on the provider's infrastructure. This software often consists of web

applications. Customers do not have to worry about installing, configuring, or operating either the software or the underlying hardware. The customer pays a usage-based fee, e.g. per month and per user. Applications are available immediately and usage can be increased or reduced at short notice if required. Customers no longer have the expense of acquiring their own software license.

Examples: Google Apps, Microsoft Office 365, and Salesforce.

8.2 FirstSpirit and the cloud: Already becoming a reality in version 5.x

Traditionally, FirstSpirit has always been an **on-premise** solution. However, right from the start customers have received installation, configuration, and operational support from the e-Spirit help desk, project consultants, and an online community. In addition, e-Spirit also offers **hosting** and managed services in cooperation with its partners. In this case, FirstSpirit is manually installed and configured on the hosting partner's hardware; operating system and FirstSpirit Server updates can be requested as a service from the provider.

FirstSpirit already offers several interfaces and functions in version 5.x that make it cloud-compatible. These are called "cloud connectors". Primarily, these allow FirstSpirit and external services to interact through the import/export of data. In this scenario, FirstSpirit (server and applications) runs locally on the customer's equipment, on a rented server (hosting), or in the cloud ("Cloud 1.0").

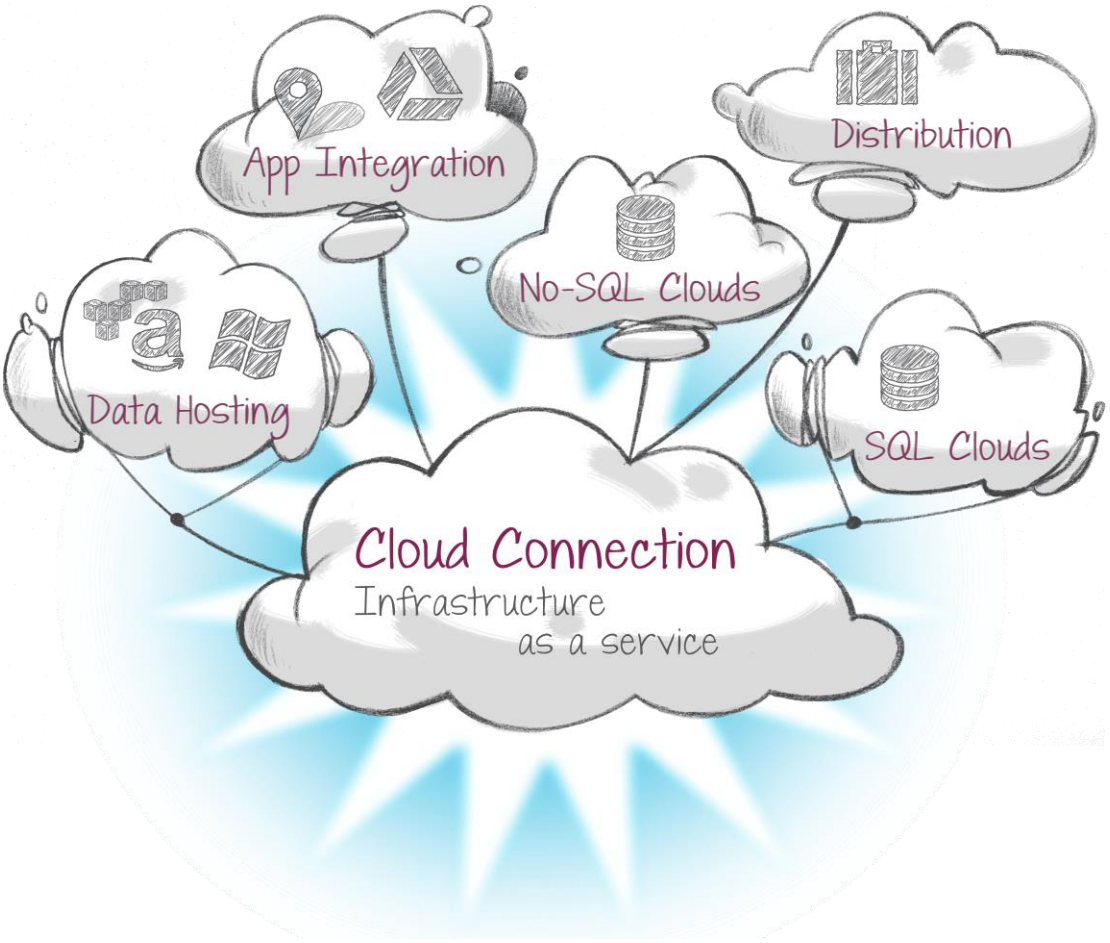


Figure 8-1: FirstSpirit cloud connectors in version 5.x

- **Application integration:** Application integration within ContentCreator and SiteArchitect provides access to cloud applications such as Google Apps for the purpose of editing Office documents, images, etc. (also refer to section 6.7 starting on page 47 for more information).
- **Deployment:** Deployment targets can be specified for the purpose of deploying sites and other content from FirstSpirit. This could take the form of Google Drive, for example. From here, delivery can take place via the Google Content Delivery Network (CDN). Amazon/Azure (etc.) can be used in a similar way.
- **Content transport:** FirstSpirit data can be grouped together into a package using the "FirstSpirit Content Transport" module function. This allows the data to be imported into other projects and onto other servers. Of course, it can also be backed up, distributed, and synchronized via cloud storage services such as Google Drive or Dropbox. Please also refer to section 6.11 page 59.

- **Database connection:** FirstSpirit supports the connection of various database systems. The database schema (involving tables plus content columns and, where applicable, the relationships between several tables) can be easily modeled via an editor. If required, editors can easily enter database content into FirstSpirit in table format and select it for use on the desired web pages. Relational cloud storage systems can also be used as database systems within this context, e.g. Google Cloud SQL and Amazon EC2 databases. One of the main advantages is that these can be made available quickly.
- **Software update:** e-Spirit offers central update management for hosting partners. This allows FirstSpirit installations that are already up and running to be updated fully automatically as soon as a new software version becomes available, thereby keeping them constantly up to date.

FirstSpirit at the touch of a button

Current efforts in the area of FirstSpirit cloud operation are aimed at achieving a higher level of automation when setting up a ready-to-use FirstSpirit environment and updating it (e.g. importing new software versions, updating the operating system and modules). Within this context, it would be conceivable to have FirstSpirit pre-configured according to the application concerned, e.g. FirstSpirit for the Internet, for an intranet, for portals, etc. The associated build process can be compared to the process of creating an MS SQL instance with Amazon Web Services: This involves querying various parameters with the help of a wizard. On completion of the wizard, the user has a solution that can be used straight away. Scaling can be requested as a bespoke service. FirstSpirit usually runs on a virtual machine (VM). A suitable delivery system can be offered alongside this or a CDN can be used.

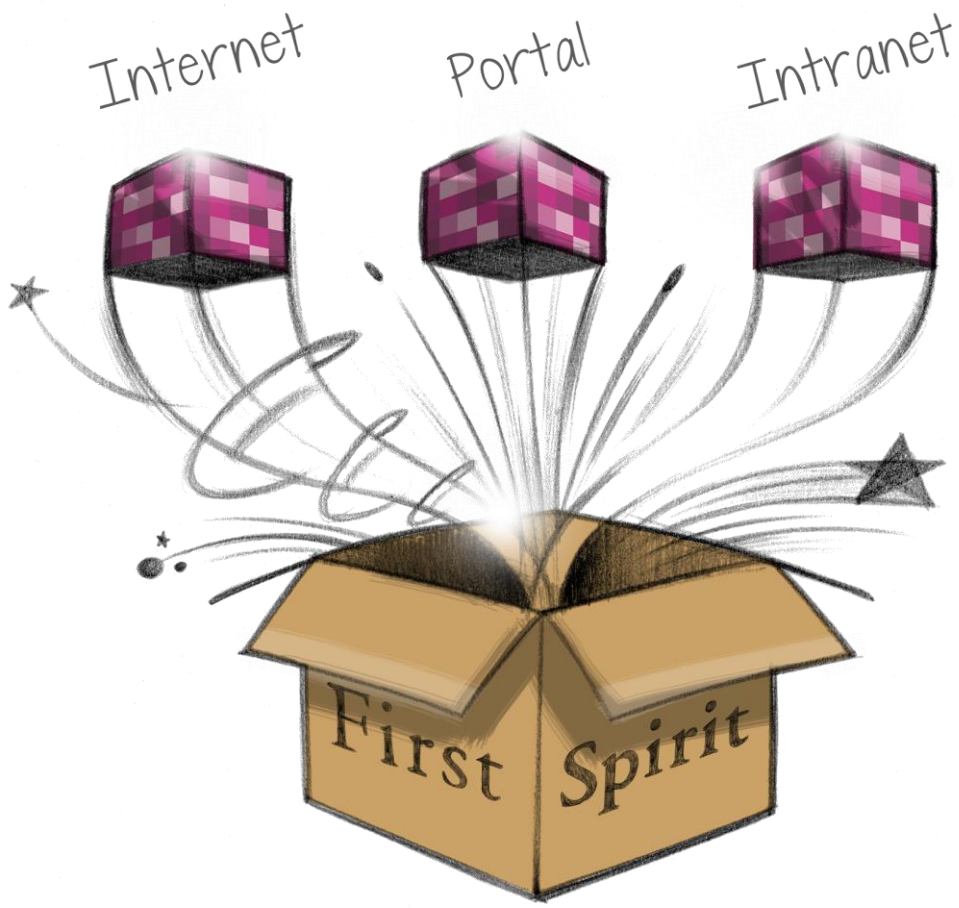


Figure 8-2: FirstSpirit "at the touch of a button"

8.3 FirstSpirit and the cloud: To be implemented as of version 6.x

FirstSpirit version 6.x is to feature additional **cloud connectors** (see section 8.2 page 74) so that FirstSpirit can be connected to services from the cloud. Among other things, the back end is to be replaceable for the purpose of storing FirstSpirit content that is not kept in databases ("repository"). This will allow NoSQL cloud storage systems such as Amazon DynamoDB or Apache Hadoop to be quickly integrated in its place.

Furthermore, Version 6.x aims to see FirstSpirit used as a cloud edition that is based on the PaaS (or SaaS) model.

If the **PaaS** model is adopted, FirstSpirit will no longer be operated as an IaaS solution using virtualization technology. Instead, it will run natively on a cloud platform, which will itself provide far superior services compared with IaaS solutions (e.g. persistence, CDN, and scalability).

If a database is to be used for FirstSpirit and FirstSpirit takes the form of an IaaS solution, operating system virtualization will (for example) be required. The database can then be installed on this. By contrast, if FirstSpirit is used as a PaaS solution, persistence as a service will be included by the cloud provider as standard. In an IaaS environment, the scaling must be undertaken by the application developer or operator while in a PaaS environment it will be performed automatically by the cloud operator according to FirstSpirit's current performance and storage space requirements. FirstSpirit customers do not have to worry about providing additional resources or about configuring FirstSpirit to ensure efficient operation on the modified resources. In fact, they do not even have to have the expertise required for this. Ideally, the fact that cloud services are being used should not be apparent to any FirstSpirit user group (see section 5.1 page 18 for more information).

Other examples of services offered by the provider when FirstSpirit takes the form of a PaaS solution: Addition of instances for preview and generation servers, system updates, provision of a FirstSpirit API. The delivery system will usually take the form of the provider's CDN. Using FirstSpirit ContentCreator as a true cloud solution seems a highly feasible option and should be easy to implement.

8.4 Which model for which application?

If FirstSpirit version 6.x is to be made available on the basis of various usage models, the following question remains: Which model is the right one to use at a particular time? The following represent important decision criteria in this regard:

- Is confidential/sensitive data being managed? ("Privacy")
- Are back end systems (e.g. MAM or SAP) connected? ("Integration")
- Are new FirstSpirit projects created on a frequent basis?
- Are new FirstSpirit Servers frequently required?
- Should it be possible to create new servers and projects at the touch of a button? ("One-click servers and projects")
- Is a scalable delivery/editing system required? ("High traffic/edit sites")

		On-premise	Hosting	IaaS	PaaS
Privacy	Management of sensitive data (Operation must be isolated from the Internet.)	+	+	o	-
Integration	Connection of back end systems (It must be possible to access back end system (LAN connection).)	+	+	+	o
One-click projects	Generating new (ready-made) FirstSpirit projects at the touch of a button (On-premise is only possible/advisable in "exceptional cases".)	o	+	+	+
One-click server	Generating new FirstSpirit Servers at the touch of a button	-	o	+	+
High edit sites	Scalable editing system	-	o	+	+
High traffic sites	Scalable delivery system	-	-	+	+

Conclusion:

The larger and more bespoke the projects concerned, the more likely they are to involve the integration of systems such as SAP or MAM and the higher the level of confidentiality, the stronger the argument for an IaaS solution.

The PaaS/SaaS solution, on the other hand, is primarily characterized by its short time to market: It is no longer necessary to conclude time-consuming contracts for various services with providers and the solution already includes a fully functional project. The investment and running costs are low. PaaS and SaaS are ideal for companies that have already transferred a large number of services to the cloud, e.g. by using Google Cloud technology for Office/groupware, etc.