

A person wearing a headset is shown from the side, looking at a computer monitor. The monitor displays the Salesforce interface, which includes a sidebar with navigation options and a main content area with a table of data. The person's hand is visible near the headset. The background is slightly blurred, showing a desk with a keyboard and some papers.

Reshape Customer Support

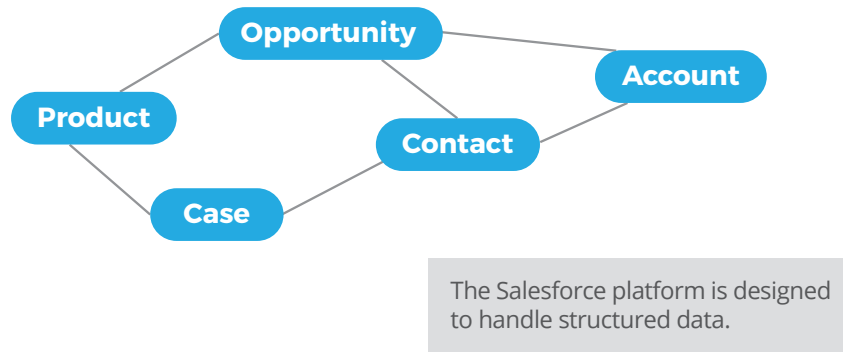
With Technical Content in Salesforce

Salesforce Information Model

Reaching far beyond its origins as a Customer Relationship Management (CRM) solution, Salesforce now delivers and supports many customer-centric processes through several solutions, including:

- **Salesforce Service**, a helpdesk tool to manage cases (support tickets);
- **Community Cloud** for building self-service portals to engage with users, partners and employees;
- **Field Service** to organize maintenance activities and related mobile workforces.

All capabilities are built around a comprehensive and extensible data model, centered on customers and on the products or services they have purchased.

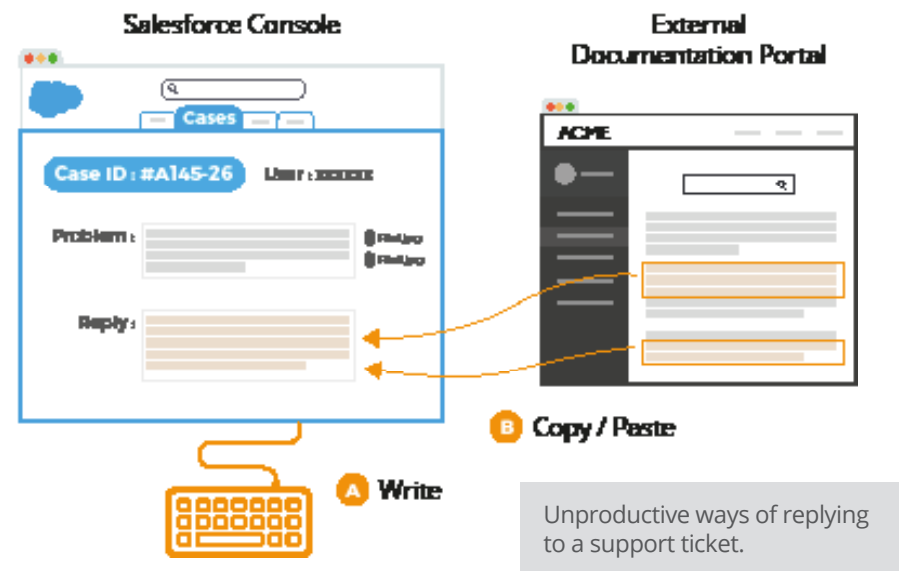


Handling documents with Salesforce

Salesforce is not intended for storing documents. Marginally, it is possible to store content as notes or file attachments to core objects. For example, a purchase order can be attached to a deal, or a service contract can be stored in PDF form as part of the customer record.

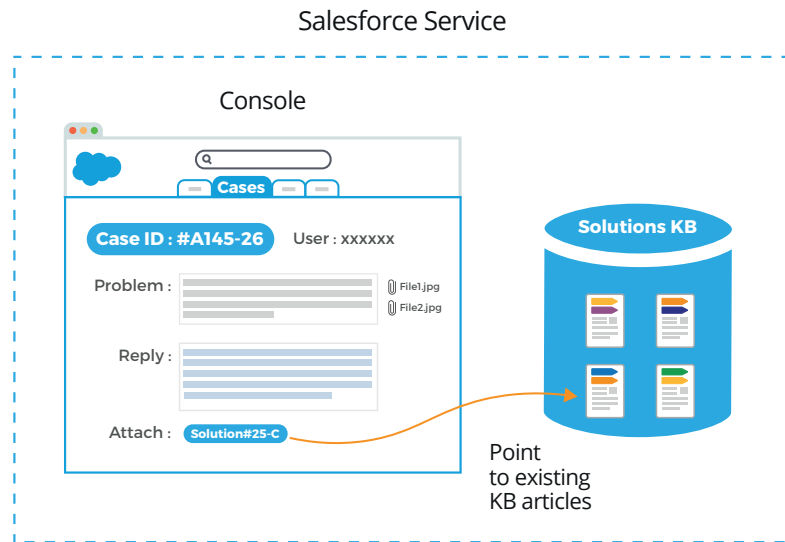
Clearly one thing Salesforce has not been designed for though, is the management of textual content such as technical documentation. As a result, when a support agent services a customer by replying to a ticket, all she can do is:

- A write an answer from scratch, which is time consuming and highly inefficient for recurring subjects, or
- B copy and paste fragments of text from existing repositories of content.



The Salesforce Knowledge Base

In an effort to increase efficiency, Salesforce has added to their platform an internal Knowledge Base system called “Solutions”, that lets support agents link existing solutions (KB articles) to cases.



Replying to a ticket by creating links to Salesforce Solution records.

And yet, the Salesforce KB content model is fairly basic, with records that simply consist of a title, a text body and a few default system metadata (creation date, author, status, etc.). It is theoretically possible to add more metadata through model extension, but this is a cumbersome task, unable to accommodate sophisticated enterprise taxonomies.

Edit Solution UI languages

Solution Edit [Save] [Save & New] [Cancel]

Solution Information [Required Information]

Solution Number: 00000005 Public: ☒

Status: Valid Visible in Public Knowledge Base: ☒

Language: English

Detail Information

Solution Title: UI languages

Body:

Formatting Controls: [How to use this]

Font: [Size: 12] [Bold] [Italic] [Underline] [Link] [Unlink] [List] [Table] [Image] [Video] [Audio] [Code] [Quote] [Table of Contents]

Fluid Topics user interface is available by default in 10 languages:

- English,
- French,
- Spanish,
- Portuguese,
- Italian,
- German,
- Japanese,
- Korean,
- Chinese Simplified,
- Chinese Traditional.

All labels and strings can be localized. It is possible to modify the existing translations, to add new languages.

[Save] [Save & New] [Cancel]

Solution articles editing in Salesforce.

Solution Fields

This page allows you to specify the fields that can appear on the Solution page. You can create up to 500 Solution custom fields.

Note that deleting a custom field will delete any filters that use the custom field. It may also change the result of Assignment or Escalation Rules that rely on the custom field data.

[Set History Tracking]

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed	Track History
	Created By	CreatedBy	Lookup(User)			<input type="checkbox"/>
Edit	Language	SolutionLanguage	Picklist			<input type="checkbox"/>
	Last Modified By	LastModifiedBy	Lookup(User)			<input type="checkbox"/>
Edit	Master Solution	Parent	Lookup(Solution)		<input checked="" type="checkbox"/>	<input type="checkbox"/>
Edit	Master Solution Details	ParentNote	Rich Text Area(32000)			<input type="checkbox"/>
Edit	Master Solution Title	ParentIssue	Text(255)			<input type="checkbox"/>
Edit	Out of Date	IsOutOfDate	Checkbox			<input type="checkbox"/>
Edit	Public	IsPublished	Checkbox			<input checked="" type="checkbox"/>
Edit	Solution Details	SolutionNote	Rich Text Area(32000)			<input checked="" type="checkbox"/>
Edit	Solution Number	SolutionNumber	Auto Number			<input type="checkbox"/>
Edit	Solution Title	SolutionName	Text(255)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Replace Edit	Status	Status	Picklist			<input checked="" type="checkbox"/>
Edit	Visible in Public Knowledge Base	IsPublishedInPublicKb	Checkbox			<input checked="" type="checkbox"/>

Default fields for tagging Solutions.

The Salesforce Knowledge Base (KB)

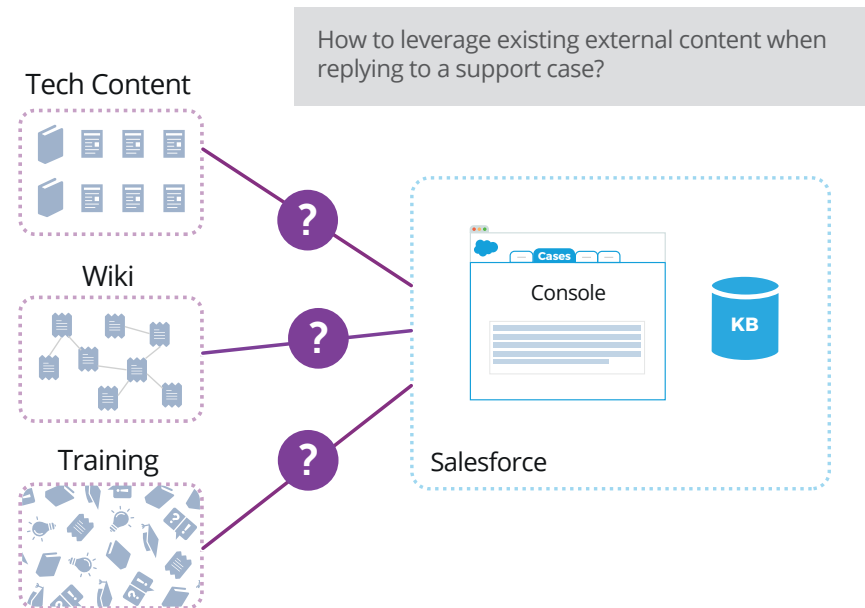
Furthermore, the KB record-based paradigm is suitable for modeling simple information where subjects are independent, and replies are short. But it is unable to support a more sophisticated content architecture where:

- Information is better conveyed through small or large manuals organizing content into chapters and sections.
- Content is inter-linked through cross-references.
- A piece of content may exist in multiple variants depending on product option, version or configuration.

The low level of semantics that is conveyed with simple HTML content also makes it difficult, if not impossible, to craft rich information that can be used in more advanced delivery channels, for example through chatbots. There is a good reason why enterprises choose structured authoring tools, and more sophisticated formats such as DITA to create and manage their technical documentation.

The content needed to efficiently support users may also be spread over multiple sources such as wikis contributed by subject matter experts, discussion forums, spare parts catalog, etc. In a nutshell, expecting the Salesforce KB to handle and manage all content needed for customer support, is simply unrealistic.

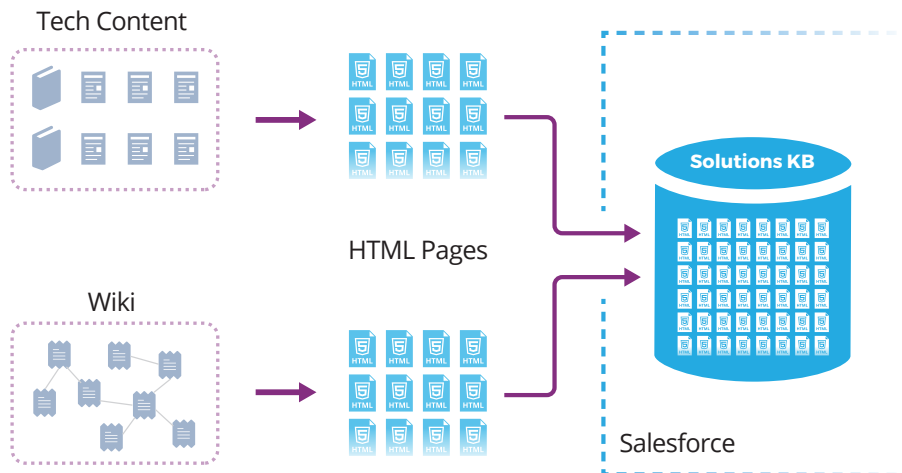
And yet, to efficiently leverage content when handling cases, support agents need to be able to find the right piece of information and to dynamically link content to tickets in order to avoid the obvious absurdity of content duplication.



The following sections of this white paper review the options available for solving this challenge, evaluate their efficiency, and discuss the most effective solutions.

Challenges of loading content into the Salesforce KB

One seemingly obvious solution to getting content into the Salesforce Solutions could be to turn various documents into HTML records and to load them into the Knowledge Base.



Dump all existing content as HTML and load it as Solution records is a daunting challenge.

This looks simple on the surface but in reality, it generates a number of problems that will end up impairing the entire support process. Let's review some of the difficulties.

- If content includes manuals, which typically range from few pages to hundreds of pages, generating HTML files for rendering each manual and load it into Salesforce would not work:
 - If a single (very) long page were generated for the entire manual, it would contain thousands of keywords and hence would match a large number of search queries, resulting in (very) weak relevance as those long untargeted documents would always be returned in search results.
 - The alternative option would be to chunk each manual into bits and to generate multiple short pages - per chapter, per section. Because the Salesforce KB is intended for storing independent articles, recreating such book structure would be close to impossible. But by doing so, the structure of the manual and the reading order would be lost, providing out-of-context information.
- If products exist in multiple versions (such as software in v1, v2.1, v2.2) or have variants (e.g. with optional components), each piece of content would have to be uploaded multiple times in the Salesforce KB, one per version, multiplied by the number of variants, resulting in highly redundant content that Salesforce is not designed to handle because it lacks techniques such as semantic clustering. Here again, the search relevance and the user experience would be severely impacted.
- When using advanced authoring tools that foster content reusability (such as DITA), the reused content components would be uploaded into the Salesforce KB as many times as they are used in different documents. This would result in redundant search results, further impairing search efficiency and findability.

Challenges of loading content into the Salesforce KB

- When dealing with more than one data source, extract-transform-load-into-Salesforce mechanisms would need to be developed for each source, each of the processes having to generate consistent HTML and to align metadata from these various silos before loading, so that the final knowledge base remains consistent.
- Loading content into Salesforce is complex – but updating this content is even more difficult - it would require that each uploaded piece of content is matched with the identifier that Salesforce has set during the initial upload, so that the proper article is replaced. Any change in the way content was chunked, transformed in HTML or organized would then require to drop all the articles from the KB before reloading, therefore breaking all existing links between content and cases.
- Providing gated access to content based on user credentials requires creating and maintaining a map of user profiles to the content metadata scheme. It is highly uncommon for content to be tagged with metadata values that match user groups and roles. Hence the need for a rule-based system that would transform and enrich existing content metadata with user group IDs recognized by Salesforce, and this must be done before loading content into Salesforce. Each modification of the security rules would require reprocessing and reloading the content, with the complexity that was highlighted before.

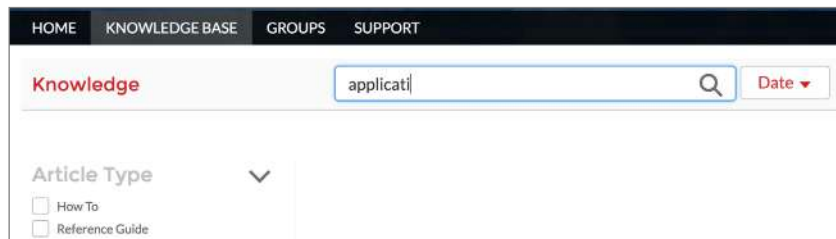
Conclusion

To cope with accelerating business dynamics, content is modified or added every day and the capability of an organization to adapt rapidly needs to be reflected in content updating workflows. The complexity and all of the challenges described above are impossible to solve without an industrial data processing platform and a modern content delivery platform, which Salesforce is not.

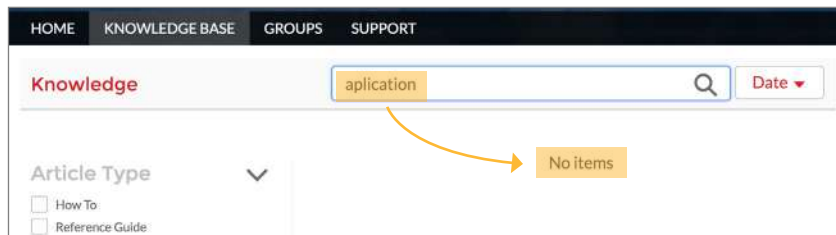
Searching for textual content in Salesforce

When it comes to delivering content, search has a huge impact on user experience. Salesforce's internal search engine is designed and optimized for looking into structured content (contacts, accounts, opportunities, etc.) and it has none of the linguistic capabilities required for searching into textual content, nor the features expected by the users. In order to illustrate these limitations, the following screenshots were taken from an actual Salesforce Community portal giving access to Solution articles.

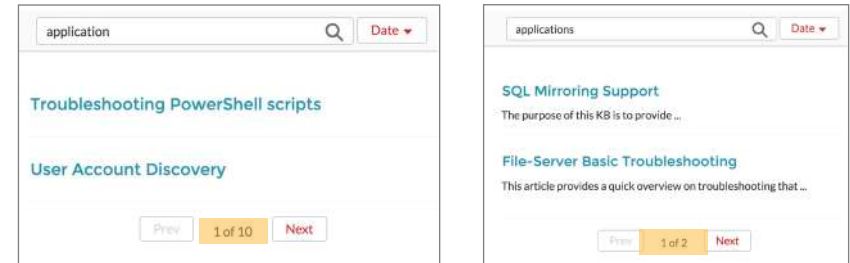
- Lacking type-ahead suggestions.



- Lacking spell-checking and typo-tolerance.

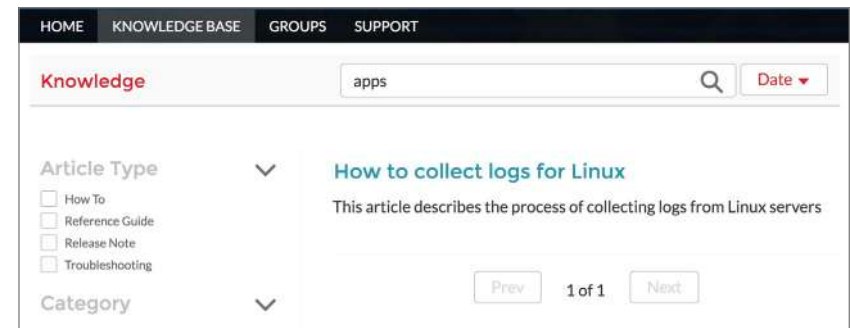


- Lacking linguistic processing and stemming (plural, compound splitting, alternate forms, ...).



Searching for "application" and "applications" does not return the same number of results.

- Lacking support for synonyms.

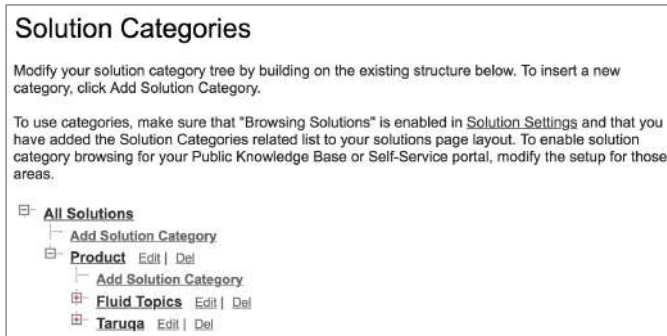


Searching for "apps" should return the same results as when searching for "applications".

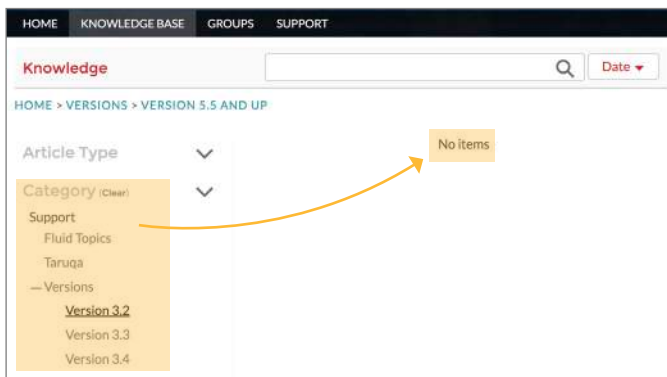
Searching for textual content in Salesforce

- Lacking dynamically faceted search.

Taxonomies are defined externally instead of being created dynamically from actual available content. Hence proposed values do not necessary match content metadata, which leads to “no items” replies.

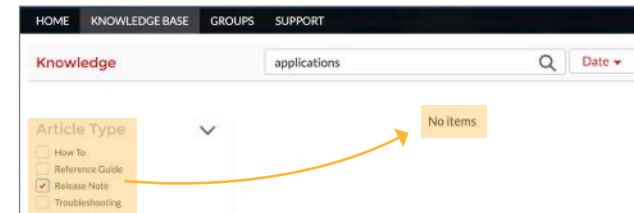


Metadata values for Solution articles are managed in Salesforce.



And they are statically displayed in search screens, even if there is no associated content.

- Facets are static (derived from taxonomies) and not search-driven: all filter values are always proposed, regardless of the existence of content matching the query, which again rapidly leads to frustrating “no items” replies.



Refining a search using statically displayed metadata leads to irrelevant searches.

- Lacking interaction.

Salesforce does not propose any feature so that users can interact with content, such as:

- Bookmark favorite articles.
- Be alerted when content matching certain context and products is added or modified.
- Send feedback to writers.

Not to mention that Salesforce provides no analytics on what is searched and read by users.

Conclusion

This weak search and access experience inevitably leads to:

- User frustration.
- More support cases, which means increased support costs.
- Lack of agent productivity who struggle to find the right answer, resulting in longer time to resolution and lower service quality.
- Missed opportunity to learn from user behavior to enhance content and understand product usage.

How to make content available in Salesforce

When analyzing the complexity and limitations of relying on Salesforce for hosting and serving content, it becomes obvious that a different strategy is needed. This section describes a solution where content is handled by a dedicated delivery platform that is seamlessly interconnected with Salesforce using web services.

1 Build a **unified hub of information** that:

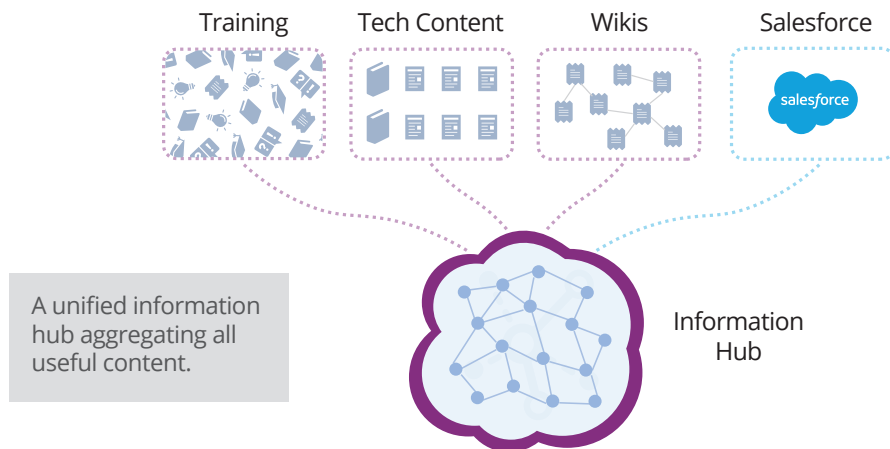
- Aggregates all content necessary to support the different stakeholders in their needs and assignments.
- Processes the various content pieces to create consistency, such as aligning the metadata.
- Gives access to this content through specialized web services, including advanced search capabilities.
- Enforces security rules in order to ensure that each user can see only what they are entitled to.

This hub of information must also give access to content managed by Salesforce, such as:

- Solution articles created by support agents to answer recurring questions that are not specifically handled by other sources.
- Past cases and replies.

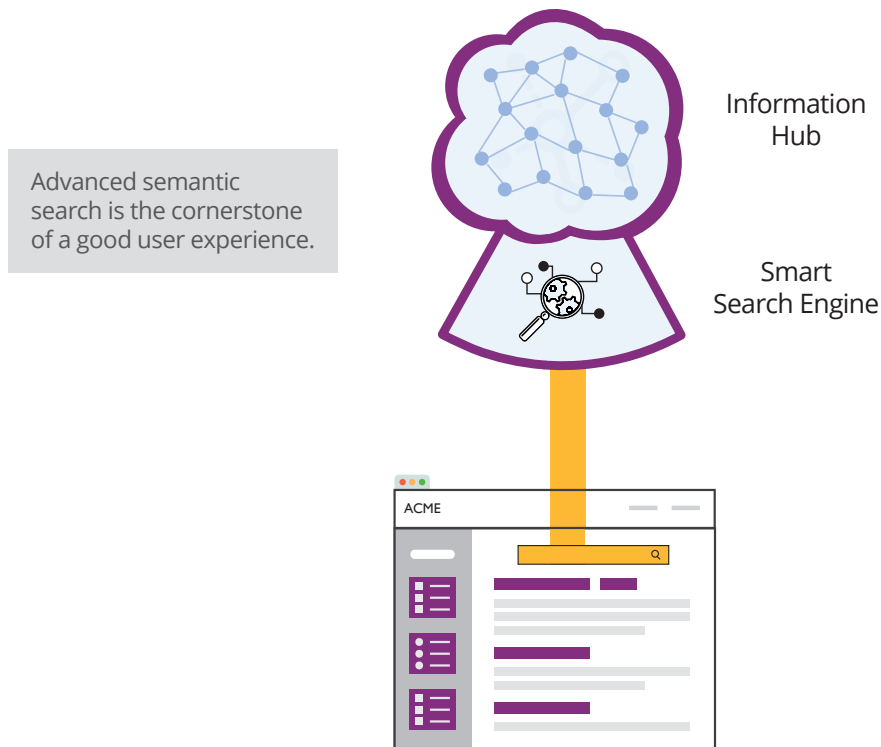
2 Offer a **search engine** that has the relevance, speed and sophistication that users expect:

- Type-ahead suggestions.
- Typo tolerance and automatic spell-checking.
- Semantic capabilities with support for synonyms and taxonomies.
- Full language support (flexions, decompounding, etc.).
- Smart snippets for highlighting search terms.
- Facets for filtering.

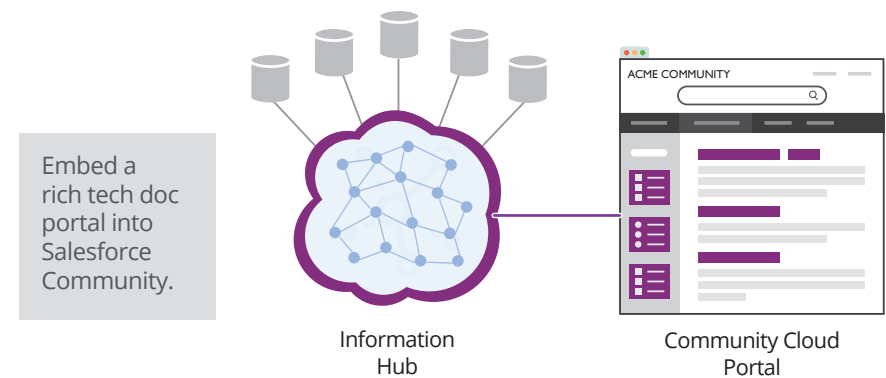


How to make content available in Salesforce

- 3 Create a **technical documentation portal** that offers a tailored and efficient searching and reading experience, including ways for users to interact with content (create bookmarks, alerts, comments, custom manuals, etc.). An efficient search engine is the cornerstone of such portal.



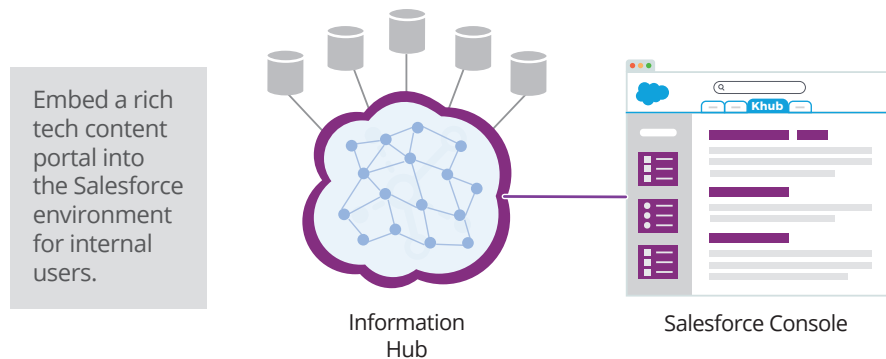
- 4 When using **Salesforce Community Cloud** to create customer or partner portals, then the Tech Doc Portal needs to be embedded in these community sites. This way, users benefit from an outstanding experience for finding the information they need through a single point of contact.



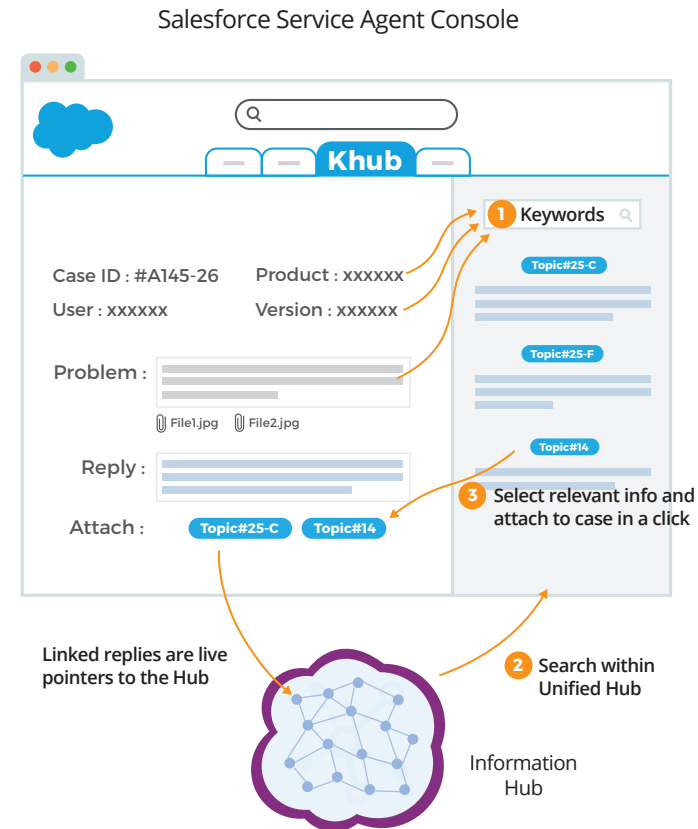
How to make content available in Salesforce

5 Empower agents by integrating the information hub in their **Salesforce work environment** to fuel the support process. This can be done in two ways.

- By integrating a Tech Content Portal in the Salesforce console, in a way similar to the one described above for Community Cloud. The portal would appear in a Salesforce tab as an application. Because they have different profiles, agents could access more content than standard users (for example, searching in past Cases).



- By integrating the information hub in the agent console. So that when handling a case, the agent can:
 - 1 Search in the unified hub,
 - 2 Find appropriate content regardless of its source, including Solutions or past Cases,
 - 3 Attach these elements to the case as part of the reply, without having to copy-paste or duplicate any content.



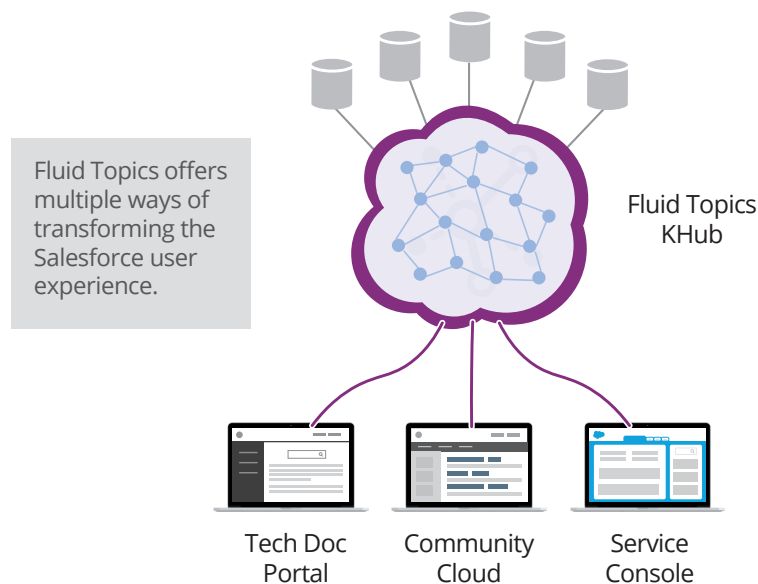
Transform the support process by creating live links to existing content.

The initial search in the hub can be automatically seeded using the keywords of the case description and some of its metadata as facets to immediately provide more targeted results.

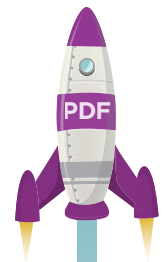
What Fluid Topics can do for you

Fluid Topics is a dynamic content delivery platform that solves all the challenges of making content available into Salesforce.

- It offers unique capabilities to combine and enrich any type of content from multiple sources, and create a unified hub of information.
- Fluid Topics includes a state-of-the-art semantic search engine optimized for technical content.
- It creates one or multiple Tech Content Portals, fully branded to any corporate look & feel, connected to any existing authentication back-end.
- Fluid Topics has ready-to-use Salesforce add-ons to integrate the Knowledge Hub into Community Cloud portals and Salesforce Service console.
- It captures and analyzes all user actions and provides advanced analytics so that one knows what people search and read.



Benefits of Fluid Topics are numerous: efficiency, agility, consistency, security, compliance.



ABOUT THIS GUIDE

Reshape Customer Support with Technical Content in Salesforce

This white paper explains the challenges of making technical content available through Salesforce. It also explains how to handle that content and interconnect it with Salesforce.

Table of contents

RESHAPE CUSTOMER SUPPORT WITH TECHNICAL CONTENT IN SALESFORCE

Handling documents with Salesforce.....	2
The Salesforce Knowledge Base (KB)	3
Challenges of loading content into the Salesforce KB	5
Searching for textual content in Salesforce.....	7
How to make content available in Salesforce	9
What Fluid Topics can do for you	12



Salesforce and the Salesforce logo are registered trademarks of Salesforce.com, Inc.

What’s behind Fluid Topics

Fluid Topics is a perfect name for a dynamic publishing product. “Fluid” conjures images of effortless motion, smooth and easy flow, and graceful simplicity in movement or execution. “Topics” evokes conversation, communication, learning, thinking, reasoning, education, and ideas. Together, **Fluid Topics** conveys a smooth action or movement of reaching for and shaping information, quickly and effortlessly without limit or barriers.

That is what we do.

Contact us

E-mail:
info@fluidtopics.com

Website:
www.fluidtopics.com

Blog:
www.fluidtopics.com/blog

Twitter:
[@fluidtopics](https://twitter.com/fluidtopics)

Reshape Customer Support

With Technical Content in Salesforce



www.fluidtopics.com
info@fluidtopics.com

Revision: March 2019
© 2019 Antidot, All rights reserved.

Fluid  Topics