Optimizely 12 Connector Documentation

Terminologies

Connector — the set of tools and configurations that automate localization flow between a CMS and a translation tool.

Gadget — a module that is deployed to CMS to automate the localization process with CMS.

Localization Hub — the software platform aimed to connect various CMS systems and different translation tools to facilitate and manage the localization process.

Source/Master version — the main language version that is used as a source for localization.

Target version — the localizable version of the source version.

Content item — is the content building block of the Optimizely page, block, or product.

Overview

The purpose of the iLangL Optimizely Connector is to facilitate the localization flow between Optimizely and a set of supported translation tools — Phrase (Memsource), memoQ Server, etc. Localization Hub is a middleware platform that transfers content between Optmizely and the translation tools.

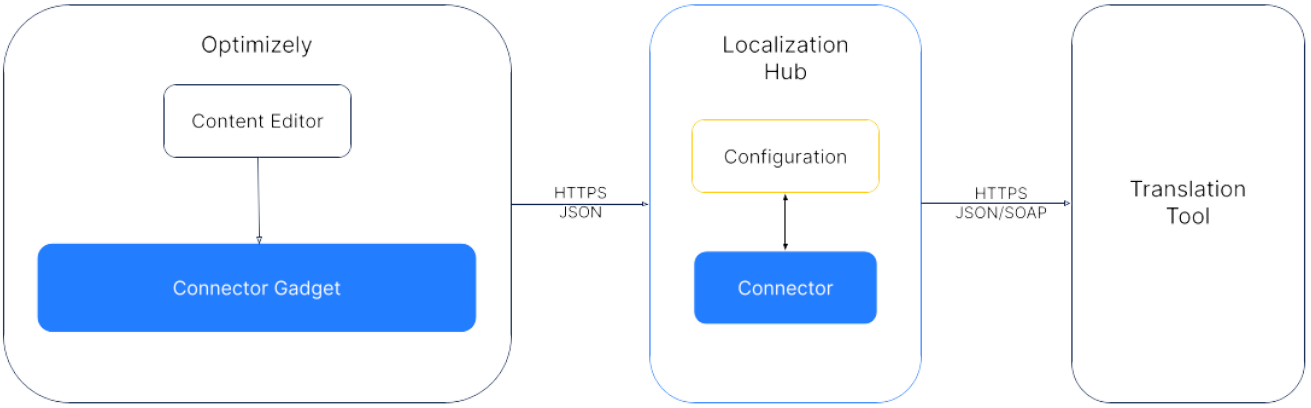

Figure 1. General Infrastructure

The connector gadget is accessible from Optimizely Content Editor and allows ordering selected items for translation and tracking the order statuses.

The connector gadget communicates with Localization Hub via REST API and exchanges data in JSON format.

Localization Hub handles orders, localization workflow and integration with translation tools. Communication with translation tools is performed via REST API (JSON).

Optimizely Gadget

DESCRIPTION

Optimizely gadget is a part of the Optimizely connector package which adds push technology to the Optimizely connector. It pushes content to the Localization Hub and pulls translated content from the Localization Hub back to Optimizely.

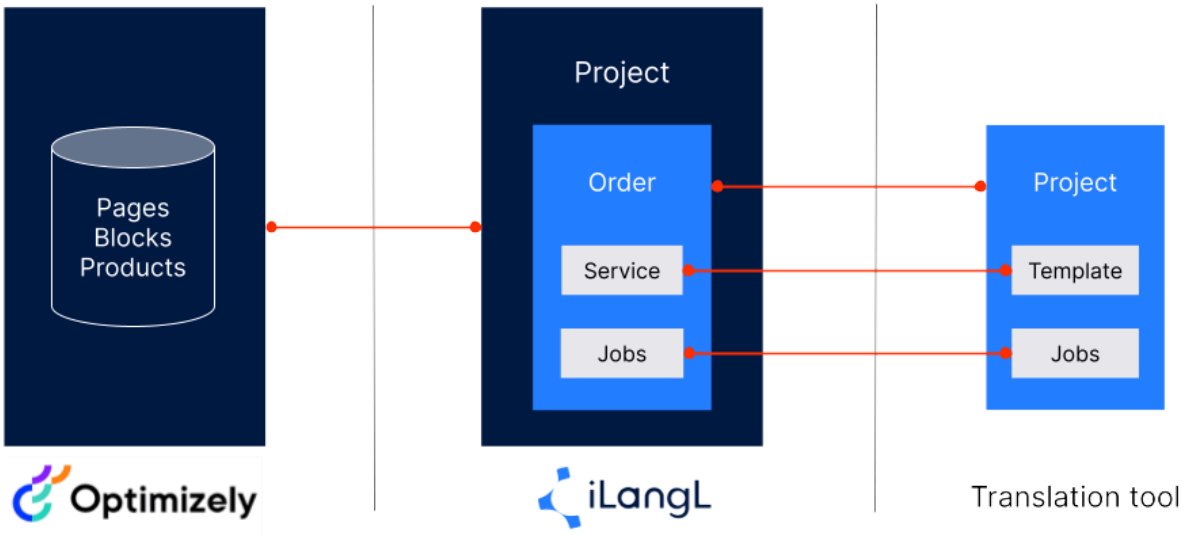

Figure 2 - Optimizely connector components

The main data operations between the Optimizely gadget and Localization Hub are:

- Ordering the content and uploading localizable content to the Localization Hub
- Checking the order status of items
- Fetching the translated content once it is completed
- Publishing page and all related items in one click

Scheduled jobs are used to check completed translation jobs and pull them back to Optimizely.

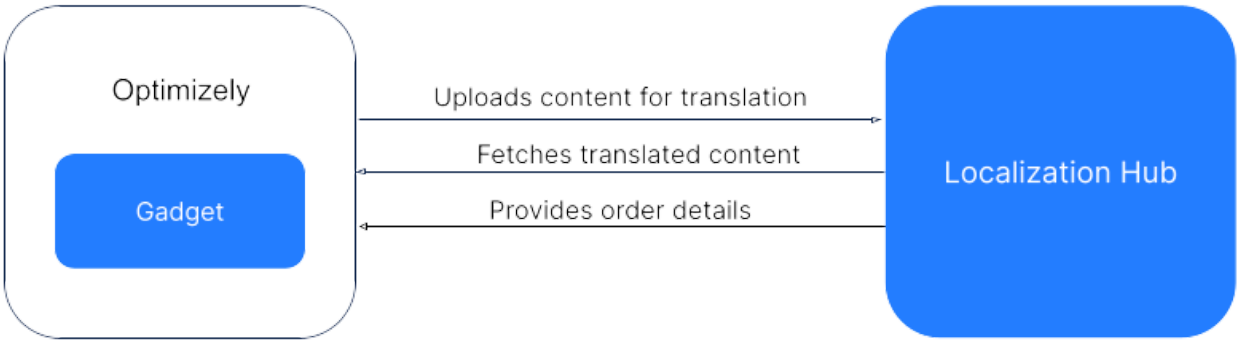

Figure 3 - Optimizely gadget data flow

Optimizely gadget supports the direct assignment of language services within Optimizely without having to leave Optimizely. It supports simple compilation of the Optimizely order through multi-selection of individual items. Users can export selected items with the latest versions of item components.

Optimizely language codes are properly aligned with translation tools in Localization Hub settings. Different source languages can be used to order content. For each source language in the Localization Hub, the client can set up a service and define available target languages and translation project templates that will help to automate translation project creation and auto-assignment of linguists, translation memories, termbases and other settings.

During the translation of Optimizely items, the Optimizely gadget shows the information that the current item is ordered and what the current order status is. Localization Hub remembers all ordered content and provides a view of the recent state of all ordered content. The system also tracks the versions of exported content and users can check what localizable versions are already outdated and reorder them.

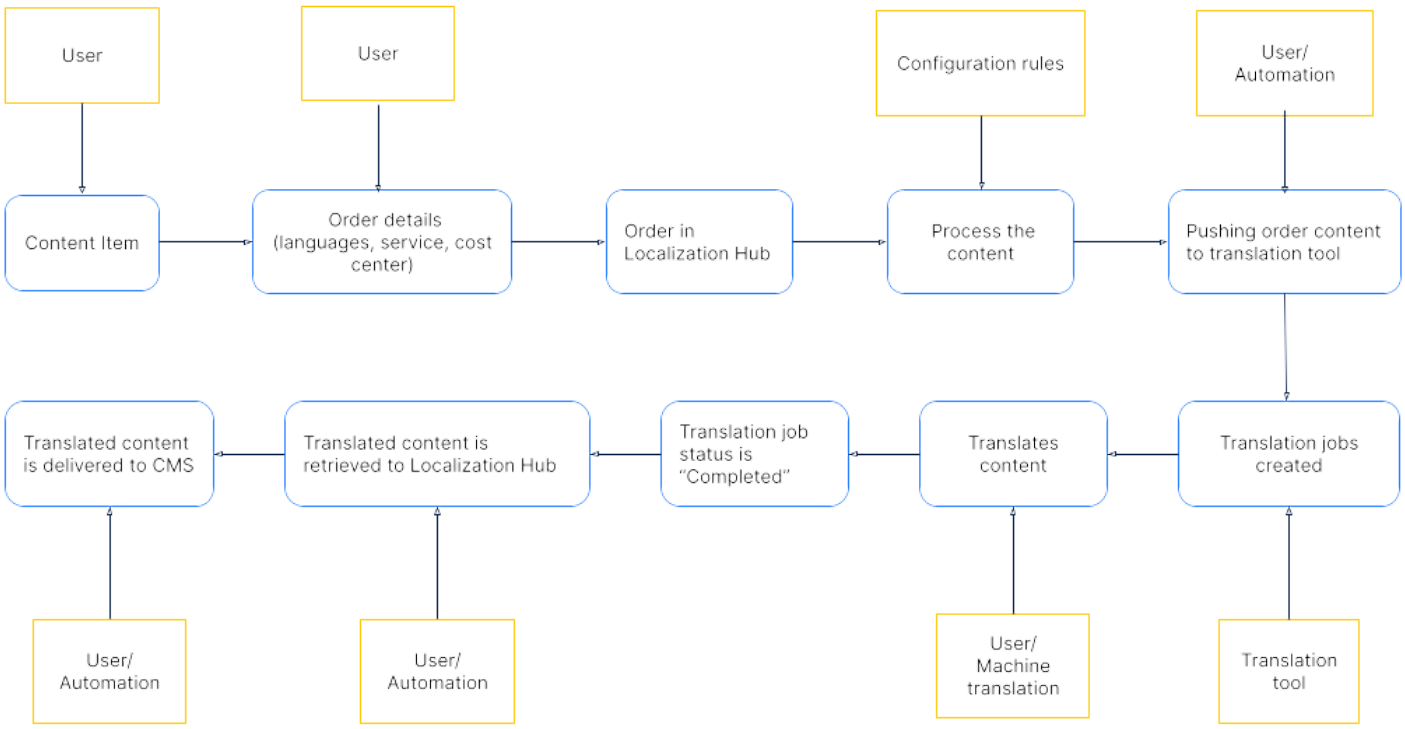

Figure 4 - Data flow diagram

SECURITY

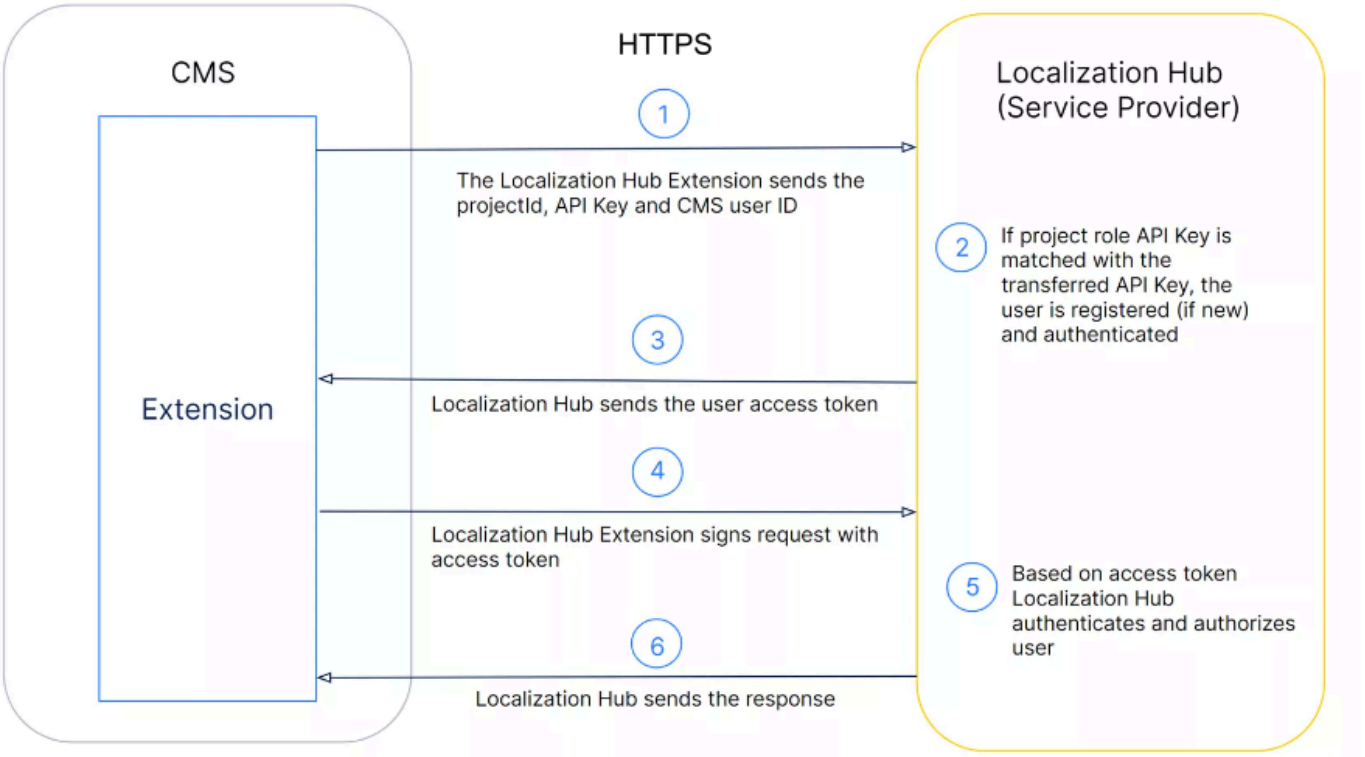

Figure 5 - Optimizely user authentication in Localization Hub

The Optimizely gadget authenticates Optimizely users in the Localization Hub by sending the Project ID, API key, and Optimizely User Name. Project ID and API key are configured in Localization Hub settings and help to authenticate and authorize all requests from gadget to Hub. Localization Hub auto-registers and authenticates users in the system by Optimizely User Name. The iLangL Localization Hub gadget allows sending orders to the Localization Hub.

INSTALLATION

If you use only CMS without Commerce, you need to install NuGet Package
iLangl.Extension.Optimizely.V12

If you use Optimizely CMS with Commerce, you need to use NuGet Package

iLangl.Extension.Optimizely.V12.Commerce

1. Install NuGet package.
2. Register iLangl.Extension.Optimizely.V12 as a ProtectedModule
3. In appsettings.json add the iLangl.Extension.Optimizely.V12 as a ProtectedModule:

```
   "EPiServer": {
  "CmsUI": {
    "ProtectedModule": {
      "Items": [
        {
          "Name": "iLangl.Extension.Optimizely.V12"
        }
      ]
    }
  },
...
}
```

4. Modify Startup.cs. Method ConfigureServices.

Add to services: Localization Extension AddLocalizationExtension()

In case of Commerce support, it should be: AddLocalizationExtensionWithCommerce()

Example:

```
services
    .AddCmsAspNetIdentity<ApplicationUser>()
    .AddCms()
    .AddAlloy()
    .AddAdminUserRegistration()
    .AddEmbeddedLocalization<Startup>()
    .AddLocalizationExtension();
```

For Development Environment, copy the iLangl.Extension.Optimizely.V12.zip to the Project to follow path \modules_protected\iLangl.Extension.Optimizely.V12\iLangl.Extension.Optimizely.V12.zip

Possible errors

CatalogContent type is not supported — when you select the product and try to order but get this message.

Resolution

Make sure that you install the right NuGet package — iLangl.Extension.Optimizely.V12.Commerce — and enabled the “Commerce” gadget.

ENVIRONMENT SPECIFICS

When the Optimizely databases need to be copied from the production environment to staging or backward, environment-specific items should not be overwritten. Anything that is environment-specific should be externalized into custom deployment patch files.

The environment-specific settings for the Optimizely gadget can be found in the DB view:

VW_iLangl.Extension.Optimizely.V12.Settings.SettingsModel in tblBigTable with StoreName equal ‘iLangl.Extension.Optimizely.V12.Settings.SettingsModel’

PREPARE OPTIMIZELY FOR LOCALIZATION

Before making the translation orders, make sure that all languages are enabled in Optimizely Settings: *Settings/Manage Website Languages.*

Once all languages are enabled in the localization gadget, open “Settings”, enter the Localization Hub URL, Workspace ID or create a new workspace. Press “Connect”. When you press the “Connect” button the first time, this operation creates a new workspace or connects to existing one in the Localization Hub. Find a step-by-step guide here.

After this, if you press “Sync with Localization Hub”, the operation updates the workspace and syncs the list of languages between Optimizely and Localization Hub. Learn more on how to sync Optimizely with Localization Hub here.

For the Commerce part, the Localization Hub will show all languages enabled in Optimizely Web. To successfully deliver the translated information about Commerce products, make sure that in the “Catalog” where the product is located corresponding target languages are enabled in the “Available languages” property. Remember to publish these changes for the “Catalog” to make them up-to-date.

TRANSLATION ORDER

Connector supports localization of such Optimizely items like Page, Block and Product. When you order the item, the connector extracts all reference items automatically and it’s possible to send all of them in one translation package.

Connector extracts the content from the fields that are marked as “Unique value per language” and supports localization of the following property types:

- String
- Long String
- LinkItem
- LinkCollection
- StringDictionary
- StringList
- Embedded Block
- XHTML string

Connector fetches the most recent master version. For example, if a user publishes a master version and creates a new draft version, the content from the draft version will be ordered.

TRANSLATION DELIVERY

When the translation is delivered to Optimizely, it's important to understand how the new version is created.

If there is no language version for the translated item, the connector creates a new language version and copies all required properties from the latest master version. After this, it updates the new version with the translated content.

If a language version already exists the connector creates a new language version and copies all required properties from the latest language version. For all required fields with empty values, the connector copies the values from the latest master version. After this, it updates the new version with translated content.

The new version is created as a Draft version that is required to be reviewed before publishing. Once all is reviewed, it’s possible to publish the page and related blocks with the standard Optimizely publish method or use the “Publish” functionality from the iLangL gadget that allows the batch publishing of the current item with all related items for all languages.

For the “LinkCollection” property type, the item in the Collection is identified by its position in the Collection. If there is no item in the target collection at the specific position, the connector copies the item from the master version. If the position of items was changed in the master version, it’s required to sync this change with target versions. Otherwise, the translated link title and name might be delivered to the wrong target link item.

Localization Hub

DESCRIPTION

iLangL Localization Hub is a middleware that connects different content management systems and translation tools in one place.

Optimizely connector is a mix of integrated Optimizely gadget and Localization Hub — the portal where localization flow can be managed for Optimizely and for other localization platforms.

The navigation from the Optimizely gadget to Localization Hub does not require Optimizely users to log in to Localization Hub (see Fig.5).

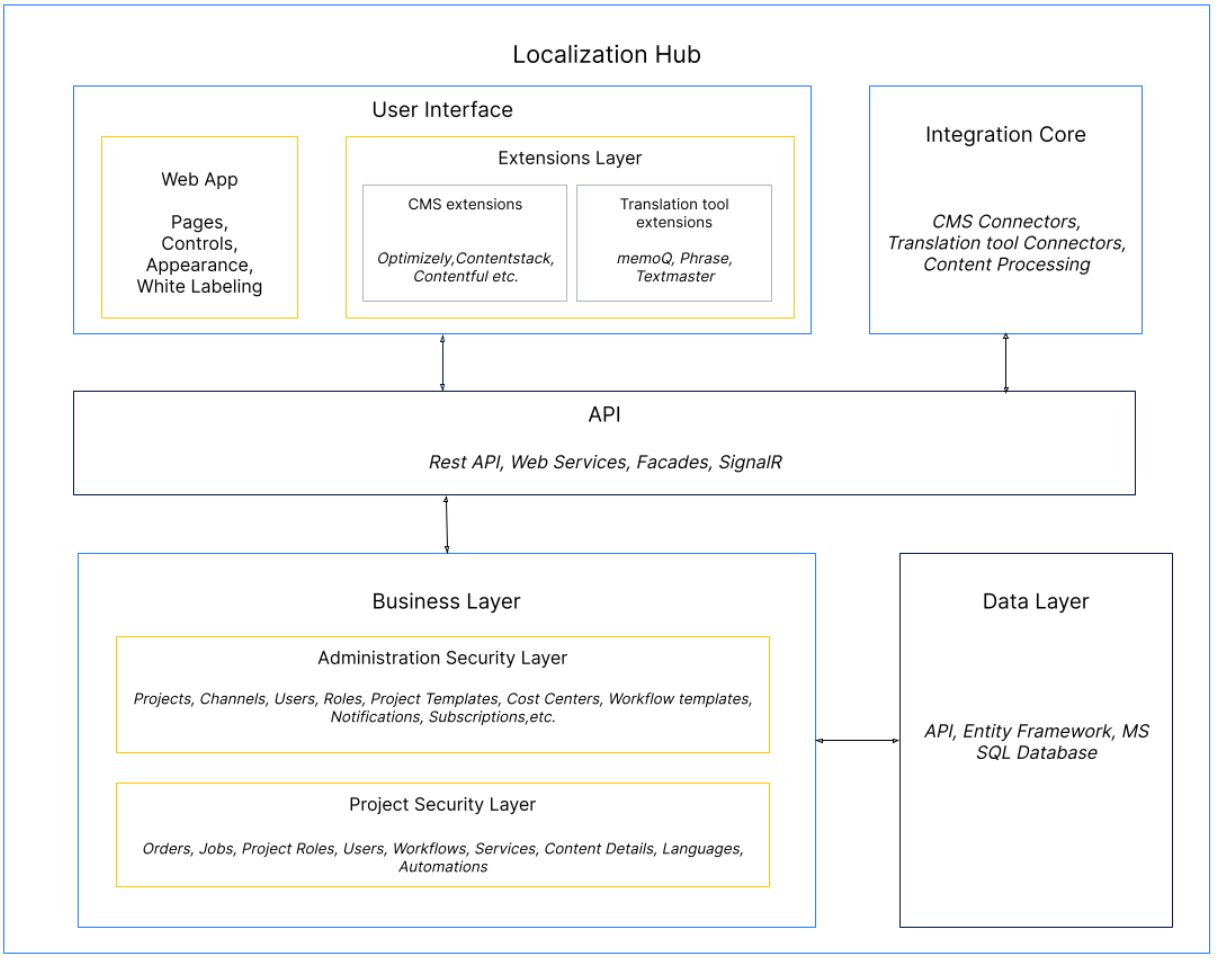

Figure 6 - Localization Hub Infrastructure diagram

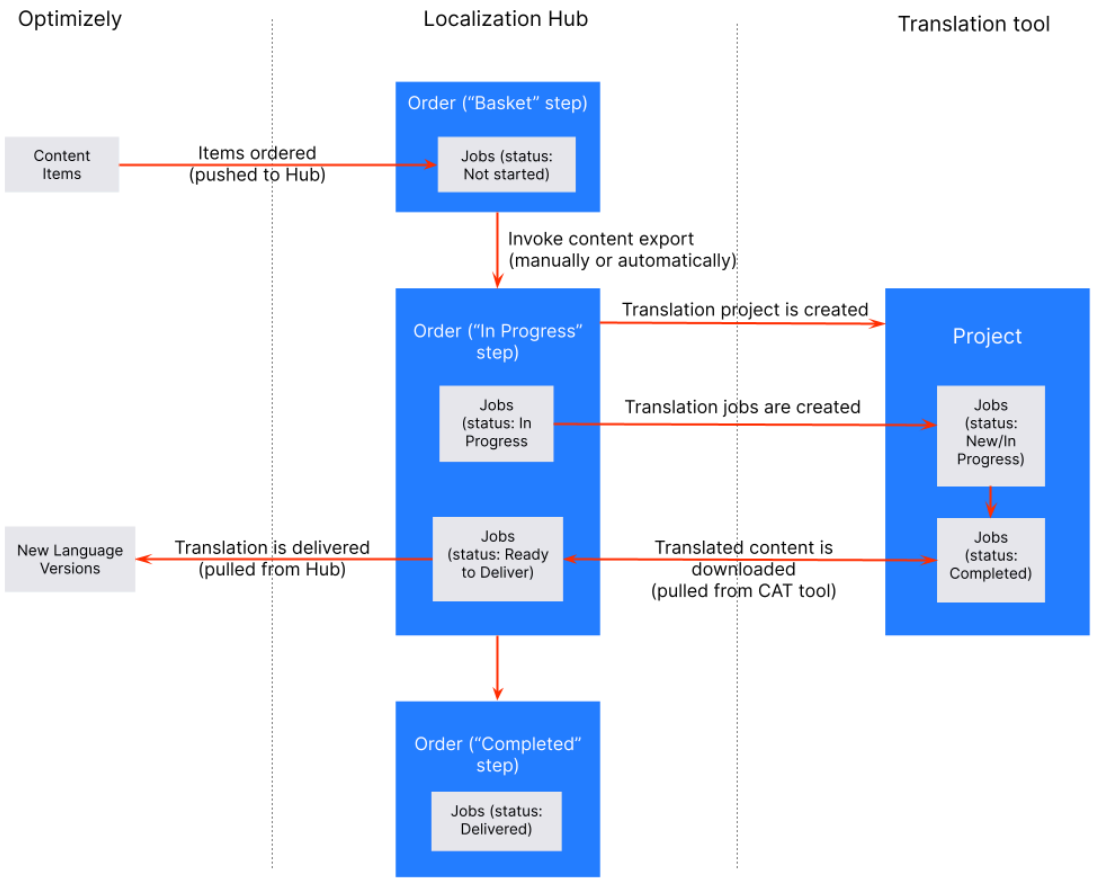

Figure 7 - Ordering process

Security

Localization Hub supports role-based security. Permissions are assigned to roles and roles are assigned to users.

Localization Hub supports two levels of role-based security:

The first level defines the permissions at the application level (Global Roles).

The second level defines the permissions at the project level (Project Roles).

Each project has its own security context.

iLangL offers real-time protection for your data. To assure data security, iLangL has implemented the following measures:

- Server. We use Hetzner Data Center and our primary servers are located in Germany. To read more about Hetzner, please follow the link: https://www.hetzner.de/cy/hosting/unternehmen/rechenzentrum
- Data security. All communication through iLangL channels is performed securely through the encrypted TLS 1.2 connection using AES_128_GCM and RSA(2048) as the key exchange mechanism.
- Backups. We perform daily backups of all data.
- Logs. We are able to go back historically in our system and application logs to forensically identify causes of the possible breach in both production and corporate IT. Logs are available for the last 3 months.
- Incident Response Plan. iLangL has an Incident Response Plan with designated staff allocated to a response team when required.
- RBAC. Our application supports RBAC with all separation roles.
- Production administrators protection. We use duo as an MFA application, formal privileged account management program, and access via security bastions.
- Vulnerability Reporting Program. We have VRP operated with Nessus https://www.tenable.com/.
- System updates. Our systems are fully patched and application dependencies are up to date. Every first week of the month we check updates for servers.
- User sessions. We manage user sessions via tokens and cookies securely.
- Confidentiality. All data in Localization Hub is protected and absolutely confidential. Access to such data is only limited to authorized users.