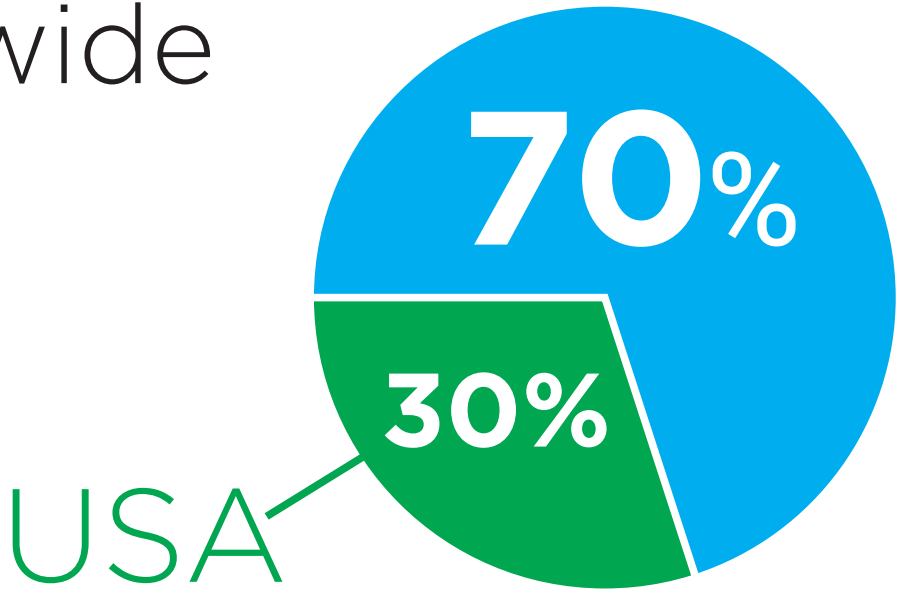
Continuous Localization at Evernote

using Open Source

150,000,000
Users Worldwide



15,000 words/month

Software

Android, iOS
Mac OS
Windows, Web

Marketing

Website
Google Docs
Zendesk KB

35 Projects
(Components)

25
languages

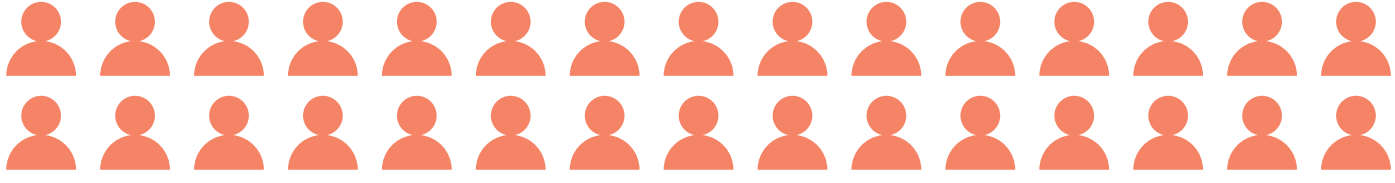
Localization Team:

Director of Localization

Localization Manager

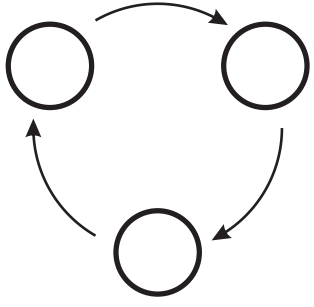
Software Engineer

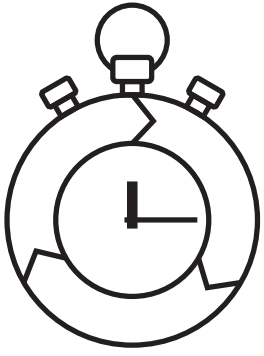
Translation Team (30 people):

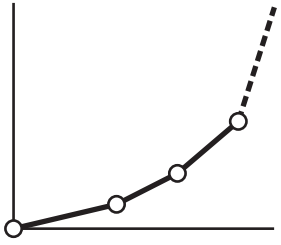


Continuous Localization

Continuous Localization

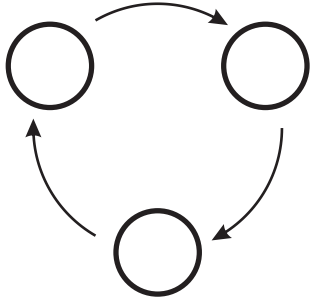


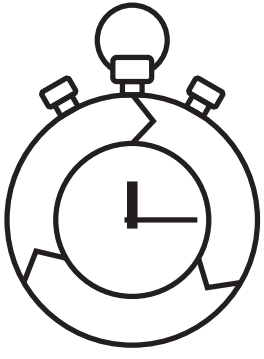


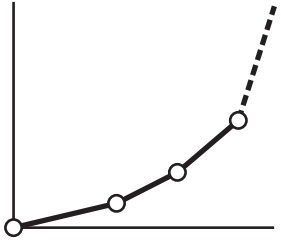


Simplifies Workflow

Continuous Localization

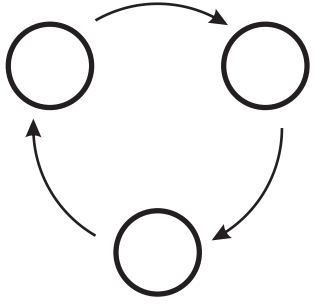


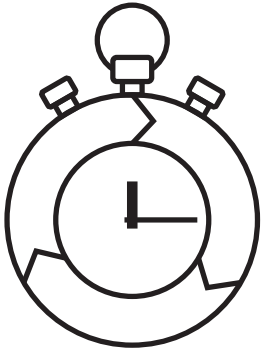


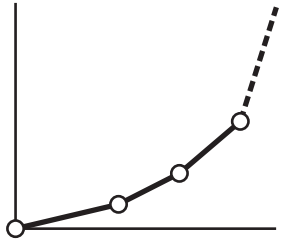


Improves Agility & Quality

Continuous Localization

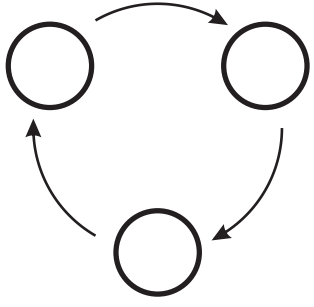


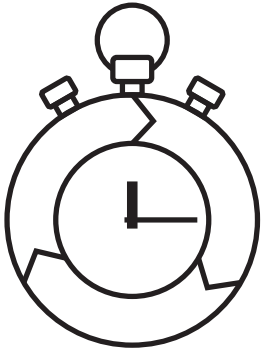


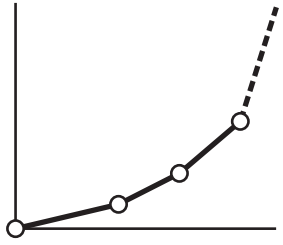


Ensures Scalability

Continuous Localization

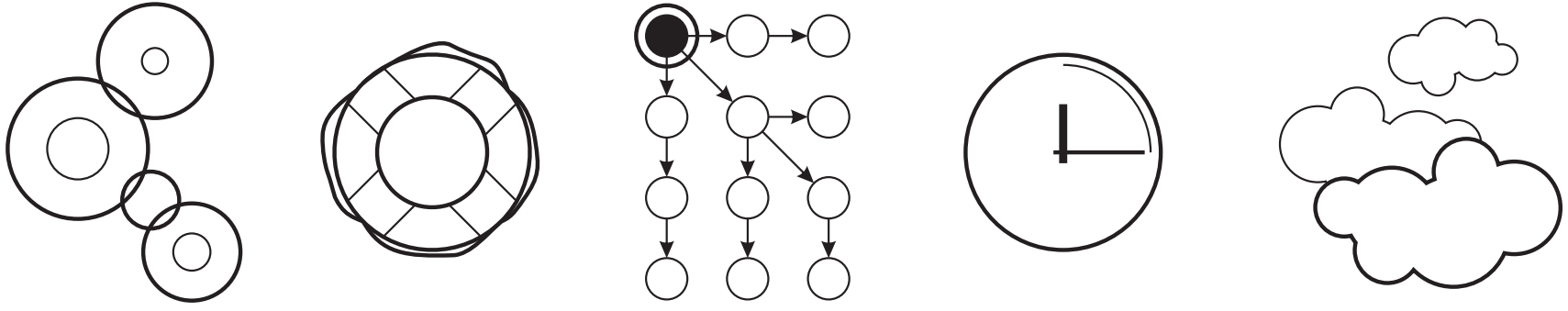






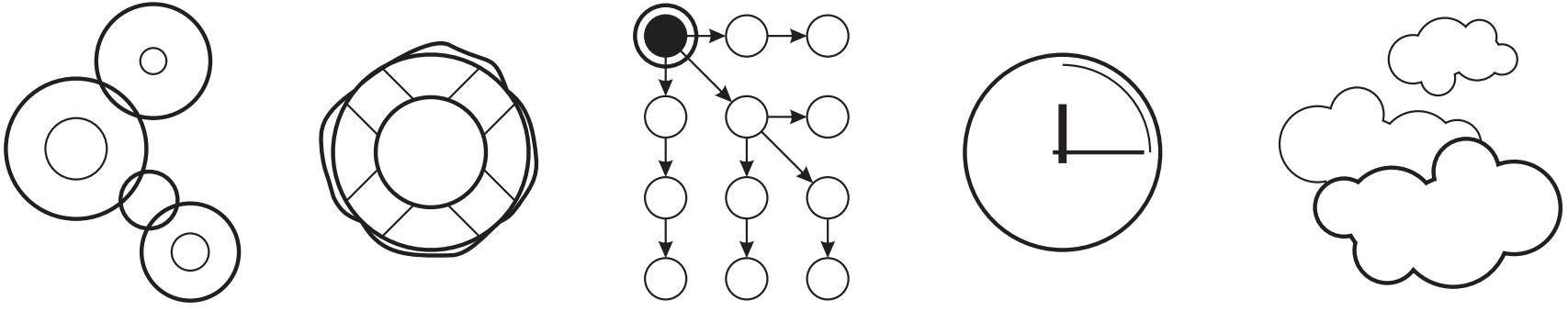
Saves Money & Resources

Our Localization System



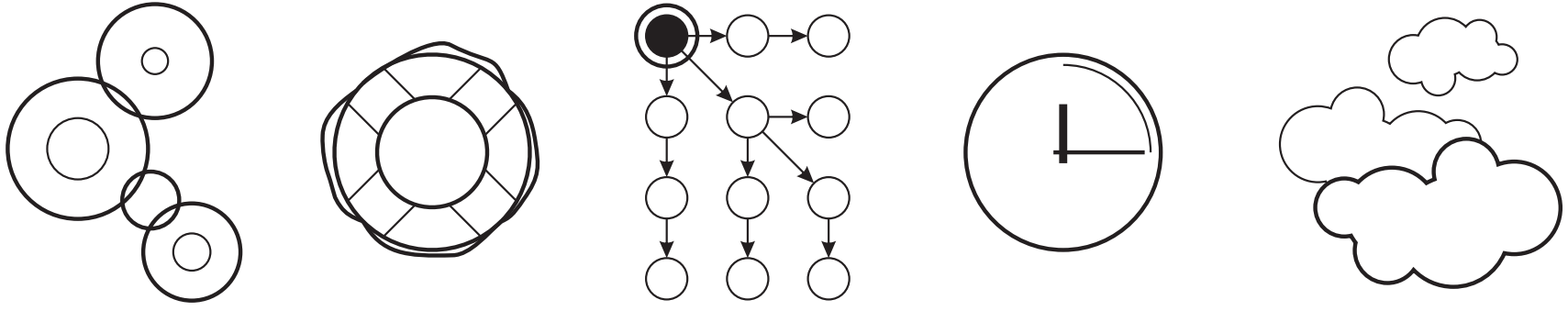
Fully Automated

Our Localization System



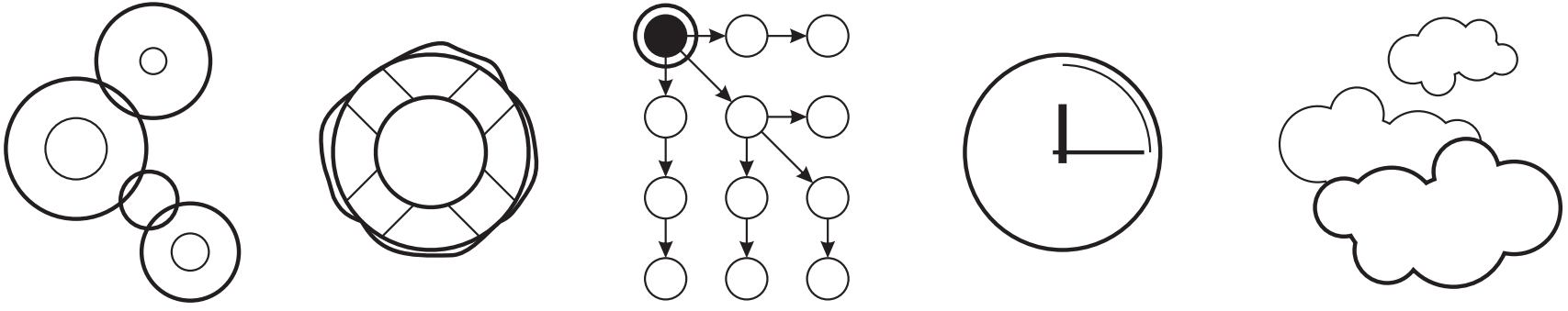
Localization Manager Connects Engineers & Translators

Our Localization System



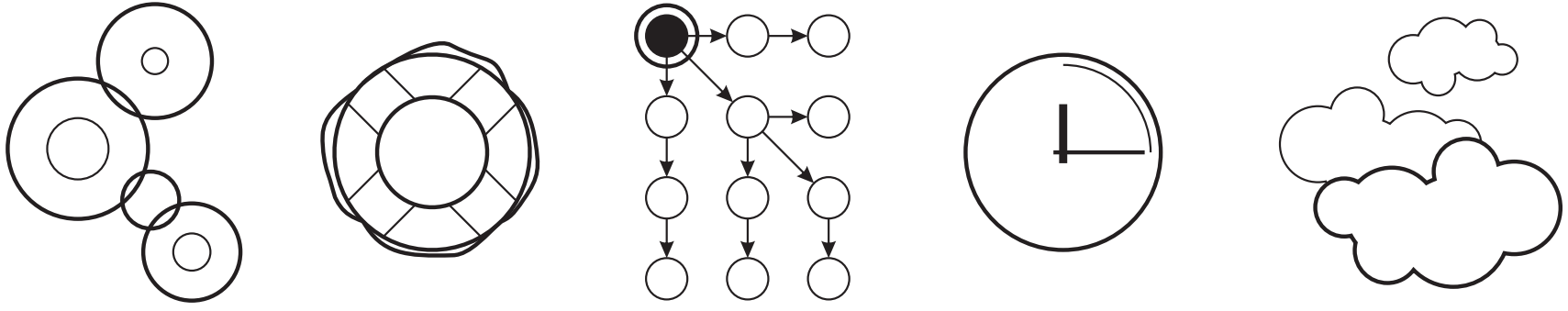
Engineers Work Only
with English Resources

Our Localization System



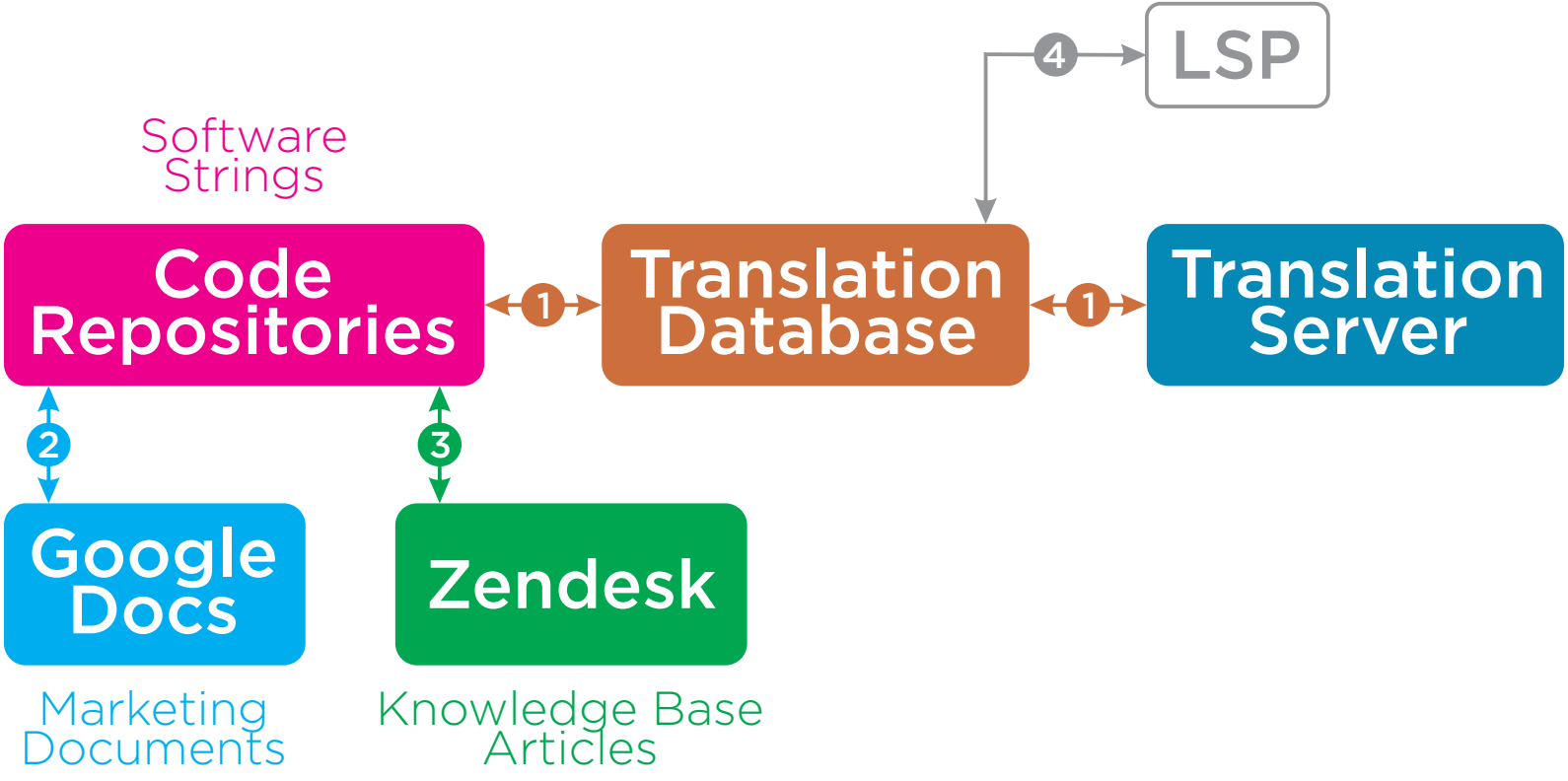
Cycle Takes ~15-20 Minutes

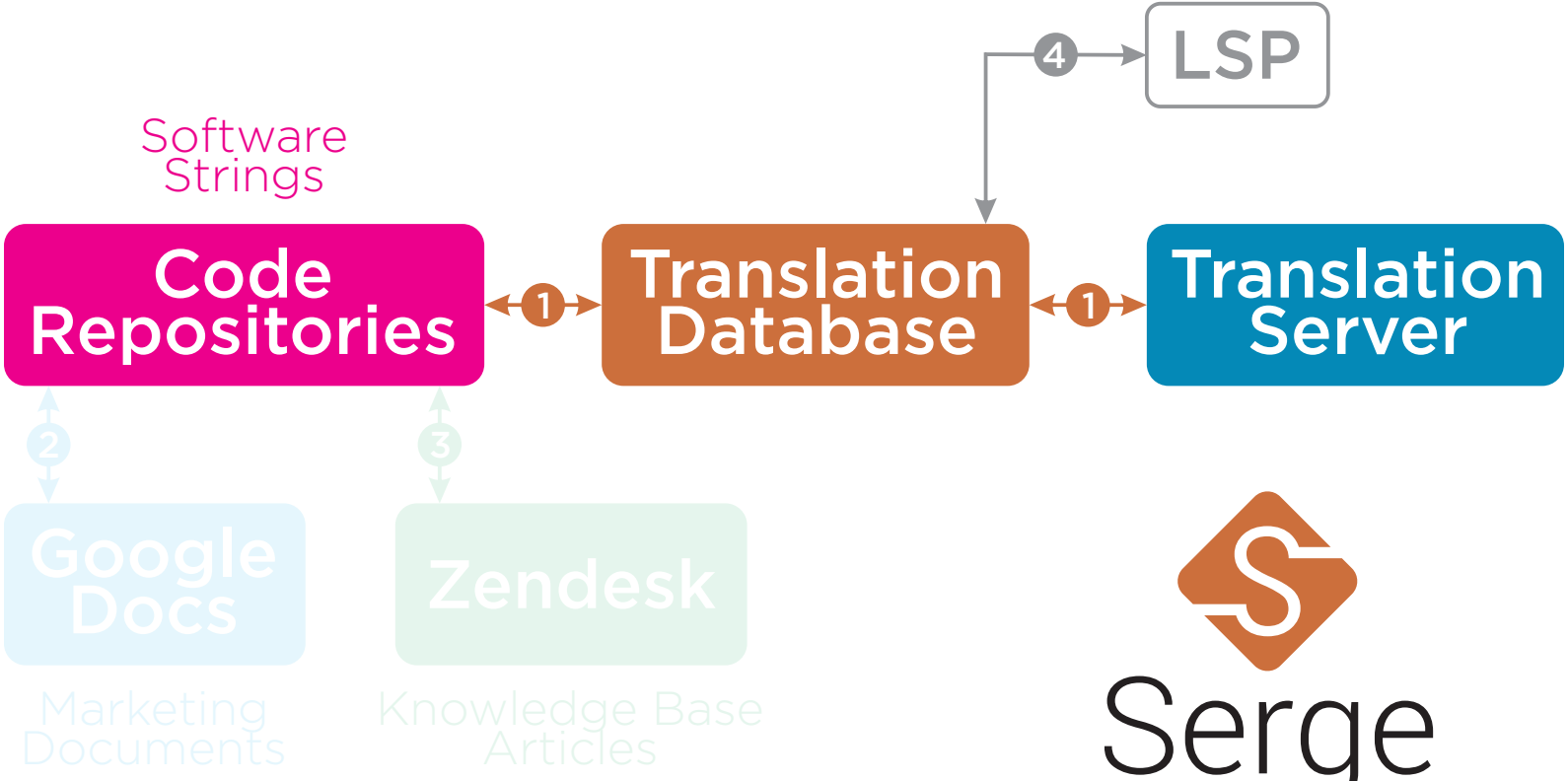
Our Localization System



Translators work Online

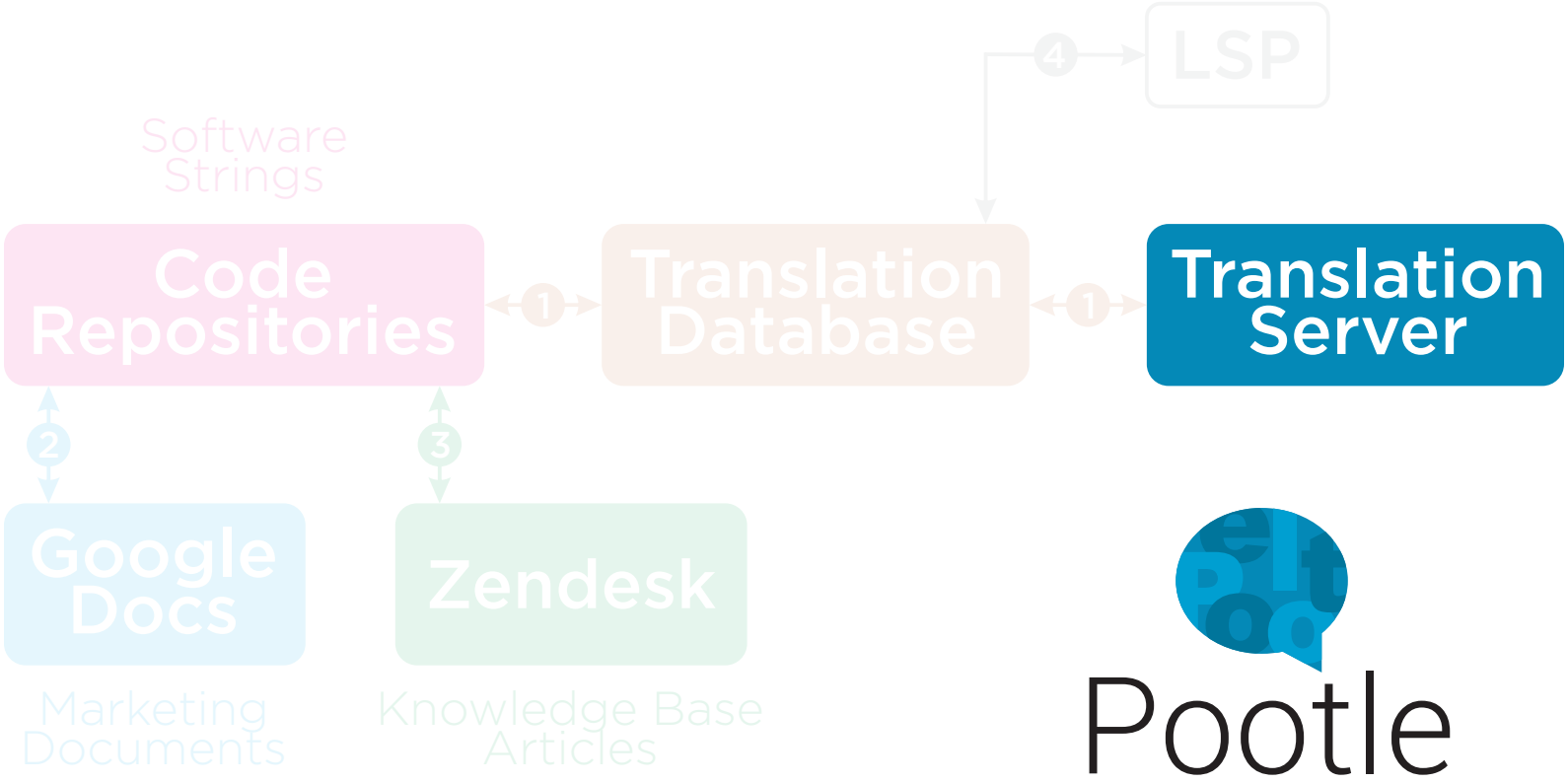
Key Components


LSP
4
Software
Strings
Code
Repositories
1
Translation
Database
1
Translation
Server
2
3
Google
Docs
Zendesk
Marketing
Documents
Knowledge Base
Articles



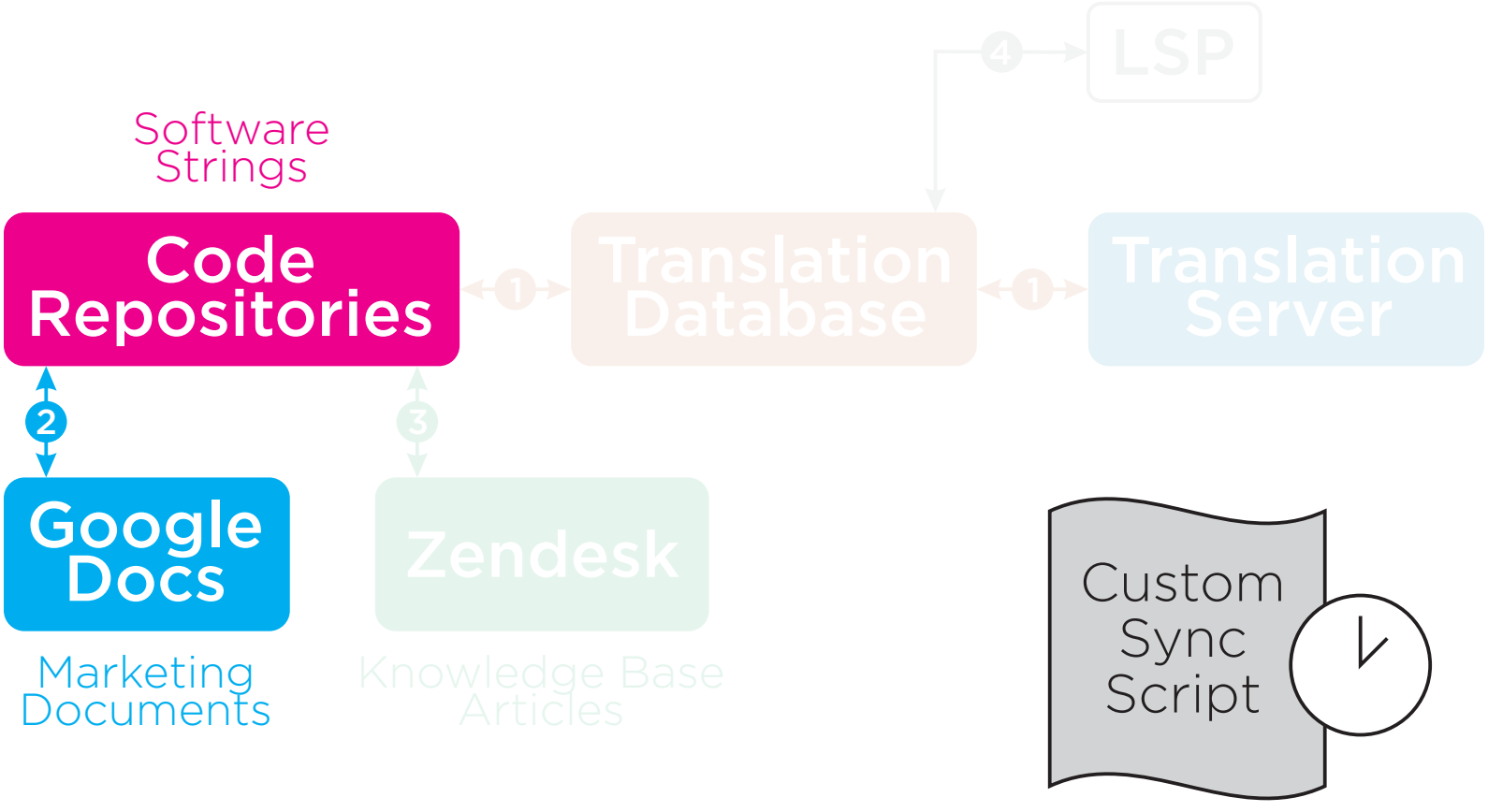
Serge

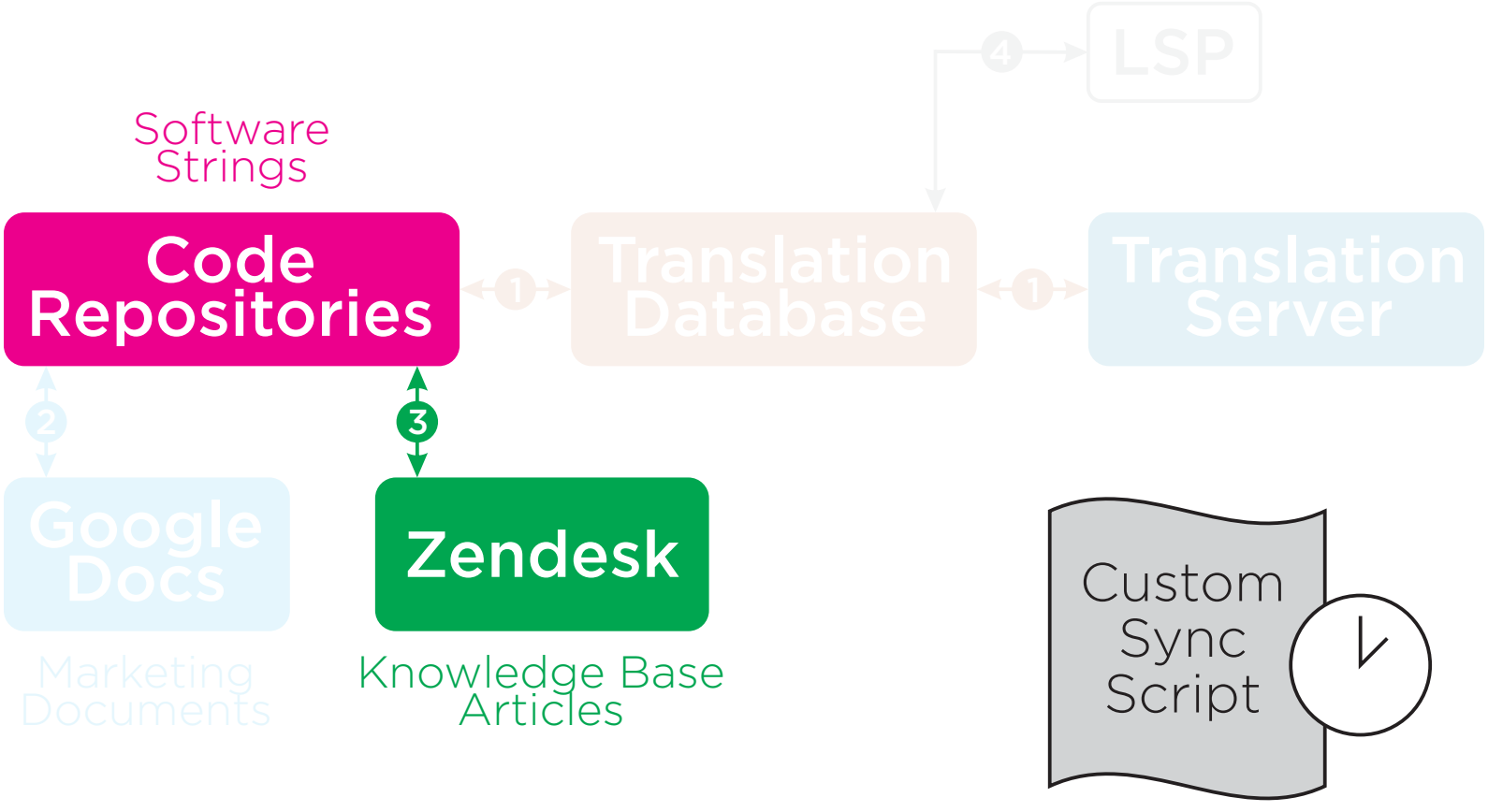
String Extraction and Resource Generation Engine



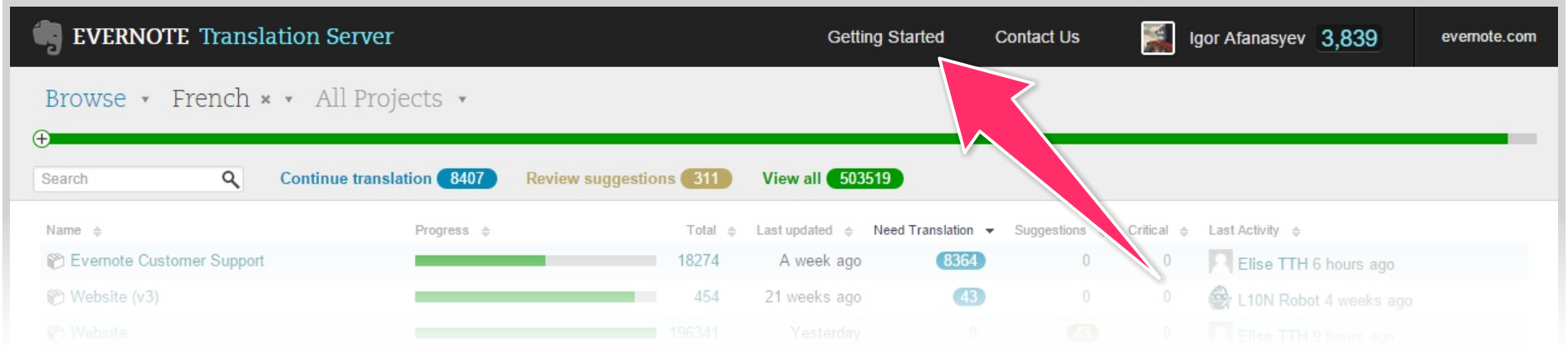
Pootle

PO-based Online
Translation/Localization Engine

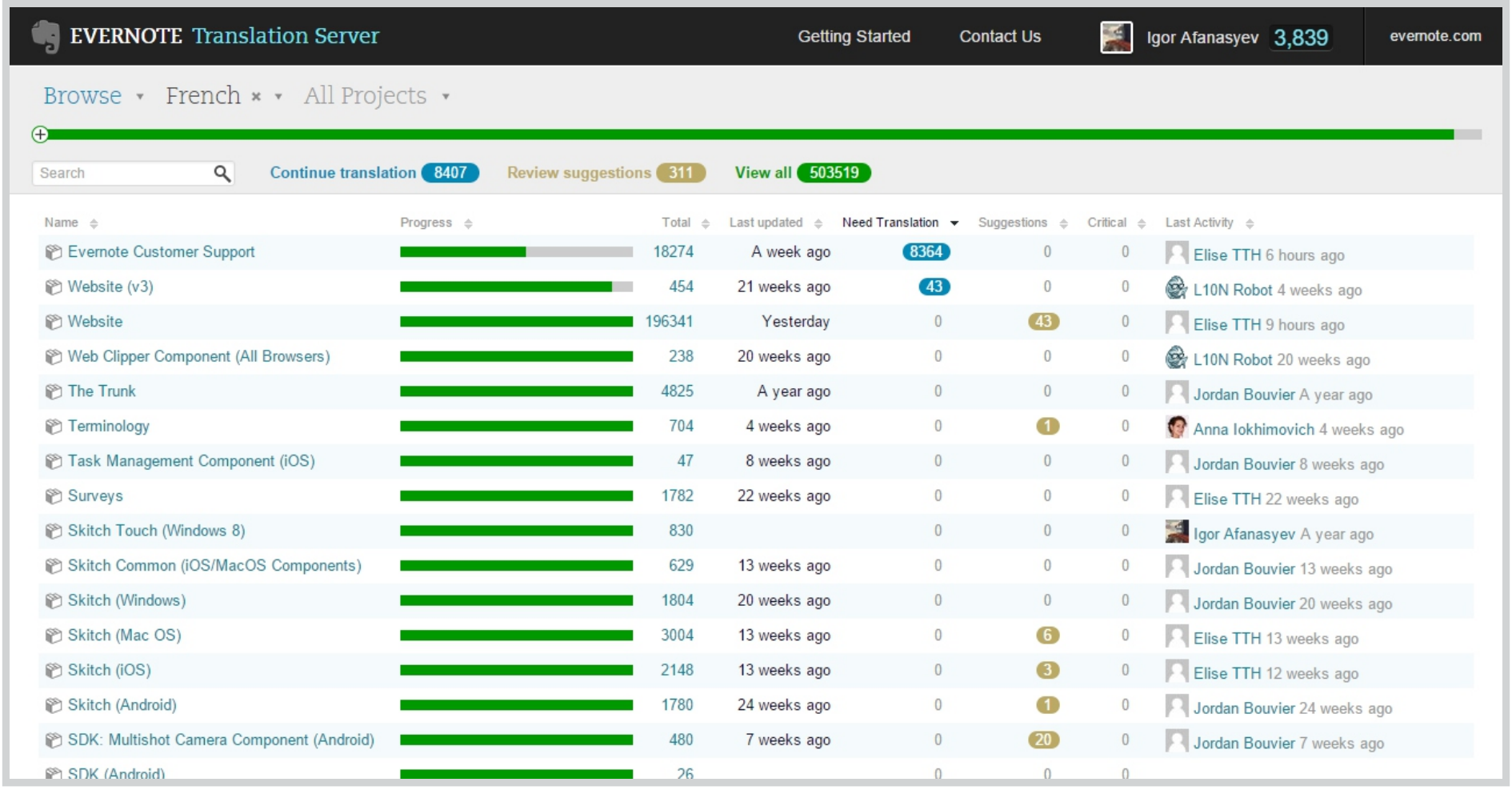

Software Strings
Code Repositories
1
Translation Database
1
Translation Server
4
LSP
2
3
Google Docs
Marketing Documents
Zendesk
Knowledge Base Articles
Custom Sync Script

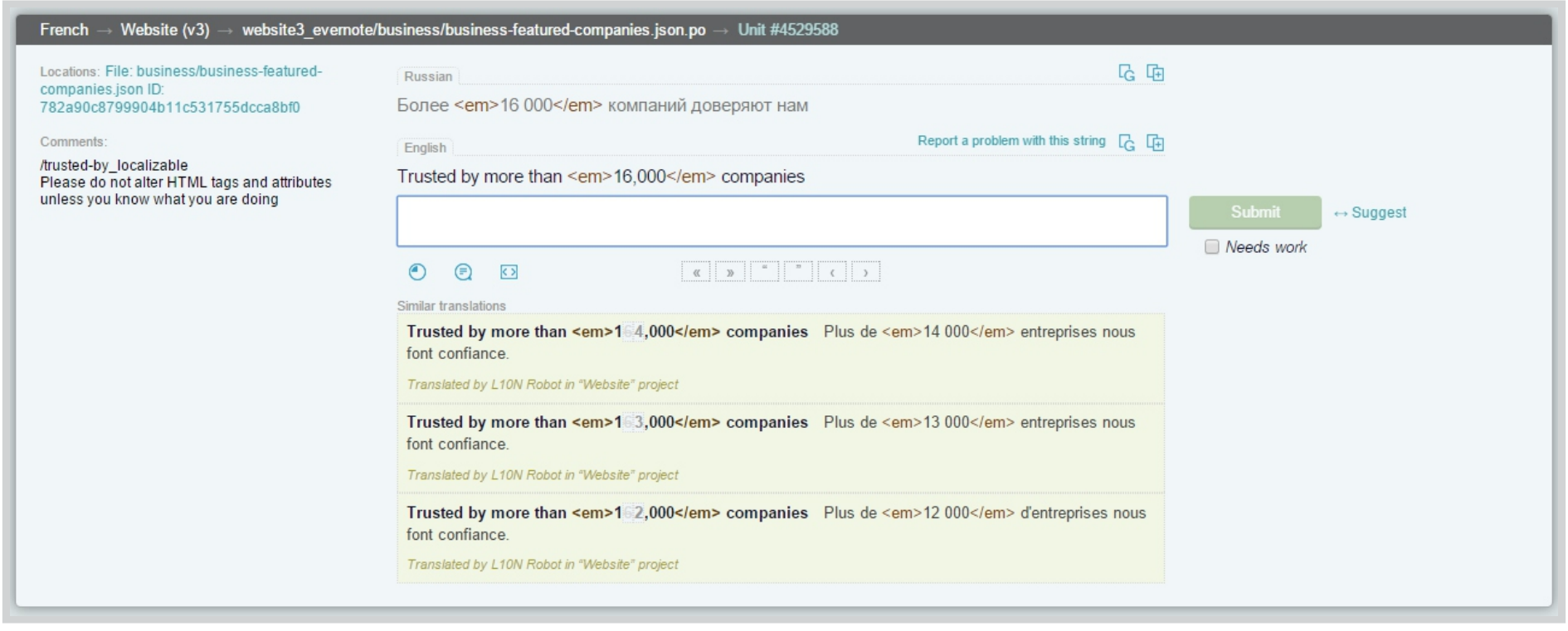

LSP
4
Software
Strings
Code
Repositories
1
Translation
Database
1
Translation
Server
2
3
Google
Docs
Zendesk
Marketing
Documents
Knowledge Base
Articles
Custom
Sync
Script

Pootle



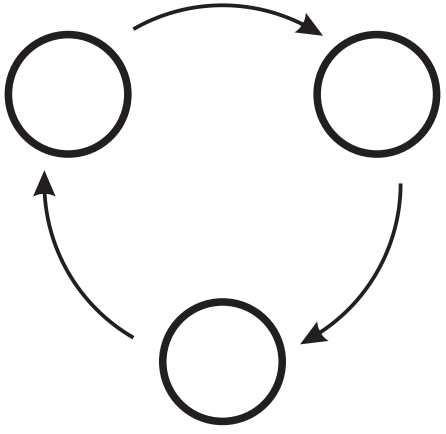
translate.evernote.com


EVERNOTE Translation Server
Getting Started
Contact Us
Igor Afanasyev 3,839
evernote.com
Browse
French ×
All Projects
Search
Continue translation 8407
Review suggestions 311
View all 503519
Name
Progress
Total
Last updated
Need Translation
Suggestions
Critical
Last Activity
Evernote Customer Support 18274 A week ago 8364 0 0 Elise TTH 6 hours ago
Website (v3) 454 21 weeks ago 43 0 0 L10N Robot 4 weeks ago
Website 196341 Yesterday 0 43 0 Elise TTH 9 hours ago
Web Clipper Component (All Browsers) 238 20 weeks ago 0 0 0 L10N Robot 20 weeks ago
The Trunk 4825 A year ago 0 0 0 Jordan Bouvier A year ago
Terminology 704 4 weeks ago 0 1 0 Anna Iokhimovich 4 weeks ago
Task Management Component (iOS) 47 8 weeks ago 0 0 0 Jordan Bouvier 8 weeks ago
Surveys 1782 22 weeks ago 0 0 0 Elise TTH 22 weeks ago
Skitch Touch (Windows 8) 830 0 0 0 Igor Afanasyev A year ago
Skitch Common (iOS/MacOS Components) 629 13 weeks ago 0 0 0 Jordan Bouvier 13 weeks ago
Skitch (Windows) 1804 20 weeks ago 0 0 0 Jordan Bouvier 20 weeks ago
Skitch (Mac OS) 3004 13 weeks ago 0 6 0 Elise TTH 13 weeks ago
Skitch (iOS) 2148 13 weeks ago 0 3 0 Elise TTH 12 weeks ago
Skitch (Android) 1780 24 weeks ago 0 1 0 Jordan Bouvier 24 weeks ago
SDK: Multishot Camera Component (Android) 480 7 weeks ago 0 20 0 Jordan Bouvier 7 weeks ago
SDK (Android) 26 0 0 0


French → Website (v3) → website3_evernote/business/business-featured-companies.json.po → Unit #4529588
Locations: File: business/business-featured-companies.json ID: 782a90c8799904b11c531755dcca8bf0
Comments:
/trusted-by_localizable
Please do not alter HTML tags and attributes unless you know what you are doing
Russian
Более 16 000 компаний доверяют нам
English
Report a problem with this string
Trusted by more than 16,000 companies
Submit
↔ Suggest
Needs work
Similar translations
Trusted by more than 14,000 companies
Plus de 14 000 entreprises nous font confiance.
Translated by L10N Robot in "Website" project
Trusted by more than 13,000 companies
Plus de 13 000 entreprises nous font confiance.
Translated by L10N Robot in "Website" project
Trusted by more than 12,000 companies
Plus de 12 000 d'entreprises nous font confiance.
Translated by L10N Robot in "Website" project

Serge

```
$ serge localize android.serge
```



We write configuration files (once)
and put them on our localization server

Serge runs on localization server
in a continuous loop

Serge generates/updates
translation interchange files and
pushes them to our translation server

Serge generates and pushes localized
resource files to our Git repos;
these commits trigger CI builds

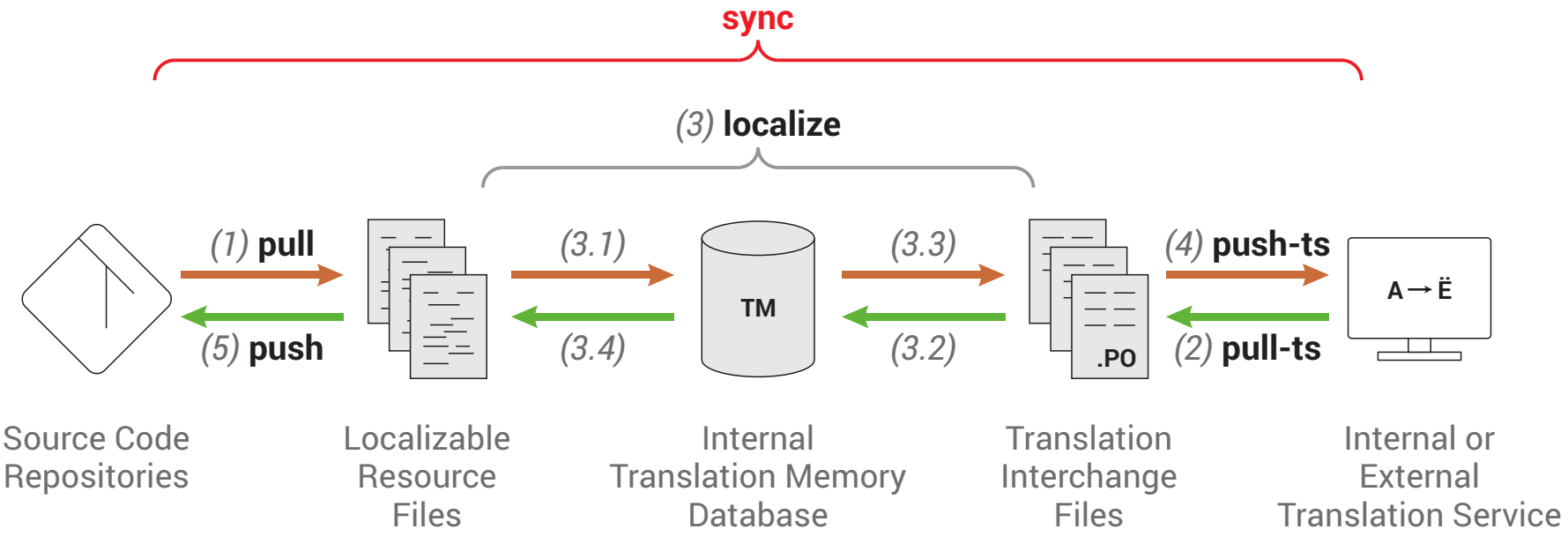
Serge Configuration Files

Configuration files contain all the information on what files to process, how to process them, and how to generate localized files

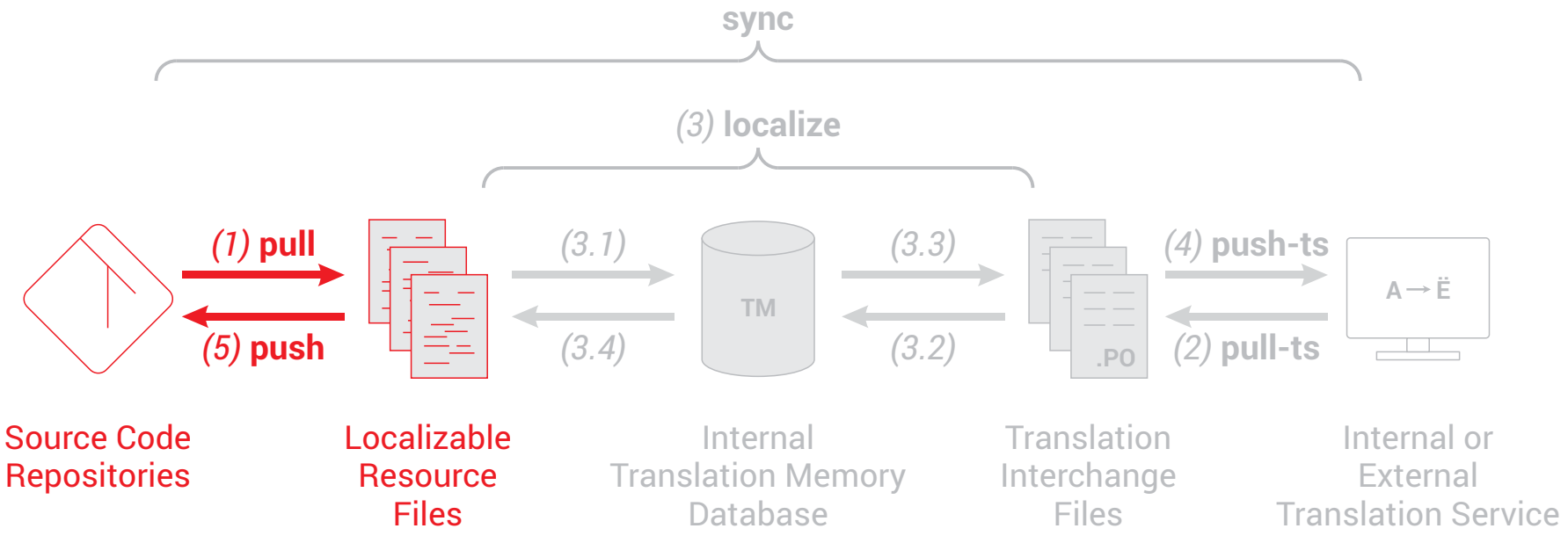
They have a clean syntax and allow inheritance (reusing common settings)

```
sync
{
    ts
    {
        # Translation Service parameters
        # ...
    }

    vcs
    {
        # Version Control System parameters
        # ...
    }
}

jobs
{
    {
        # Job 1 parameters
        # ...
    }

    {
        # Job 2 parameters
        # ...
    }
}
```

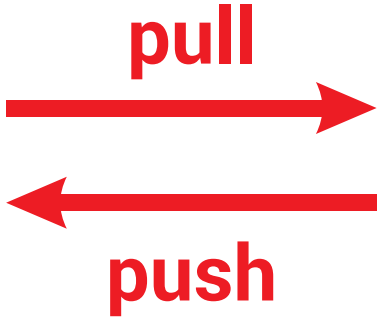
```
$ serge sync myproject.serge
```



```
$ serge pull myproject.serge
$ serge push myproject.serge
```



Integration with Version Control Systems



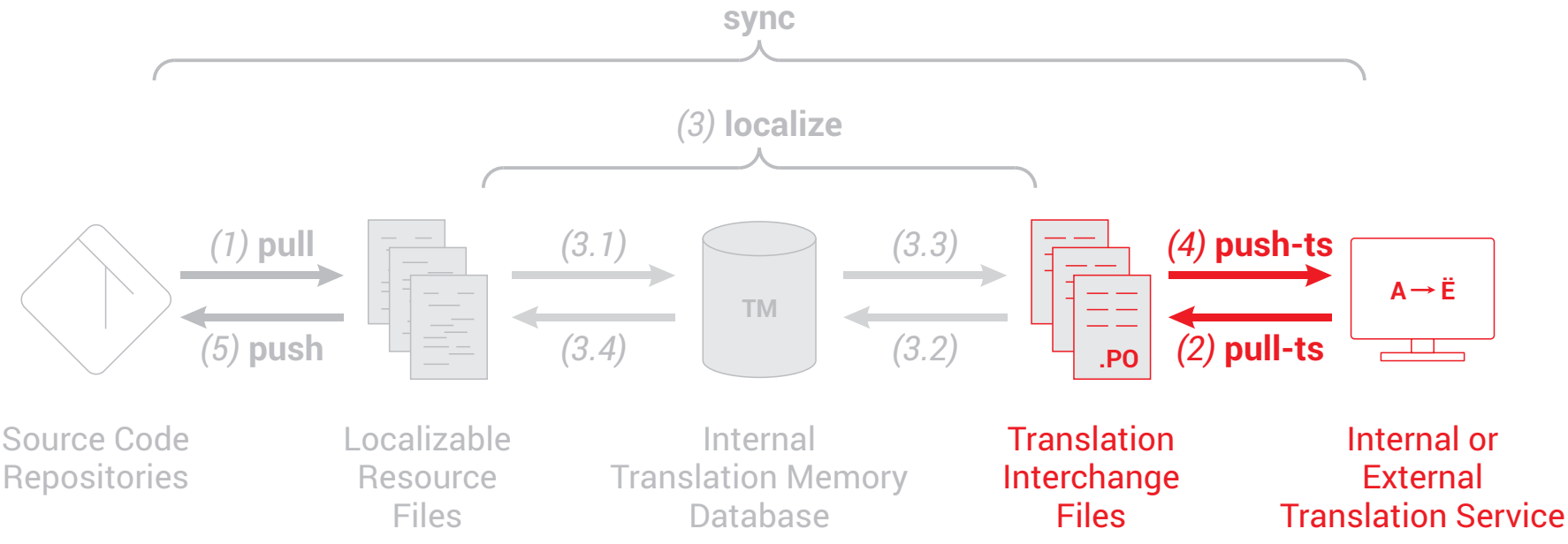
Plugins: Git, Gerrit, Subversion, Mercurial

Isolation from different VCS logic

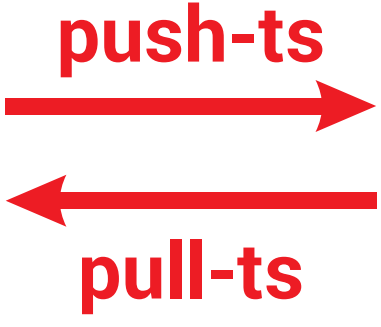
Multiple repositories as one logical internationalization project

Automatic conflict resolution

```
$ serge push-ts myproject.serge
$ serge pull-ts myproject.serge
```



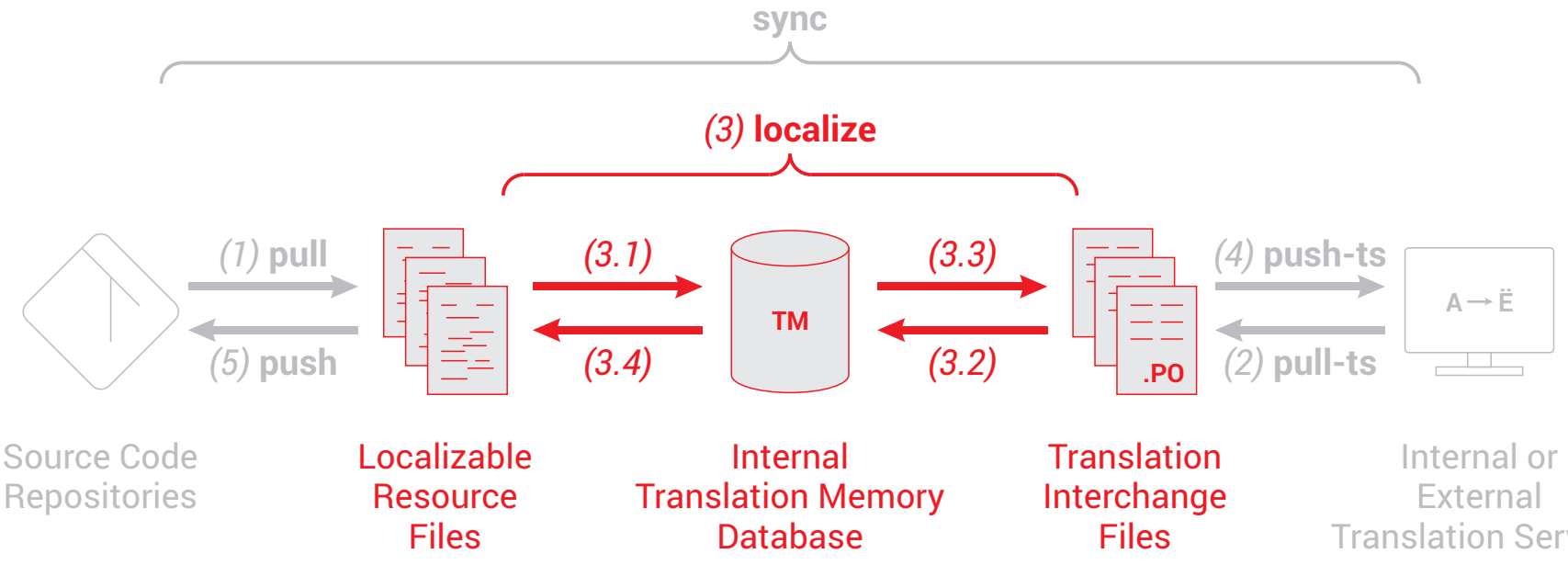
Integration with Translation Services



Plugins: Pootle

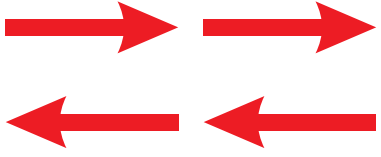
Alternative: publish .po files
for offline translation,
integrate with
localization vendor's API

```
$ serge localize myproject.serge
```



Core Functionality

localize



Parsing and generating resource files

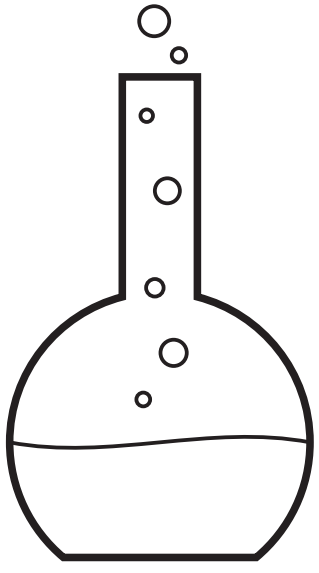
Generating and parsing
translation interchange files (.PO)

Many built-in configuration options
(what and how to process,
how to use TM)

Advanced control plugins

20+ Parsers Out of the Box

Android .strings, Chrome Extension .json files, .DTD, Perl/PHP/Ruby hashes, JavaScript (dictionary objects), generic JSON files, INI-like key-value files, PHP/XHTML, iOS .plist files, .pot files, Java .properties, Windows .rc files, .Net framework .resx files, Blackberry .rrc files, MacOS/iOS .strings files, QT Linguist .ts files, WiX installer .wxl files, XLIFF 1.x/2.x files, XML files (generic, Android-specific, InDesign-specific), generic and Rails YAML files, .master files (arbitrary plaintext markup), generic regex-based parser



Advanced Control Plugins

`process_if` Plugin

Do not process certain source files
(**`can_process_source_file`** callback phase)

Do not extract certain strings for translation
(**`can_extract`** callback phase)

Do not generate/save certain localized files
(**`can_generate_localized_file`** callback phase)

Criteria: file path, file content, string content, string comment, target language, previously set flag

process_if recipe: translate only files that have special “LOCALIZABLE” flag

```
# job definition
{
    # ...

    callback_plugins
    {
        {
            plugin                          process_if
            phase                           can_process_source_file

            data
            {
                if
                {
                    content_matches         \bLOCALIZABLE\b

                    then
                    {
                        return              YES
                    }
                }
            }
        }
    }
}
```

process_if recipe: do not translate a string if it starts with “!!”

```
# job definition
{
    # ...

    callback_plugins
    {
        {
            plugin                          process_if
            phase                           can_extract

            data
            {
                if
                {
                    content_doesnt_match    ^!!

                    then
                    {
                        return              YES
                    }
                }
            }
        }
    }
}
```

`replace_strings` Plugin

Rewrite source file content before parsing
(**`after_load_file`** callback phase)

Rewrite localized file content before saving
(**`before_save_localized_file`** callback phase)

Rewrite translation on the fly
(**`rewrite_translation`** callback phase)

Rewrite input/output file paths
(**`rewrite_path`** callback phase)

replace_strings recipe: statically rewrite a variable

strings.js → `var lang = "en";`

strings.de.js → `var lang = "de";`

strings.pt-br.js → `var lang = "pt-br";`

```
# job definition
{
    # ...

    callback_plugins
    {
        {
            plugin              process_if
            phase               before_save_localized_file

            data
            {
                #               replace what    replace with
                replace         `lang = "en"`   `lang = "%LANG%"`
            }
        }
    }
}
```

`limit_languages` Plugin

Gives an ability to specify the list of target languages on a per-file basis.

By default, each file is localized into all languages specified in the job's `destination_languages` parameter.

Setting the list of languages (default rule):
`L10N_LIMIT_DESTINATION_LANGUAGES=es,de,pt-br`

Excuding languages (default rule):
`L10N_EXCLUDE_DESTINATION_LANGUAGES=ja,zh-cn`

limit_languages recipe: specify languages by regions

```
default      → empty list of languages

LANG_EUROPE  → add: da de es fi fr it nl
                    pl pt ru sv tr

LANG_APAC    → add: id ja ko zh-cn zh-tw

LANG_ALL     → add all languages
```

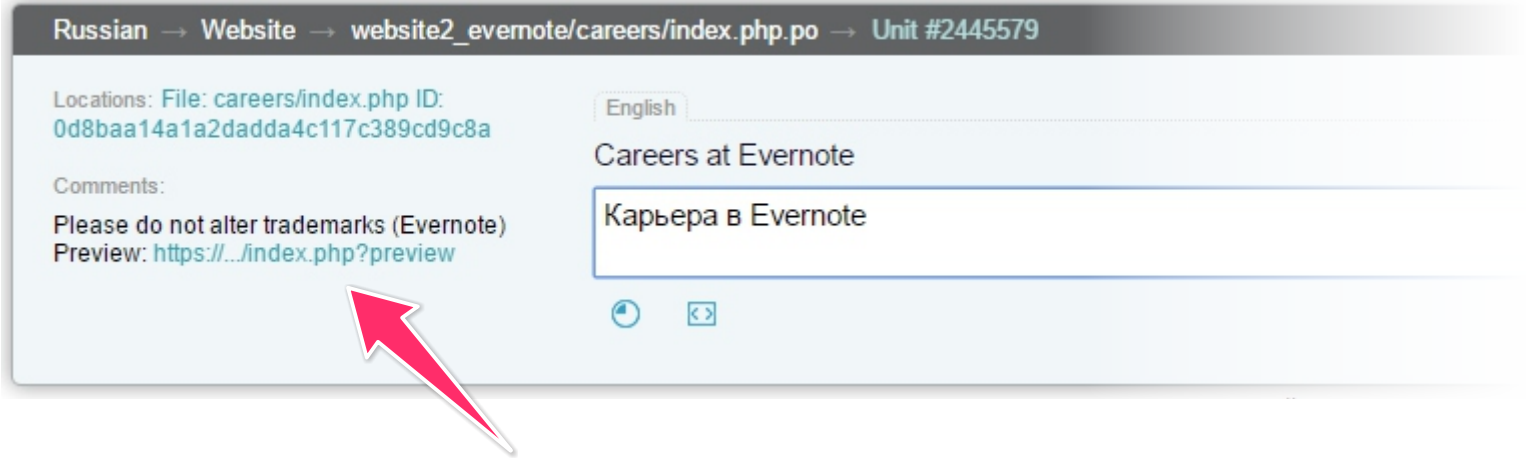
---

global.js → `// LANG_ALL`

feature_x.js → `// LANG_EUROPEAN LANG_APAC`

`append_hint_message` Plugin

Add arbitrary comments/link to translation units



limit_languages recipe: website preview links

```
# job definition
{
    # ...

    callback_plugins
    {
        {
            plugin                  append_hint_message

            data
            {
                    message         Preview (English):
                                    https://preview-site/en/%FILE%

                    message         Preview (localized):
                                    https://preview-site/%LANG%/%FILE%
            }
        }
    }
}
```

`feature_branch` Plugin

Optimize localization of multiple branches by translating only deltas.

`transform` Plugin

"Guess" translations by using existing ones and applying known transformations (case, whitespace, punctuation, wrapper tags).

Hello, world! → Привет, мир!

HELLO, WORLD → ???

Hello, world! → HELLO, WORLD

`original string → upper case → trailing punctuation`

Привет, мир! → ПРИВЕТ, МИР

`test_language` Plugin

Generate pseudolocalized files on the fly, with optional string expansion and character replacement.

This is an example

Ŧĥĩš ĩš áŋ ēχáḿṕḽē

[Ŧĥĩš ĩš áŋ ēχáḿṕḽē xxxxx xx]

`completeness` Plugin

Create, update or delete localized resource files only when they are translated above or below the certain threshold.

```
# job definition
{
    # ...

    callback_plugins
    {
        {
            plugin                          completeness

            data
            {
                    create_threshold        0.9
                    update_threshold        0.3
                    can_delete              YES
            }
        }
    }
}
```

serge.io

Documentation, download and installation instructions

github.com/evernote/serge

`t/` folder contains additional examples:
test configuration files for different parsers

Questions?