

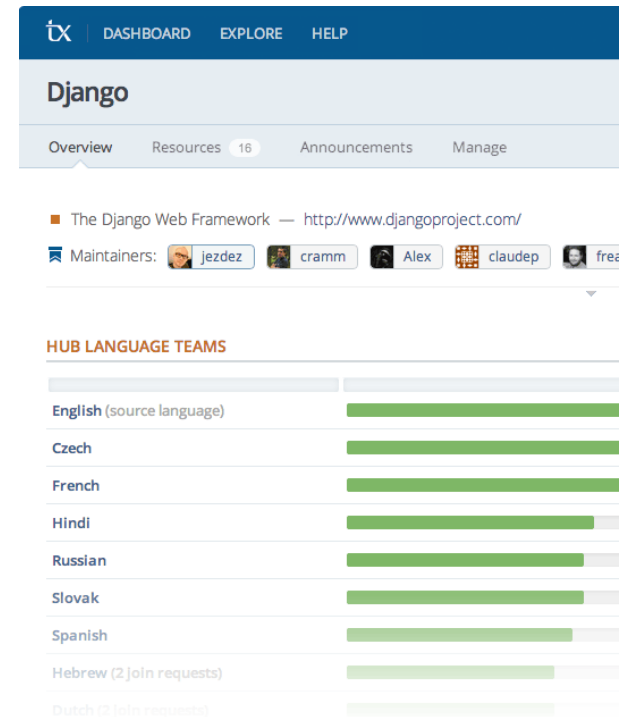
Transifexによる Sakai CLE翻訳

名古屋大学情報連携統括本部情報戦略室
出口 大輔

Transifexとは？

Transifex とは？

- ◆ Webを利用した翻訳プラットフォーム (OSS)
 - WEBが利用できれば翻訳可能
 - オフライン翻訳にも対応
 - 翻訳ワークフローの自動化が可能
- ◆ 多様な言語をサポート
 - 英語, フランス語, 日本語, 他
- ◆ 複数のフォーマットをサポート
 - Gettext (.po), Java property files
Windows resource files (.resx), etc.
- ◆ オープンソースプロジェクトは無料



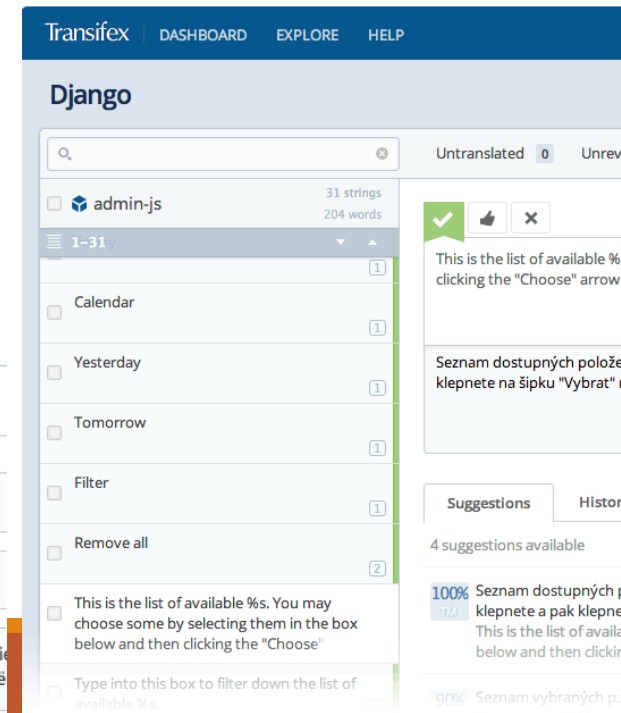
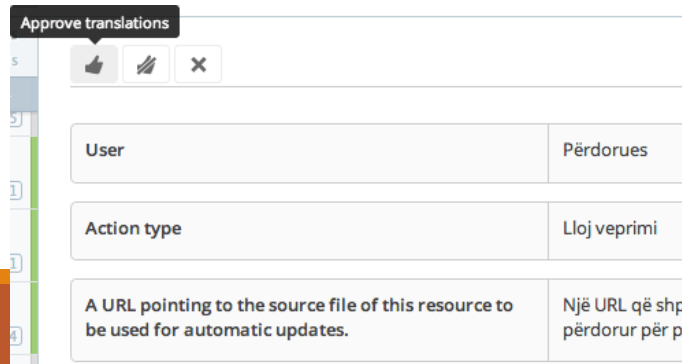
Transifex の利点(1)

◆ コラボレーション機能

- Transifex上で全てのリソースを管理可能
- 翻訳者間で翻訳を常に最新に保つことが可能
- WEBインタフェースを利用した同時翻訳が可能

◆ 複数ロール

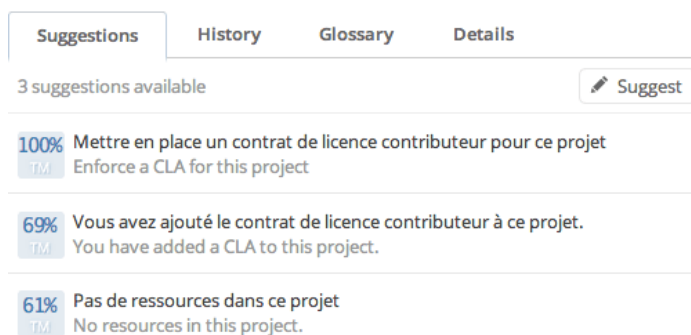
- 翻訳者
- レビューア
- コーディネータ
- ...



Transifex の利点 (2)

◆ 便利な翻訳ツール

- 翻訳メモリ
 - リソース間で翻訳を共有可能
 - 翻訳の一貫性を保つことが可能
- 用語集
- バージョン管理



The screenshot shows the 'Suggestions' tab of the Translation Memory interface. It features a navigation bar with 'Suggestions', 'History', 'Glossary', and 'Details'. Below the bar, it indicates '3 suggestions available' and includes a 'Suggest' button. Three suggestions are listed with their match percentages and descriptions:

Match %	Description
100%	Mettre en place un contrat de licence contributeur pour ce projet Enforce a CLA for this project
69%	Vous avez ajouté le contrat de licence contributeur à ce projet. You have added a CLA to this project.
61%	Pas de ressources dans ce projet No resources in this project.

Translation Memory



The screenshot shows the 'Glossary' interface with a '+ Add new term' button. It displays a table with two columns: 'ENGLISH' and 'FRENCH'. The table contains two entries:

ENGLISH	FRENCH
contributor license agreement No comment yet.	contrat de C'est l'équiva Générales d'l
entry No comment yet.	entrée No comment

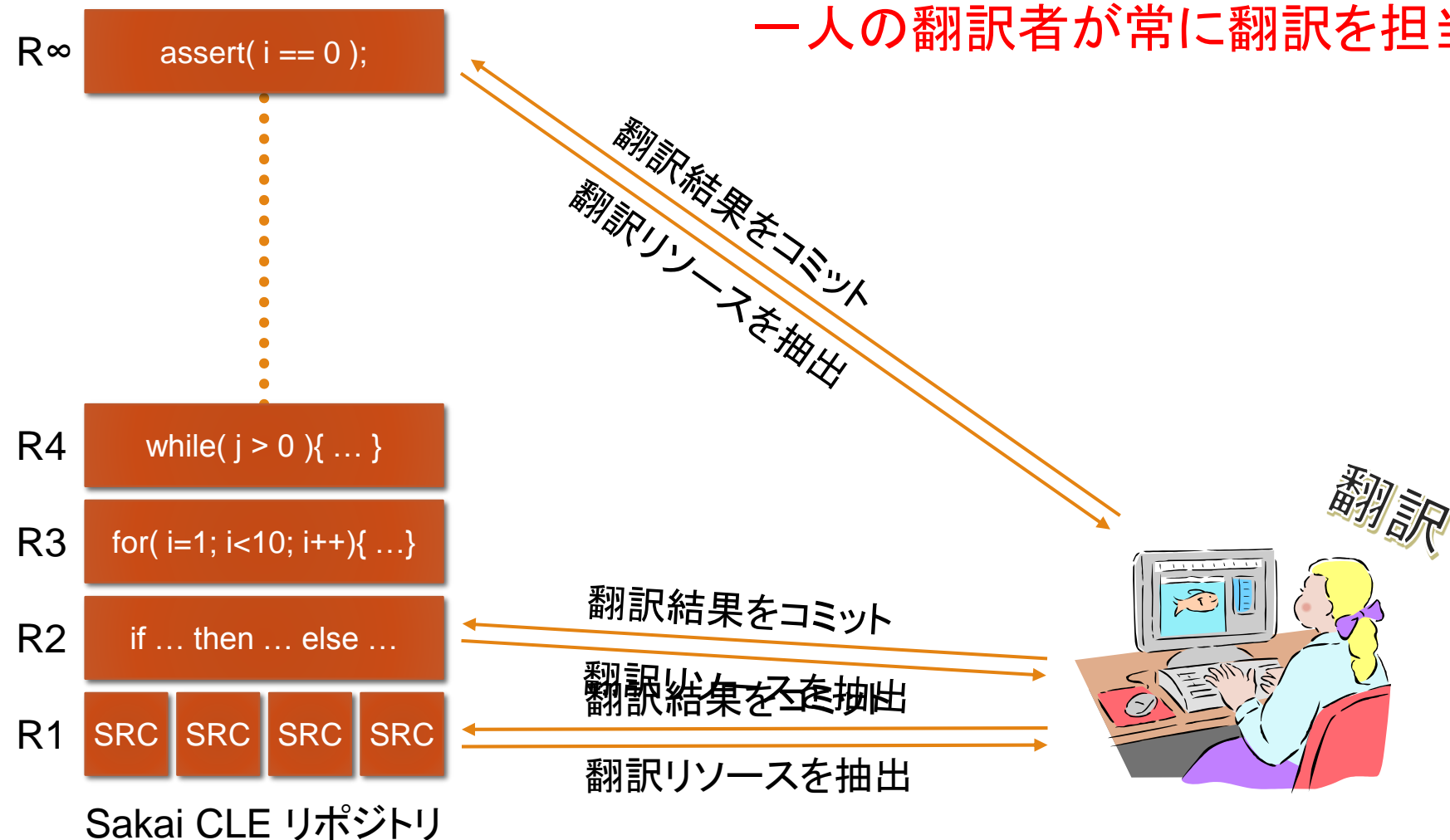
Glossary

Sakai CLEの翻訳

～Transifex編～

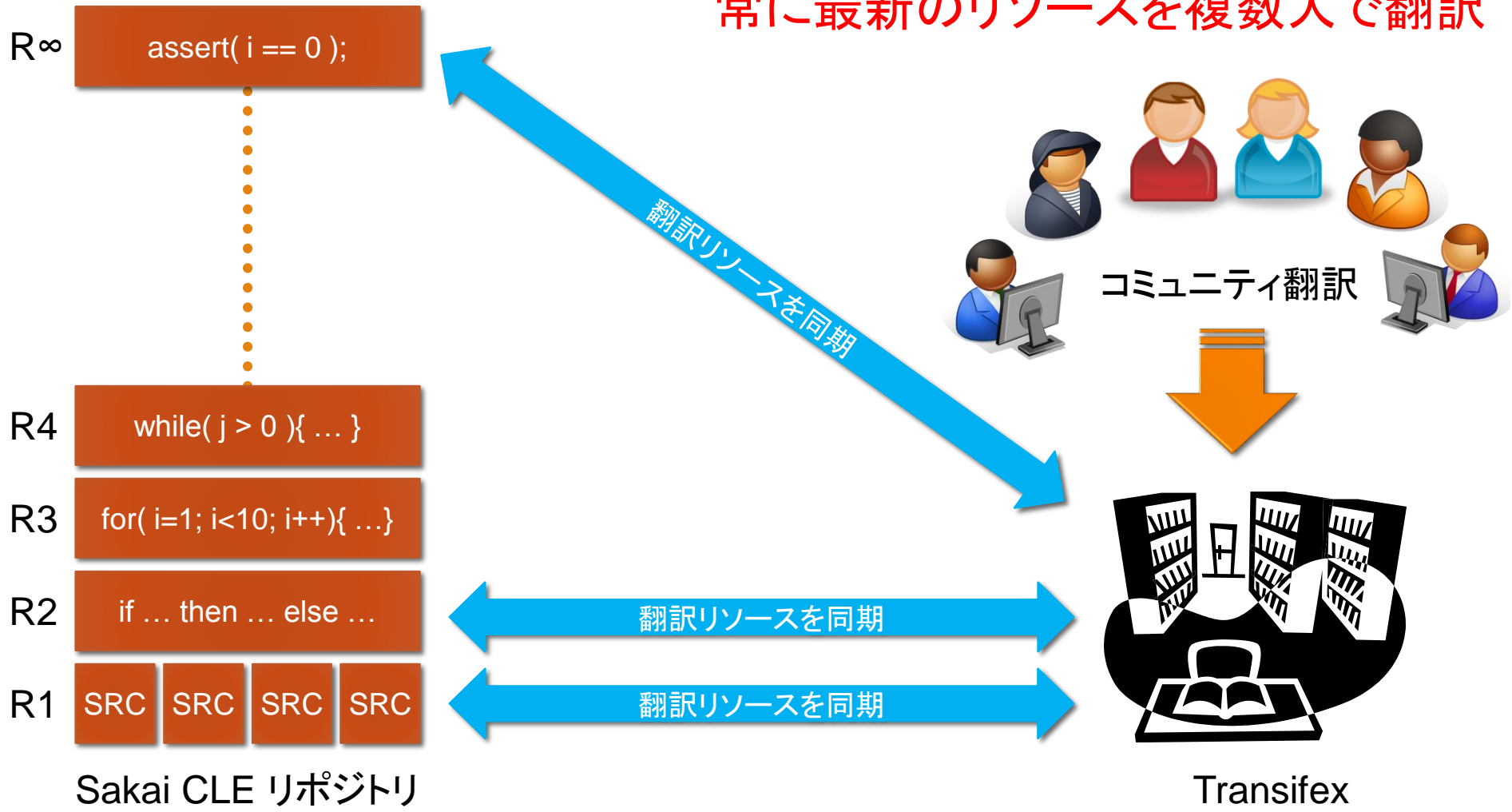
Sakai CLE 翻訳の現状

一人の翻訳者が常に翻訳を担当！



Transifex による Sakai CLE の翻訳

常に最新のリソースを複数人で翻訳



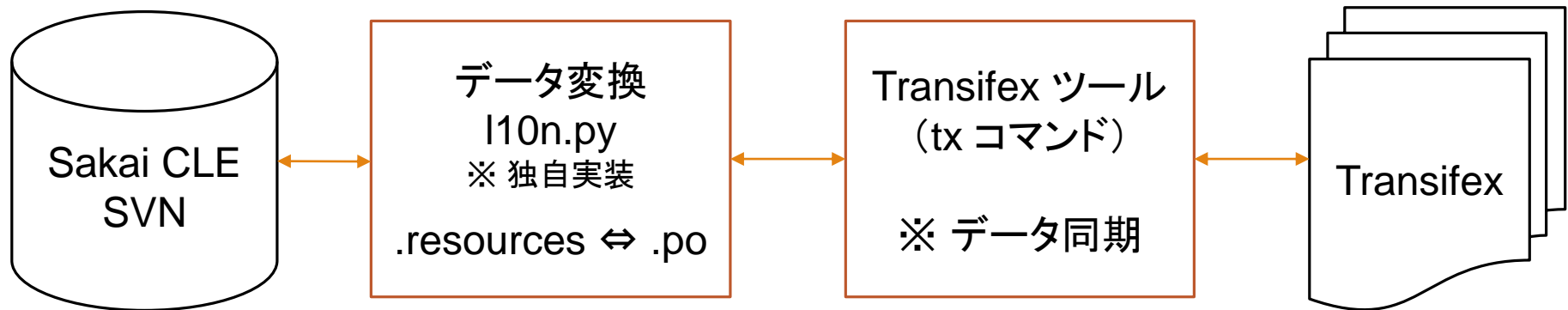
Sakai と Transifex 連携の問題点

◆ 翻訳単位の違い

- Sakai CLE
 - 1モジュールに複数の .properties ファイル
- Transifex
 - 1 リソース = 1 .properties ファイル

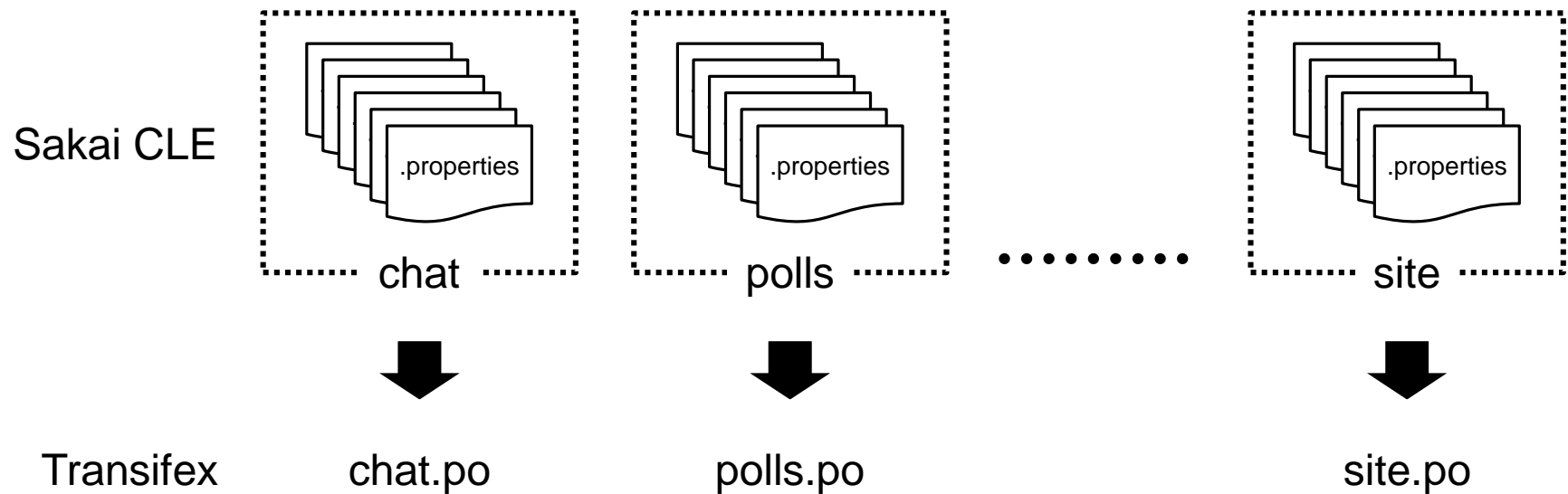
◆ モジュール (Sakai) とリソース (Transifex) の対応

- Sakaiの「.properties」の集約／展開が必要



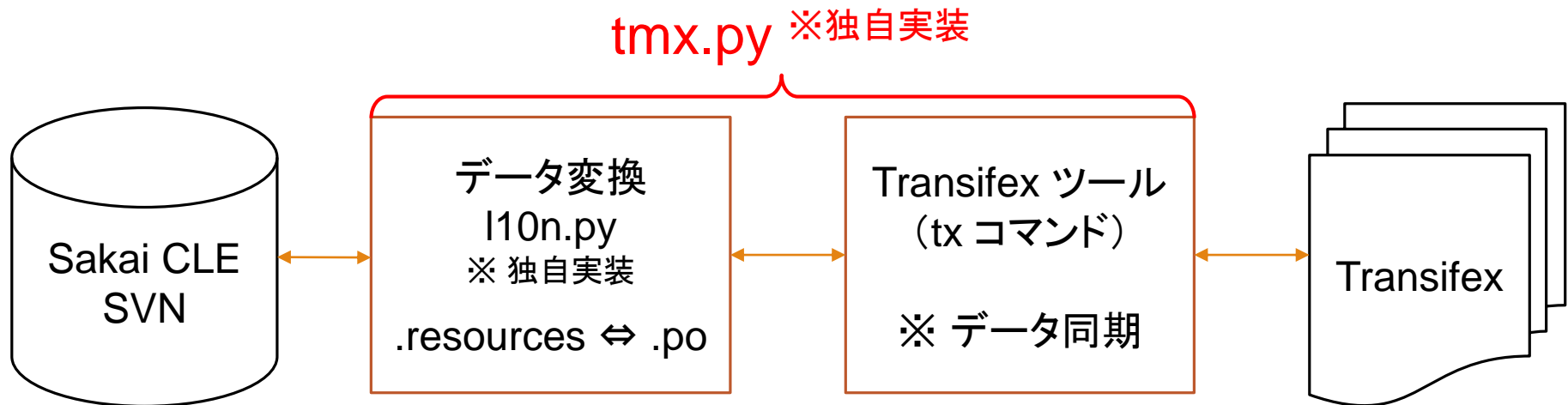
モジュール単位でのリソース変換

- ◆ Sakai CLE で翻訳対象となる .properties
 - モジュールに複数の .properties が存在
 - Transifex で大量のリソースを扱うことは困難
 - 複数の .properties をまとめて単一の .po に変換



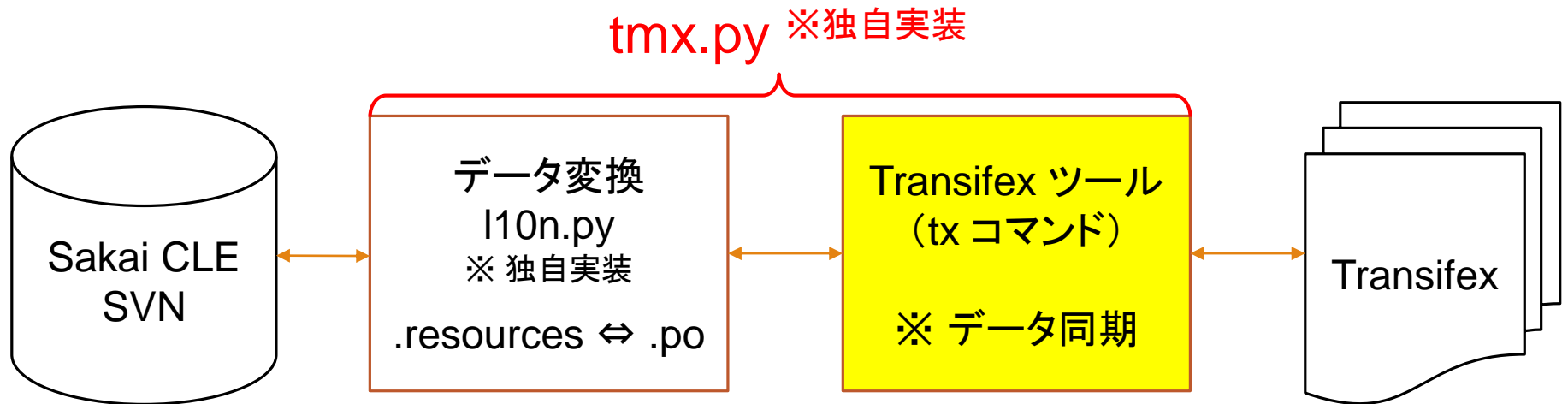
Sakai CLEの翻訳ワークフロー

1. SakaiのSVNリポジトリからソースをチェックアウト
2. 各モジュールから翻訳対象の .properties を抽出
3. .properties を集約して .po (gettext形式) に変換
4. Transifex へアップロード／翻訳／ダウンロード
5. .properties へ反映



Sakai CLEの翻訳ワークフロー

1. SakaiのSVNリポジトリからソースをチェックアウト
2. 各モジュールから翻訳対象の .properties を抽出
3. .properties を集約して .po (gettext形式) に変換
4. Transifex へアップロード／翻訳／ダウンロード
5. .properties へ反映



Transifex Command-line Client

Transifex Command-line Client

◆ tx: サーバとクライアント間のリソース同期ツール

```
hoge hoge$ tx -h
Usage: tx [options] command [cmd_options]
```

This is the Transifex command line client which allows you to manage your translations locally and sync them with the master Transifex server. If you'd like to check the available commands issue `tx help` or if you just want help with a specific command issue `tx help command`

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
-d, --debug	enable debug messages
-q, --quiet	don't print status messages to stdout
-r ROOT_DIR, --root=ROOT_DIR	change root directory (default is cwd)
--traceback	print full traceback on exceptions
--disable-colors	disable colors in the output of commands

TXツールのインストール

◆TXのインストール

```
pip install transifex-client
```

◆TXの更新

```
pip install --upgrade transifex-client
```

◆PIPのインストール

```
curl -O https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py  
python get-pip.py
```

TXの使い方(初期化)

◆ tx init

- 翻訳を行うプロジェクトの初期化 & 認証情報の設定
- Transifexのパスワードは「~/.transifexrc」に平文で保存

```
Creating .tx folder...
Transifex instance [https://www.transifex.com]: ← 何も入力しない
Creating skeleton...
Creating config file...
No configuration file found.
No entry found for host https://www.transifex.com. Creating...
Please enter your transifex username: <...> ← TransifexのユーザID
Password: <...> ← パスワード
Updating /home/username/.transifexrc file...
Done.
```


TXの使い方(リソースの登録)

- ◆ tx set --auto-local -t PO -r PRJ.RES <lang>/RES.po
-source-lang en -source-file templates/RES.pot -execute
 - -r PRJ.RES
 - Transifexの翻訳リソースを指定
 - プロジェクト名 (PRJ) + 「. 」 + モジュール名 (RES)
 - 例) tmx-test-project.chat
 - --auto-local
 - ローカルのリソースを元にTransifexプロジェクトを初期化
 - -t PO
 - リソースのフォーマットを gettext 形式に指定

TXの使い方(リソースの登録)

- ◆ tx set --auto-local -t PO -r PRJ.RES <lang>/RES.po
-source-lang en -source-file templates/RES.pot -execute
 - <lang>/RES.po
 - 言語でフォルダを作成し, その下にリソースを配置(ローカル)
 - -source-lang en
 - 翻訳元の言語を英語に設定
 - -source-file templates/MMM.pot
 - 翻訳元のファイルのパスを指定
 - -execute
 - コマンドを実行

TXの使い方(アップロード)

◆ tx push -s

- 翻訳元リソースを全てアップロード

◆ tx push -s -r PRJ.RES

- 翻訳元リソース RES のみをアップロード

◆ tx push -t -l ja

- 翻訳済みの ja のリソースを全てアップロード

◆ tx push -t -l ja -r PRJ.RES

- 翻訳済みの ja のリソース RES のみをアップロード

TXの使い方(ダウンロード)

◆ tx pull -a

- 全ての言語の翻訳済みリソースを全てダウンロード

◆ tx pull -l ja

- 言語 ja の翻訳済みリソースを全てダウンロード

◆ tx pull -l ja -r **PRJ.RES**

- 言語 ja の翻訳済みリソース RES のみをダウンロード
 - 例) PRJ.RES = tmx-test-project.chat

Transifexでの翻訳

検索

未翻訳 65 レビュー前 264

- announcement
- 264 件の文字列 / 1424 語
- opted out
- None - No notification
- Permissions
- Merge
- Reorder
- Back to Announcements
- You have to select the announcement first!
- Please choose only one announcement at a time to edit
- You don't have permission to edit this announcement!
- You need to fill in the title!
- You need to fill in the body of the announcement!
- You don't have permissions to create this announcement - {0}
- you don't have permission to delete the messages.
- you don't have permissions to delete this announcement - {0}

翻訳対象

保存 0

You don't have permission to edit this announcement!

オリジナル

このアナウンスを編集する権限がありません!

翻訳

by ddeguchi, a month ago

開発者メモ: java.alert.youdont

説明: 文字列の説明を追加するにはここをクリック

タグを追加

↑ 詳細を閉じる

キー: You don't have permission to edit this announcement!

サイズ: 8 語

出現場... announcement/announcement-tool/tool/src/bundle/announcement.prop

コンテ... announcement/announcement-tool/tool/src/bundle/announcement.prop

提案 9 履歴 1 用語集 コメント

9 件の提案があります

100% このアナウンスを編集する権限がありません!
 You don't have permission to edit this announcement!

翻訳メモリ

92% このお知らせを削除する権限がありません!
 you don't have permissions to delete this announcement!

89% このお知らせを削除する権限がありません
 you don't have permissions to delete this announcement -

77% このチャットを作成する権限がありません。
 You do not have permission to create this chat.

76% このチャットを表示する権限がありません。
 You do not have permission to view this chat.

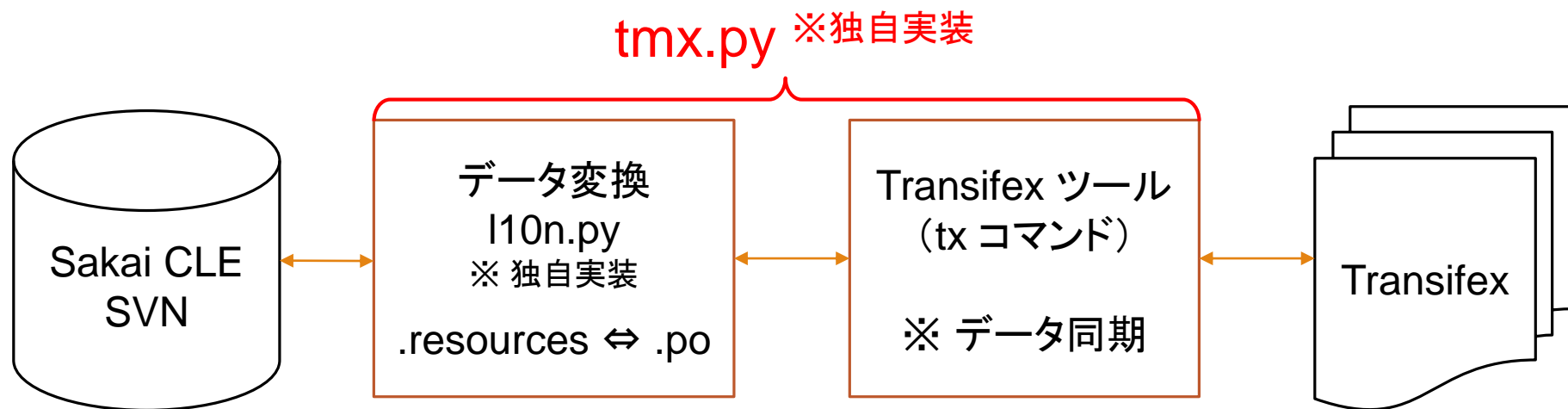
74% このチャットを投稿する権限がありません。
 You do not have permission to post to this chat.

71% メッセージを削除する権限がありません。
 you don't have permission to delete the messages.

TMX.PY

Sakai CLEの翻訳ワークフロー

1. SakaiのSVNリポジトリからソースをチェックアウト
2. 各モジュールから翻訳対象の .properties を抽出
3. .properties を集約して .po (gettext形式) に変換
4. Transifex へアップロード／翻訳／ダウンロード
5. .properties へ反映



tmx.py の基本

◆ 初期設定

- Sakai CLEのルートフォルダにツール一式 (l10n) を配置

◆ プロジェクトの初期化

- `python tmx.py init`

◆ リソースの変換 & Transifexへのアップロード

- `python tmx.py upload -mu`
 - `.properties` → `.po` (モジュール単位) → Transifex

◆ Transifex からダウンロード & リソースの変換

- `python tmx.py download -u -l ja`
 - Transifex → `.po` (モジュール単位) → `.properties`

.properties → .po

.properties → .po の詳細

◆ .po (gettext形式) のコメント & コンテキストを活用

#. alias.alert ← **.properties のキー**

#: alias/alias-tool/tool/src/bundle/admin.properties:1
← **ソースファイル名 + 番号**

msgctxt "alias/alias-tool/tool/src/bundle/admin.properties:alias.alert"

msgid "Alert:" ← **翻訳対象となるテキスト**

msgstr "Alert:" ← **翻訳後のテキスト**

← **ソースファイル名 + キー**
※ 別ファイルの同じ翻訳テキスト
を区別するため