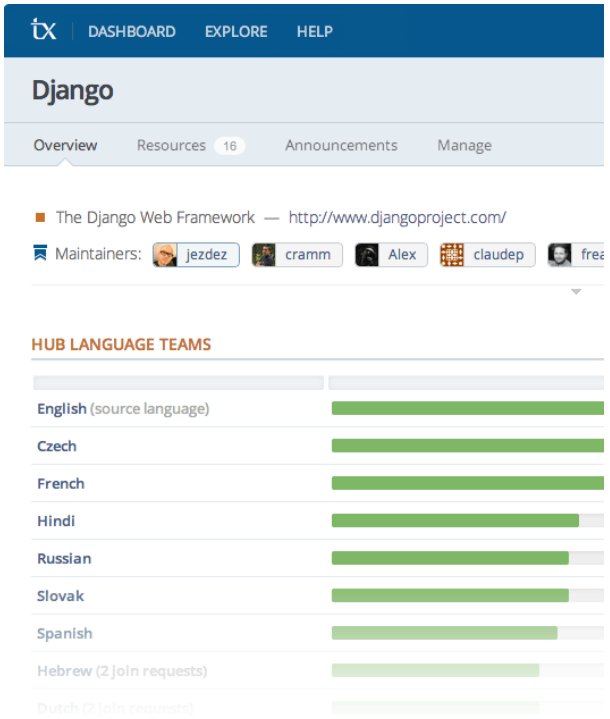
TransifexによるSakai翻訳環境の設定

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？

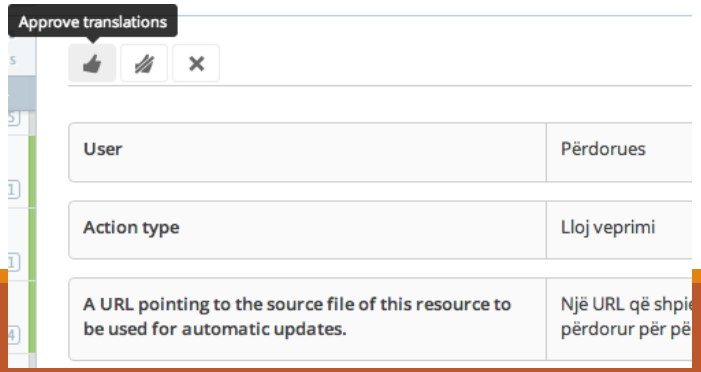
Transifex とは？

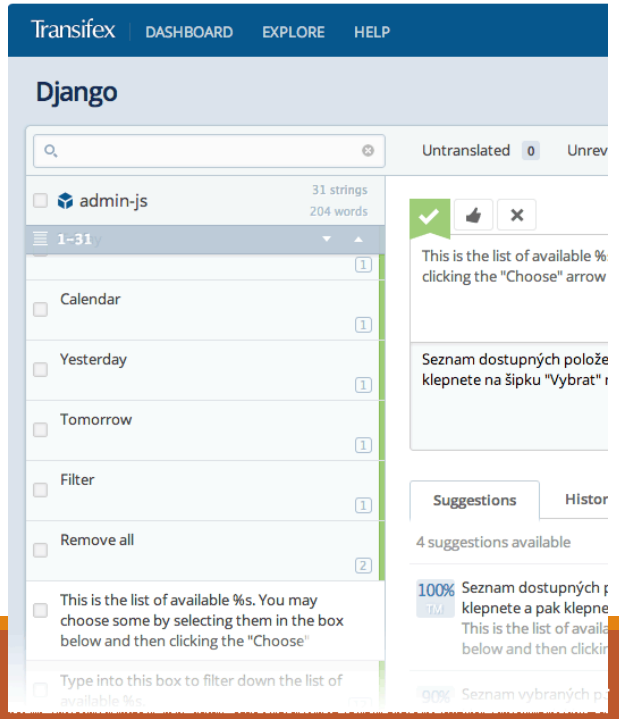
- Webを利用した翻訳プラットフォーム（OSS）
  - WEBが利用できれば翻訳可能
    - オフライン翻訳にも対応
  - 翻訳ワークフローの自動化が可能
- 多様な言語をサポート
  - 英語，フランス語，日本語，他
- 複数のフォーマットをサポート
  - Gettext (.po), Java property files
    Windows resource files (.resx), etc.
- オープンソースプロジェクトは無料



Transifex の利点（1）

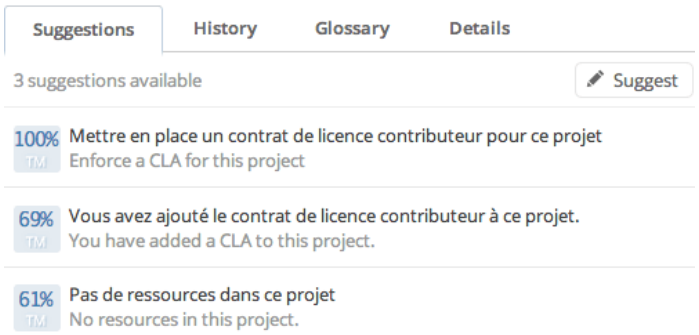
- コラボレーション機能
  - Transifex上で全てのリソースを管理可能
  - 翻訳者間で翻訳を常に最新に保つことが可能
  - WEBインタフェースを利用した同時翻訳が可能
- 複数ロール
  - 翻訳者
  - レビューア
  - コーディネータ
  - …





Transifex の利点（2）

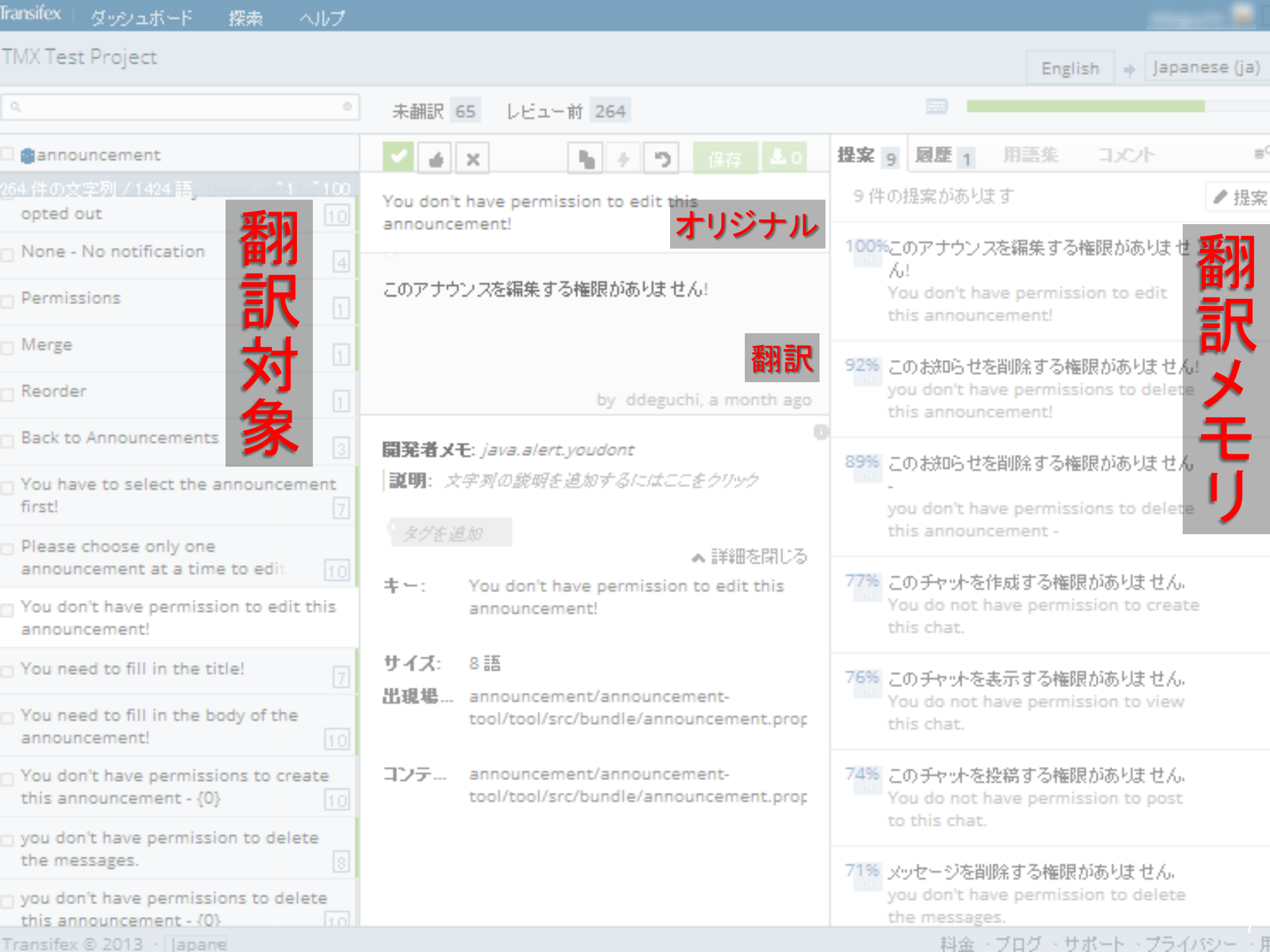
- 便利な翻訳ツール
  - 翻訳メモリ
    - リソース間で翻訳を共有可能
    - 翻訳の一貫性を保つことが可能
  - 用語集
  - バージョン管理



Translation Memory

Glossary

Transifexでの翻訳

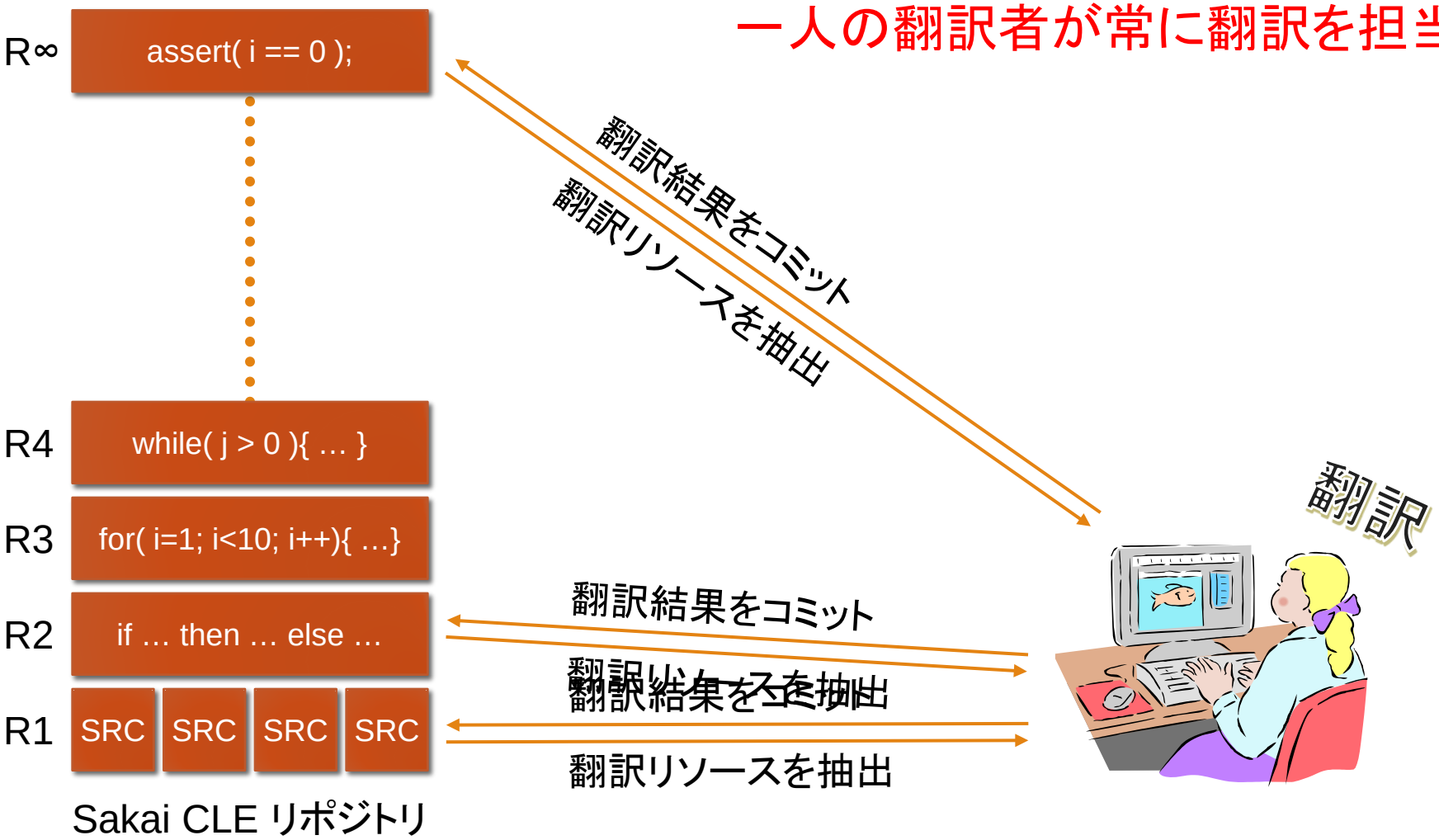

Transifex
ダッシュボード
探索
ヘルプ
TMX Test Project
English
Japanese (ja)
未翻訳 65
レビュー前 264
announcement
264 件の文字列 / 1424 語
opted out
None - No notification
Permissions
Merge
Reorder
Back to Announcements
You have to select the announcement first!
Please choose only one announcement at a time to edit
You don't 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
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
提案
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!
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.
翻訳メモリ
Transifex © 2013
料金 · ブログ · サポート · プライバシー

Sakai CLEの翻訳

～Transifex編～

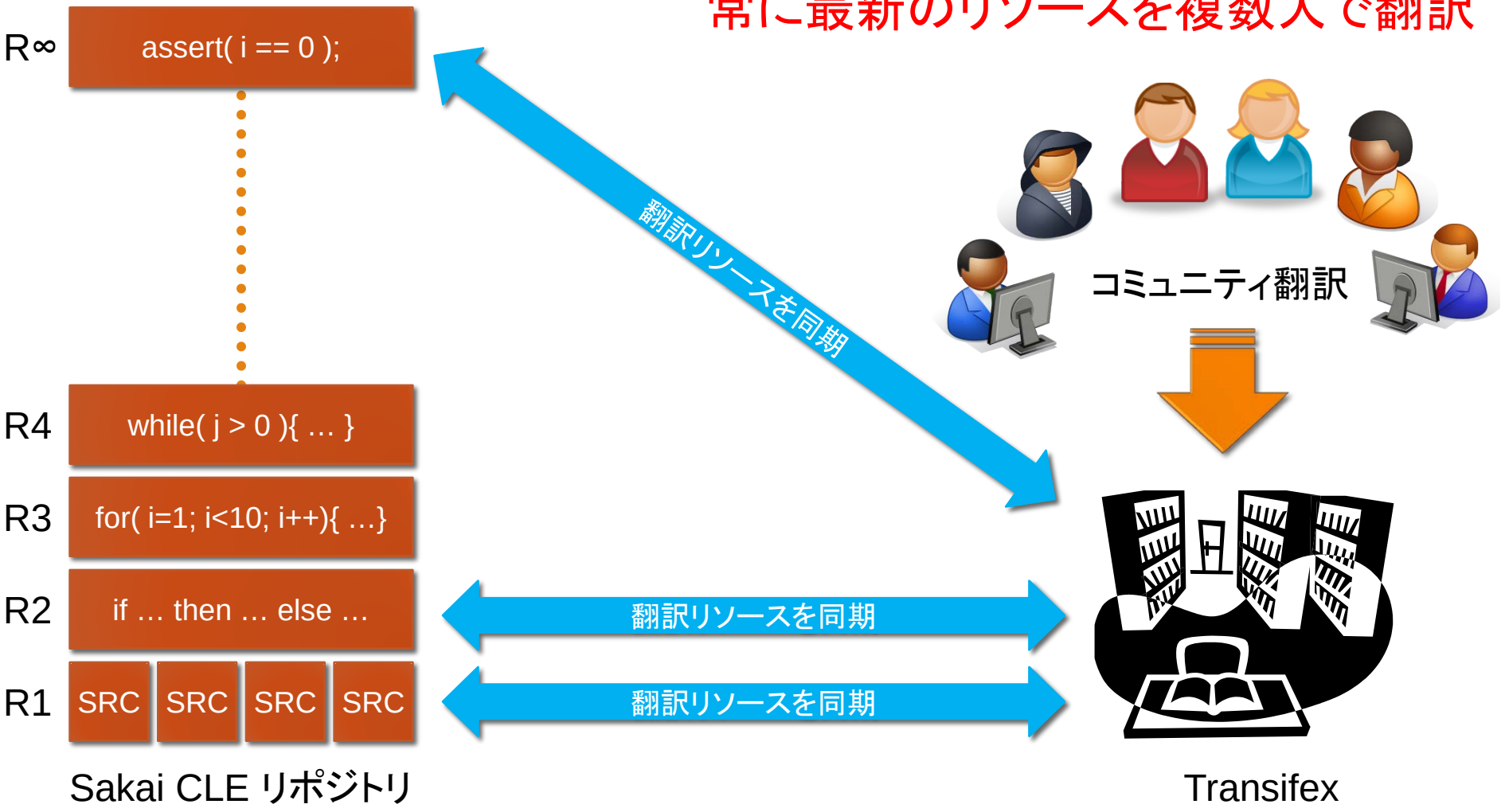
Sakai CLE 翻訳の過去の取り組み

一人の翻訳者が常に翻訳を担当！



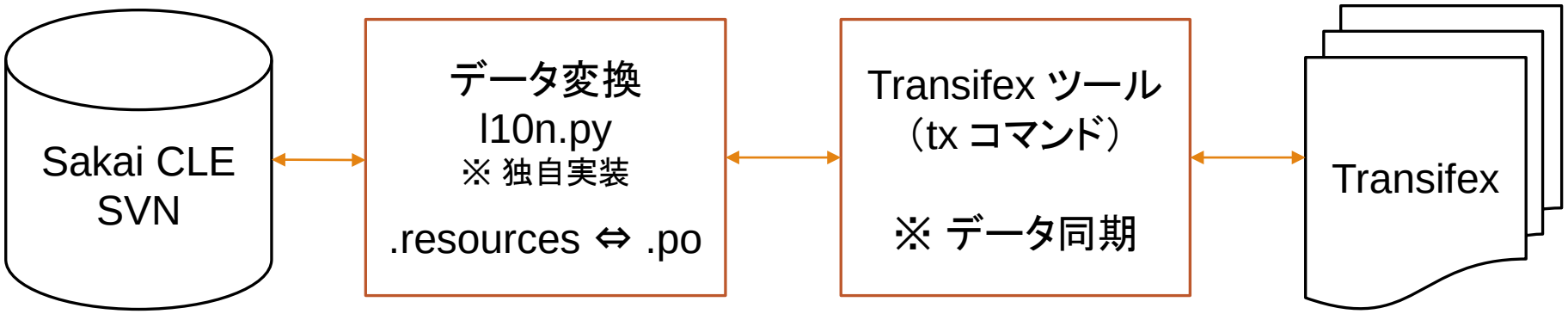
Transifex による Sakai CLE の翻訳

常に最新のリソースを複数人で翻訳



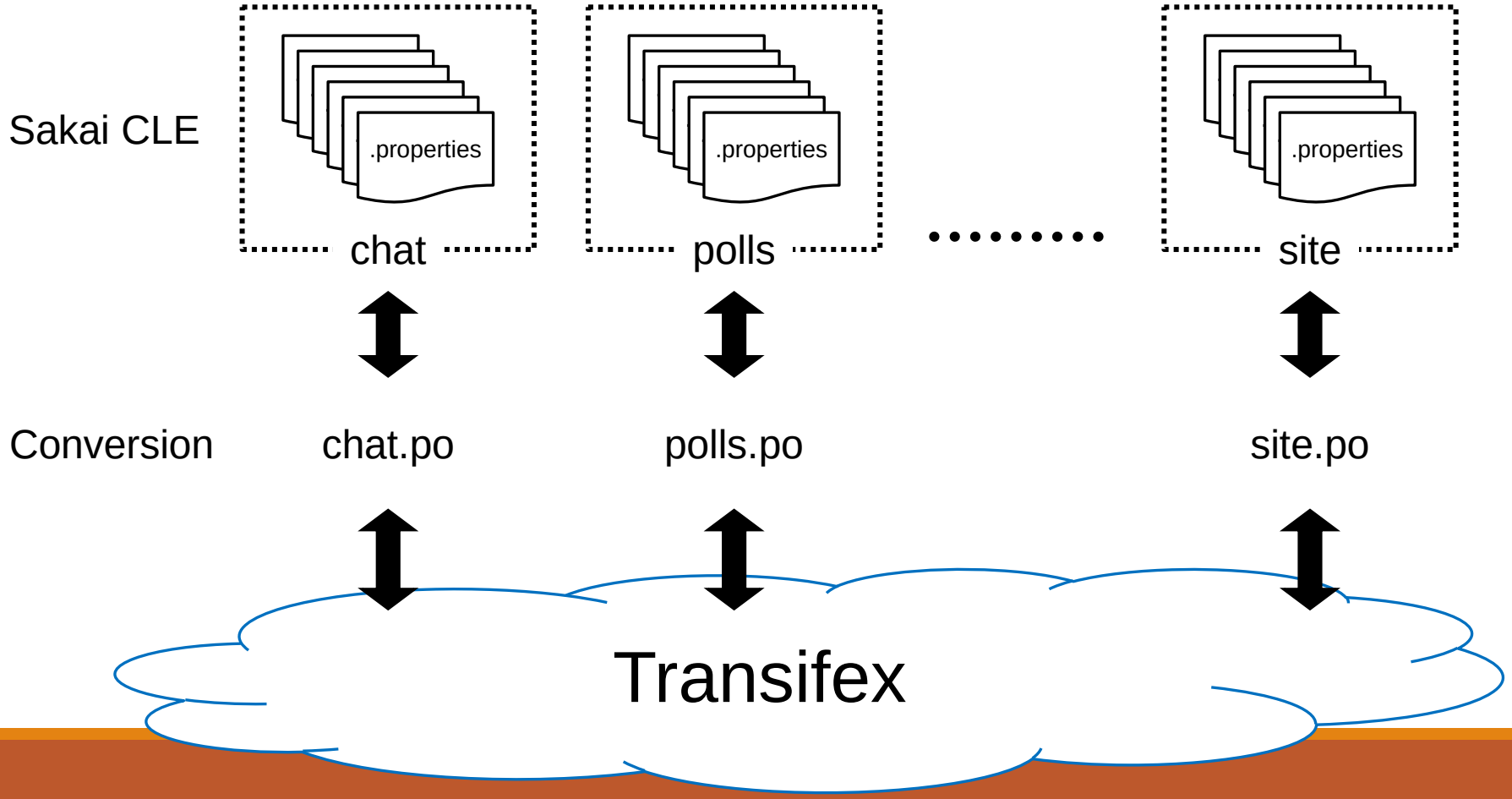
Sakai と Transifex 連携の問題点

- 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



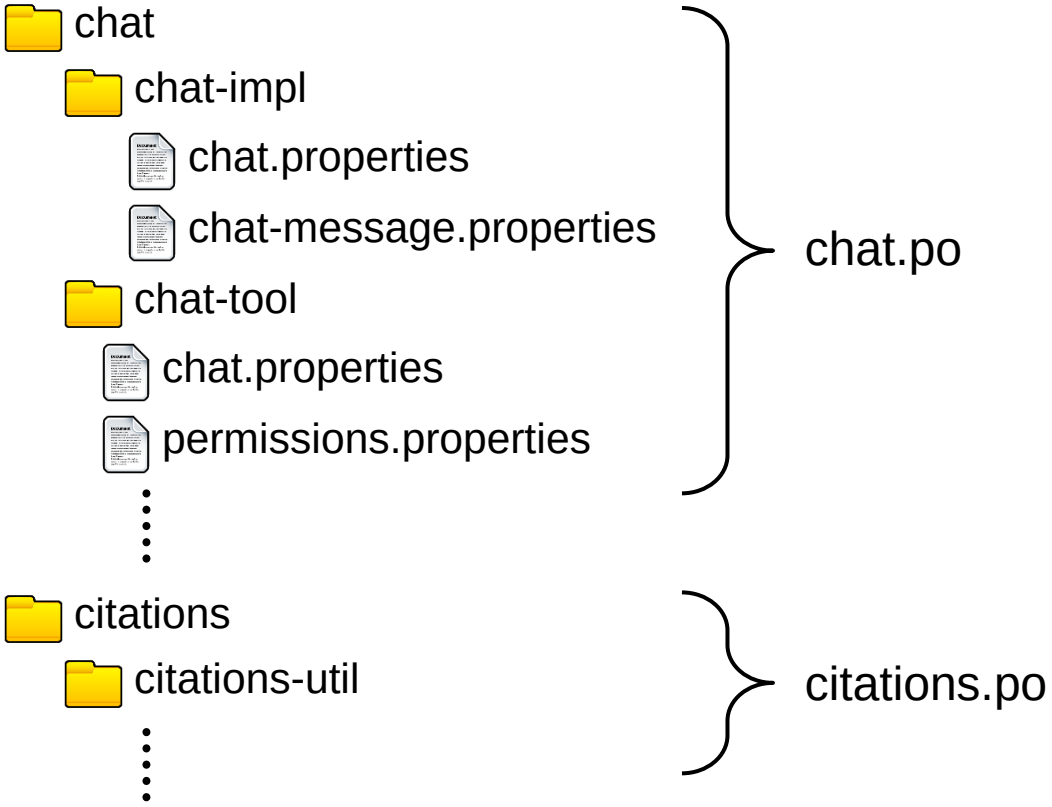
モジュール単位でのリソース変換

- Sakai CLE で翻訳対象となる .properties を集約
  - モジュール内の .properties を束ねて単一の .po に変換



モジュール単位でのリソース変換

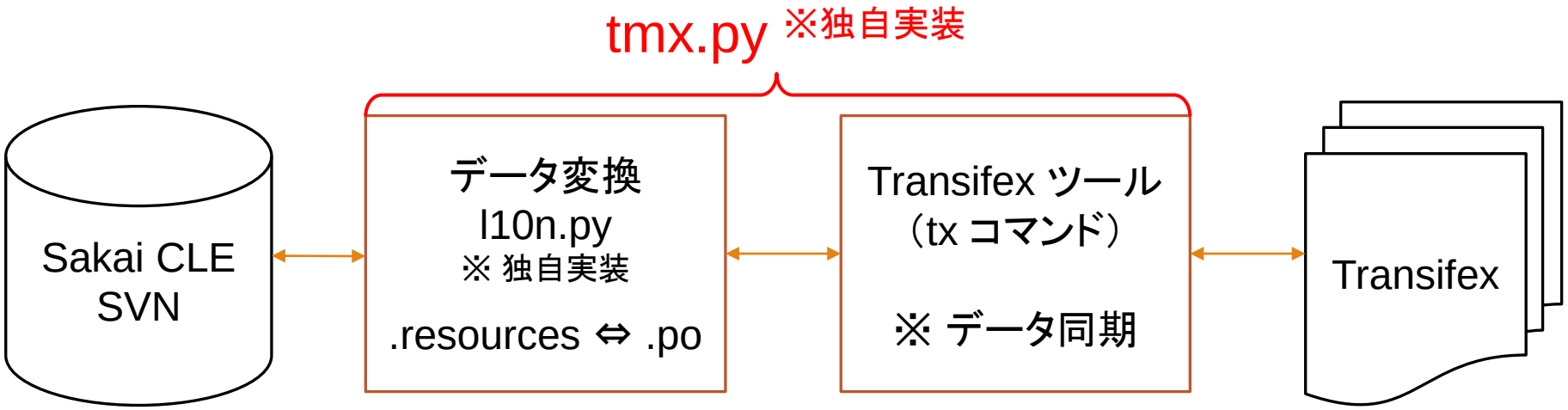
- Sakai CLE で翻訳対象となる .properties を集約
  - モジュール内の .properties を束ねて単一の .po に変換



Sakai CLE ⟵ Format conversion (gettext) ⟶ Transifex

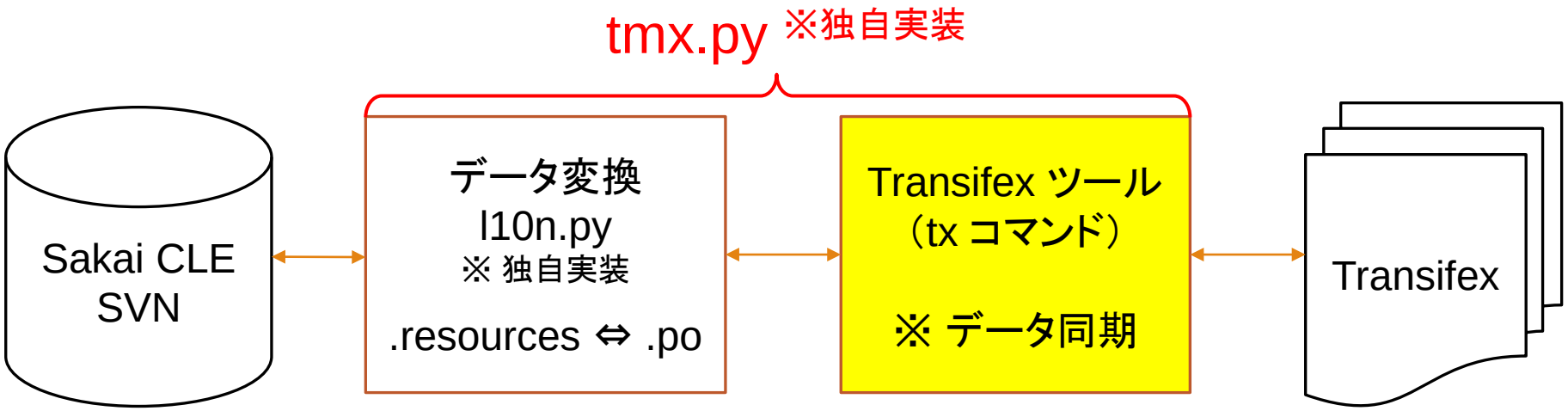
Sakai CLEの翻訳ワークフロー

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



.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

.properties のキー

#. alias.alert

ソースファイル名＋番号

#: alias/alias-tool/tool/src/bundle/admin.properties:1

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

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

```
hogehoge$ 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
  - https://pip.pypa.io/en/latest/installing.html

```
easy_install pip
```

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]:
Creating skeleton...
Creating config file...
No configuration file found.
No entry found for host https://www.transifex.com. Creating...
Please enter your transifex username: <...>
Password: <...>
Updating /home/username/.transifexrc file...
Done.
```

何も入力しない

TransifexのユーザID

パスワード

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

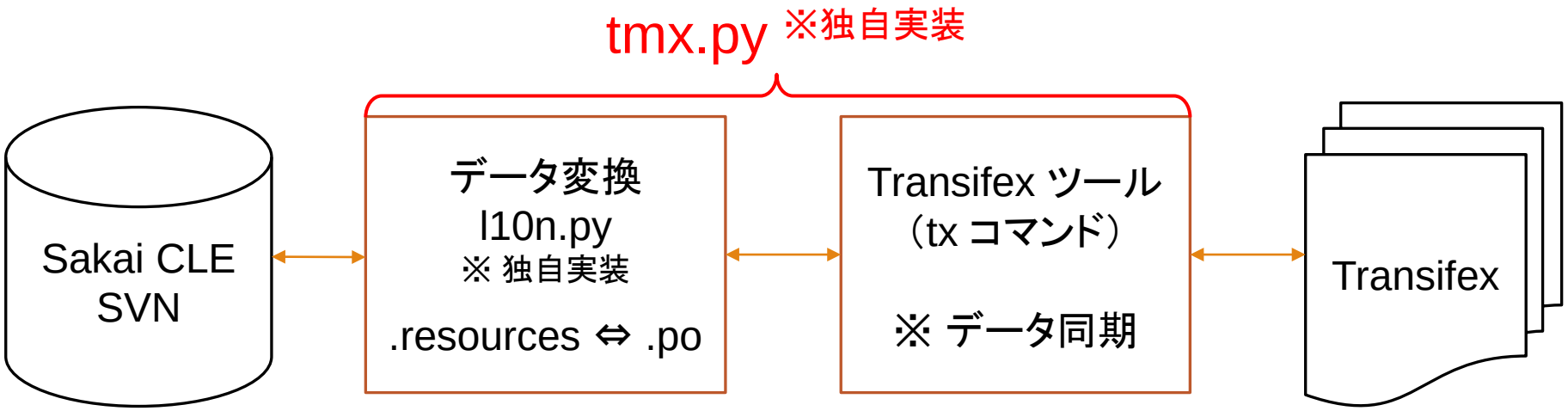
TXの使い方(ダウンロード)

- 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

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

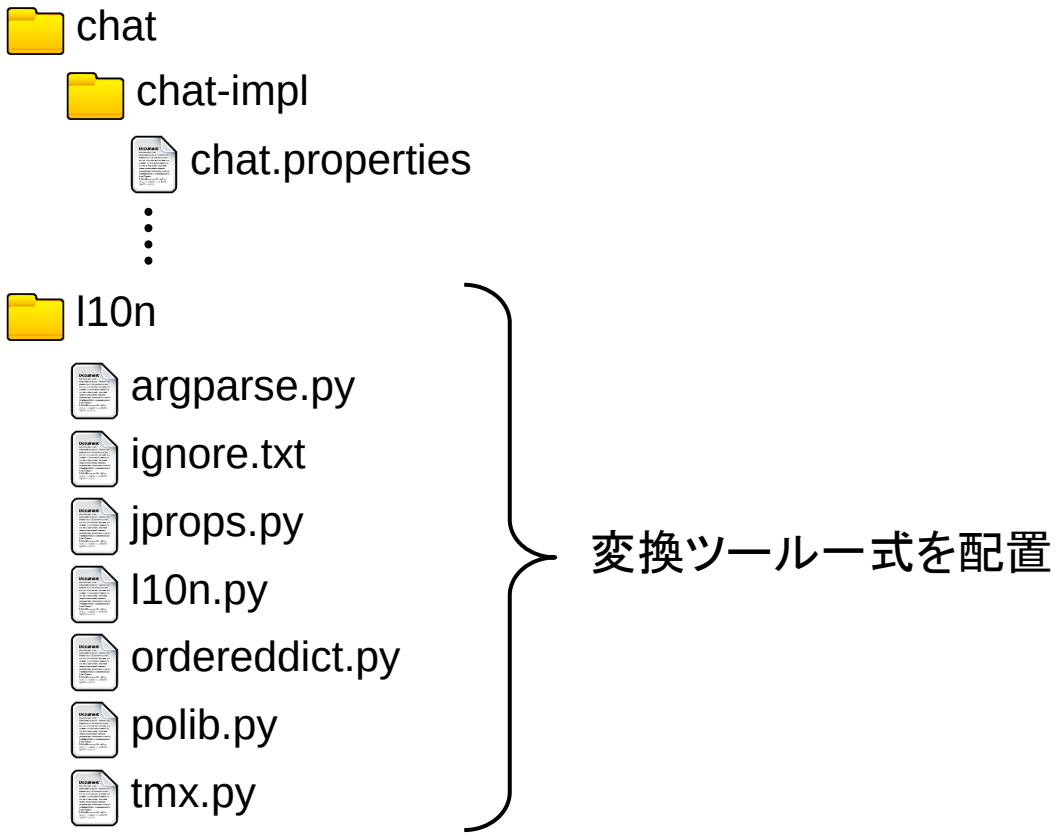


tmx.py の基本

- 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

tmx.py の使い方（初期設定）

- Sakai のルート直下に l10n フォルダを作成
  - tmx.py 等のファイル一式を配置



ヘルプの表示

- python tmx.py –help
  - 使い方がわからない時の確認方法

```
$ python tmx.py --help
usage: tmx.py [-h] {download,init,upload} ...

Tool for TMX Project to transfer translation resources between Sakai CLE and
Transifex

positional arguments:
  {download,init,upload}
                        transfer mode ...
    init                initialize Transifex settings and create .tx
                        directory.
    upload              upload local resources to Transifex
    download            download resources from Transifex to local

optional arguments:
  -h, --help            show this help message and exit
```

初期化

- python tmx.py init
  - 翻訳を行うプロジェクトの初期化＆認証情報の設定
  - Sakaiの翻訳で使用するリソースの登録
  - Transifexのパスワードは「~/.transifexrc」に平文で保存

```
Creating .tx folder...
Transifex instance [https://www.transifex.com]:
Creating skeleton...
Creating config file...
No configuration file found.
No entry found for host https://www.transifex.com. Creating...
Please enter your transifex username: <...>
Password: <...>
Updating /home/username/.transifexrc file...
Done.
Read .properties from announcement...
Updating source for resource sakai-293.announcement ( en ->
templates/announcement.pot ).
```

何も入力しない

TransifexのユーザID

パスワード

リソースのアップロード（ヘルプ）

- python tmx.py upload –help

```
$ python tmx.py upload --help
usage: tmx.py upload [-h] [-u] [-m] [-l LANGUAGE] [-v] [modules [modules ...]]

positional arguments:
  modules               specify which .po/.pot file you want to upload to
                        Transifex (empty means all modules)

optional arguments:
  -h, --help            show this help message and exit
  -u, --update          update .po/.pot file from .properties before upload is
                        finished
  -m, --master          upload .pot (master) file to Transifex
  -l LANGUAGE, --language LANGUAGE
                        upload .po file of each language (e.g. ja_JP) to
                        Transifex
  -v, --verbose         output many messages.
```

リソースのアップロード

- オリジナルリソースのアップロード
  - python tmx.py upload -mu
    - Sakai の .properties を収集して gettext 形式に変換
      - templates/ 以下に .pot ファイルを生成
    - -m オプション：オリジナル（マスター）ソースを指定
    - -u オプション：Sakaiのソースからリソースを抽出
- 特定の言語リソースをアップロード
  - python tmx.py upload -u -l ja
    - Sakai の ja リソースを収集して gettext 形式に変換
      - ja/ 以下に *.po ファイルを生成
    - -l オプション：抽出対象のロケールを指定

リソースのダウンロード（ヘルプ）

- python tmx.py download –help

```
$python tmx.py download --help
usage: tmx.py download [-h] [-u] [-l LANGUAGE] [-v] [modules [modules ...]]

positional arguments:
  modules               specify which .po/.pot file you want to download from
                        Transifex (empty means all modules)

optional arguments:
  -h, --help            show this help message and exit
  -u, --update          update .properties file from .po/.pot files after
                        download is finished
  -l LANGUAGE, --language LANGUAGE
                        download .po file of each language (e.g. ja_JP) from
                        Transifex (defaults to all)
  -v, --verbose         output many messages.
```

リソースのアップロード

- 特定の言語の .po ファイルをダウンロード
  - python tmx.py download -l ja
    - Transifex から ja のリソースを gettext 形式でダウンロード
      - ja/ 以下に .po ファイルを生成
    - -l オプション : 抽出対象のロケールを指定
- 特定の言語リソースをダウンロードして反映
  - python tmx.py download -u -l ja
    - Transifex から ja のリソースを gettext 形式でダウンロード
      - ja/ 以下に .po ファイルを生成
    - .po ファイルから .properties ファイルを生成して Sakai のフォルダに配置

TMX.PY（上級編）

特定モジュールのリソース抽出

- 特定モジュールのオリジナルリソースの抽出
  - python l10n.py -m -r announcement
    - Announcement モジュール の .properties を収集して gettext 形式に変換
      - templates/ 以下に announcement.pot ファイルを生成
    - -m オプション：オリジナル（マスター）ソースを指定
    - -r オプション：リソースの抽出を指定
- 特定モジュールの言語リソースを抽出
  - python l10n.py -r -l ja announcement
    - Announcement モジュールの ja リソースを gettext 形式に変換
      - ja/ 以下に *.po ファイルを生成
    - -l オプション：抽出対象のロケールを指定

特定モジュールのリソース反映

- 特定モジュールの言語リソースを反映
  - python l10n.py -w -l ja announcement
    - Announcement モジュールの翻訳結果（gettext 形式）をSakaiに反映
    - -w オプション：gettext 形式を .properties ファイルに変換
    - -l オプション：抽出対象のロケールを指定

Transifexへ特定リソースのみ送信

- python l10n.py -r -l ja announcement
  - Announcement モジュールの翻訳を gettext 形式に変換
    - .properties → .po（gettext）

- tx push -t -l ja -r［リソースID］
  - ［リソースID］=［プロジェクトID］.［モジュール名］
    - 例）sakai-293.announcement

Transifexから特定リソースのみ反映

- tx pull -l ja -r［リソースID］
  - ［リソースID］=［プロジェクトID］.［モジュール名］
    - 例）sakai-293.announcement

- python l10n.py -w -l ja announcement
  - Announcement モジュールの翻訳結果をソースに反映
    - .po（gettext）→ .properties

Q&A

確認事項

- .metaprops ファイルの扱いはどうするのか？
- HTMLファイル等はどうするか？