Here comes localize.drupal.org

Gábor Hojtsy, Acquia, Drupalcon Paris

ME

- Needed a translated Drupal in 2003
- Became locale.module maintainer, Drupal 6 core committer
- Developed all kinds of translation tools
- Working for Acquia

YOU

- **Running translated Drupal sites?**
- **Translating modules for “self”?**
- **Contributing to drupal.org translations?**
- **Writing & translating modules to “self”?**

Take a walk

2001 Jan 22
t() is born

Drupal

1.0.0

2001 Feb
locale.module,
translation interface

2.0.0

3.0.0

Drupal.org

Disclaimer: time scale not exact

2002 Jan
Locale cache

2003 Sep
extractor.php

2004 Aug
Gettext based sharing,
web admin, plurals

4.0.0

4.3.0

4.5.0

4.7.0

Disclaimer: time scale not exact

2006 Sep
Translatable installer

5.0

2006 Aug
@ !

2007 May
Automated import

Disclaimer: time scale not exact

2007 Jun
JS translation

6.0

2009 Jun
Contexts

7.0

2009 Aug
localize.drupal.org

Disclaimer: time scale not exact

All lined up

TRANSLATING THE BUILT-IN INTERFACE

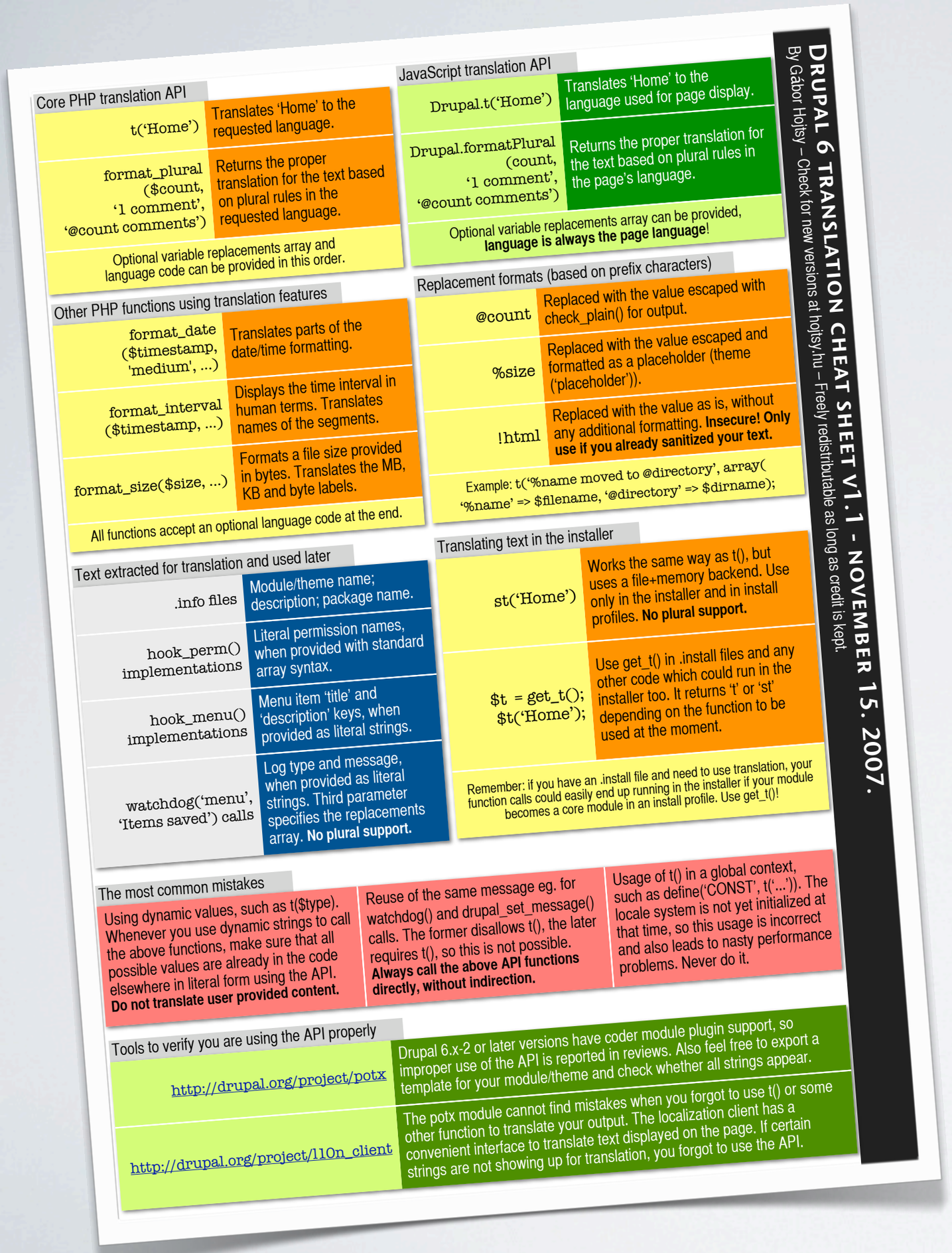
TEXT COMES FROM SOURCE CODE

```
t('Go to %page',
array('%page' => $p));
```

```
Drupal.t('Go to %page',
         {'%page': num});
```

```
format_plural(
    $count,
    '1 year', '@count years'
);
```

```
Drupal.formatPlural(
    count,
    '1 year', '@count years'
);
```



DRUPAL 6 TRANSLATION CHEAT SHEET V1.1 - NOVEMBER 15. 2007.

By Gábor Hojtsy – Check for new versions at hojtsy.hu – Freely redistributable as long as credit is kept.

Core PHP translation API

`t('Home')`	Translates 'Home' to the requested language.
`format_plural ($count, '1 comment', '@count comments')`	Returns the proper translation for the text based on plural rules in the requested language.

Optional variable replacements array and language code can be provided in this order.

JavaScript translation API

`Drupal.t('Home')`	Translates 'Home' to the language used for page display.
`Drupal.formatPlural (count, '1 comment', '@count comments')`	Returns the proper translation for the text based on plural rules in the page's language.

Optional variable replacements array can be provided, **language is always the page language**!

Other PHP functions using translation features

`format_date ($timestamp, 'medium', ...)`	Translates parts of the date/time formatting.
`format_interval ($timestamp, ...)`	Displays the time interval in human terms. Translates names of the segments.
`format_size($size, ...)`	Formats a file size provided in bytes. Translates the MB, KB and byte labels.

All functions accept an optional language code at the end.

Replacement formats (based on prefix characters)

`@count`	Replaced with the value escaped with check_plain() for output.
`%size`	Replaced with the value escaped and formatted as a placeholder (theme ('placeholder')).
`!html`	Replaced with the value as is, without any additional formatting. **Insecure! Only use if you already sanitized your text.**

Example: `t('%name moved to @directory', array( '%name' => $filename, '@directory' => $dirname);`

Text extracted for translation and used later

.info files	Module/theme name; description; package name.
`hook_perm()` implementations	Literal permission names, when provided with standard array syntax.
`hook_menu()` implementations	Menu item 'title' and 'description' keys, when provided as literal strings.
`watchdog('menu', 'Items saved')` calls	Log type and message, when provided as literal strings. Third parameter specifies the replacements array. **No plural support.**

Translating text in the installer

`st('Home')`	Works the same way as t(), but uses a file+memory backend. Use only in the installer and in install profiles. **No plural support.**
`$t = get_t(); $t('Home');`	Use get_t() in .install files and any other code which could run in the installer too. It returns 't' or 'st' depending on the function to be used at the moment.

Remember: if you have an .install file and need to use translation, your function calls could easily end up running in the installer if your module becomes a core module in an install profile. Use get_t()!

The most common mistakes

Using dynamic values, such as t($type). Whenever you use dynamic strings to call the above functions, make sure that all possible values are already in the code elsewhere in literal form using the API. **Do not translate user provided content.**

Reuse of the same message eg. for watchdog() and drupal_set_message() calls. The former disallows t(), the later requires t(), so this is not possible. **Always call the above API functions directly, without indirection.**

Usage of t() in a global context, such as define('CONST', t('...')). The locale system is not yet initialized at that time, so this usage is incorrect and also leads to nasty performance problems. Never do it.

Tools to verify you are using the API properly

http://drupal.org/project/potx	Drupal 6.x-2 or later versions have coder module plugin support, so improper use of the API is reported in reviews. Also feel free to export a template for your module/theme and check whether all strings appear.
http://drupal.org/project/l10n_client	The potx module cannot find mistakes when you forgot to use t() or some other function to translate your output. The localization client has a convenient interface to translate text displayed on the page. If certain strings are not showing up for translation, you forgot to use the API.

GET THE CHEAT SHEET FROM HOJTSY.HU

ALSO SEE D.O/NODE/322729

TEXT IS FOUND AND REPLACED RUNTIME, WHEN USED

	Hungarian	Croatian
Go to %page	%page. oldalra ugrás	...
1 year	1 év	
@count years	@count év	

Uhm, not so simple

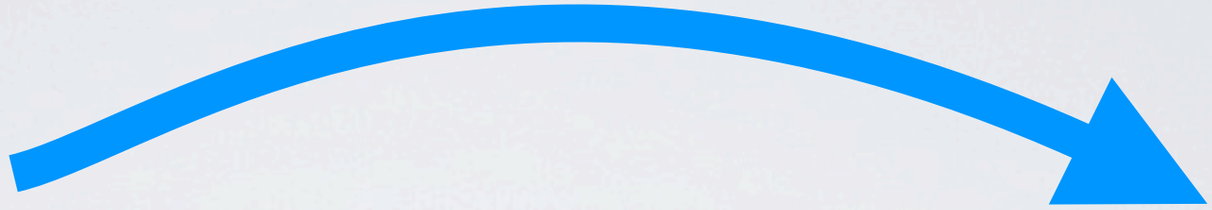
TRANSLATIONS ARE STORED IN THE DATABASE

DRUPAL PROCESS

Running source code

DRUPAL PROCESS

Running source code



Found text stored in DB

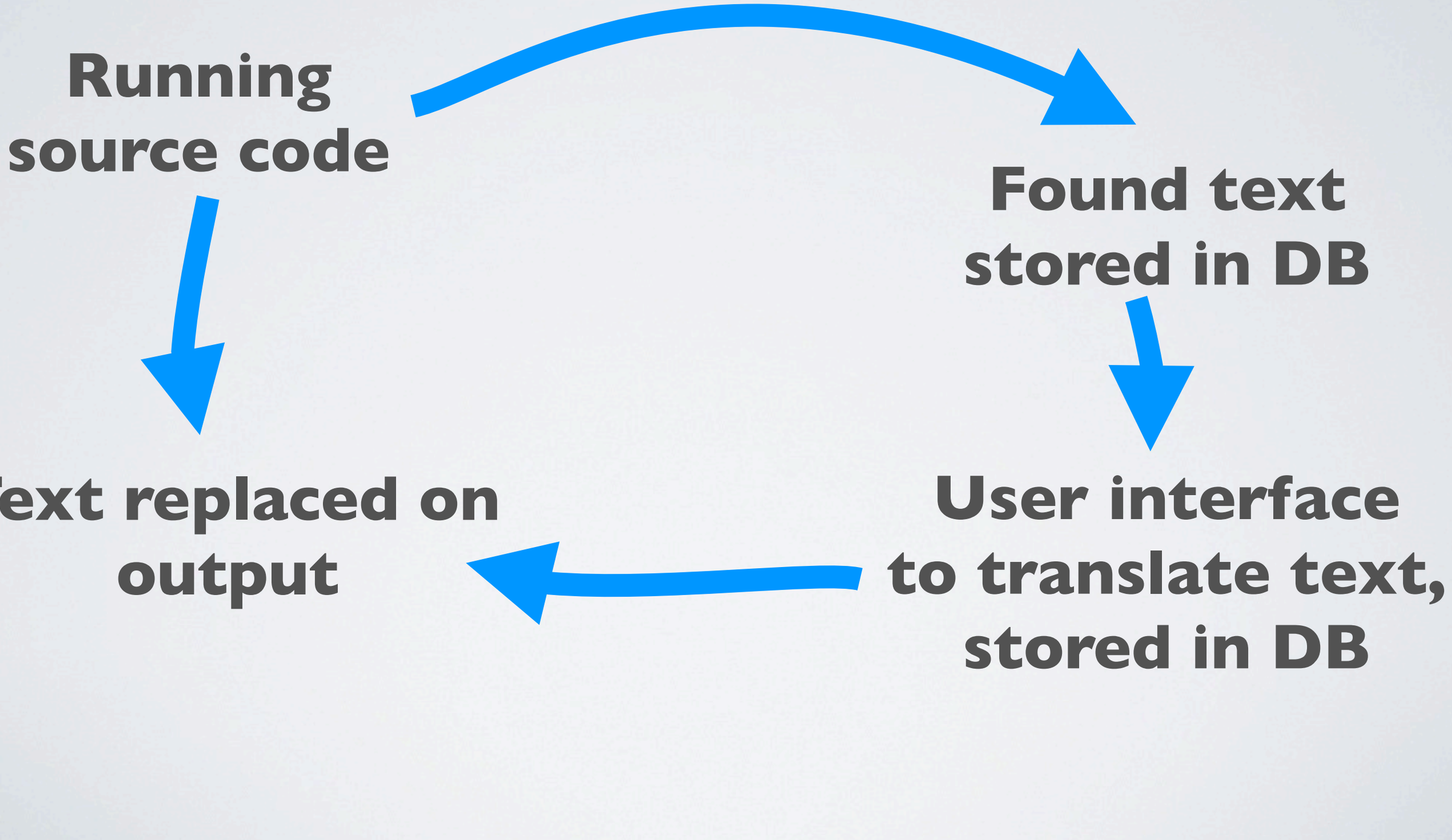
DRUPAL PROCESS

Running source code

Found text stored in DB

User interface to translate text, stored in DB

DRUPAL PROCESS

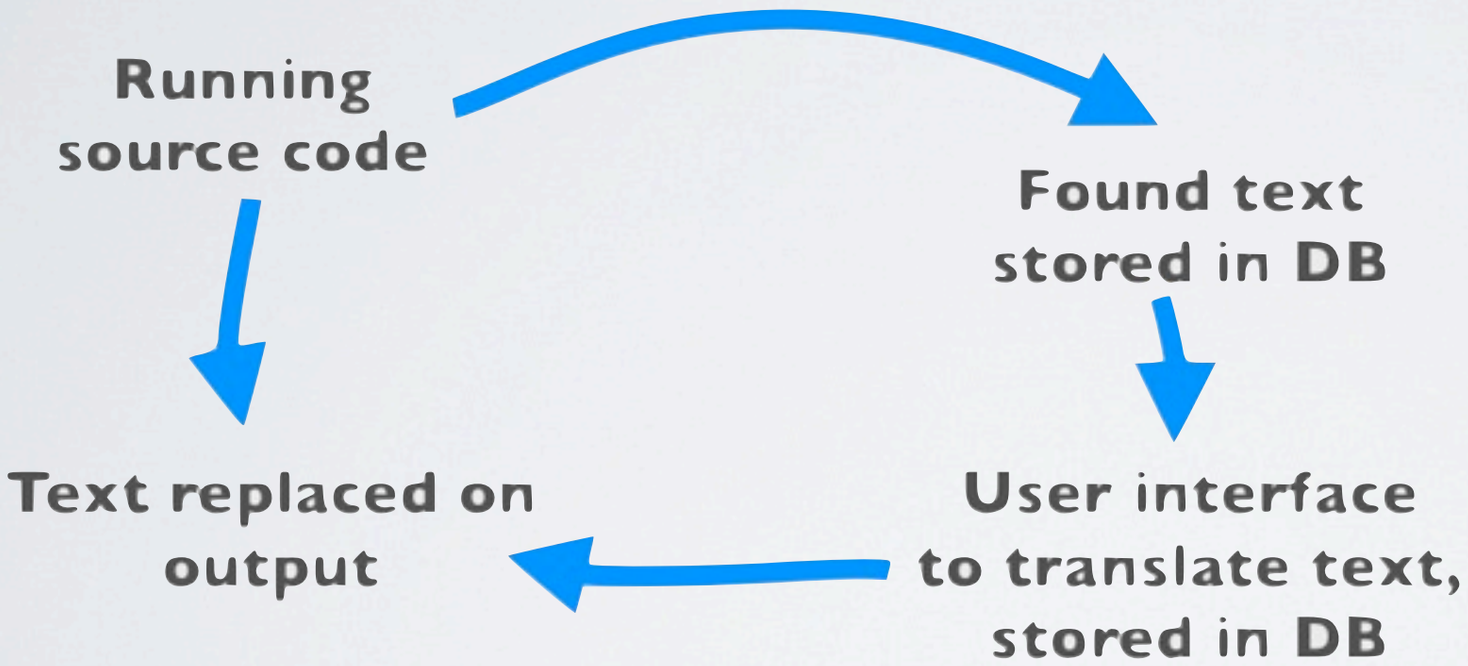


Where's the happy family?

TRANSLATIONS ARE SHARED IN GETTEXT .PO

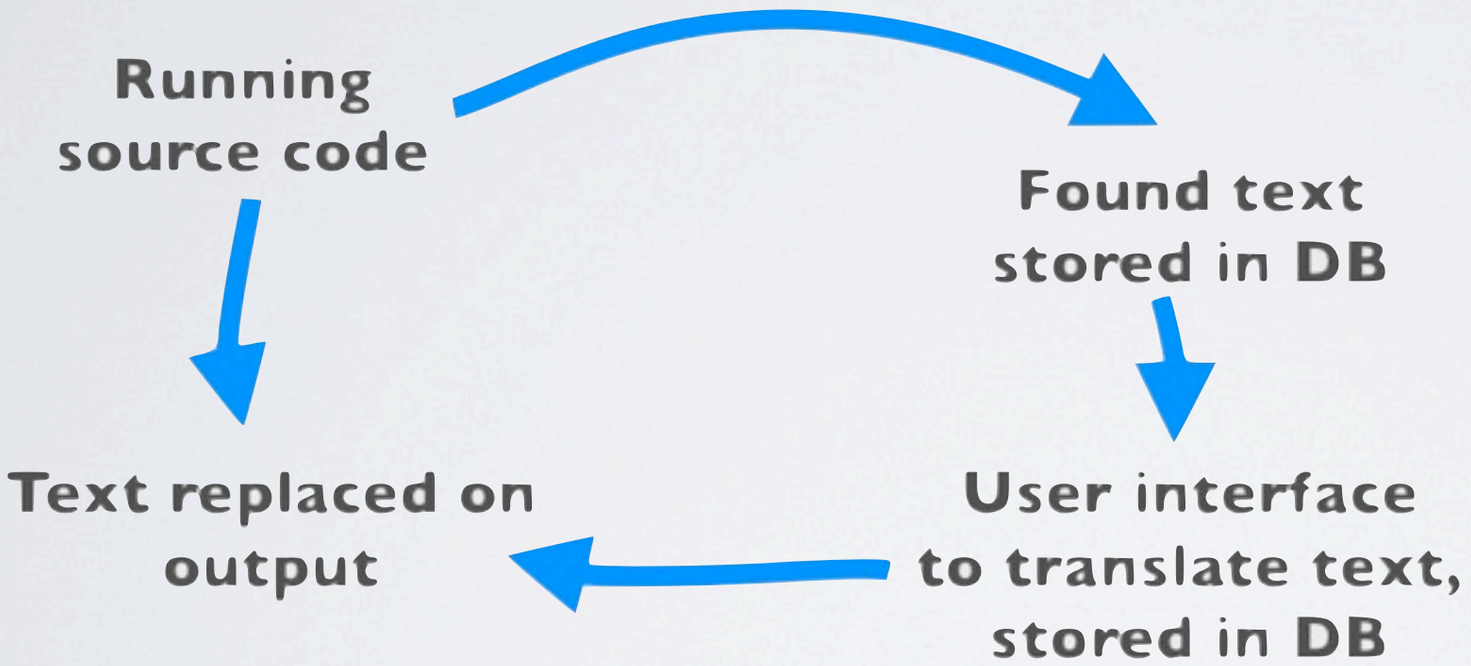
SHARING PROCESS

SHARING PROCESS

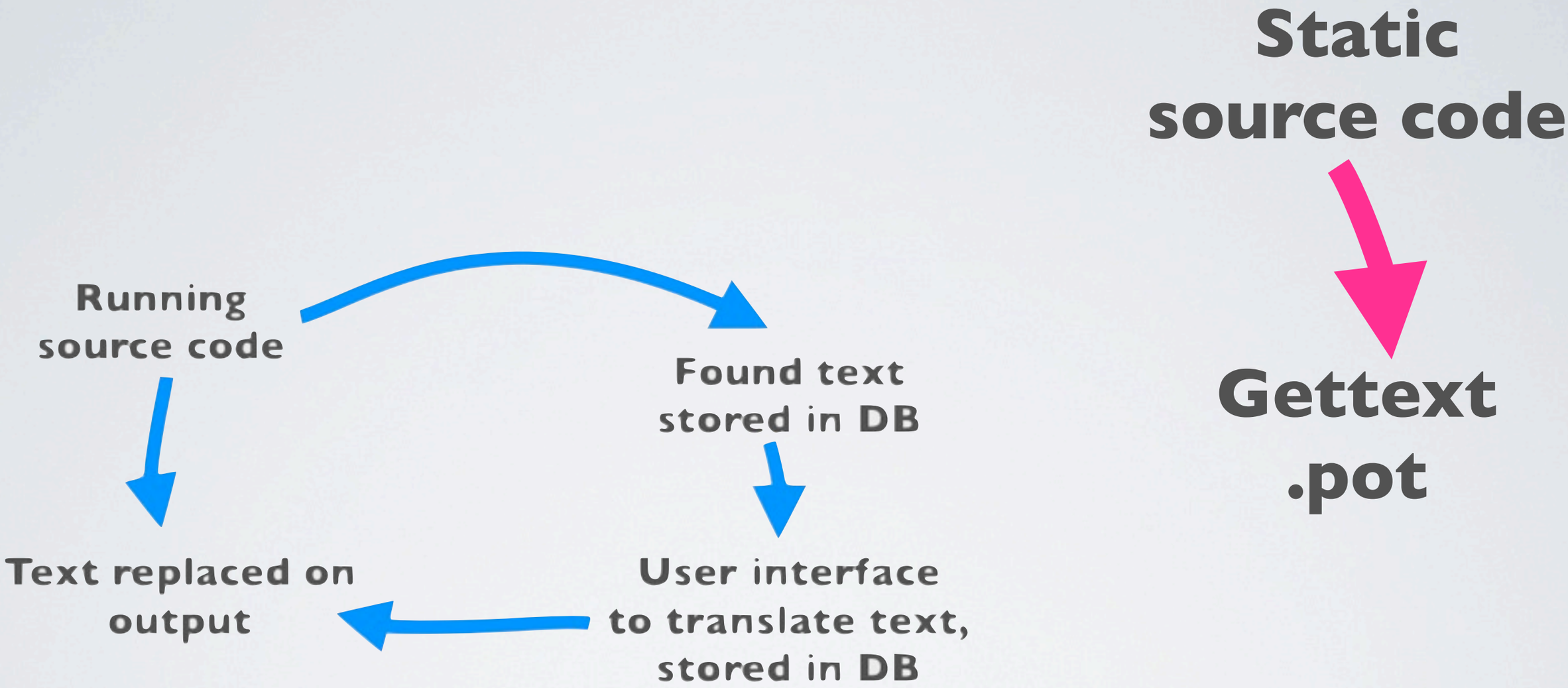


SHARING PROCESS

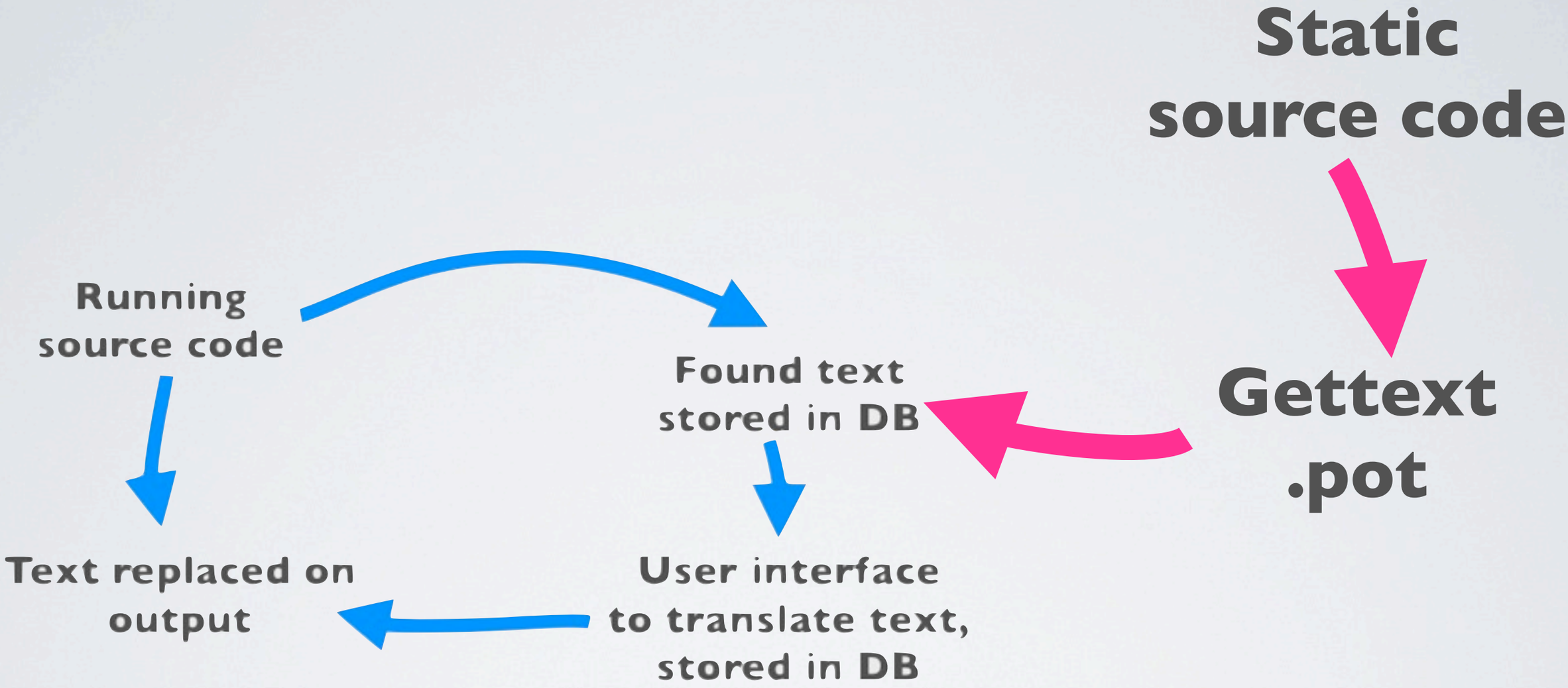
Static source code



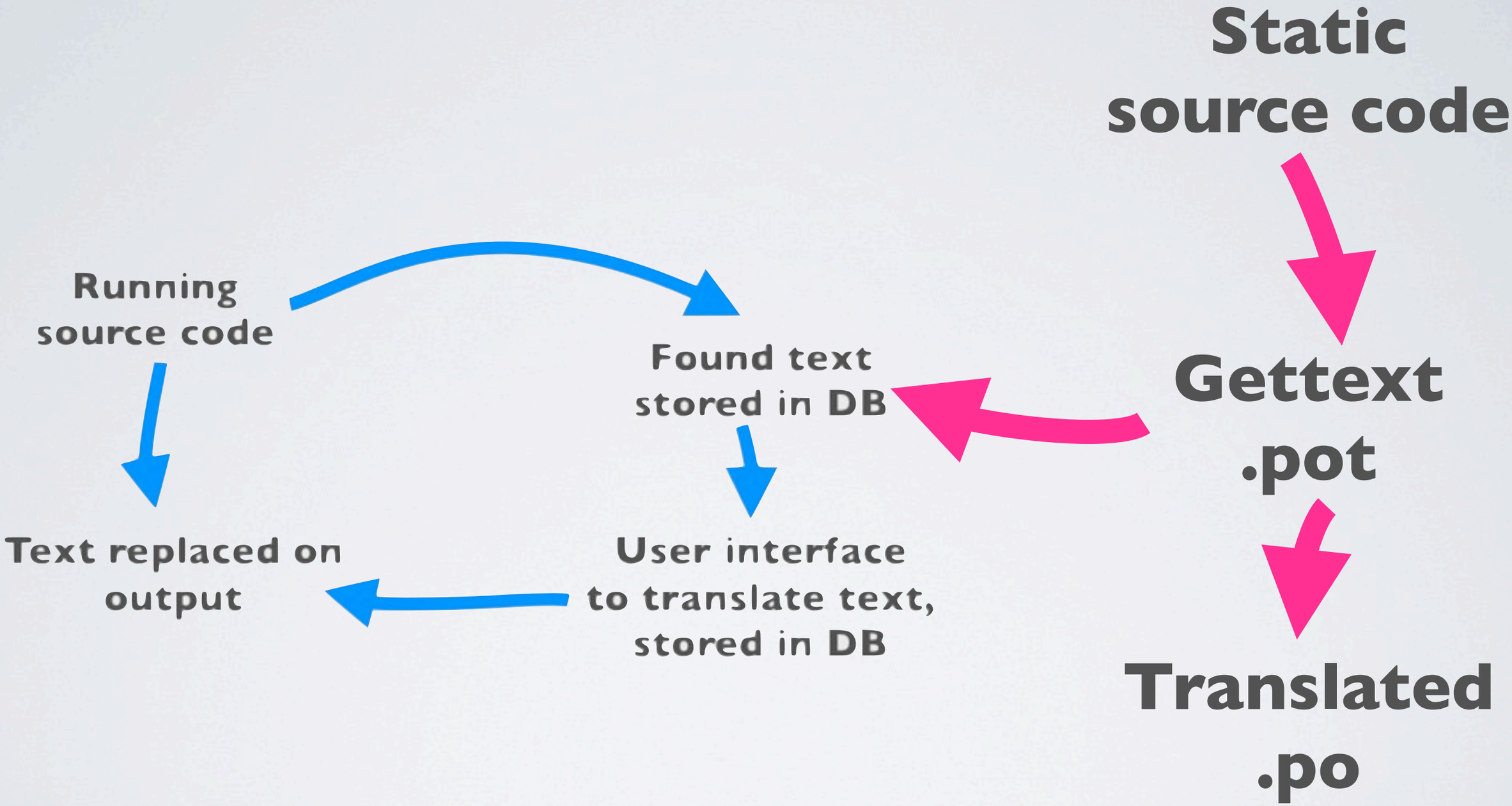
SHARING PROCESS



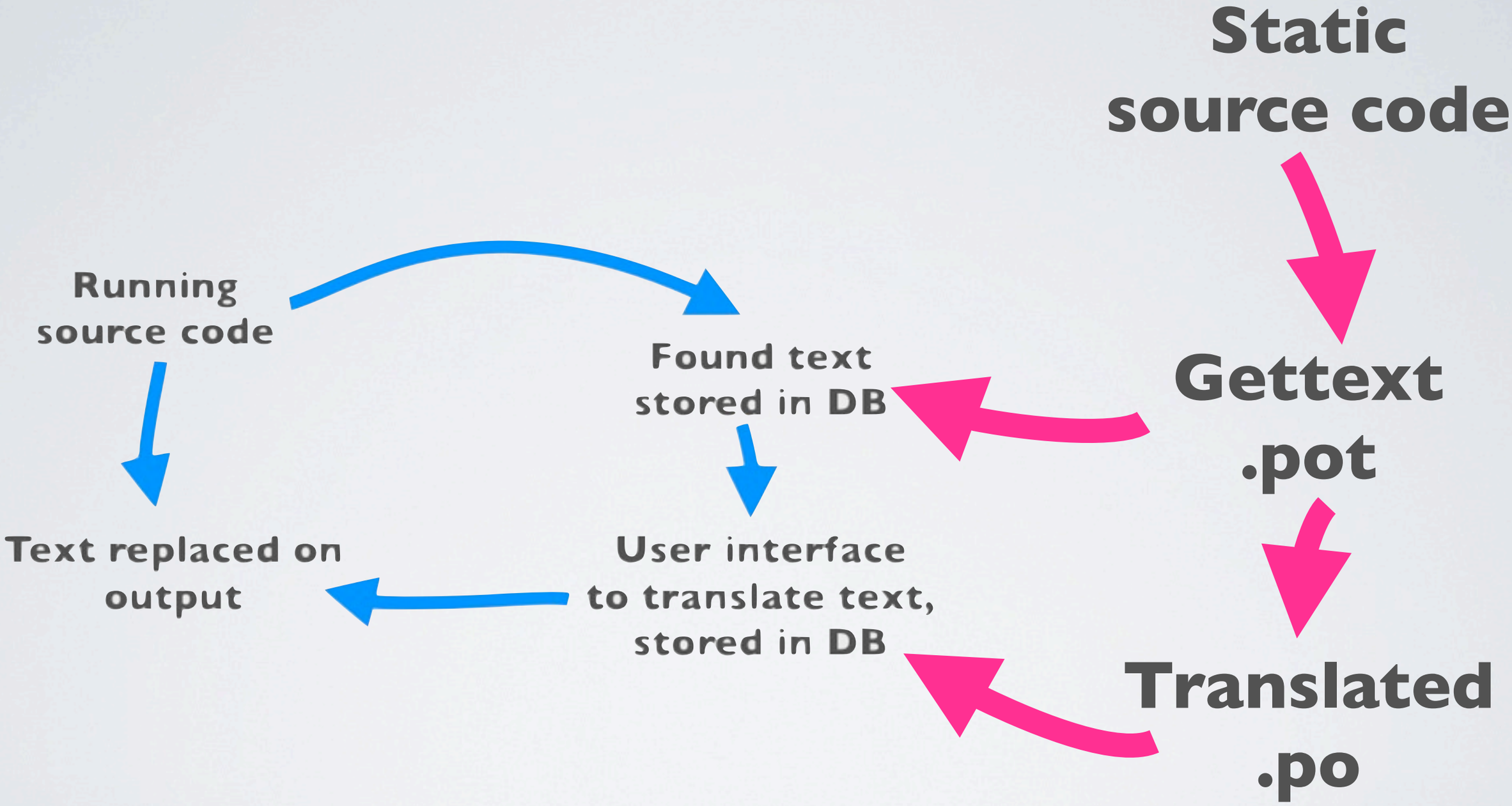
SHARING PROCESS



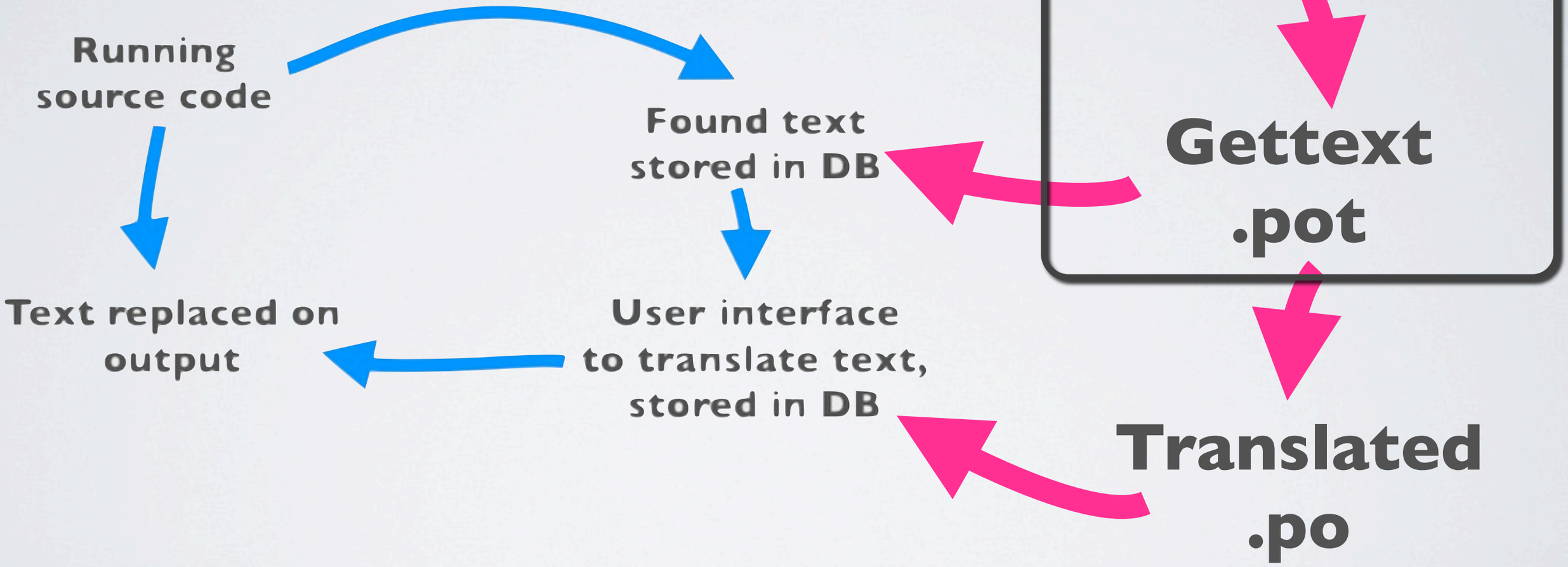
SHARING PROCESS



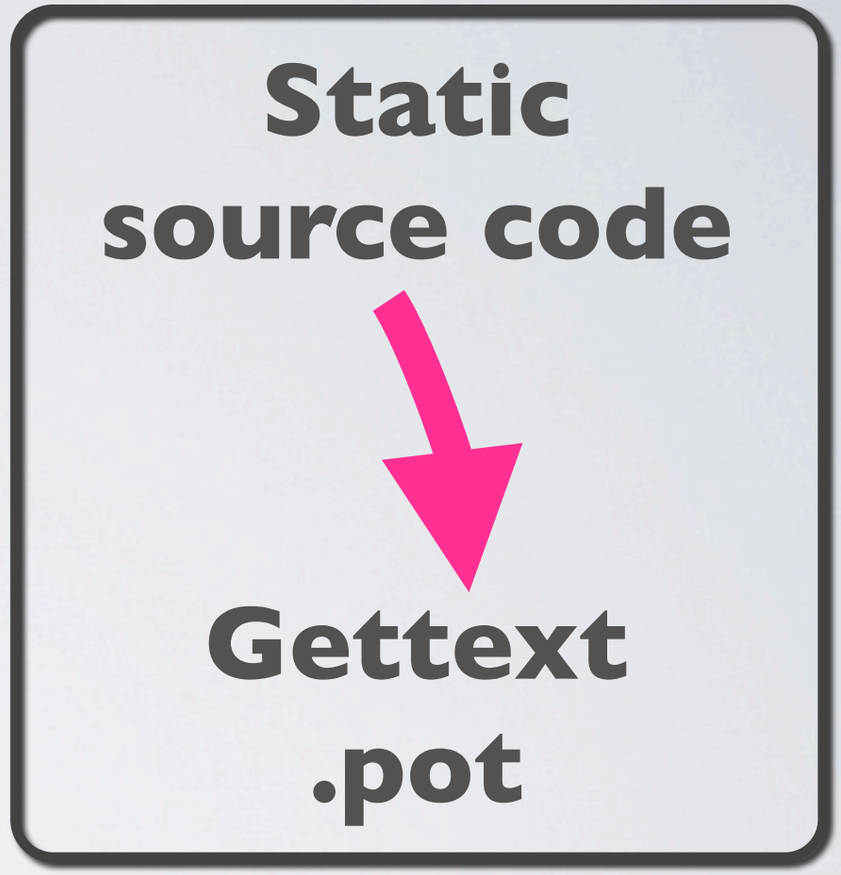
SHARING PROCESS



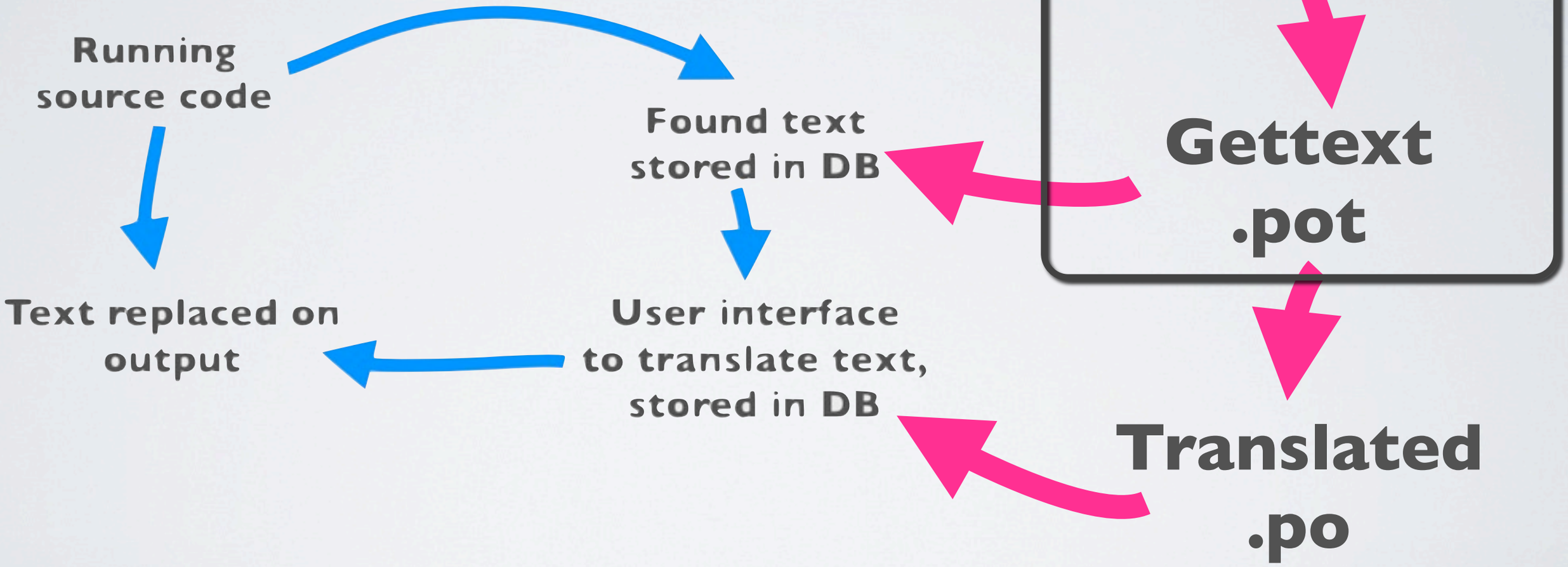
SHARING PROCESS



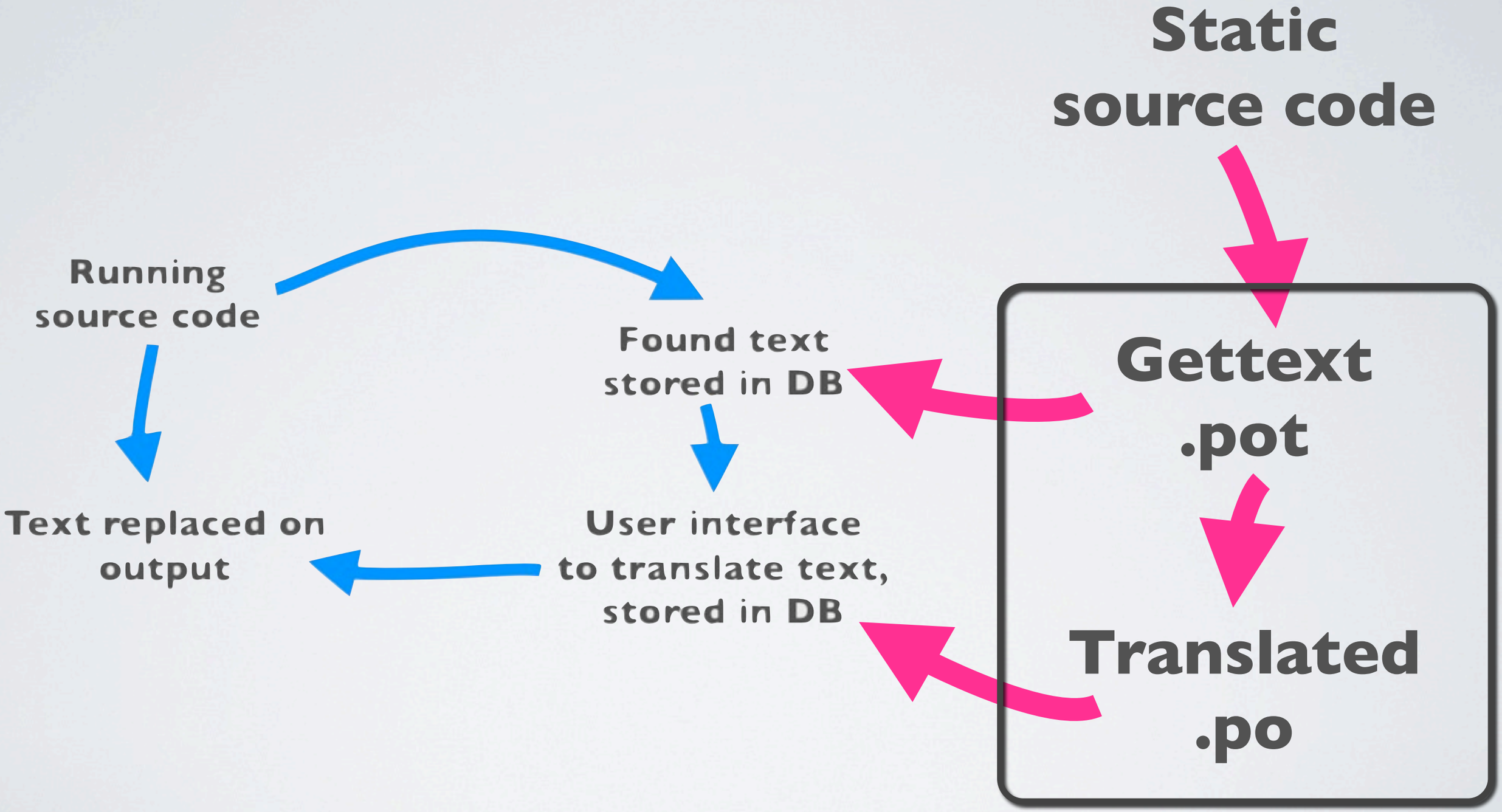
- Translation template extractor
- http://drupal.org/project/potx
- Web interface in your Drupal install
- Interfaces with coder review module to help developers



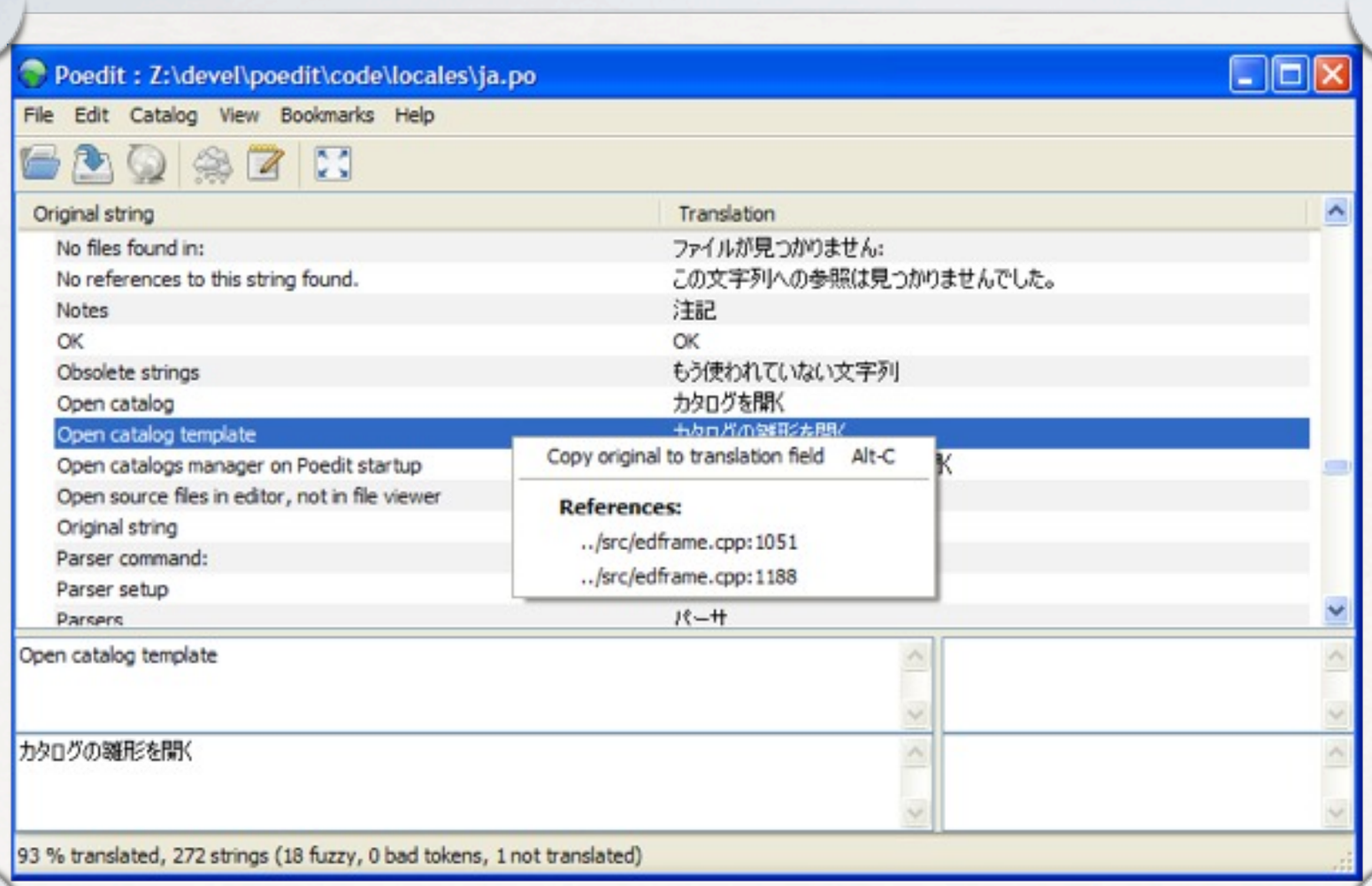
SHARING PROCESS



SHARING PROCESS



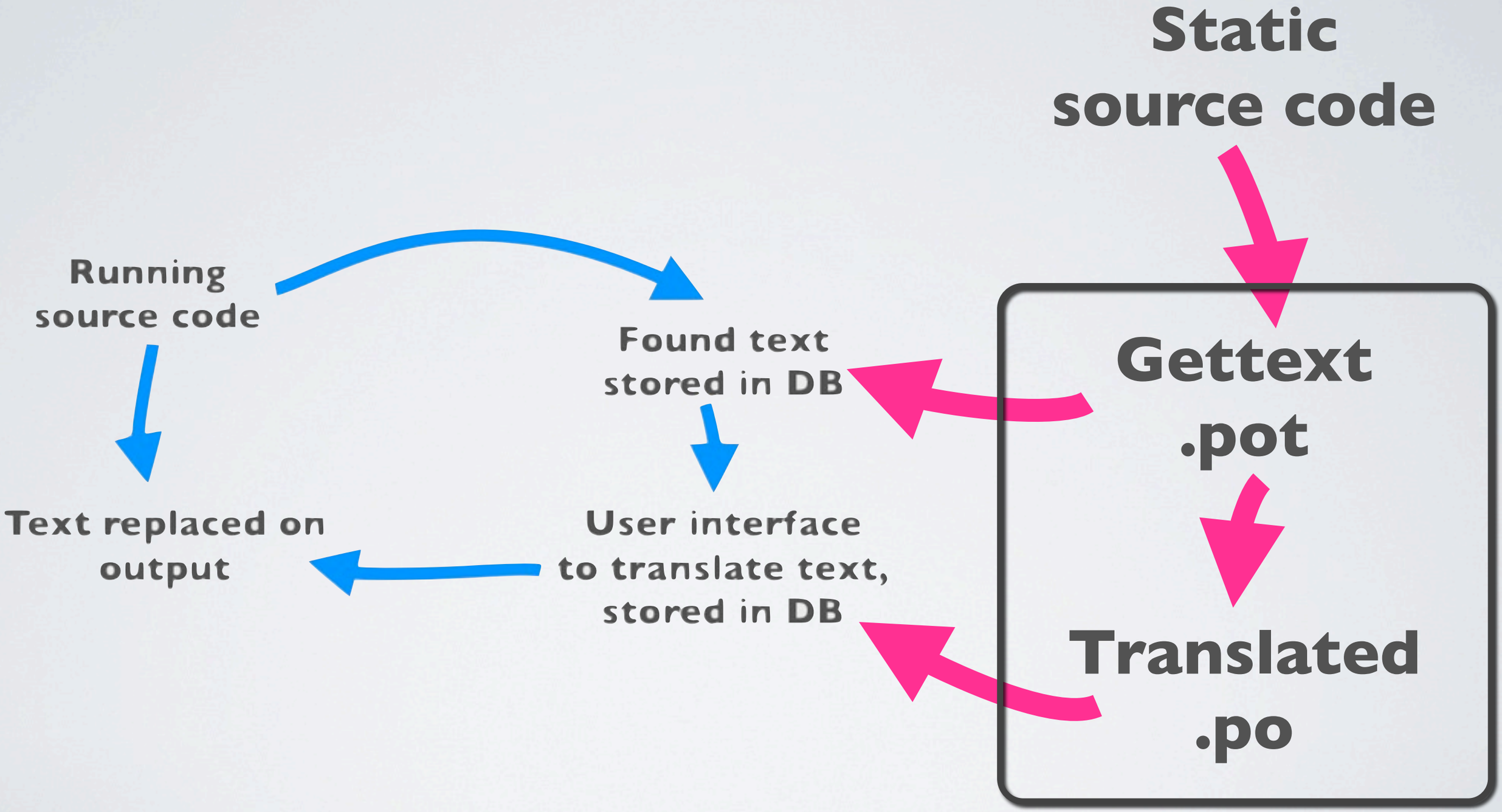
Desktop tools like PoEdit or just Notepad



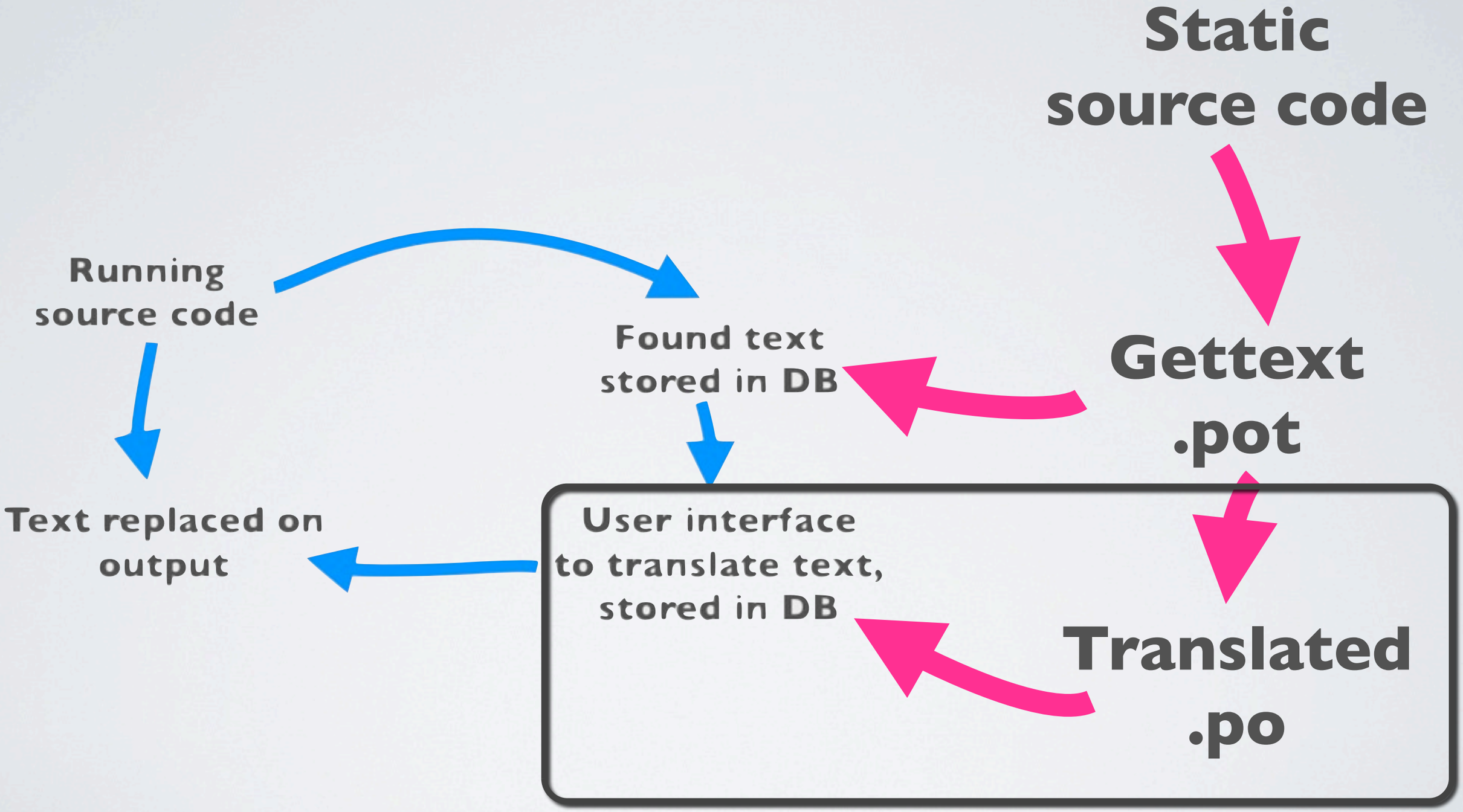
Gettext .pot

Translated .po

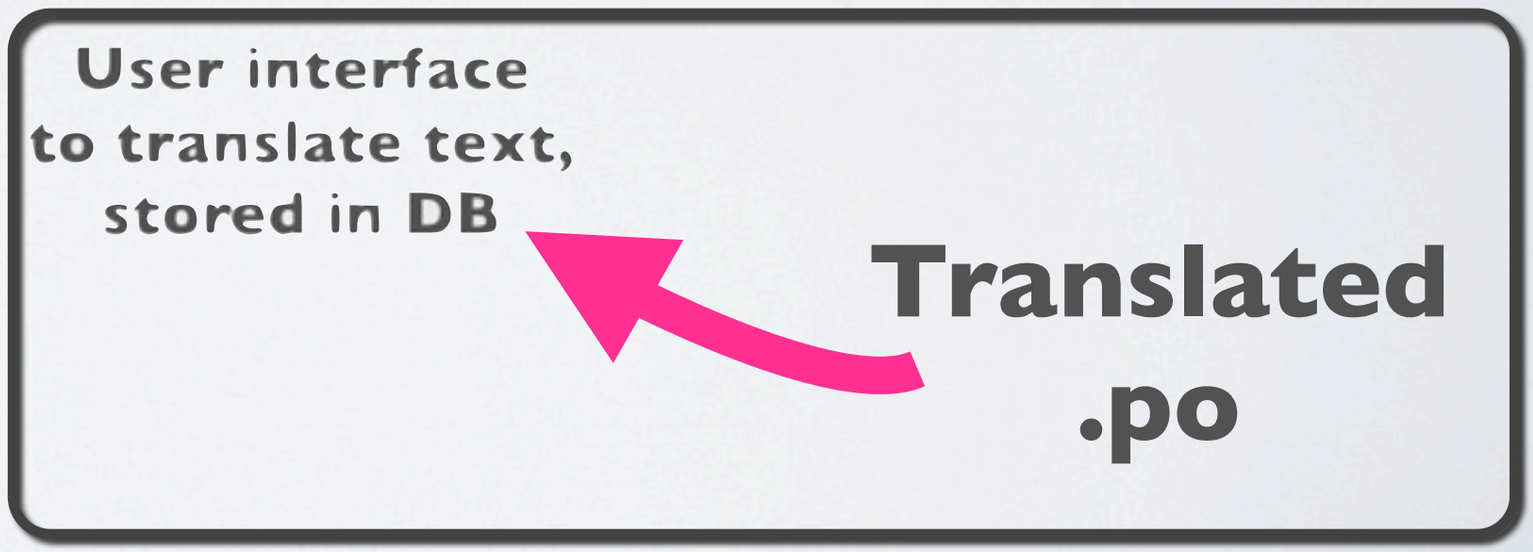
SHARING PROCESS



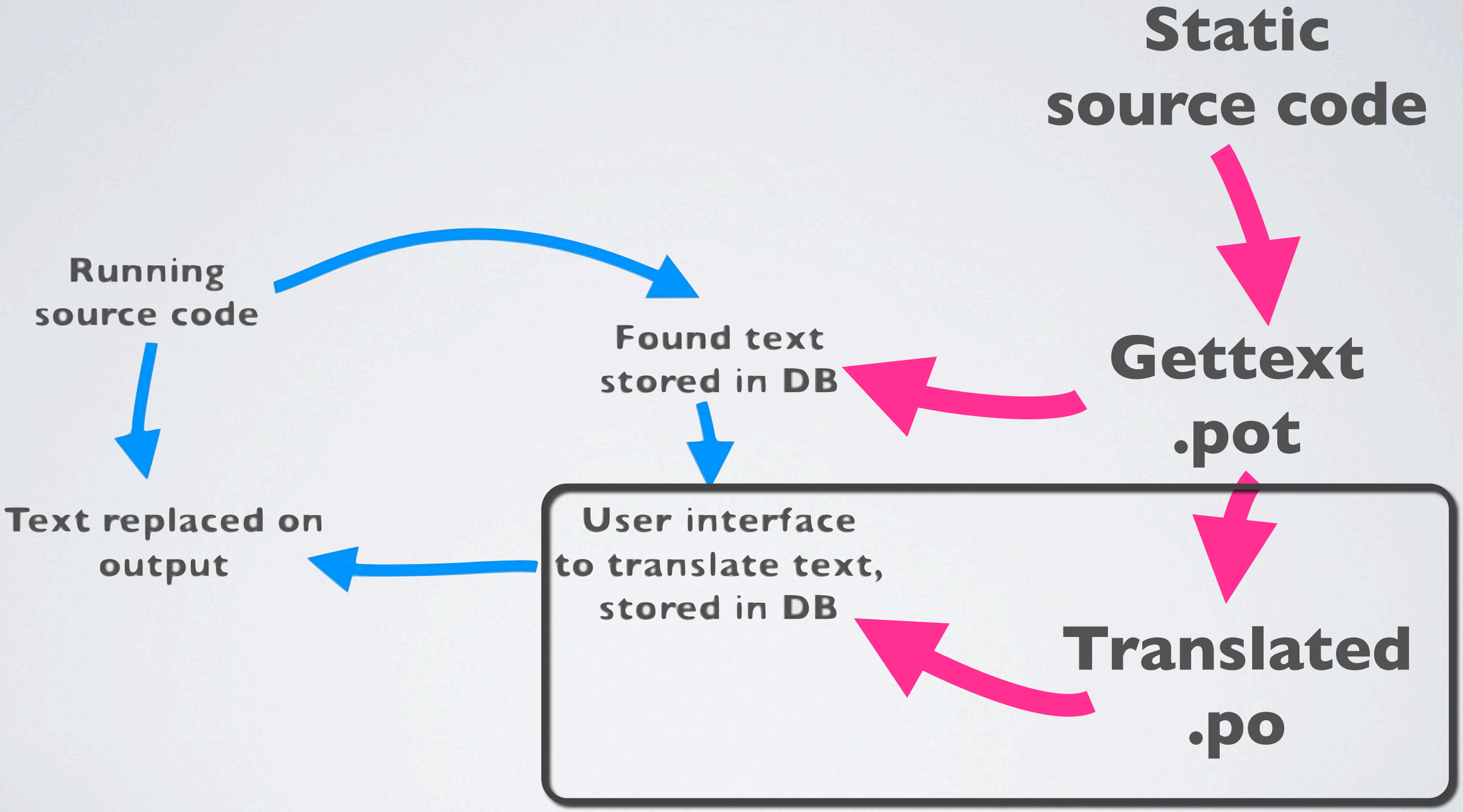
SHARING PROCESS



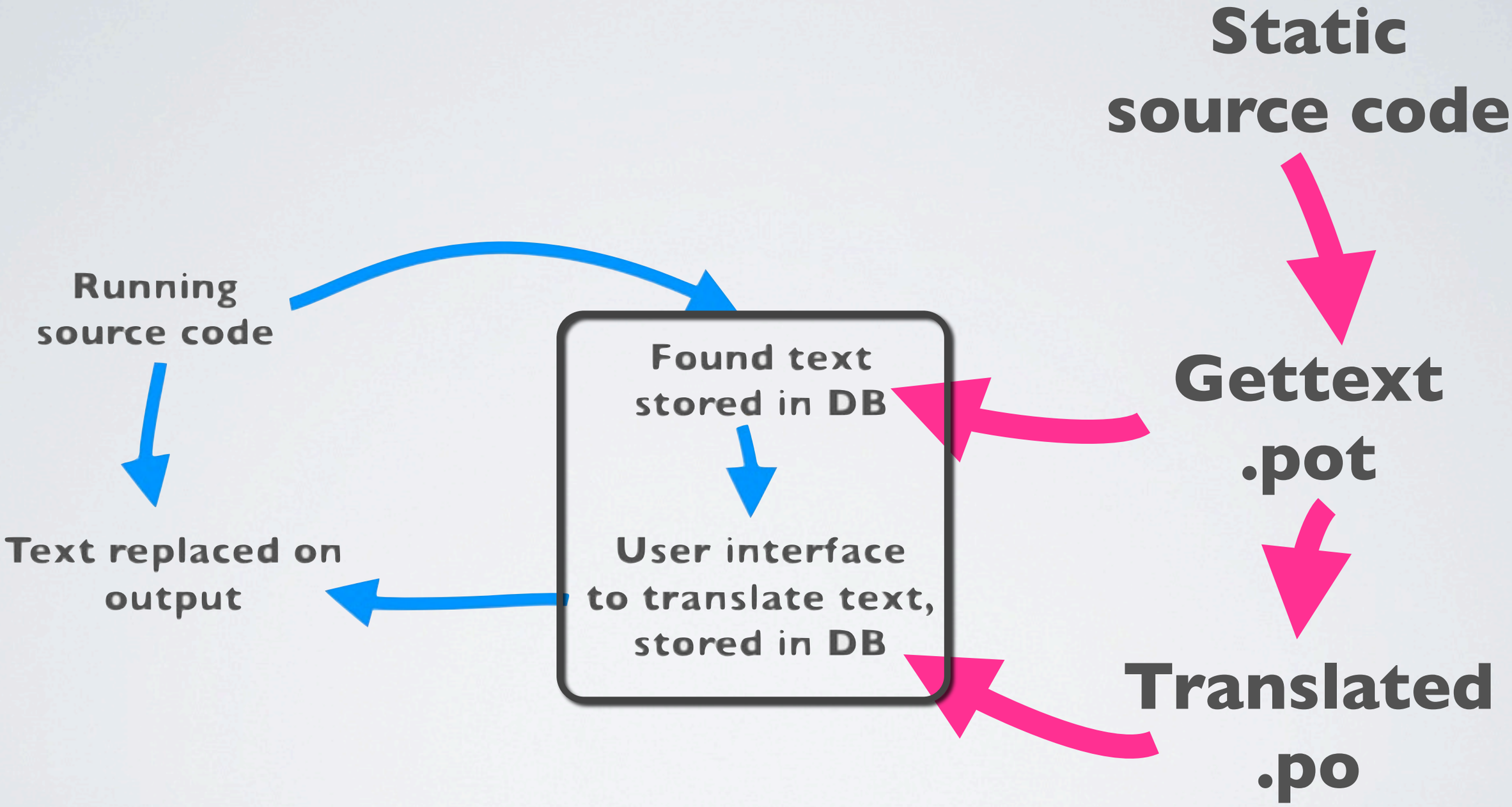
- Translation teams commit .po files to CVS
- http://drupal.org/project/Translations and under contributed projects
- Released on the parent project's timeframe
- Drupal imports automatically



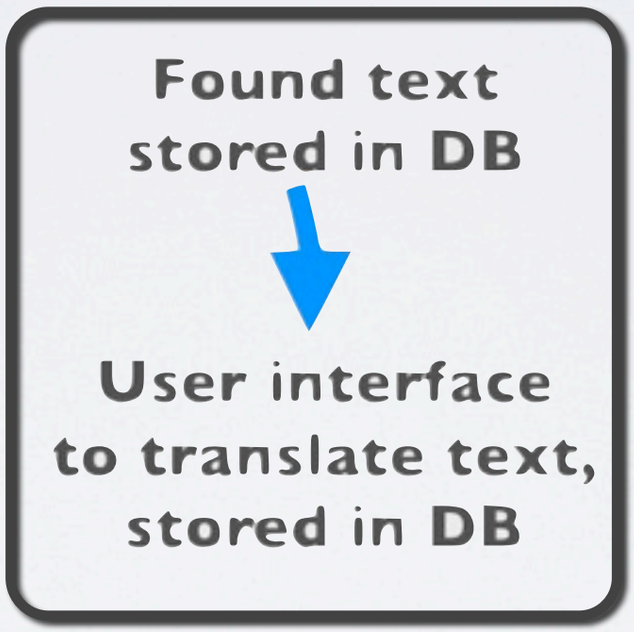
SHARING PROCESS



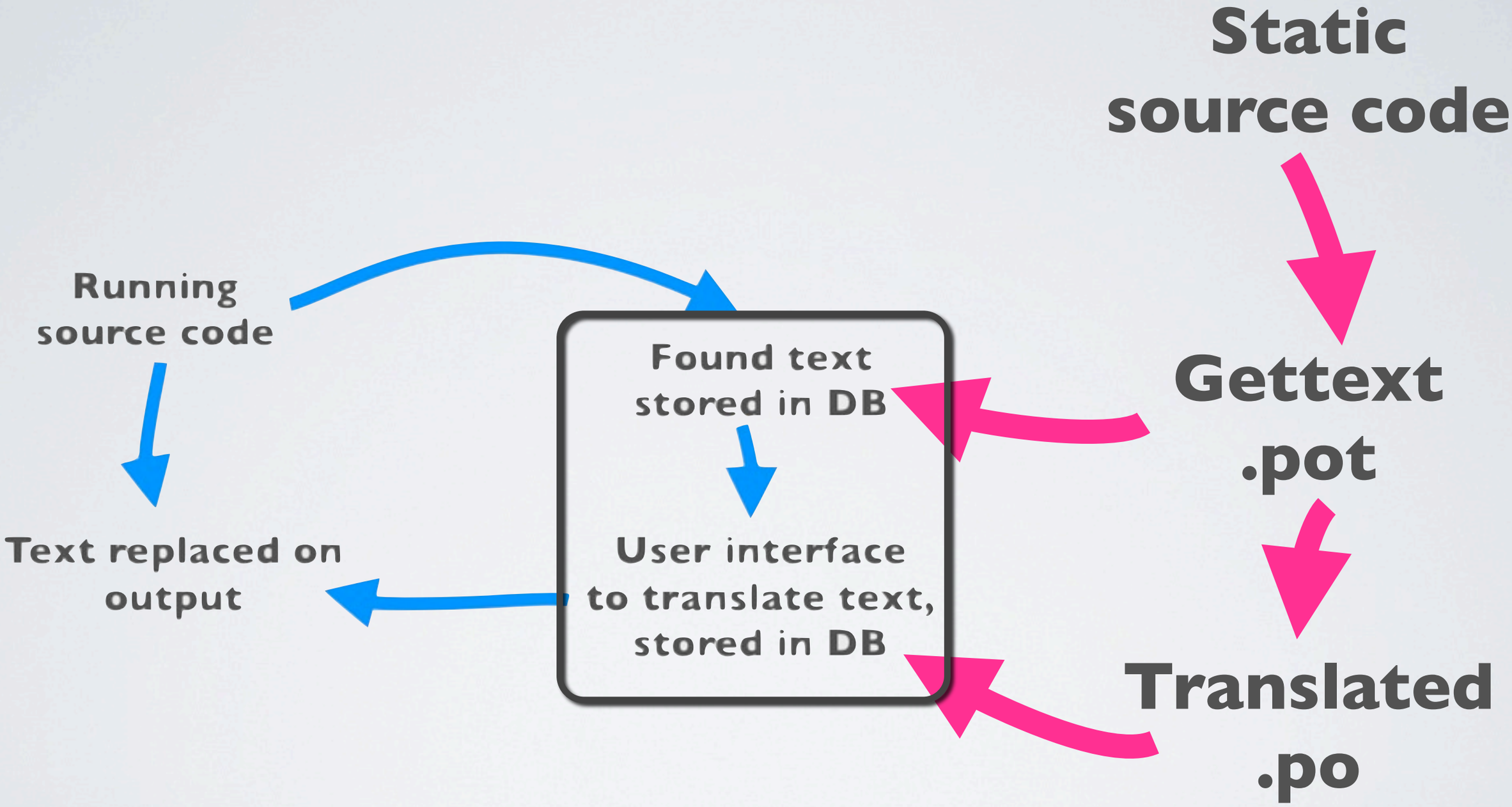
SHARING PROCESS



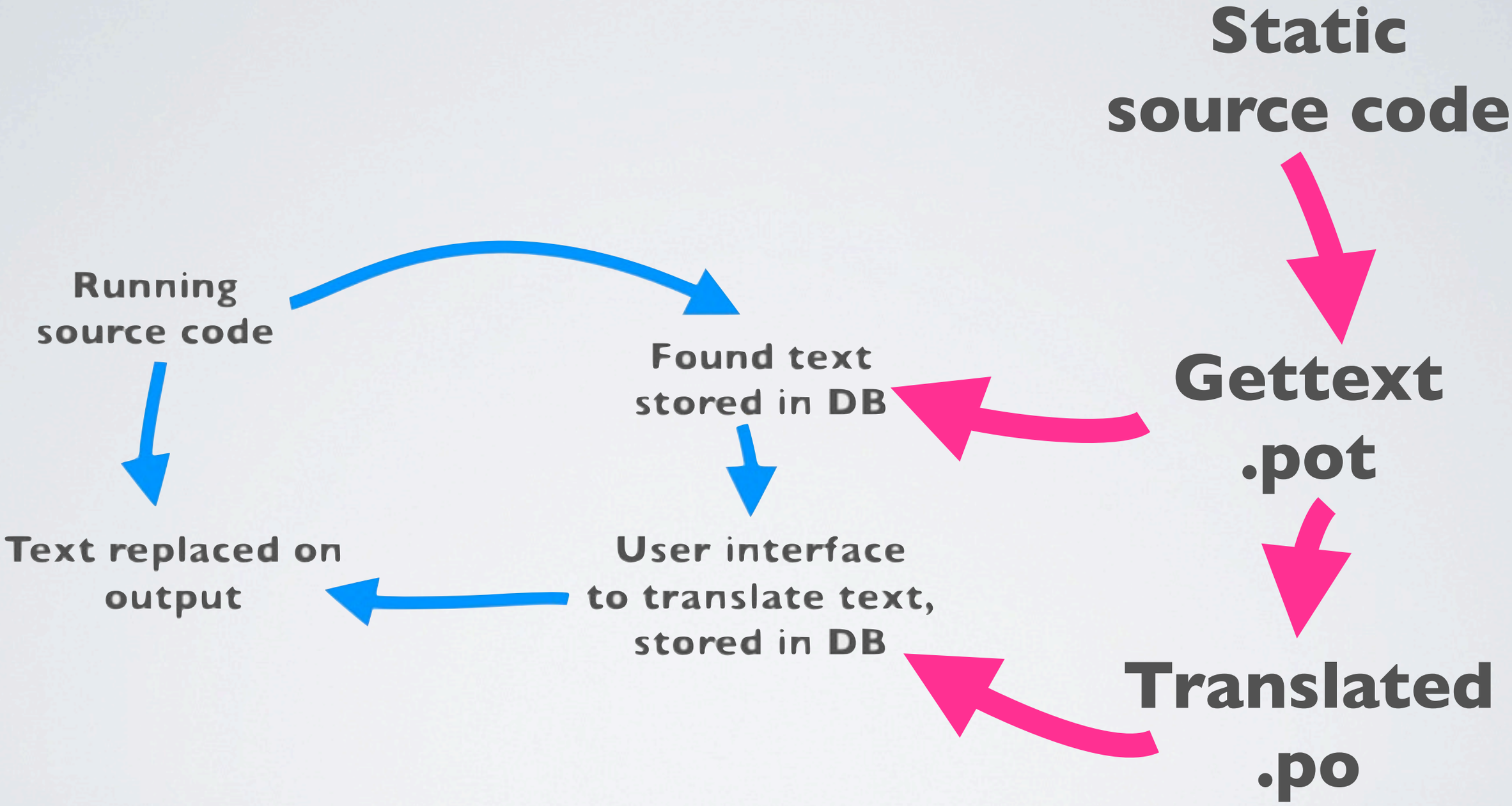
- Built-in translation interface
- http://drupal.org/project/l10n_client
- Import and export .po(t) files manually



SHARING PROCESS



SHARING PROCESS



All come together

GIVING AWAY (THIS WAY) IS HARD

POTX

POTX .PO

CVS

POTX .PO

CVS

POTX .PO

RELEASES

CVS

POTX .PO

RELEASES

BRANCHES

CVS

PLURAL FORMS

POTX

.PO

RELEASES

BRANCHES

CVS

PLURAL FORMS

POTX

.PO

SCHEDULING

RELEASES

BRANCHES

CVS

PLURAL FORMS

POTX

.PO

SCHEDULING

RELEASES

BRANCHES

ISSUE QUEUES

ENOUGH!

LOCALIZE.DRUPAL.ORG

Web UI for Drupal.org project translations

LOCALIZE.DRUPAL.ORG

Automated parsing of Drupal.org releases on publication

LOCALIZE.DRUPAL.ORG

Skips CVS, branching, templates, hides potx and plural forms, does not require desktop tools

LOCALIZE.DRUPAL.ORG

Single-sign-on with Drupal.org

LOCALIZE.DRUPAL.ORG

Organic groups based translations

LOCALIZE.DRUPAL.ORG

Different permission levels (open/controlled)

LOCALIZE.DRUPAL.ORG

Keeps import/export support with Gettext formats

LOCALIZE.DRUPAL.ORG

Provides a remote submission web service for Localization client

Oh, those stats!

almost

5000

projects known

over

5600

releases parsed

over

149000

strings to translate

149000

IT IS LIKE

TRANSLATING DRUPAL

43 TIMES

15

TRANSLATION TEAMS LAUNCHED IN BETA (MORE IN QUEUE)

Take a ride!

Envious?

SOFTWARE ON L.D.O

- **Translation template extractor**
  http://drupal.org/project/potx

- **Localization server:**
  http://drupal.org/project/l10n_server

- **Organic groups**
  http://drupal.org/project/og

TRANSLATE **YOUR** CODE

- **Open Atrium and Gallery 2 uses Localization server to translate their code**
- **You can set up l10n_server to parse your own code and translate in-house**
- **Eg. https://translate.openatrium.com/**

Long way to go

Color the future!

THESE SLIDES ARE (CC)

Photo credits:

http://www.flickr.com/photos/sinkdd/2853334715/

http://www.flickr.com/photos/chavals/2978360981/

http://www.flickr.com/photos/koesbong/125933266/

http://www.flickr.com/photos/cheishichiyo/3637712921/

http://www.flickr.com/photos/paky/52194393/

http://www.flickr.com/photos/patrickgage/3738107746/

http://www.flickr.com/photos/zhzheka/1240470939/

http://www.flickr.com/photos/fchouse/3270958643/

http://www.flickr.com/photos/colinj/490622069/

http://www.flickr.com/photos/brartist/313272029/

This presentation is © Gábor Hojtsy
Licensed: http://creativecommons.org/licenses/by-nc-sa/2.0/

THANK YOU!

Gábor Hojtsy, Acquia
gabor@hojtsy.hu
http://hojtsy.hu/