Universal PHP Framework

toKernel is a fully Object-Oriented extensible framework.
By design, the framework's kernel is monolithic which brings a lot of advantages. It is NOT a loader for a mere collection of classes. The versatility of this framework allows you to create web applications, command line applications and run the same application in either mode including interactive mode.

The main goal is to 'Make website or CMS developments simple and flexible'.

http://www.tokernel.com
framework@tokernel.com

Features

To this end, this project provides all necessary functions and features for creating websites / CMS or Command line application including:

- Multiple instances of applications / websites
- Templates, widgets, views, themes (as in many CMS)
- CLI with interactive mode
- Add-ons, modules, Libraries
- Aliasing for CLI and HTTP
- Flexible Error handling and logging
- Security: Filter, Validation
- Debugging

And more...

Requirements

GNU/Linux or other Unix like OS / Windows 2000 or newer.

PHP version 5.2.x or newer.

Apache (httpd) Web server version 2.x.

PHP CLI SAPI for command line interface mode.

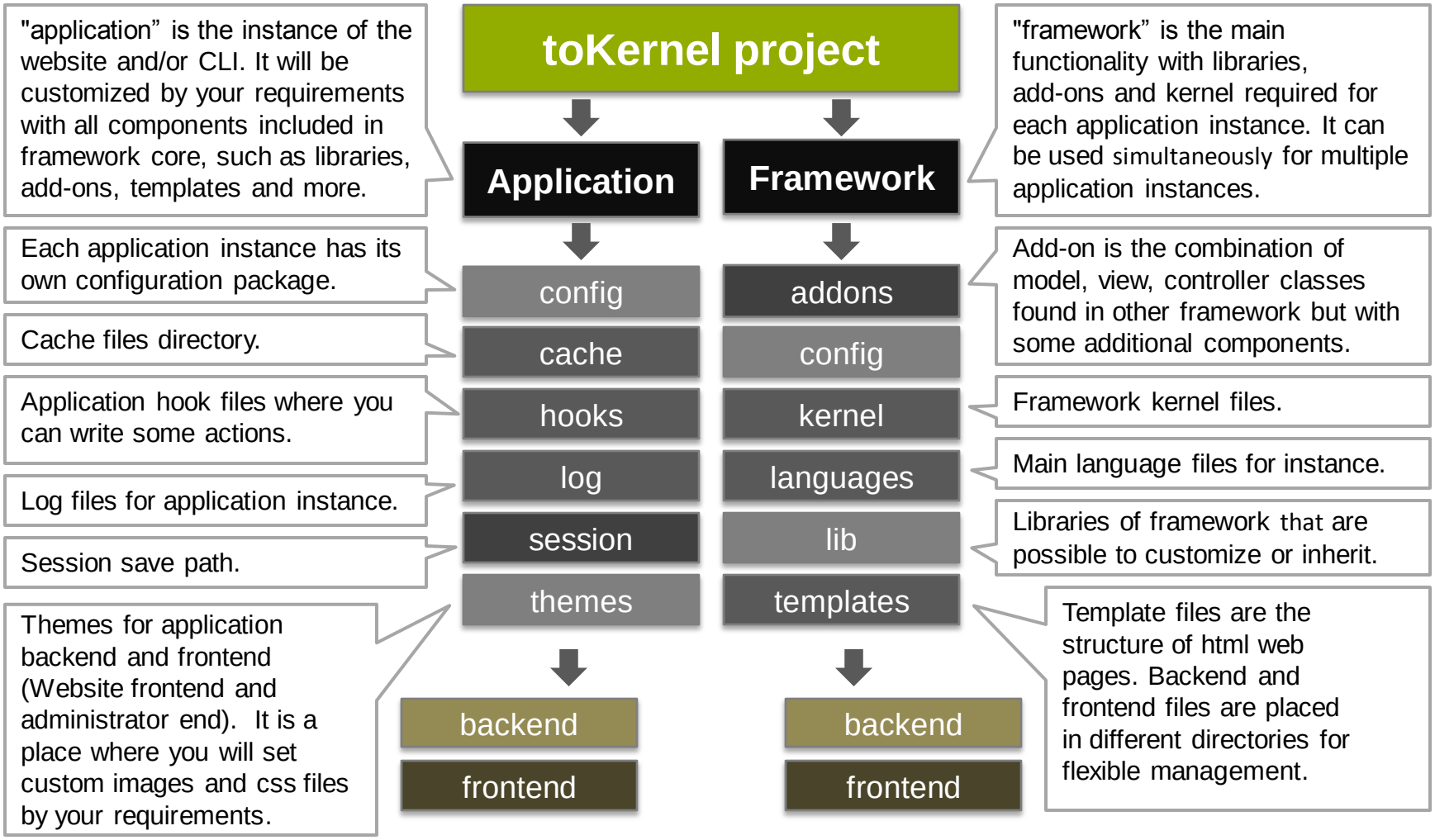
Apache module mod_rewrite for web interface mode.

PHP extensions: mysql, zlib, gd, mbstring.

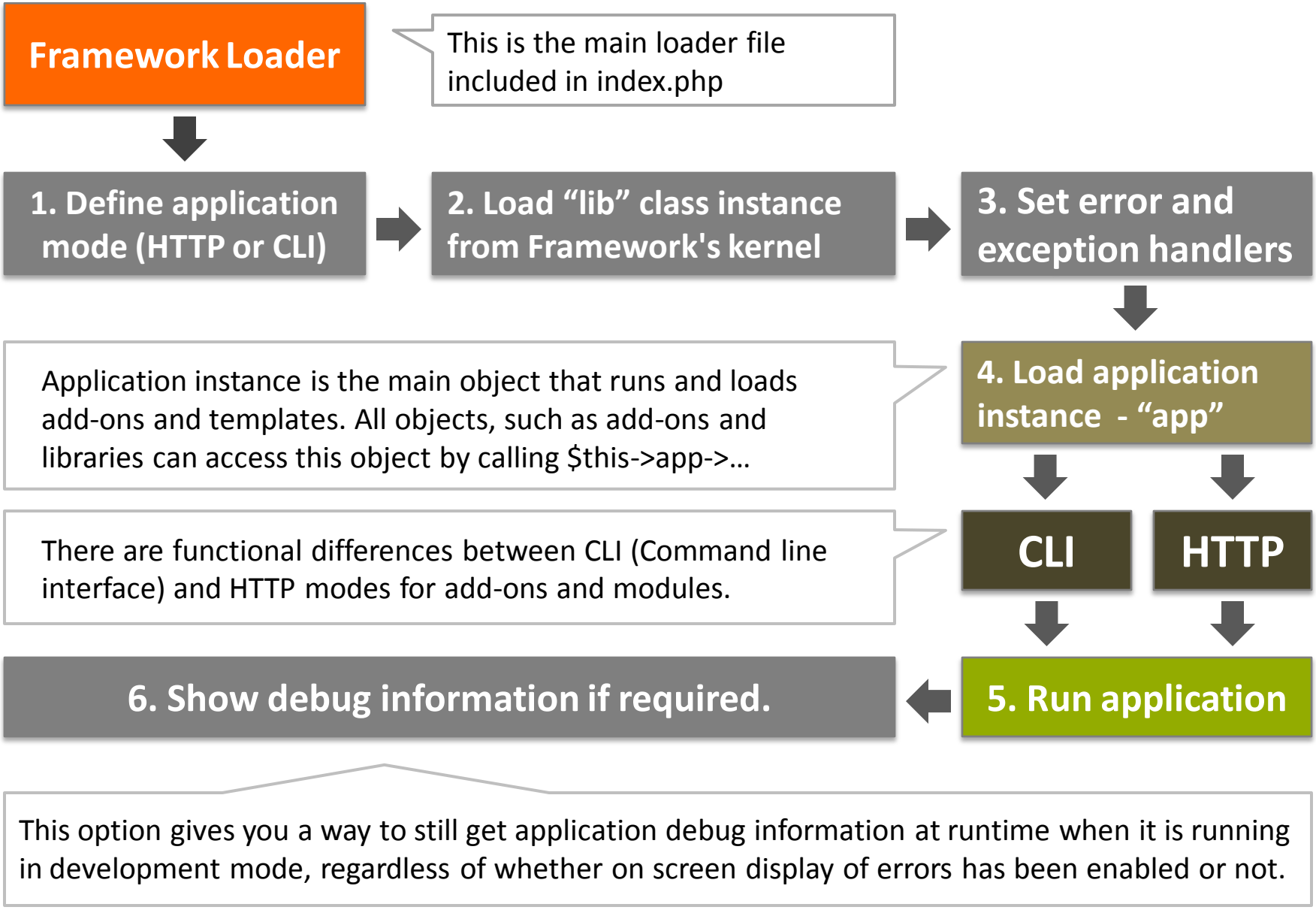
Project license

What is in the project?

There are two main directories in project: application and framework. Some directories such as "addons", "lib", "languages", "templates" can be placed in both directories, in application and framework. If the file exists in both, then by default system loads a file from application.



How does it work?



What is “lib” class instance?

The “lib” class instance is the object which is accessible from any part of application, as “$this->lib”. It is the loader of all class libraries.

To access library call "$this->lib->lib_name" expression, without “include” operation.

Example: Accessing URL Parameter named “offset”.

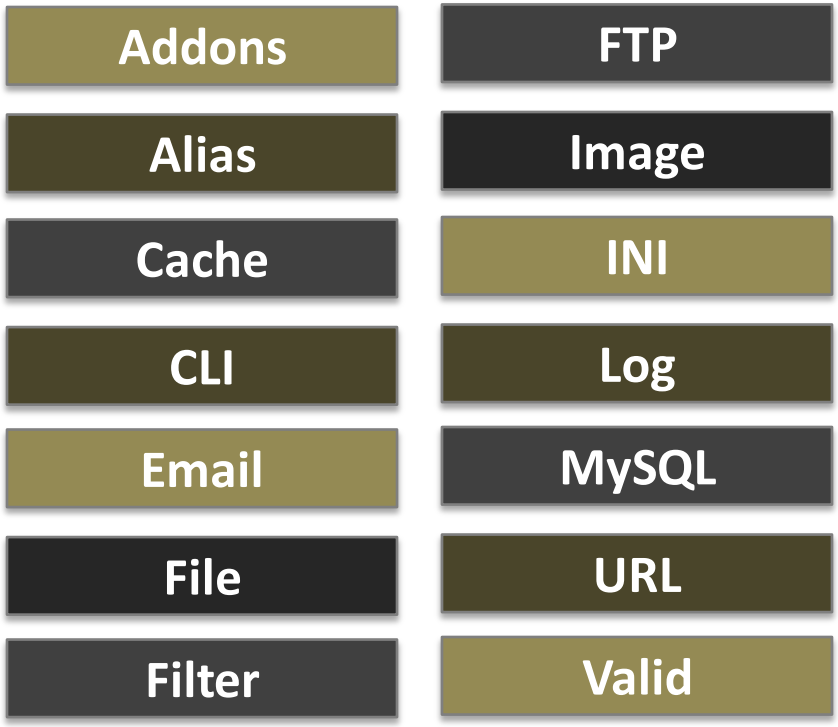
```
$offset = $this->lib->url->params(‘offset’);
```

Some possibilities.

1. Overriding existing library.
   If the library exists in framework and application (custom) libraries directories, then library loader loads from application directory.

2. Integrate library which inherits standard existing library.

Some libraries of toKernel framework.



3. It is also possible to develop and integrate new libraries.

What is application instance, and how is it loading?

The “app” class instance is the main singleton object of application, which is accessible from any part as “$this->app”.

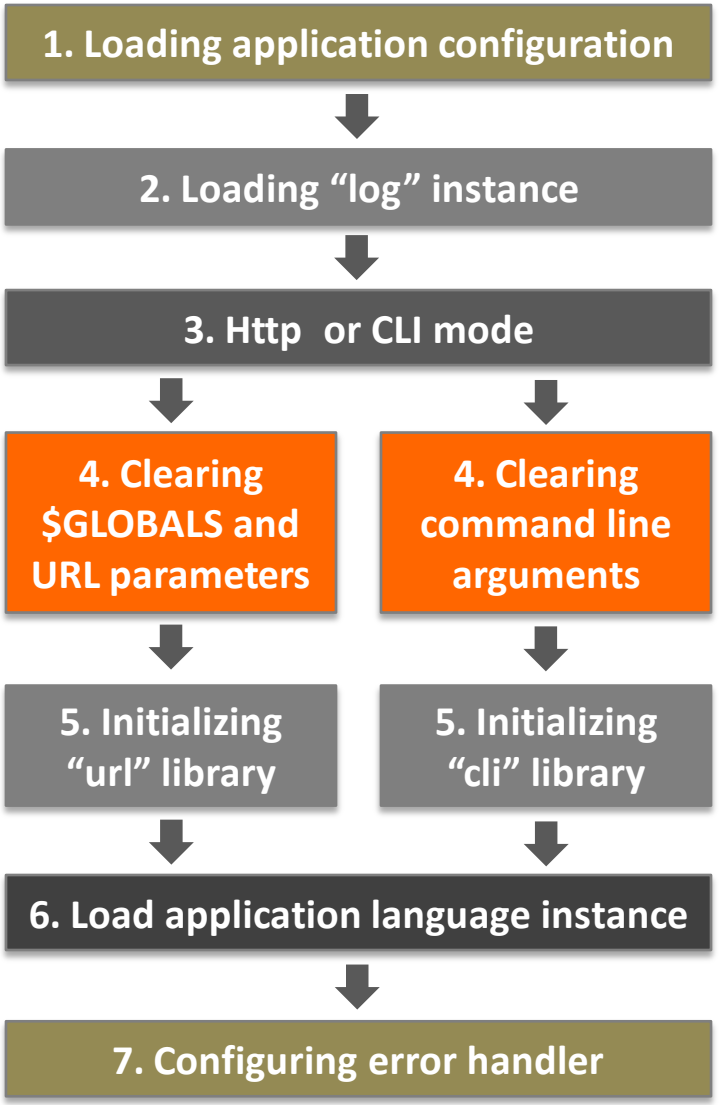
Example: Call redirect function of application.

```
$this->app->redirect($url);
```

Application Instance() method, is the main initialization step, which is called at once from framework loader.

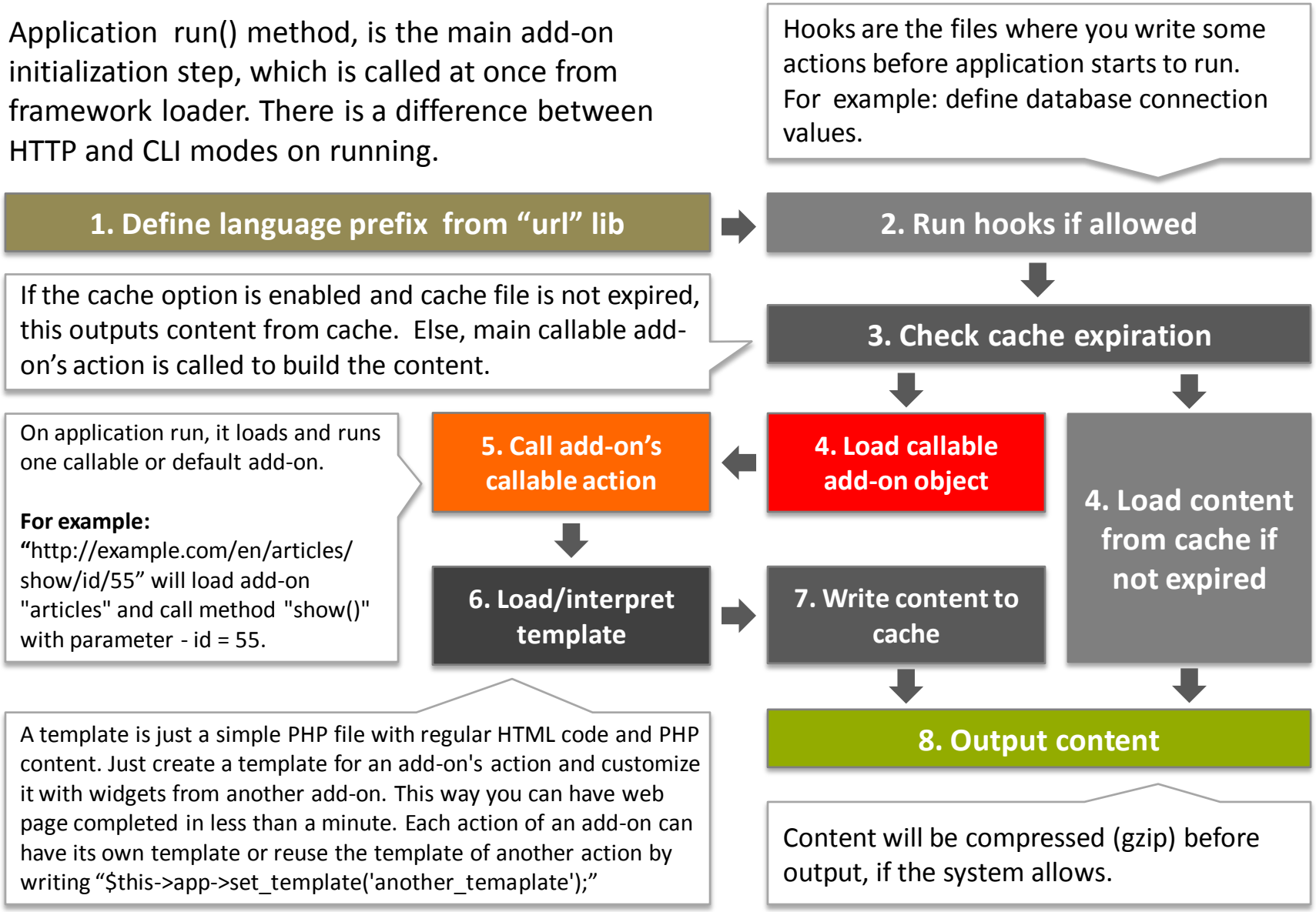
"url" library parses all arguments received from URL as - language prefix, main callable add-on’s id, add-on’s action (method which should be called) and other arguments.

"cli" library parses the same values as "url" library, but from command line interface.



How is the application instance running in HTTP mode?

Application run() method, is the main add-on initialization step, which is called at once from framework loader. There is a difference between HTTP and CLI modes on running.



How is the application instance running in CLI mode?

Application run() method, is the main add-on initialization step, which is called at once from framework loader. There is a difference between HTTP and CLI modes on running.

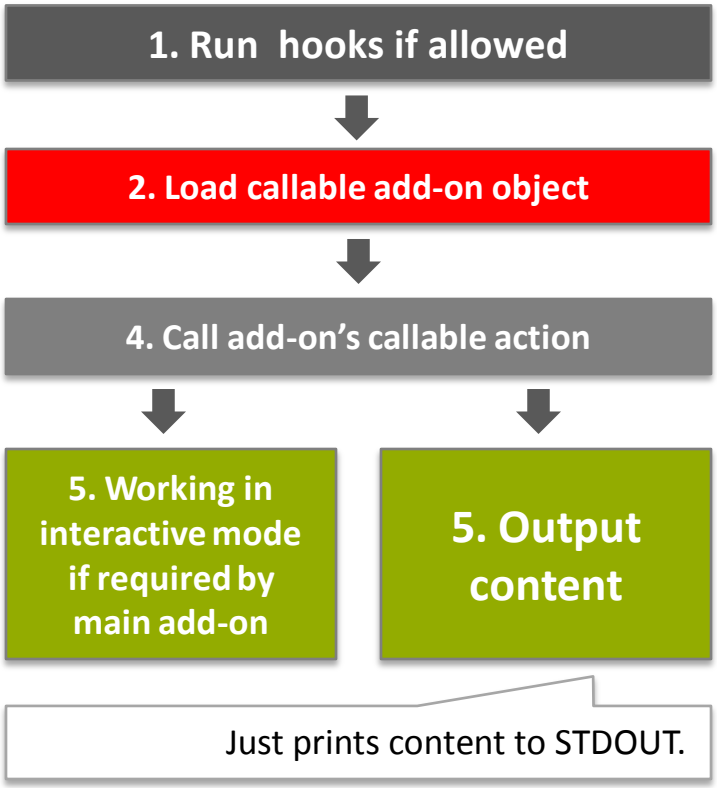
Using toKernel you can develop applications working in command line mode very simply.
For example, if you have the add-on "home" with action "cli_welcome", you can write your own add-on "my_addon" and run it by calling:

/usr/bin/php index.php --addon my_addon --action my_func --param1 param1_value --param2 param2_value

With toKernel, interactive mode applications are quite simple as well, which is quite an advantage.
How does it work? Have a look at the code below, which should be more or less self-explanatory.

```
$this->lib->cli->out('Welcome to toKernel', 'red');
$this->lib->cli->out("\n"); // or TK_NL constant
$this->lib->cli->out('Please enter your name:', 'green');
$name = $this->lib->cli->in();
$this->lib->cli->out('Hello ' . $name, 'yellow');
```

Hooks are the files where you write some actions before application starts to run.
For example: define database connection values.



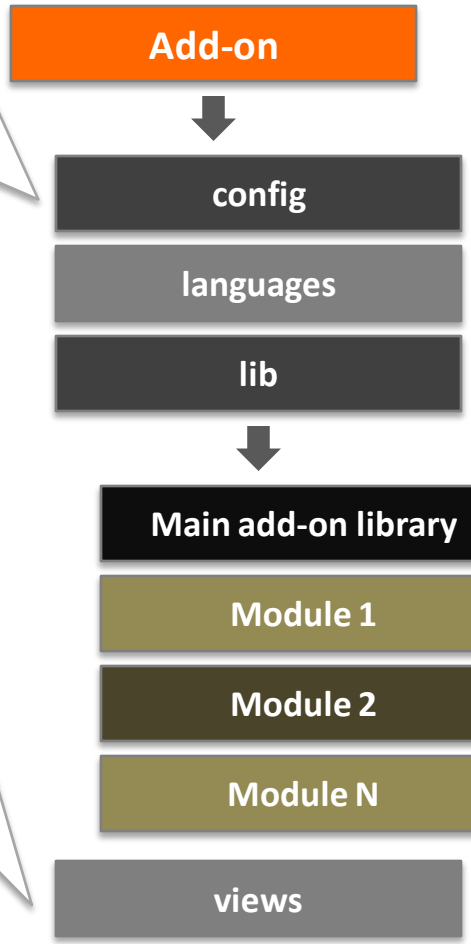
What is an add-on?

An add-on in toKernel is the combination of model, view, controller classes found in other framework but with some additional components. Each add-on has one main controller class and as many modules as required. A module is a just class in the add-on package that can be used only in this add-on.

Each add-on has own configuration files. They have content and extension similar to *.ini files.

Example: config.ini

```
;admin - configuration file
[CORE]
version=1.0.0
allow_cli=0
allow_http=1
```



Each add-on has own language files in “languages” directory. They has an *ini (config) like format for flexible editting.

Example: en.ini

```
;language file for admin add-on
error=Error
some_message=Some message!
```

A view is a standard (X)HTML coded file that may contain scripts. The scripts are used to add dynamic content to view.

Example in library:

```
$view = $this->load_view(‘my_view’);
$view->name = ‘David’;
echo $view->run();
```

Example in view file:

```
<p><?php echo $this->name;  ?></p>
```

Each add-on has one main controller class, named *.addon.php
Each add-on class can have as many modules as required.
A module is a just class in the add-on package that can be used only in this add-on. For example, you can create a module for working with the database, an other module for working with files and yet an other module for working with some functions.

toKernel team members

David Ayvazyan
Patrick Isbendjian
Arshak Ghazaryan
Razmik Davoyan
Karen Grigoryan

http://www.tokernel.com
framework@tokernel.com