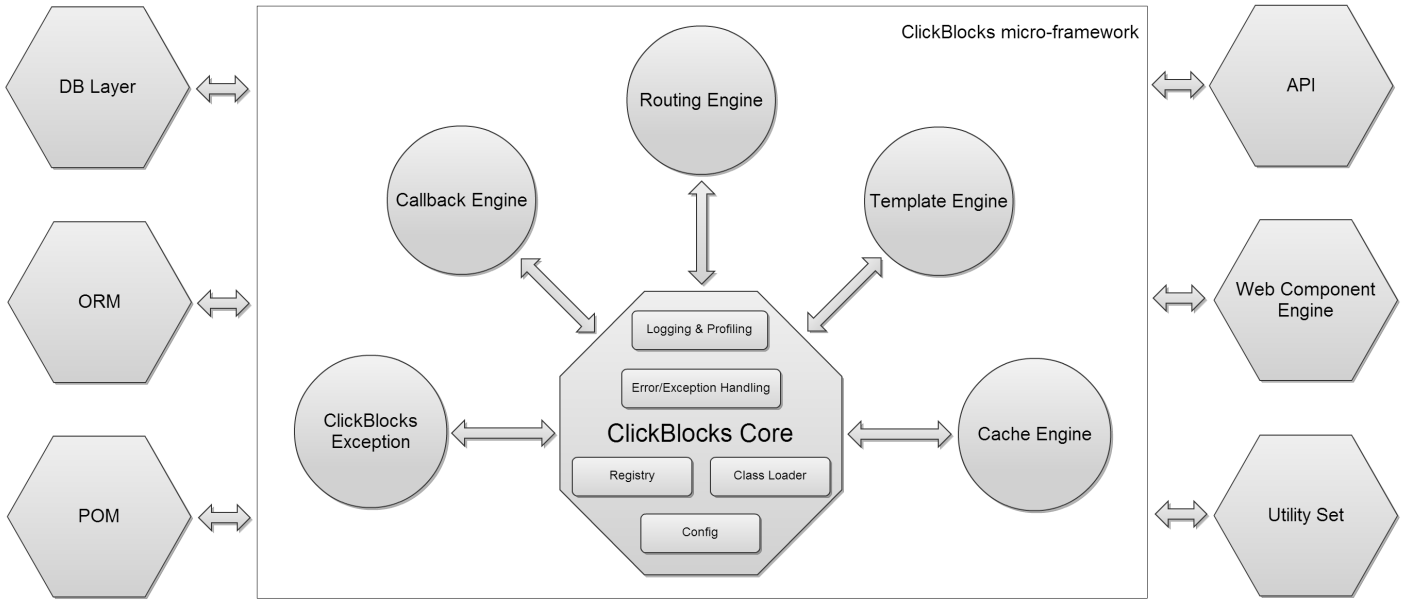
Overview

The picture below shows a functional scheme of the framework:



Framework can be divided into two parts: internal and external. The inside of the framework is a full-featured micro-framework and the foundation for the construction of the outer part implemented as a set of different modules.

Micro-framework consists of the base class **CB**, which is the core of the framework, and additional classes, without which the framework does not work. The core class provides the following functionality:

- Error and exception handling as well as error analysis.
- Logging information about errors and exceptions, as well as a simple means of code profiling.
- Automatic loading of classes, interfaces or traits.
- Configuration tools of the framework and applications built on its basis.
- Centralized storage of global objects (design pattern Registry).

Additional classes of the framework's core can be combined into 5 groups by functional sign.

- **Class of exception handling**. All framework's classes generate exceptions of this type. You can use it to determine the specific type of error and the class in which the exception occurred.
- **Class for work with delegates**. This class allows you to call functions, as well as various methods of classes through the use of a specific format strings, called delegates. These strings represent nothing more than a string record (in the specific format) of calling of class methods or functions.
- **The caching system**. The cache is used by any framework's modules for storing various types of information, such as: page templates, localization data, the structure of the business model of the application (ORM), SQL queries, etc.
- **The template engine**. ClickBlocks has an easy-to-use and effective the template engine using php as the template language. The template engine is part of the XHTML-parser of the POM module.
- **The class provides the routing of HTTP-requests**. With the help of these classes you can implement a full MVC application or API application.

The exterior of the framework can be classified by functional sign into 6 modules (groups of classes):

- **Layer interaction with a relational database**. The classes of this module uses PDO extension for access to the main types of relational databases and provide basic functionality for working with the database (sampling, editing, and deleting data).
- **ORM (Object-relational mapping) layer**. With this module, it is possible to create an intermediate layer between the database itself and the web-based application that meets the required business logic and which is a collection of automatically generated classes of appropriate logical entities in the logic of the application. Using this layer it becomes possible to transparently convert logical (abstract) entity in the real database objects.
- **POM (Page Object Model) module**. Includes templates parser XHTML-pages and control systems which are created by web form interface

and web 2.0 (using the Ajax approach). A key feature of the controls is that each control exists on both the client (in the form of a function) and the server (as an object Meet the requirements of the class), and thus may be transparent manipulation of the interface site is not only the client, but also on the server side.

- **API module**. Consists of classes designed for the construction of various types of RESTful API. In addition to the base class API classes are available for filtering and data validation. This module is linked with the kernel proper framework through the router requests.
- **Module of web components**. Represents compiler web components from their templates and javascript function library for web components on the client. Under the web component means a collection of html, js and css code, bound by a single logic and existing as a whole. Thus, unlike the POM, the web components exist only on the client side and embedded directly in the DOM.
- **Utilities module**. A set of unrelated helper classes to facilitate various tasks: working with date and time, location, non-standard operations on arrays and strings, splitting php scripts into tokens, getting information about the php sites, etc. It is possible that in the future some of these classes become full-featured modules.