

# Architecture

- Overview
  - Symfony Full stack
  - eZ Platform
  - eZ Studio
- eZ Platform Architecture details

This page describes the new generation eZ CMS architecture introduced as of eZ Platform and eZ Studio in 2015. However, this architecture has already been battle-tested by large multinational companies as part of eZ Publish Platform 5.x series since 2012. For further information on the previous generation, see [eZ Publish 5 Architecture - Introduction & Overview](#).

## Overview

TODO: Illustration of layers: Symfony > eZ Platform > eZ Studio / Services

## Symfony Full stack

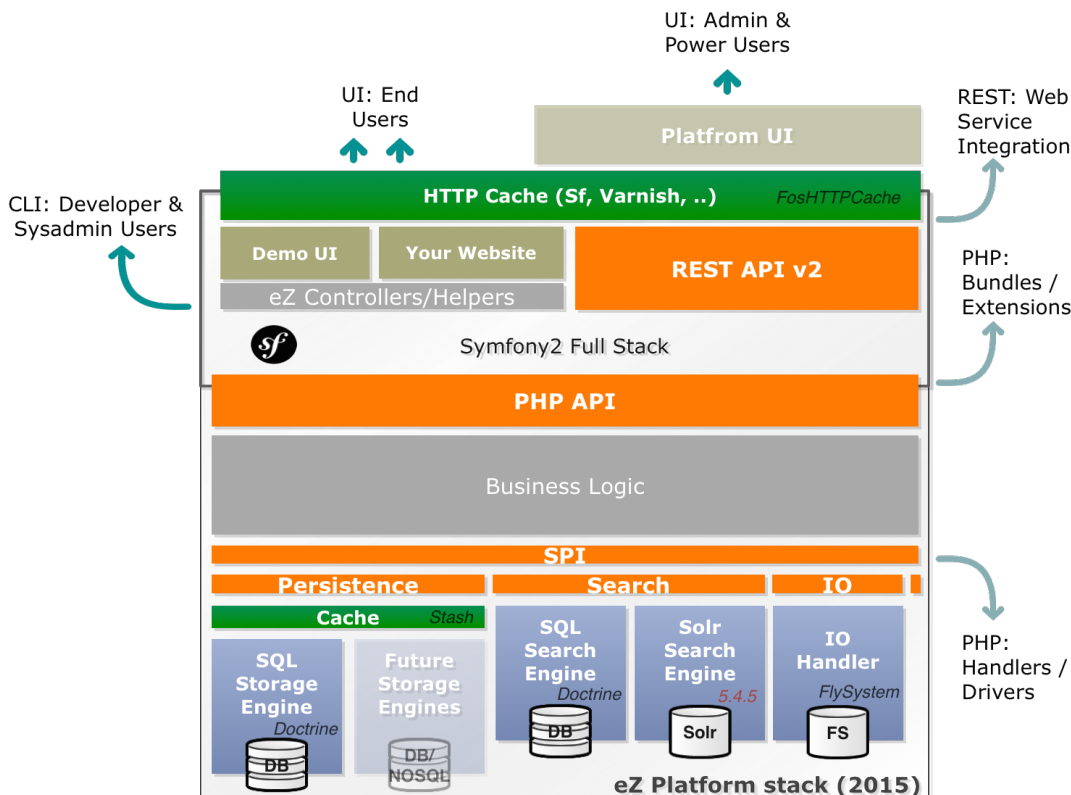
## eZ Platform

## eZ Studio

## eZ Platform Architecture details

The first important thing to understand about the new architecture is to explain it standalone, without considering the old legacy architecture.

The following diagram shows a detailed view of this new architecture.



The new architecture is layered and uses clearly defined APIs between the layers.

- The **business logic** is defined in a new kernel. This business logic is exposed to applications via an API (the [Public API](#)). Developers rely on this to develop websites and web applications using Symfony to organize the way they develop the user interface layer.
- User interfaces are developed using the Twig template engine but directly querying the [Public API](#).
- Integration of eZ Platform in other applications is done using the [Rest API](#), which itself relies also on the [Public API](#).
- Finally, the development of extensions for eZ Platform is done using the Symfony framework when it comes to the structure of the code, and once again relying on the Public API when it comes to accessing content management functions.

At a lower level, the new architecture also totally redefined the way the system stores data. While this is not finalized in version 5.0 (where the new storage system is only shipped with MySQL support), the architecture, when finalized will rely on a storage API that will be used to develop drivers to any kind of storage subsystem.

A motto for this new architecture is to **heavily use APIs** that will be maintained on the long term to **ease upgrades and provide lossless couplings** between all parts of the architecture, at the same time improving the migration capabilities of the system.