

# Web4thejob

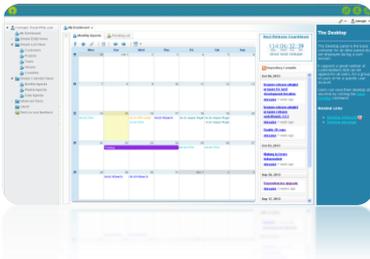
*Web4thejob is a technology company that offers software vendors and developers a reliable and affordable strategy to port their applications to the web using FOSS and Cloud Computing technologies.*

## About Web4thejob

Web4thejob is a technology company that offers software vendors and freelance developers a reliable and affordable strategy to port their applications to the web using FOSS and Cloud Computing technologies.

The company's flagship product is called *web4thejob CE* - a full-stack open source java framework for the rapid development of data-centric web applications. The key differentiation of web4thejob compared to other java web development frameworks lies in its data-centric approach for solving web application development. More precisely, the general idea of web4thejob CE resembles to that of Ruby on Rails. However, instead of Ruby, web4thejob uses Java and instead of Rails it uses a combination of cutting-edge open source java technologies with ZK being one of them.

*"ZK has been the missing piece for developing the web layer of a full-stack web development framework like web4thejob"*



## ZK x Web4thejob

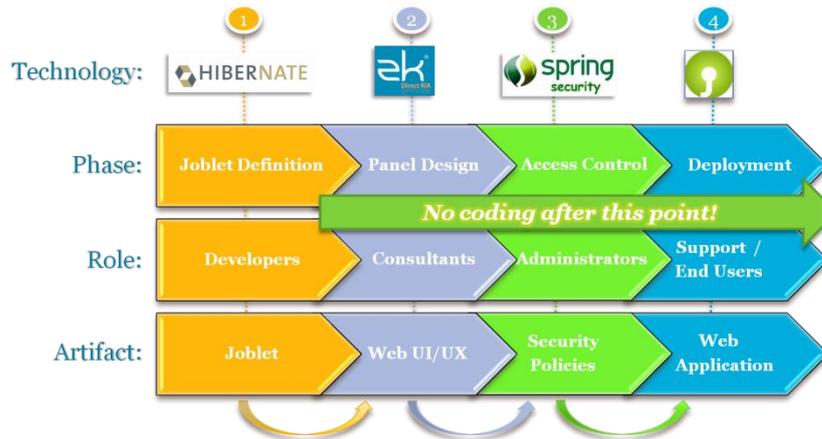
Web4thejob CE is an open source java framework for the rapid development of data-centric web applications. It is ideal for those who seek a component-based event-driven full-stack web development framework, capable of producing elegant desktop-style web applications that can automate any business domain.

The key innovation of web4thejob lies in the concept of "joblets". Joblet is the equivalent of an app in the mobile platform. Similarly to mobile apps that can turn a cellular phone to a social meeting point, a game console, or TV set (i.e. the smartphone), joblets can turn web4thejob into a web application suitable for any business domain. Developers can build and combine web4thejob joblets in a way that promotes the development of high quality Enterprise Software to a higher level.

On top of an innovative technology, web4thejob offers its adopters an Agile Development Lifecycle inspired by the [Manifesto for Agile Software Development](#). Consequently, the Agile Development Lifecycle of web4thejob comprises of 3+1 sequential phases, where each phase requires different IT skills and results to an output that is

used as input for the next phase. These first three phases are dubbed Joblet Definition, Panel Design and Access Control and are carried out by Developers, Consultants and System Administrators respectively. This means that the UI/UX and the security policies of any web application can be delivered code free! The fourth phase is dubbed "Deployment" and it's performed by the support personnel or even an advanced end-user.

## Agile Development Lifecycle



As you can see from the image above, each phase of the development lifecycle is governed by an underlying technology. ZK is heavily engaged in the web layer which produces the user interface of the framework. Thanks to ZK, web<sub>4</sub>thejob provides a WYSIWYG UI/UX design experience with visual inheritance support. The rendering engine is currently based on ZK CE edition but it can be easily extended to include PE and EE editions of ZK, as well as, any other third-party ZK component.

The idea behind web<sub>4</sub>thejob has been in my mind for many years. However, it was only three years ago when I accidentally discovered ZK, that I realized that all the pieces of the technological puzzle were right there in front of me. Since then our work has been to write the sophisticated code that glues all the pieces together.



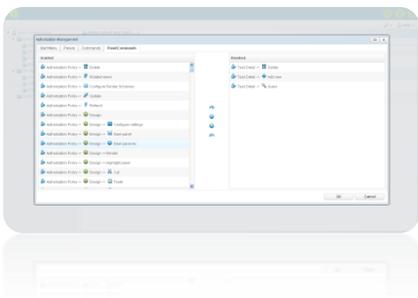
## Architecture

From a physical point of view, the framework consists of four jar files that form a two level dependency hierarchy. At the top of the hierarchy there is the core jar that contains interface declarations, abstract classes and utility functions used throughout the framework. It also holds a reference to the Spring application context which is stored internally by a wrapper class.

The remaining three jars act as implementers of the three framework layers, namely **orm**, **web** and **security**. The layer jars are

aware of the core but are agnostic of one another. Each layer jar has an external primary dependency on another open source java framework specialized on the layer's functional domain. The naming pattern of each jar is based on (a) the layer it implements, and (b) the name and major version of its primary dependency.

Jar Name	Description
org.web4thejob.core	The core jar holding the Spring application context
org.web4thejob.orm.h4	The implementer of <u>orm</u> layer using Hibernate 4
org.web4thejob.web.zk6	The implementer of web layer using ZK 6
org.web4thejob.sec.ss3	The implementer of security layer using Spring Security 3



From a logical point of view the framework consists of ten java packages that span across the four jars, with each package representing a component of the framework. Three of the ten packages are homonymous to the layer names; hence they contain classes relevant to the respective functional domain. The remaining packages define key concepts of the framework's engine, like messages, settings and commands.

Package Name	Description
command	Defines the command execution mechanism
context	Defines wrapper classes for the Spring context
message	Defines the message exchange mechanism
module	Defines the module registration mechanism
<u>orm</u>	Defines the <u>orm</u> layer
print	Defines the printing mechanism
security	Defines the security layer
setting	Defines the setting management mechanism
<u>util</u>	Defines utility classes
web	Defines the web layer

## Challenge

The main goal of web4thejob is to relieve the developer from the burden of coding the ordinary so he/she can focus on the extraordinary. Developers that have coded a data-centric application more than once, most probably have identified similar use cases regardless of the underlying core business. For example, the need for query screens, list view screens, entity screens, master/detail screens, CRUD operations, security policies and others are some very representative use cases common to all data-centric applications. The problem with all these use cases is that while they

constitute irrelevant technicalities for the core business, on an average basis they consume 80% of the development effort.

The main challenge for web<sub>4</sub>thejob has been to combine cutting-edge technologies and agile procedures, so that this 80% ceases to be a developer's task. In other words, web<sub>4</sub>thejob **quadruplicates** the time that developers have to focus on the other 20% of the effort, i.e. the part that is highly related to their core business and makes them stand out as skilled professionals of their business domain.

ZK's contribution towards that end has been fundamental from the right beginning thanks to its clean architecture and predictable behavior. ZK is easy to learn and easy to extend and this is why web<sub>4</sub>thejob has succeeded in providing to developers a flexible web layer that can sustain a great degree of customization with a small amount of effort.

## Why ZK

Our main expectations from a web development framework in order of significance are the following:

1. Openness: it can be used under the terms of an OSI approved license.
2. Simplicity: the amount of work not needed is maximized.
3. Functionality: a rich set of widgets readily available.
4. Flexibility: can be used from a component based (using java) and/or a URL based (using xml) approach for developing a web application.
5. Aesthetic: elegant UI is not a difficult task.
6. Vendor Reliability: reliable support on time.

*“ZK has helped me master web application development much faster than I expected”*

When building a framework like web<sub>4</sub>thejob that aims to deliver a codeless WYSIWYG UI design experience with Visual Inheritance support, you are required to go through a lot of prototyping on different approaches until you are sure that you have identified the one that best serves your needs. Practically speaking, you are building a web rendering engine (web<sub>4</sub>thejob's web layer) on top of another web rendering engine (ZK). Hence, you are stressing the limits of the underlying engine (ZK) both in terms of architectural design and implementation quality.

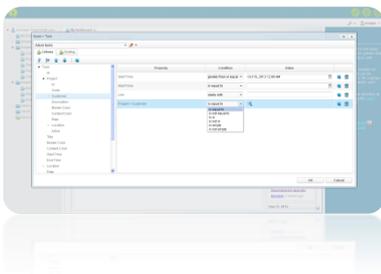
ZK captured my attention from the start thanks to its rich set of widgets and its hybrid nature capable of serving both java and xml definitions. When you are at the beginning and you want to experiment, xml can help you move fast and get familiarized with the specifics of ZK as soon as possible. Then, when you have verified that everything you need is there, you can switch to java and start

building your product straight away, enjoying the full set of features that java development has to offer (type safe compiling, debugging, sophisticated IDEs and more).

## Best of ZK

My personal favorite among the ZK CE components is the Border Layout. To the best of my knowledge no other web framework offers out-of-the-box the ability to split up your screen in multiple splittable and collapsible regions like the nesting of Border Layout instances can do. If you add to this the Tabbox component, you end up having an amazing raw material for designing a highly ergonomic and elegant web UI for your users.

Web4thejob materializes this greatly concept by introducing the concept of [panels](#). Panels are distinguished in two functional categories: The Layout panels (like Border Layout and Tabbox Layout) and Content panels (like List Views, Entity Views, Calendar Views and more). During the second phase of the Agile Development Lifecycle of web4thejob, the Consultant (or even an advanced user) can build the UI/UX of a web application by combining intuitively Layout and Content panels through a WYSIWYG visual editor that is running in the browser. Check out [this video](#) for more details.



## The Result

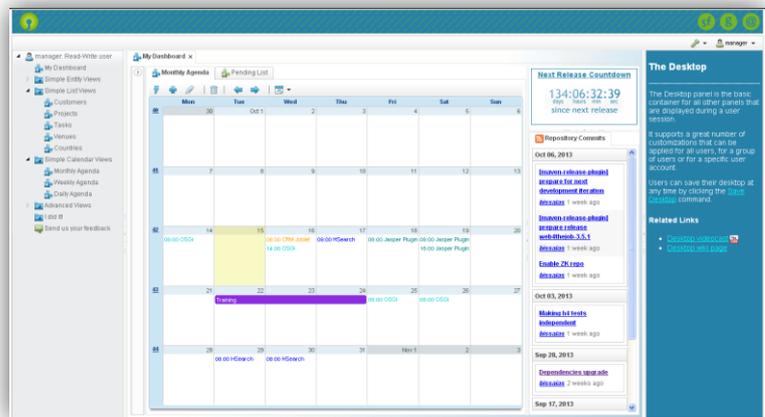
The competitive advantage of web4thejob is that it inspires a new culture for developing data-centric web applications. A culture based on cutting-edge open source technologies on one hand and an agile development methodology on the other, that (a) facilitates the release of working software early in the development cycle, (b) favors collaboration thanks to the joblets and the codeless UI design and (c) is tolerant to changes even late in the development lifecycle.

Going forward, web4thejob should become fully modularized in order for the concept of joblets to work in its full potential.

Currently, there is only one technology that is proven to attach this dynamic nature to a java project; which is the [OSGI Alliance](#). Imagine an environment similar to that of the Eclipse IDE where joblets can be installed and uninstalled on-demand by end-users. Sure it's a challenging task because we are talking about modules of Enterprise Software that span functionality across all layers of the system, with the data layer complicating our goal a bit. However, if this vision sounds fascinating, I am sure that the trip to get there will be even better.

Consequently, the next logical step would be to create the **joblet marketplace**. An online repository where developers publish their joblets under the licensing terms of their preference (free or

proprietary) and end-users are able to try, review and purchase joblets online. Upon acquisition, the end-user could opt either to run the joblet on the **cloud** or on-premises. It's as simple as that, no closed platforms, **no vendor lock-in**.



-Veniamin Isaias, Founder & CEO at [web4thejob](#).

## About ZK

ZK is the leading enterprise Java Web framework with more than 1,500,000 downloads. ZK is deployed by a large number of Fortune Global 500 companies, including Barclays, Allianz, Swiss RE, Roche, Deutsche Bank, Sony, Sun Microsystems, and Toyota, providing them with the ability to rapidly create rich Ajax enterprise level applications.

## Contact us

Potix Corporation

[info@zkoss.org](mailto:info@zkoss.org)  
[www.zkoss.org](http://www.zkoss.org)