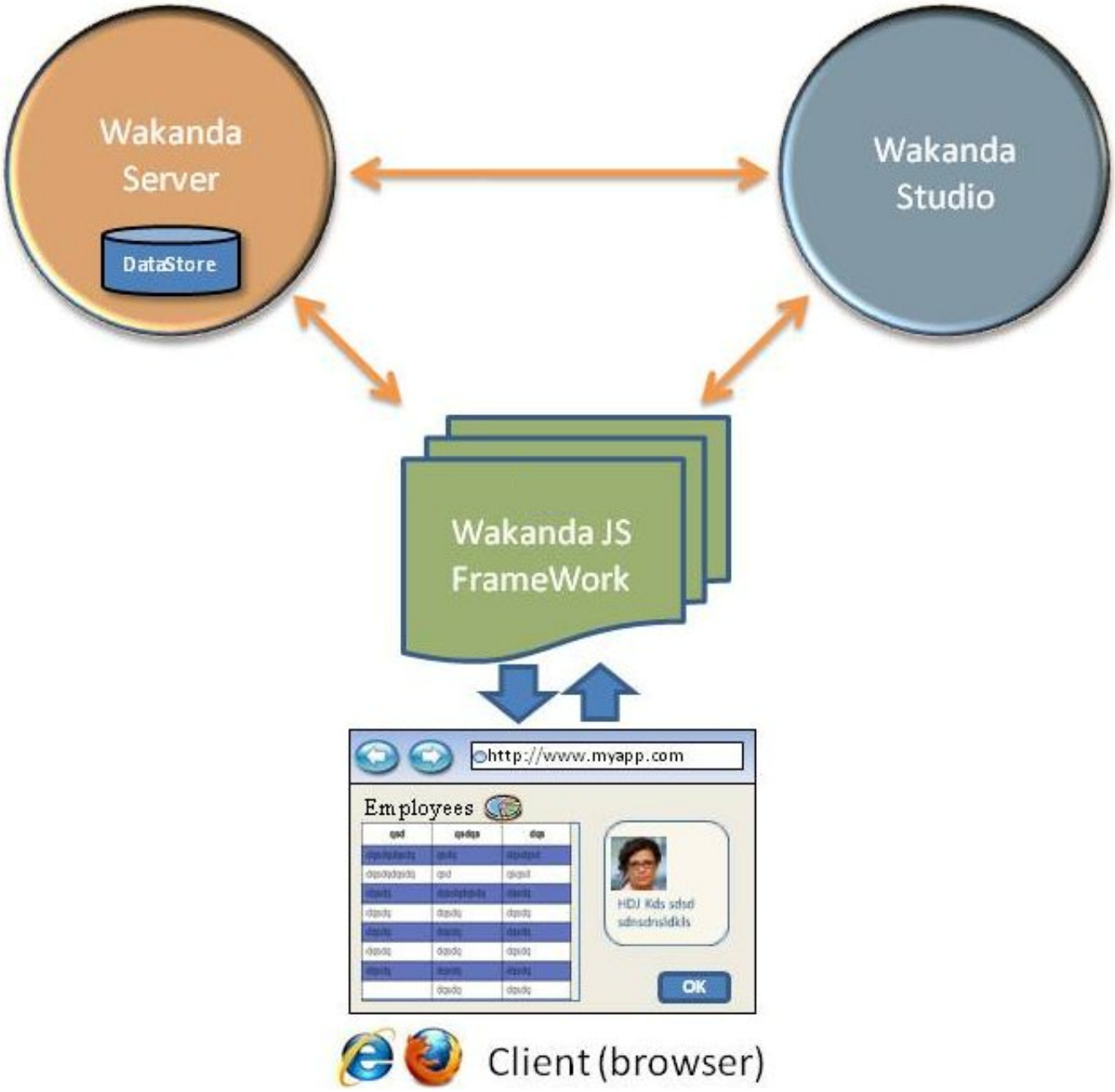
Wakanda Studio Reference Guide

Wakanda Studio Overview

Wakanda is an integrated, complete and open development platform, making it possible to design, build and release business-oriented Web applications or Business Internet Applications (BIAs). Business Internet Applications are desktop-like applications that are hosted on a server and accessible through a web browser, tablet, or smartphone.

The Wakanda platform consists of three main components:

- Wakanda Server, a Web application server tied to a database engine (datastore),
- Wakanda Studio, a desktop application providing an IDE, and
- WAF, a JavaScript framework.



The fully integrated Wakanda platform provides the necessary tools to create and run your Web

applications. While other development environments require various technologies and tools (application server, HTTP server, database server, a framework, etc.) that must work together through multiple layers of code, Wakanda provides an **all-in-one environment**, thus eliminating the intermediary code and hazards of inter-application interfacing. Both Wakanda Server and the Web applications use the same language: JavaScript.

The application's logical structures (entities) are available as directly editable objects through JavaScript code, making the access to data immediate and natural. Building Wakanda applications is done using business logic, found at all stages of the creation process, making the application easy to design as well as easy to maintain.

The Wakanda application's data structure is inspired from the entity-relationship model. The data model as well as its attributes and relations are defined using datastore classes. Datastore classes make it possible to define the database structure in a natural way, befitting the user's needs and letting go of the conceptual constraints of the relational model.

Wakanda Features

Wakanda provides you with an end-to-end JavaScript platform, where you get:

- A zero-configuration server with built-in HTTP services,
- A client proxy that automatically replicates the logic and services from the server,
- Client-side access to all the data objects,
- Multi-layered architecture,
- Datastore classes in the domain layer (with XML model file),
- Robust infrastructure with integrated database,
- Server-side JavaScript,
- REST data services with automatic JSON transport using JSON-RPC,
- Fully evolutionary by inheritance to add behaviors and events, and
- Dynamic behaviors: no static code generation.

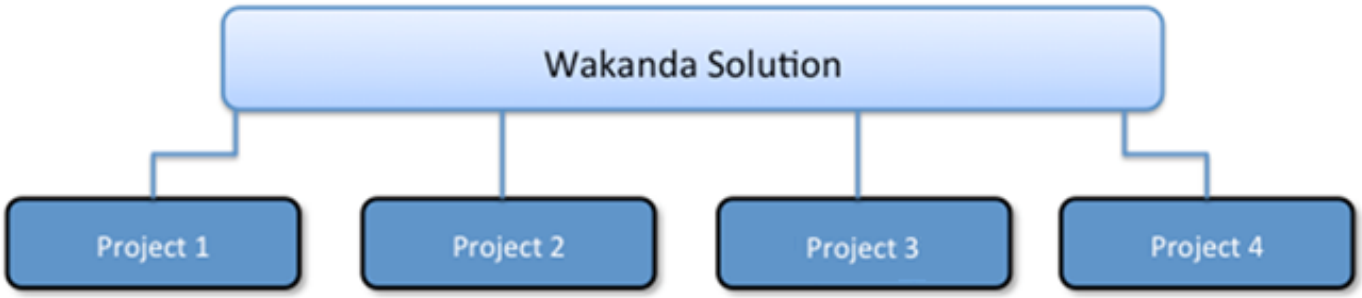
Wakanda Server

Wakanda Server is a faceless application intended to run on Windows, Mac and Linux, comprised of the two principal components:

- **Data server**: Wakanda Server benefits from a complete, robust, and powerful Object Data Storage engine, on top of which is built the concept of entity-relationship model, thus allowing you to formalize the hierarchical data structures in a simple and efficient manner using datastore classes, staying as close as possible to the requirements of your application logic.
- **Web server**: Facilitating and optimizing the interactions between the user interface and data by way of the technologies implemented by the development environment (datastore classes, Server-Side JavaScript, RESTful Data Services, etc.), Wakanda's Web server offers a platform dedicated to deploying powerful and evolutionary Business Internet Applications, using the latest industry standards.

Wakanda Solutions and Projects

In Wakanda, you create a project to group the different files (HTML, JS, CSS, image files, etc.) that make up your Web application. Wakanda can group several projects together to be a part of one solution, managed by the server. The basic architecture can be illustrated like this:



Each project consists of a certain number of files, arranged in folders: text files (HTML, JavaScript, CSS, logs, etc.) and binary files (handled by Wakanda, such as the data file, or handled by you, such as images).

Projects within a solution are completely independent from one other and cannot share JavaScript code. If you want to share resources between different projects, you need to use the Wakanda connectors (refer to **Connector Pro: Wakanda/Wakanda**).

Wakanda Studio

The Wakanda Studio is the place where you can design all the parts of your Web applications. It is made up of the following components:

- **Solution Manager**
- **Datastore Model Designer**
- **GUI Designer**
- **Code Editor**
- **Debugger**

Solution Manager

The Solution Manager is the main area in Wakanda that allows you to manage your solutions, projects, and various files (Pages, HTML, JS, CSS, JSON, XML, Text, images, etc.) in each project.

Datastore Model Designer

The Datastore Model Designer is where you create the model for your Wakanda project. Models are broken down into datastore classes, containing attributes and methods.

GUI Designer

The GUI Designer allows you to design your Pages with powerful widgets to view and edit the data in your Wakanda application. Besides many built-in widgets, you can also create your own widgets that you can include in your projects.

Code Editor

The Code Editor is where you write your JavaScript code for specific events for datastore classes, attributes, Pages, widgets, and much more. It can also handle files written in XML, JSON, and CSS.

Debugger

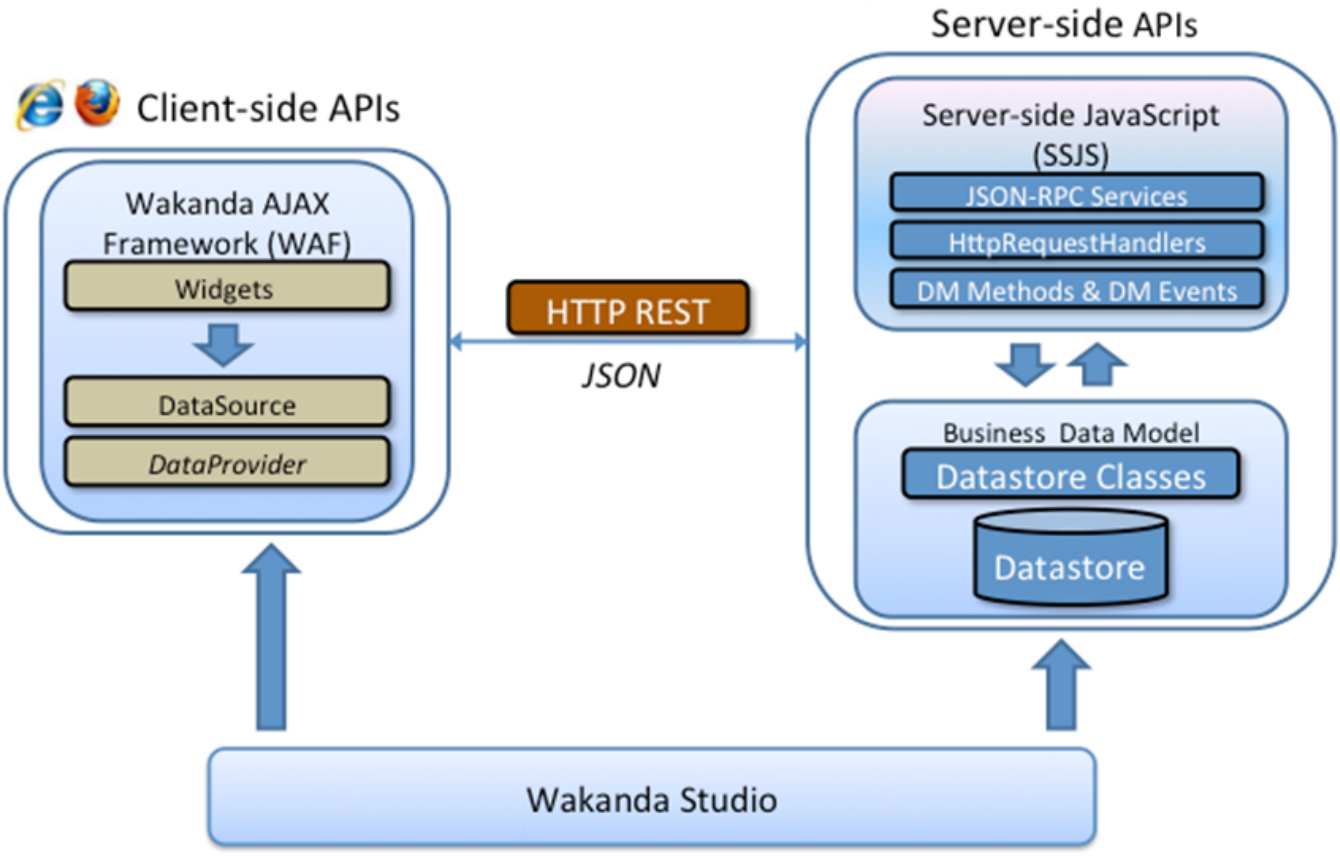
A full-featured JavaScript Debugger that will help you to test your code.

Studio Tools

A few additional tools, like the **Wakanda Server Administration** and **Data Browser**, allow you to administer your web application on the server and to manage the data in your projects. Both are accessible through your default Web browser. The **Add-ons Extension** extension allows you to install themes, widgets, and modules into your project as well as extensions into Wakanda Studio.

Wakanda APIs

A Wakanda solution relies on several JavaScript APIs available on both the server-side and client-side that allow built-in editors or interfaces from any other application to connect and use the business model. The current architecture of Wakanda data access APIs is shown here:



Server-side JavaScript APIs provide you with all the necessary tools to manage files, folders, services, and so on.

For more information about Wakanda JavaScript APIs, please refer to the APIs documentation at Wakanda Doc Center.

System Requirements

The Wakanda platform requires the minimum following operating systems and web browsers:

Wakanda Server

- Windows Vista 32-64 bit, Windows 7, or Windows Server 2008
- Mac OS X 10.6 (Snow Leopard) with a 64-bit processor machine
- 64-bit Linux (Wakanda Server is tested on Ubuntu 2.6)

Wakanda Studio

- Windows Vista 32-64 bit, Windows 7, or Windows Server 2008

- Mac OS X 10.6

Web Browsers

- Google Chrome
- Firefox
- Safari
- Internet Explorer 9

Installation

Install **Wakanda Server** and **Wakanda Studio** onto your hard disk after downloading it from the Downloads section of the Wakanda website.

You can place Wakanda Server and Wakanda Studio in the same folder/directory; however, it is not necessary. If they are side by side, Wakanda Studio will find and launch the Wakanda Server automatically. If Wakanda Studio does not find Wakanda Server, it will display a dialog so that you can select which Wakanda Server to open.

Once **Wakanda Server** and **Wakanda Studio** are installed on your machine, double-click on Wakanda Studio. You don’t have to launch Wakanda Server because Wakanda Studio will automatically launch it when needed or as defined in your solution's settings. You can also start Wakanda Server by clicking on Wakanda Studio’s **Start Solution** toolbar button if it is not started.