Overview

This is a high level design document for the Unibot web platform. The Unibot web platform is a Java web application that supports cloud hosting of bots, live chat, forums, and other content.

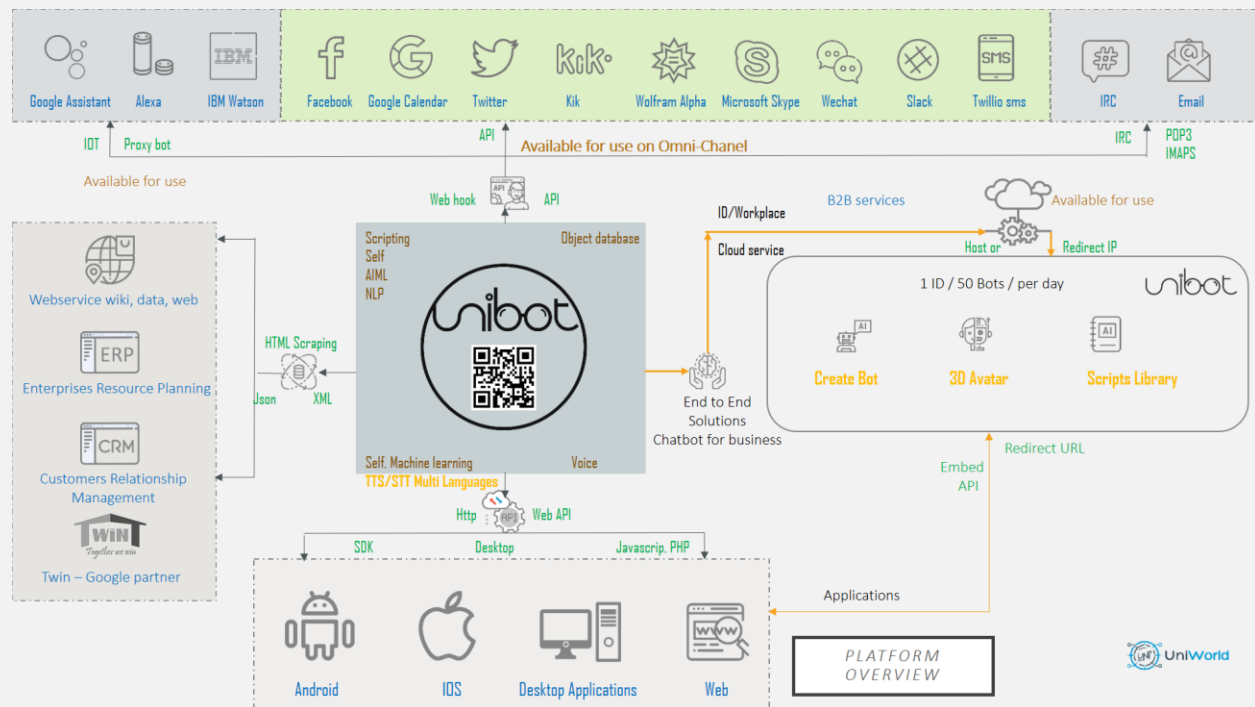The Unibot web platform uses the Tomcat web server, and the PostgreSQL database.

The Unibot web platform provides both a web HTML interface, an XML and JSON web API, and a JavaScript.

The Unibot web platform services:

• *Bot hosting*

• *Bot creation and development*

• *Bot admin and monitoring*

• *Bot embedding*

• *Facebook automation service*

• *Twitter automation service*

• *Telegram automation service*

• *Slack automation service*

• *Email automation service*

• *Timer service*

• *Live chat hosting*

• *Live chat admin and monitoring*

• *Live chat embedding*

• *Forum hosting*

• *Animated avatars*

• *Text to Speech*

• *Graphics library*

• *Script library*

• *User management*

• *User directory*

• *Isolated domain work spaces*

• *Server statistics and console*

Unibot is composed of several component layers. These include:

• *JSP files*

• *Servlets*

• *JAXRS REST service*

• *Bean classes*

• *Live Chat web sockets*

• *Object model*

• *Database service*

• *Social media and timer services Unibot uses the following libraries:*

• *Unibot AI engine – NLP, knowledge, scripting, and social media integration*

• *Mary TTS – open source Java text to speech engine*

• *EclipseLink JPA – open source Java persistence framework*

• *Jersey REST server – open source Java REST service*

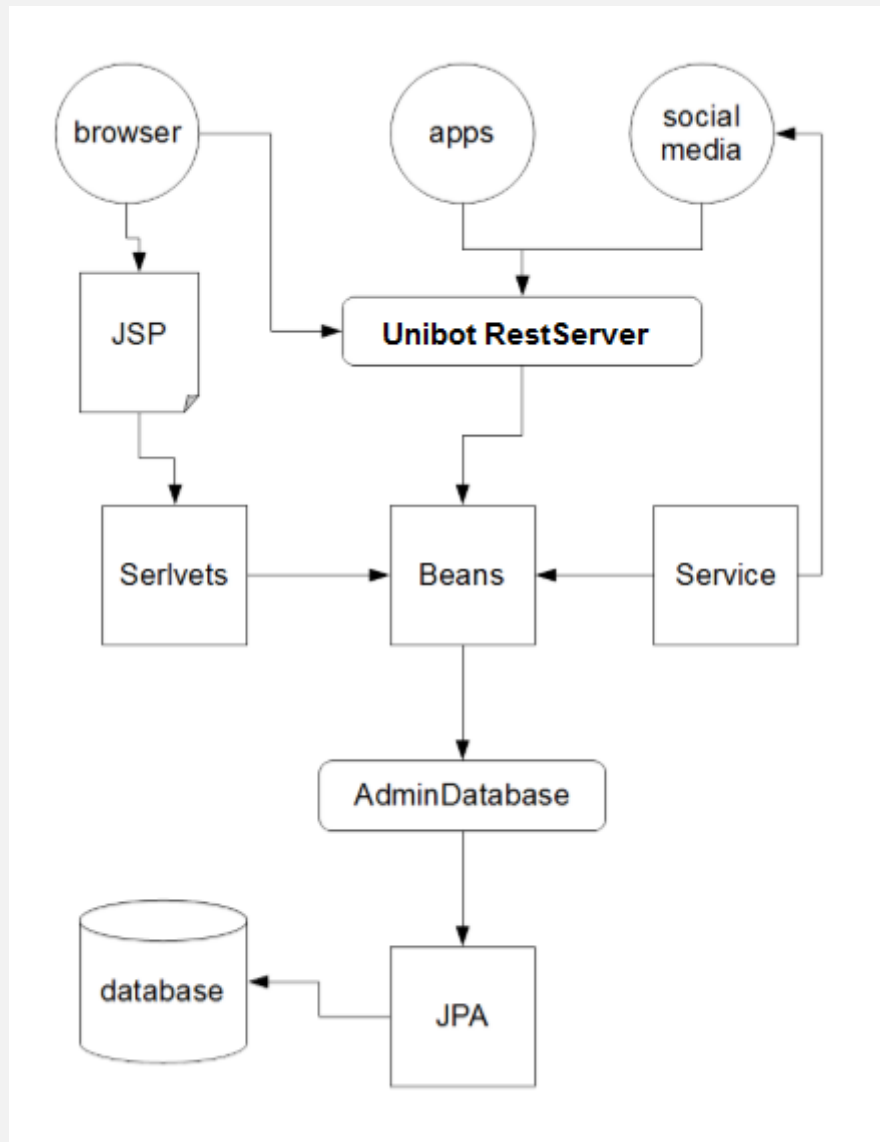• *Oswap security – open source security library for sanitizing HTML*

Process Flow

The process flow starts from a web request, web API, or service poll, and flows through the process classes.

The process classes include:

• JSP – HTML JSP files that define the web interface.

• Servlets – Java servlet classes that respond to JSP POST and GET calls. Each JSP or service defines its own servlet class.

• *UnibotRestService – REST service that defines the web API, the web API is used from JavaScript, apps, and social media web hooks.*

• *Java Beans – The bean classes implement the server functionality for requests. They process the request, access the database, and return data. Each service defines it own bean class.*

• *AdminDatabase – This class defines all database access. JPA used used to access the database.* • *Service – The service classes poll social media, email, and perform timer functions at scheduled intervals.*