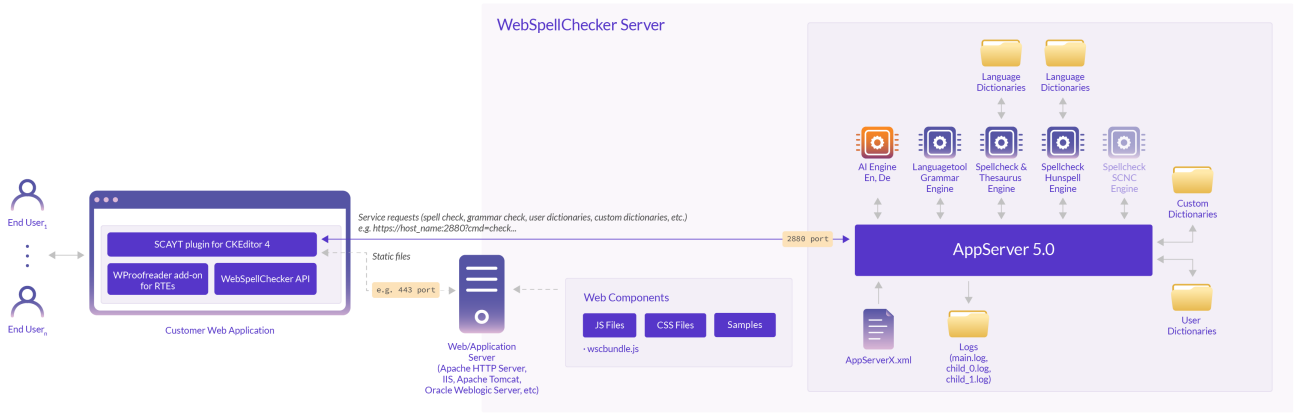
# Architecture diagram

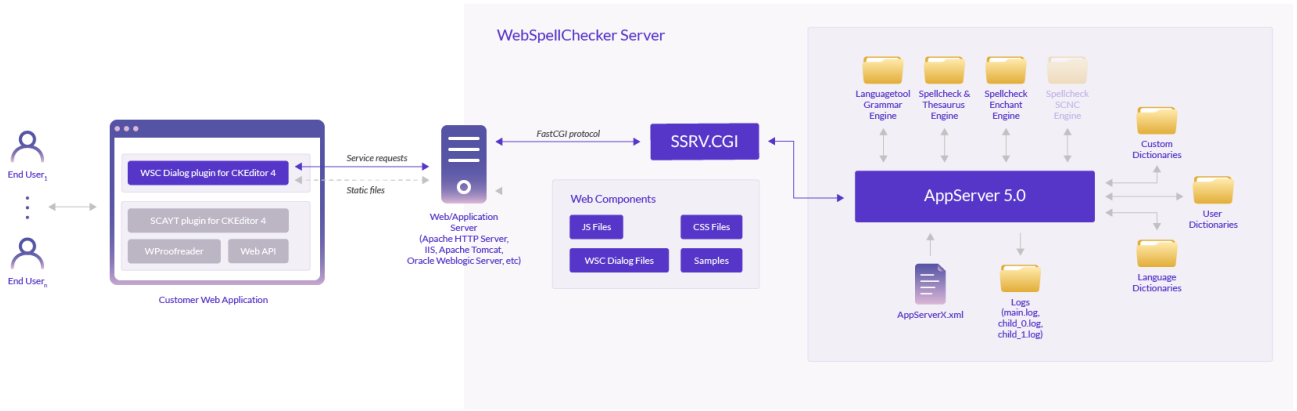
WebSpellChecker Server is a web-based application. Thus, it requires a properly configured web or Java application server for static files processing. Chec k WebSpellChecker System Requirements to find more about the supported web or application server(s).

The diagrams below illustrate the Server version of the WebSpellChecker application that is installed on a separate server (instance, VM, etc.) and communicates with the customer's web app via web or Jira application server and processing service text check requests directly with AppServer (scenario A) or via the FastCGI module (scenario B).

## Scenario A. AppServer processes service requests directly (recommended)




## Scenario B. FastCGI proxies service requests before AppServer




## Legend

**Administrator** is an administrator of both WebSpellChecker application and a customer web application. Administrator can perform or manage the following actions:

- Install and configure application on the server;
- Activate the application license (enter a Ticket ID and request a License Response string);
- Customize and adjust the default settings to meet end users' and business needs;
- Create custom wordlists with specific terms to extend the default dictionaries (company and user-level custom dictionaries);
- Monitor the application performance and troubleshoot issues.

**End User** is a user of a customer web application. End user can perform the following actions (within the scope of the WebSpellChecker application functionality):

- Send input text for spell and grammar checking;
- See and correct defined misspelled words with suggested corrections;
- Ignore words or grammar mistakes;
- Add words to a dictionary called “user dictionary”;
- Choose and change a language for spelling and grammar checking. Supported languages.

**Customer Web Application(s)** where proofreading functionality is required and will be used.

**FastCGI** is a binary protocol for interfacing interactive programs with a web server. It is aimed at reducing the overhead associated with interfacing the web server and CGI programs, allowing a server to handle more web page requests at once.

> ⚠ For proper work of the application components, the FastCGI module must be enabled for a chosen web server.

Web/application server requests JavaScript, SpellChecker Dialog plugin and CSS files, then sends all requests with text for spelling and grammar checking to AppServer via FastCGI.

Words from the input text are sent by web/application Server to AppServer for processing through a FastCGI component and returned with marked misspelling or grammar mistakes and defined corrections.

**WebSpellChecker Server Application**

**Web Components** is a set of various static files that are required for the WebSpellChecker application functioning. All these files are shipped with the standard package and installed automatically.

**AppServer 5.0** is an application server that is responsible for combining all the components of the system together to provide the multi-language proofreading functionality.

**Engines** includes a list of integrated engines that are responsible for spelling and grammar checking, and thesaurus processes. These engines are shipped with the default installation package.

**Dictionaries** combine a list of various types of dictionaries either shipped with the package or created during the application operation and use.

There are three main categories of dictionaries available:

- default language dictionaries,
- company-level custom dictionaries,
- user-level custom dictionaries.

**AppServerX.xml** is a configuration file which contains a list of settings for the WebSpellChecker application configuration. You can find a detailed description of all available

parameters in the Configuring WebSpellChecker Server Parameters guide.

**Logs** is a folder with log files that keeps a registry of events, processes, and messages that occur within the application.