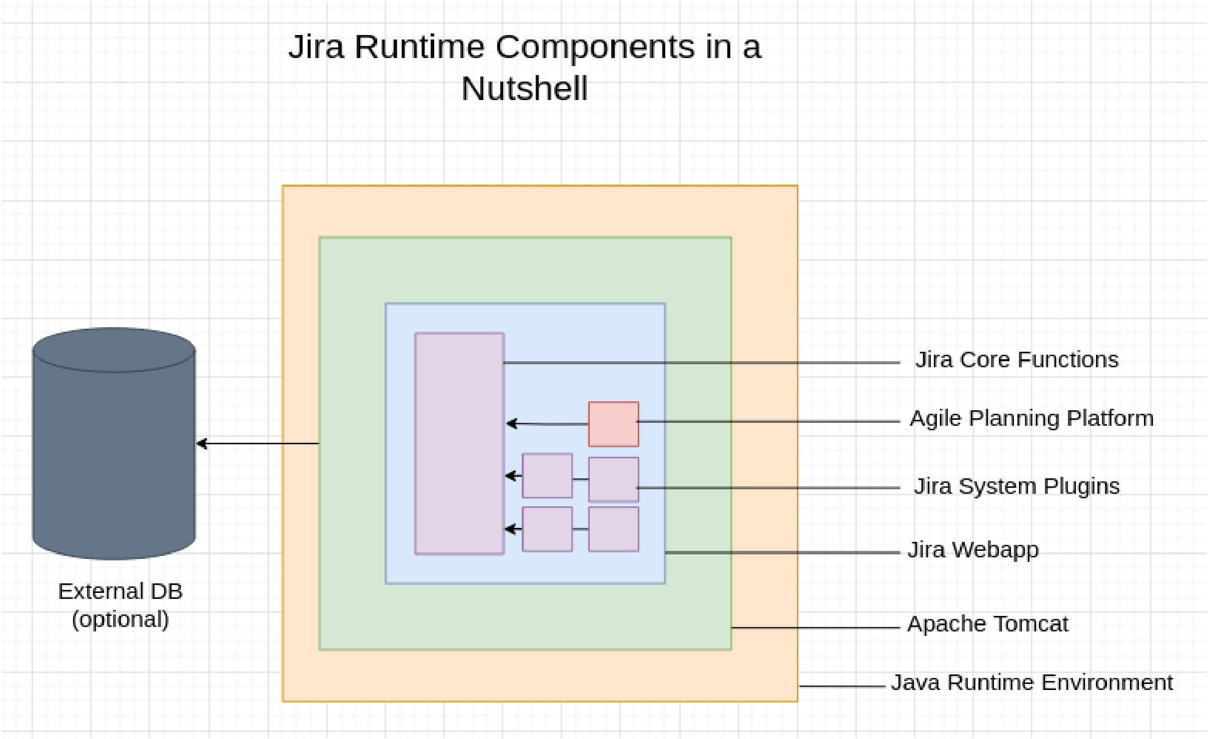
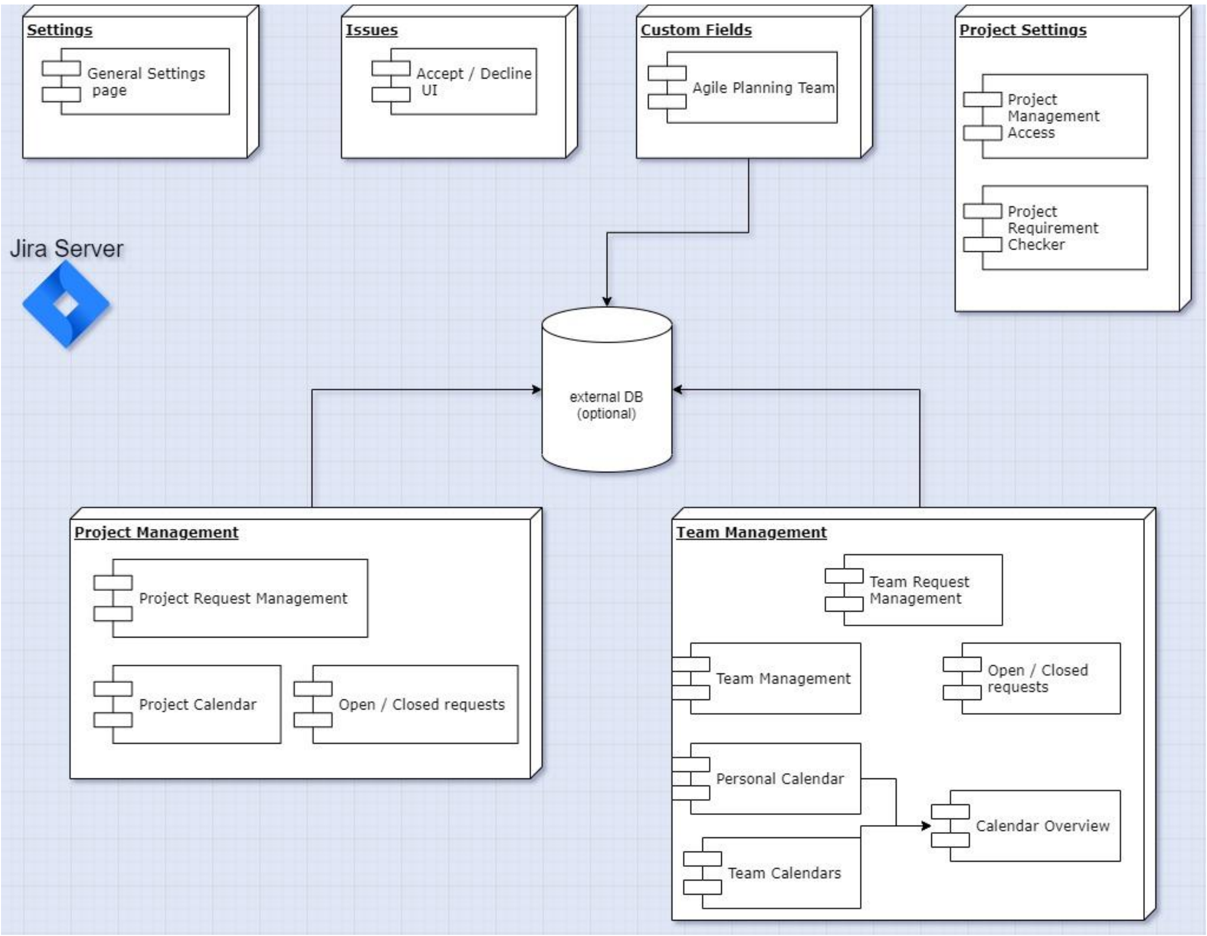
From the early analysis of the customer requirements we decided to
focus on the on premise variant. Jira on-premise is a Java application, which is bundled in a modified Version of Apache Tomcat and can feature an external database, which will be called through the Apache Tomcat. The application itself consists of some Core functions and a Plugin System with some system plugins on top. It is also possible to deploy your own plugin in this environment, so called “user-installed apps”. This is the piece of code this project aims to produce in order to extend Jira with agile planning features not included in the standard version. The Plugin is pure Java code, but can also feature HTML/Javascript/CSS and JSP code on the presentation layer.



Code Components
In our case we are not differentiating between the components within a client and a server as the server is available through Jira. We rather focus on the modules needed to implement the plug in.
As stated by Atlassian, modules are the most important components in a plug-in as they extend the functionality of the Jira platform, e.g user interface but also other parts such as workflow or permissions.
In the scope of the Jirassic park project we would develop following modules:

- Settings, implementing a general setting page
- Issues, covering the accept and decline functionality within a request
- Project management, which considers users with a project manager role and needed functionalities as: request management, project calendar and overview of open and closed requests.
- Team management offers almost the same features as the project management module but from the perspective of a team manager role.
- Furthermore a module project setting will be implemented which control settings of a project and validates requirements
- The create issue dialog should be enhanced by a custom field agile planning team



Technology Summary

We use different technologies to facilitate the “agile transformation”-plugin in Jira. Depending on the deployment (on cloud vs. on premises) we have two different technology stacks to be used in development. The deployment will be decided by Bearing Point, our business partners. It is planned to deploy on premises but a final decision is still pending.

On cloud

Github + Git: We will use Git for software version control in our development. Our repository will be hosted on Github.

Atlassian SDK: The Atlassian SDK provides a set of shell scripts for creating, installing and building plugins for Atlassian products.

Java 8 JDK: The Java 8 JDK is required for using the Atlassian SDK. It includes tools for testing and developing programs written in Java.

JavaScript: JavaScript is designed for creating network-centric applications while being complementary to and integrated with Java and HTML as well as open and cross-platform.

Maven: Maven is a project management tool, which will be used for automating builds, project dependencies and documentation.

jsr311-api: Interfaces and annotations for creating RESTful applications in Java.

JUnit: JUnit is a framework for unit tests in Java.

On premises

Includes the same technology stack as the cloud and the following:

Node.js: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Active Objects: Active Objects is a plugin implemented into Atlassian Applications, which enables scalable data access and storage. Active Objects provides the opportunity to create a plugin data storage component and persist your own data in your plugin, it provides developers with access to a real database(in Jira is H2 Database).