

Popis architektury

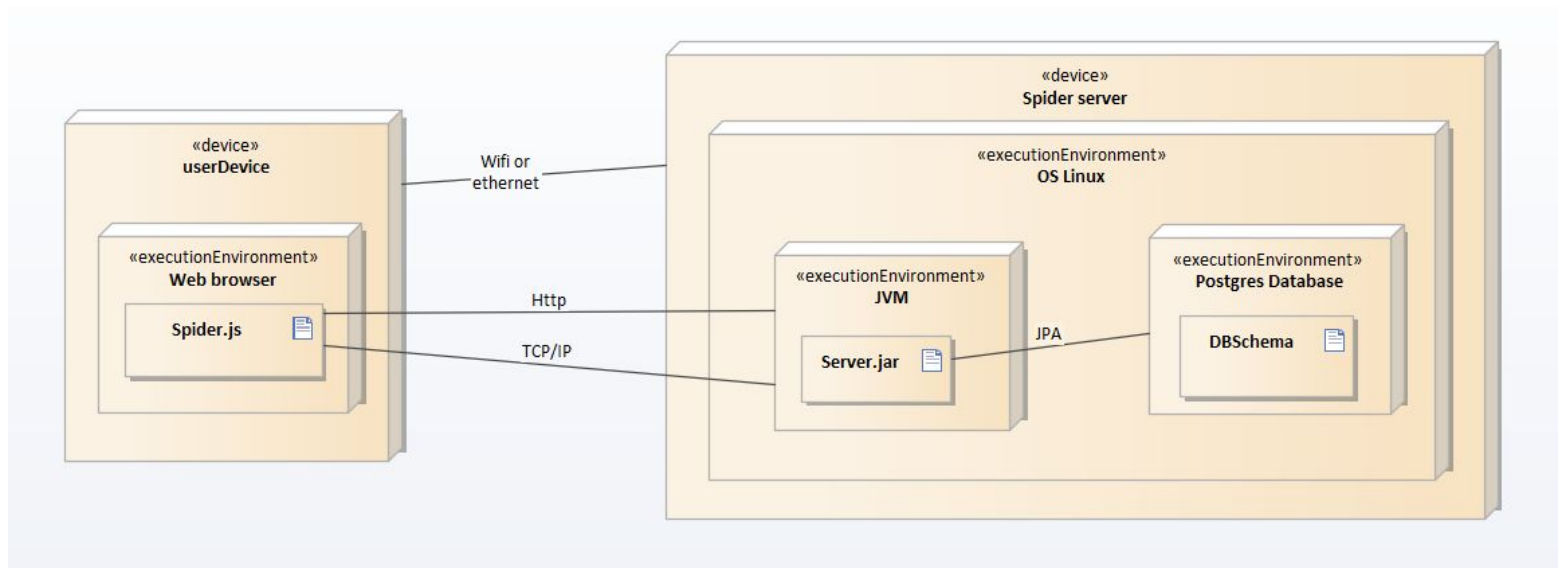
Projekt: Spider

Verze: 1.0 - 14. 5. 2019

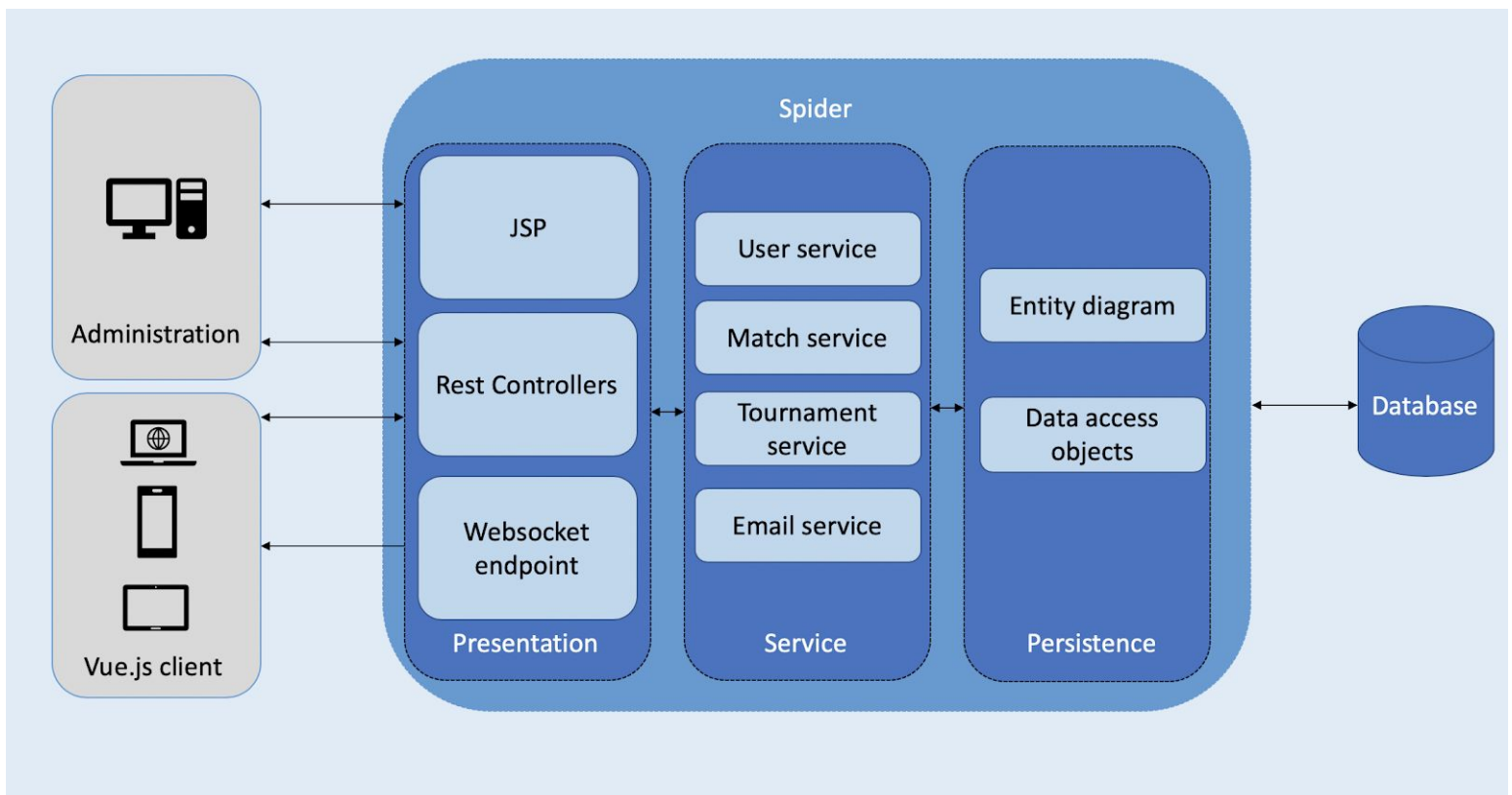
Autor: David Stražovan

Obecně

Diagram nasazení



Architektura a komunikace



Zvolená architektura je klient-server. Komunikace mezi serverem a klientem probíhá dvěma způsoby:

- Request-response
- Posílání zpráv přes websockety

Request-response komunikace je použita pro základní operace ze strany klienta na server.

Operacemi jsou například:

- Přihlášení
- Odhlášení
- Získání údajů o turnaji
- Aktualizace výsledků zápasu

Komunikace přes websockety je použita pro “živé” zobrazení turnaje. Při jakékoliv změně v turnaji, na který se uživatel dívá, server pošle klientovi zprávu obsahující změny a ten na základě nich zobrazená data aktualizuje. Tak odpadá potřeba průběžně kontrolovat, jestli se něco změnilo a lepší se UX.

Databáze

Jako datábaze byla zvolena PostgreSQL.

Aplikace - backend

Pro komunikaci s DB a modelování bylo zvoleno JPA, protože je dostatečné pro náš model a poskytlo nám možnost rychlé implementace.

V systému existují 4 služby (na úrovni kódu) poskytující základní funkcionalitu - Uživatelská, turnajová, zápasová a emailová. Dále existují další podpůrné služby jako např. služba pro vytváření QR kódů.

Velká část aplikační logiky je konfigurovatelná. V konfiguraci je možné přidávat skripty pro vytváření turnajů a zpracování aktualizací výsledků. Tyto se dají po dvojicích seskupovat do tzv. šablon ze kterých může uživatel vybírat při tvorbě turnaje. Toto nám dovoluje doplňovat nové turnajové formáty bez potřeby znovu sestavování aplikace a jejího nasazení. Stejně tak je možné konfigurovat používané emailové účty a šablony posílaných emailů a další atributy systému. Pro skriptování je použit jazyk Groovy, skripty jsou za běhu aplikace kompilovány a vyhodnocovány.

Aplikace - frontend

Administrace

Administrační část je implementovaná pomocí JSP a javascriptu a je součástí backendového balíčku. Jedná se o jednoduchou aplikaci, která se skládá převážně ze stránek s tabulkami a detaily pro jednotlivé záznamy.

Uživatelský klient

Klientská aplikace je psaná v javascriptu. Pro implementaci klientské aplikace byl framework Vue.js. Ten nám dovoluje tvořit znovupoužitelné komponenty, které můžeme využívat na více stránkách.

Aplikaci je možné nasazovat samostatně a nezávisle na backendové části na http server (např. Apache nebo NGINX).