

Jiří Fryč

frycjiri@fel.cvut.cz

České vysoké učení technické v Praze

Fakulta elektrotechnická

Prosecutor

Detekování plagiátů a porovnávání dat

Návrhová dokumentace

# Revize

|  |  |  |  |
| --- | --- | --- | --- |
| Verze | Datum | Změnil | Změny |
| 1.0 | 20.7.2016 | Jiří Fryč | Úvodní verze |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Tabulka - Tabulka revizí

# Obsah

[1. Revize 1](#_Toc456763818)

[2. Obsah 2](#_Toc456763819)

[3. Části systému 0](#_Toc456763820)

[3.1. Prosecutor API 0](#_Toc456763821)

[3.2. Prosecutor Planner 1](#_Toc456763822)

[3.3. Barrister 2](#_Toc456763823)

[3.4. Prosecutor moodle plugin 2](#_Toc456763824)

[4. Komunikace mezi částmi systému 3](#_Toc456763825)

[4.1. Vnější rozhraní 3](#_Toc456763826)

[4.2. Prosecutor API <-> Prosecutor Planner 3](#_Toc456763827)

[4.3. Prosecutor Planner <-> Barrister 3](#_Toc456763828)

[5. Hlavní ideologie 4](#_Toc456763829)

[5.1. Prosecutor Solution (Předmět) 5](#_Toc456763830)

[5.2. Entity (Student v předmětu) 5](#_Toc456763831)

[5.3. Suite (Úkol v předmětu) 5](#_Toc456763832)

[5.4. Group (Semestr/Cvičení v předmětu) 5](#_Toc456763833)

[5.5. Task (Nastavení testu) 5](#_Toc456763834)



# Části systému

## Prosecutor API

Slouží k poskytování RESTful přístupu k testovacím funkcionalitám Prosecutor systému z externích systémů. Taktéž obsahuje základní administraci formou html, pro snadnou tvorbu tokenů pro komunikaci s externími systémy a monitoring.

Technologie:

* Play Framework
* Java 8
* JAXB (XML Binding) – Export, import a validace XML dat oproti XSD
* JPA – Entity Framework

Poskytuje:

* RESTful rozhraní pro přístup externími systémy
* Html rozhraní pro spravování/monitoring správcem systému

Využívá:

* Databázi PostgreSQL
* Lokální úložiště
* Prosecutor Planner

## Prosecutor Planner

Slouží pro plánování testování s ohledem na již naplánované testy, zatížení serveru a přidělené prostředky. Veškeré testování limituje pomocí kernel funkcionalit čímž zamezuje nešetrnému přetěžování, které by jinak vyvolalo pád serveru. Zároveň kontroluje a limituje přístupy probíhajících testů.

Technologie:

* Go
* Go-ps

Požadavky:

* Root/Administrator přístup k serveru
  + Kvůli:
    - Využívání kernel funkcionalit jako cgroup a nice
    - Spouštění a případné násilné ukončování testů pod uživatelem s omezeným přístupem k datům
    - Zjišťování volných prostředků serverů a monitoring celého serveru
    - **Zajištění kontrolované prostředí pro testy**

Poskytuje:

* TCP/IP Socket

Využívá:

* Databázi PostgreSQL (nezávislou na Prosecutor API)
* Lokální úložiště
* Kernel
  + Access control - pro omezování přístupu testovacích programů k pouze pro ně určeným datům.
  + Performance control – pro omezování zátěže tvořené testovacími programy.
  + Monitoring – pro sledování aktuálního zatížení serveru.

## Barrister

Barrister provádí samotné testování zdrojových kódů. Je postaven na základu JPlagu s povznesením na vyšší úroveň. Veškeré vstupy a výstupy jsou validovány oproti XML Schema, podporoju multithreading a díky cachování si poradí i s enormním počtem dat.

Technologie:

* Java 8
* Tiling skripty JPlagu
* JAXB

Poskytuje:

* XML výstup

Využívá:

* Lokální úložiště

## Prosecutor moodle plugin

Je pluginem pro moodle který rozšiřuje funkcionalitu úkolů v předmětech o testování plagiátů.

Technologie:

* Moodle API
* PHP
* Prosecutor API

Požadavky:

* **Moodle s plně funkční officiální Moodle API**
  + Officiální API je jasně definovaná a lze na jejím základě provádět jednoduchý vývoj a testovaní.
  + U ČVUT FEL Moodle se kompletně přešlo z open-source community driven vývoje na close-source vývoj a nelze odhadnout stav, či změny které byly provedeny. Tím se zároveň systém stal zastaralým, jelikož bylo zamezeno automatickým/manuálním mergováním kódu, které by jinak bylo možné provádět pomocí fork systému.
  + Jelikož bude plugin open-source tak se zároveň tímto chrání před přímou spojitostí s FEL Moodle který se nyní nachází z právního hlediska v šedé zóně GPLv3 licence a tudíž případnou DMCA žádostí.

# Komunikace mezi částmi systému

## Vnější rozhraní

Prosecutor API umožňuje výměnu dat pomocí RESTful rozhraní.

Data jsou k dispozici v formátech json a xml

## Prosecutor API <-> Prosecutor Planner

Systémy navzájem komunikují pomocí TCP/IP socketů. Prosecutor Planner poskytuje socket server a Prosecutor API je jejím clientem.

TCP/IP sockety byly zvoleny kvůli universálnosti, snadné implementaci a zároveň mohou v budoucnosti umožnit testovaní na více serverech pokud by výkon jednoho stroje nestačil.

## Prosecutor Planner <-> Barrister

Planner využívá funkcí kernelu k tvorbě, omezení, monitoringu a případnému násilného vypnutí.

Výměna dat probíhá následně:

* Barrister ve vstupních parametrech dostane parametry/pozici souboru konfigurace, nadále do něj žádná další komunikace/výměna dat neprobíhá.
* Planner ve vstupních parametrech pošle místo výstupních souborů, které si po skončení procesu vyzvedne.

# Hlavní ideologie

Vzhledem k rozdílným představám a systémům ve kterých se bude Prosecutor používat je potřeba složitější model na kterém bude celý systém fungovat a zároveň poskytovat i možnost jednoduššího přístupu pro systémy které budou chtít využívat software jen okrajově.



Visio - Rozdílné implementace



Visio - Části Prosecutor Solution

## Prosecutor Solution (Předmět)

Páteř celého systému tvoří Solution/Projekty. Slouží k ohraničení testovacího celku.

## Entity (Student v předmětu)

Entita označuje studenta.

## Suite (Úkol v předmětu)

Suite vybírají data z Entity. Jsou od nich odděleny jelikož mohou být sdíleny mezi více Group, navíc jsou přístupné i z ostatních Solution.

## Group (Semestr/Cvičení v předmětu)

Group spojuje Entity do skupin, to co každá skupina znamená je zcela na uživateli API. V rámci uploadSystem bude znamenat semestr.

## Task (Nastavení testu)

V task se nachází výběr Suites a Groups které budou v testu použity. Dále pak nastavení testu jako citlivost testování, testované programovací jazyky a další.

Prosecutor

Detekování plagiátů a porovnávání dat

Bakalářská práce

Jiří Fryč

frycjiri@fel.cvut.cz

jirifryc.cz@jirifryc.cz