Sem vložte zadání Vaší práce.

České vysoké učení technické v Praze

Fakulta informačních technologií

Department of Katedra teoretické informatiky

Diplomová práce

Analýza bezpečnostních rizik aplikací z logů v reálném čase

Bc. Vojtěch Krákora

Vedoucí práce: Pavel Pivoňka, GWCPM

10. dubna 2017

Poděkování

THANKS (remove entirely in case you do not with to thank anyone)

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Prague dne 10. dubna 2017 .....

Odkaz na tuto práci

Krákora, Vojtěch. *Analýza bezpečnostních rizik aplikací z logů v reálném čase.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

V několika větách shrňte obsah a přínos této práce v českém jazyce.

Klíčová slova Replace with comma-separated list of keywords in Czech.

Abstract

Summarize the contents and contribution of your work in a few sentences in English language.

Keywords Replace with comma-separated list of keywords in English.

Obsah

Seznam obrázků

Introduction

[[Napsat max jednu stranku]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Kapitola 1

Úvod do problematiky

1.1 Zjišťování bezpečnostních rizik

[[Co to jsou bezp rizika]] **[[Jak se zjišťují]]** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

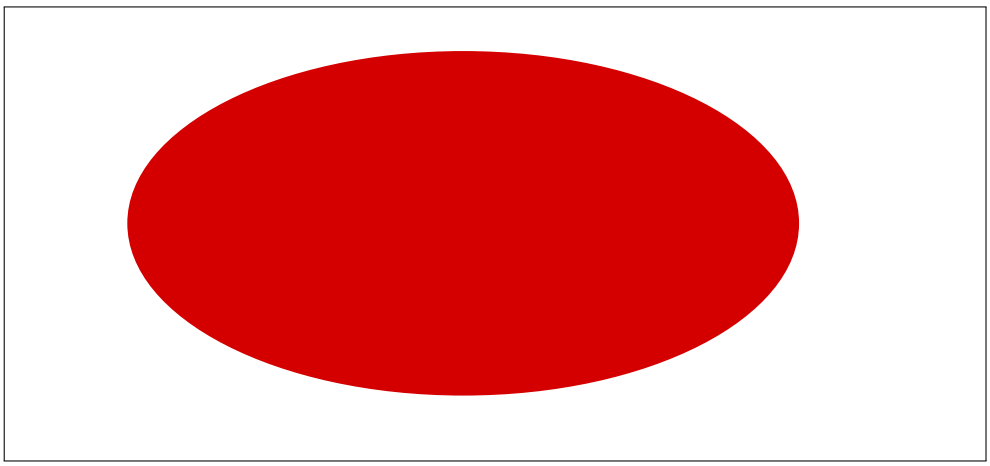
1.2 Platforma Unify

Cílem práce je prověřit jeji funkčnost na platformě Unify [5].

1.2.1 Integrační platforma

Unify je produkt, který slouží jako integrační platforma. Integrace slouží k propojení různorodých systémů a služeb.

Integrace se zpravidla zkládají ze sběrnice [**?**]. Sběrnice, která propojuje konzumující aplikaci a poskytující aplikace se nazývá *ESB* (enterprise service bus). ESB je stavěno na architektuře orientované na služby [**?**].



Obrázek 1.1: Enterprise Service Bus.

Konzumující aplikací rozumíme takovou aplikaci, která posílá dotaz na konkrétní službu. Poskytující aplikace je taková aplikace, která poskytuje rozhraní, jehož výstupem jsou data pro různé konzumující aplikace. V reálné aplikaci je možné poskytovat interface, který například spustí nějaký proces.

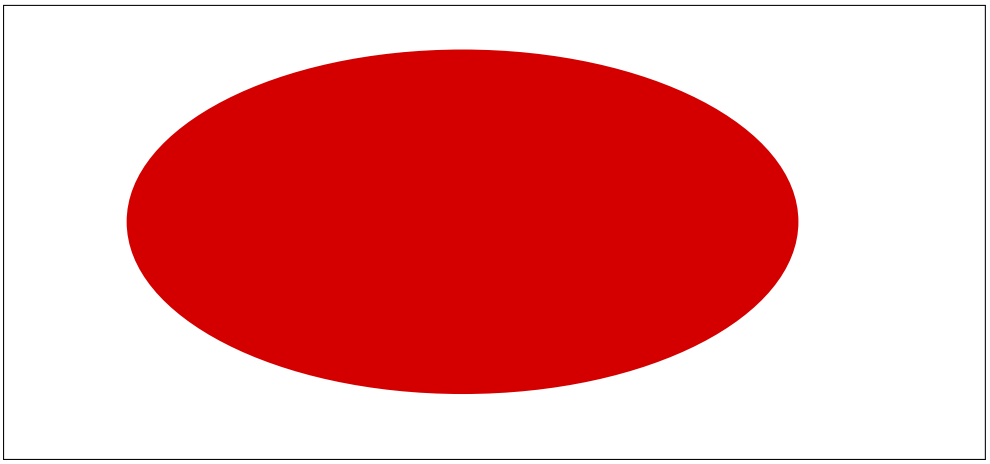
ESB je využíváno interně v rámci jedné firmy nebo logické struktury. Pokud je žádané, aby některé služby byly konzumovány aplikací takzvaně odjinud používá se sběrnice B2B (business to business) [**?**].

B2B poskytuje rozhraní a zpravidla následně samo je konzumentem ESB.

1.2.2 O Unify

Unify je integrační platforma, která je orientovaná na služby [**?**].

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis.



Obrázek 1.2: High Level Desing architektura Unify.

Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.3 Clustering

[[Proč použít clustering]] [[Druhy clusteringu]] [[Výpočet vzdáleností]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan

semper.

1.4 Outliner

[[Proč použít Outliner]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.5 Text mining

[[Proč text mining]] **[[Princip text miningu]]** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultri-

ces augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Kapitola 2

Návrh řešení

[[Nic moc tento odstavec]] V této kapitole se zabývám principy, technologiemi algoritmy, které jsem se rozhodl použít, k tomu abych splnil cíle této práce. Tedy vytvoření aplikace, jenž umožní sledovat bezpečností rizika v reálném čase.

2.1 Architektura aplikace

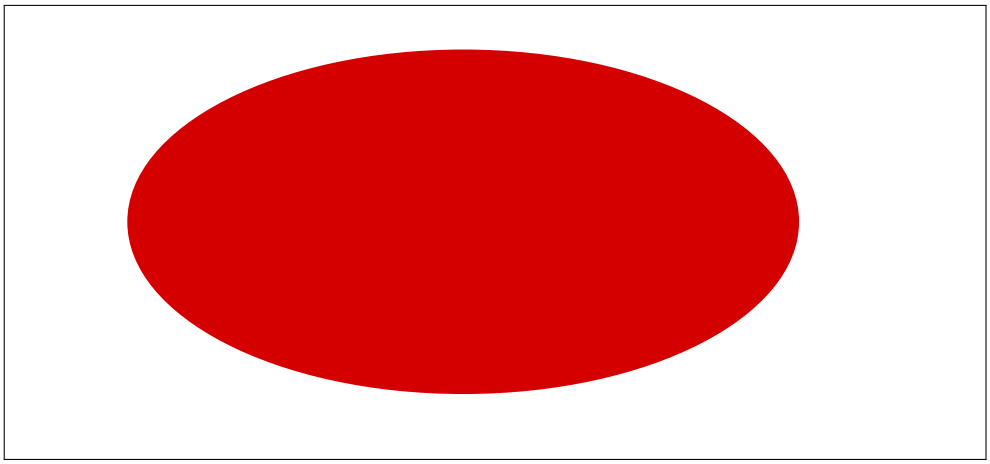
[[Pozor na duplici s odstavcem v úrovni nad]] [[Lze rozdělit na podsekce]] Požadavkem na aplikaci je, aby požadavky zpracovávala a predikovala v reálném časem. Proto je princip položen na čtení požadavků proudících přes platformu. Jejich předzpracování a odeslání do cloudového řešení Microsoft Azure (Více v sekci 2.2). Po přijetí odpovědi je výsledek uložen do databáze. Pro vizualizaci dat slouží REST API [2], které vypisuje předem definované informace ve formátu pro ideální zobrazení v Google Charts [3].

Pro snažší představu o architektuře aplikace poslouží obrázek 3.5, na kterém je vidět High Level Desing [4].

Základem celé aplikace je neohrozit stávající integrační platformu. Na základě toho jsem se rozhodl, že informace o proběhlé komunikaci získám pomocí čtení logovacích souborů. Pomocí čtení přírůstků k jednotlivým auditovým logům získám jednotlivé požadavky a pro Unify to nepředstavuje žádnou zátěž.

Dalším stavebním kamenem je použitý aplikační server Jboss (více v kapitole 2.3). Na serveru je celá aplikace. Dochází zde k předzpracování zpráv, jejich odslání do Microsoft Azure (2.2) a také k ukládání do DB.

Jak jsem již zmiňoval provoz z platformy je po zpracování odesílán do MS Azure, zde jsou definovány jednotlivé algoritmy, jejichž výsledky jsou vracený zpět do aplikačního serveru. Azure jsem se rozhodl používát, protože umožňuje rozložení výkonu na server Microsoftu a protože využití služeb v cloudu se stává stále oblíbenějším. Díky spolupráci s Microsoftem je možné i získat, popřípadě zakoupit, instanci Azure do vlastní sítě.



Obrázek 2.1: High Level Desing architektura aplikace.

Získané výsledky jsou zpracovány a uloženy do NoSQL databáze MongoDB (více v kapitole 2.5).

Aby výsledky nebyly jen hodnoty uložené v databází, je použité REST API, přes které lze výsledky sdílet. API je navržené tak, aby v případě použití Google Charts nevznikly žádné potíže. Google Charts se u cílového zákazníka pužívají již nyní například na zobrazení stavu objednávek. Proto jejich se jejich použití jeví jako další logický krok. Nicméně, není problém stejné api použít pro svoji libovolnou aplikaci, která hodnoty použije buď pro zobrazování přehledů, nebo jako jeden z dalších vstupů například do různých systémů SIEM.

2.2 Microsoft Azure

Na integrační platformě Unify [5] je předpokládaný provoz 20 požadavků za vteřinu. Vzhledem k takto silnému provozu bude potřeba i přiměřeně velký výpočetní výkon.

Spolupráce se společností Miscrosof [6] mi umožnola jako řešení vyzkoušet její cloudové služby Microsfot Azure [7].

Microsoft Azure je sada integrovaných cloudových služeb. Azure nabízí cloudová řešení pro mnoho činností. Motivací k použití této služby k detekci bezpečnostích rizik je nástroj Microsoft Azure Machine Learning Studio [8].

Microsoft Azure Machine Learning Studio je plně cloudová služba, která umožňuje vytváření prediktivních modelů pro strojové učení [8]. Výhodou studia je to, že veškerý výkon je rozprostřen vevnitř cloudu. Díky grafickému rozhraní je snadné vytvořit učící model, který je následně převeden do modelu prediktivního.

Aby měl prediktivní model smysl, je třeba mu poskytovat nějaká data, u kterých je predikce využita. K tomu se využívají webové služby. Prediktivní model se vystaví na specifické URL adrese. Zde je pak očekáván na vstupu konkrétní formát JSONu a služba vrací předem definovanou odpoveď se správnými parametry.

Výhodou využití cloudu je přenesení výpočetní zatěže mimo společnost. Naopak rizikem je problém s konektivitou, který může vytvořit výpadek služby a nebude tedy možné po tuto dobu predikovat rizika. Jednou z možností, jak řešit takové riziko je nechat přenést instanci MS Azure do své sítě.

[[Více rozepsat rozdíl mezi učícím a prediktivním modelem]]

2.3 JBoss

Platforma Unify je postavená na aplikačním server JBoss AS 7 [5]. Z toho důvodu je třeba aby aplikace byla zcela kompatibilní.

[[Nají nějaké zdroje kde je popsáno o co vlastně jde]] Jboss AS je aplikační server pro Javu EE[9].

[[Popsat proč jboss]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2.4 Logování Unify

Platforma Unify loguje veškerý průběh do souborových audit logů. Protože celá integrace je založena na Javě, je využit logovací framework Log4j [10]. Knihovna Log4j umožňuje nastavit si pattern logování [11]. Na Unify je použit pattern **[[Přidat pattern Log4j]]**.

Každý požadavek, který na platformu příde je zalogován do audit.logu. Zpráva je vždy na jedné řádce a zároveň na jedné řádce je povolena pouze jedna.

[[Ukázka logu]]

[[Popsat ukázku logu]]

Vzhledem k tomuto principu jsem se rozhodl přistoupit k tomu, že se vybrané logovací soubory budou kontinuálně číst, kde se bude ke každému

řádku přistupovat jako k samostatné zprávě, která bude následně zpracována dál.

Díky této volbě nebude nutné nijak zasahovat do integrační platformy Unify a minimalizuje se tím riziko, jakéhokoliv nebezpečí ze strany mojí aplikace.

Unify využívá pro některé služby logování do Oracle databáze. Ale vzhledem k tomu, že jde pouze o několik málo určených služeb, připojení na databázy by tak kromě případných komplikací ani nepřineslo žádaný účinek.

2.5 Ukládání dat

[[Porovnání SQL/NoSQL]] Rozhodl jsem se pro ukládání dat využít NoSQL databázi. Protože aplikace Unify momentálně ukládá veškerý svůj provoz do souborů (vyjímečně jsou některé konkrétní služby journalovány do DB), bude databáze využita i pro ukládání veškeré komunikace. To zpřístupní do budoucna snažší operování s jednotlivými zprávami. Popřípadě snažší zpětnou analýzu. **[[Schéma DB]]**

Protože se bude ukládat veškerá komunikace, rozhodl jsem se v databázi mít uvedené následující informace:

[[Byt je to na pohled jasne, tak popsat co ktery param. je]]

- ObjectId - automaticky generováno z MongoDB
- timestamp- časové razítko uložení záznamu do DB
- original-message - nezměněná zpráva
- normalized-message - zpráva po normalizaci
- platformId - jedinečný identifikátor platformy
- assignment - skupina, kterou azure vyhodnotil pro zprávu jako správnou

2.6 Předzpracování dat

[[Dopsat čištění dat]]

Pro snažší prácí s informacemi z logů jsem se rozhodl pro normalizaci jednotlivých požadavků. Normalizace je jeden z požadavků při zpracovávání textu [12].

V kapitole 2.4 jsou vidět informace, které se kromě samotné zprávy logují. V každé zprávě se objevuje takzvaná integrační hlavička. V té jsou základní údaje, jako čas odeslání, jednoznačné identifikátory, zdrojové a cilové systémy. Položka jako je timestamp bude zpravidla pro každý požadavek jiná, stejně na tom budou jednoznačné identifikátory. Z tohoto důvodu jsem se rozhodl zvolit jejich nahrazení.

Při normalizaci dat jsem se podobně jako ve zdroji [?] rozhodl použít následovně:

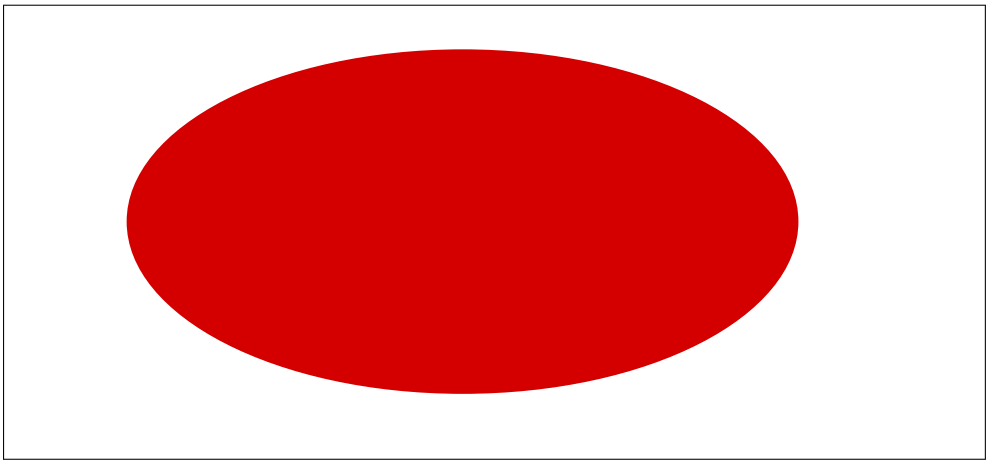
- Nahrazení všech čísel - Pomocí speciálního symbolu nahradím všechny výskyty čísel.
- Velikost písmem - Všechna písmena jsou z velkých znaků převedena na znaky malé.
- Odstranění speciálních znaků - Veškeré znaky jako jsou čárka, tečka ...jsou odstraněny
- Odstranění xml tagů - Rozhodl jsem pro zpracovávat jen obsah zpráv bez xml tagů.

Nahrazení čísel se mi jeví jako logický krok. V jednotlivejch požadavcích jsou čísla například výsledky různých měření na sítí nebo právě čas v časovém razítku. Pro další využití považuji za podstatné vědět, že se v daném místě vystkytovalo číslo, než že to bylo nějaké konkrétní číslo.

Převod písmen na malá zajistí, aby slova, lišící se právě jen ve velikosti nějakých písmen byla vyhodnocena jako stejná.

Všechna komunikace na platformě je převedena do xml (není-li již od počátku vedena v xml). Protože téměř u všech zpráv stejného druhu se používají ty samé xml tagy, nebudou pro další zpracování podstatné a budou zcela odstraněny. Algoritmus bude dále pracovat jen s reálným obsahem zprávy.

Na obrázku 2.2 je ukázka originálu zprávy a její normalizované alternativy.



Obrázek 2.2: Original and normalized message

2.7 Vytvoření vektoru

V okamžik, kdy máme předzpracovaná, znormalizovaná textová data je nutné najít vhodný způsob pro jejich převod do numerické podoby. To umožní snažší zpracování jak v případě clusteringu, tak i v případě detekce outlinerů.

Cílem tedy je vytvořit vekotr, který bude dostatečně jednotlivé zprávy reprezentovat.

2.7.1 TF-IDF algoritmus

Při clusteringu dokumentů lze využívat algoritmus TF-IDF (term frequency - inverse document frequency) [**?**].

[[Trošku lépe pořešit]]

2.7.1.1 Frekvence slova

Ten funguje na principu, že se spočíta frekvence daného slova w v dokumentu d, označujeme $TF(w, d)$. Vypočteme se tak, že se spočítá suma výskytů slova w v dokumentu d. Výšší číslo znamená častější výskyt a tedy o to více w charakterizuje d.

2.7.1.2 Frekvence dokumentu

Frekvence dokumentu pro slovo w DF(w) je počet dokumentů, ve kterých se slovo w nachází.

2.7.1.3 Inverzní frekvence dokumentu

IDF neboli inverzní frekvence dokumentu je daná následující formulí [**?**]:

$$IDF(w) = \log \frac{|D|}{DF(w)}$$

Kde $|D|$ je počet souborů.

2.7.1.4 TF-IDF

Samotný vzorec na výpočet TF-IDF je [**?**]:

$$TFIDF(w, d) = TF(w, d) * IDF(w)$$

2.7.1.5 Použití

Pro své účeli budu předpokládat, že jednotlivé zprávy jsou soubory a slova budou mezerou oddělený obsah zprávy.

Protože slov může být velké množství, rozhodl jsem se najít nějakou hranici, například takovou, že do výsledného vektoru zanesu TF-IDF pro taková slova, která se vyskytují nejvíce v 95% zpráv, ale minimálně v 10%.

2.7.2 Forma vektoru

V sekci 2.7.1 jsem navrhl, jak textová data převést do vektoru. Tím je zaručené, že bodou-li se data přenášet přes internet do Microsoft Azure, budou anonymizována. Z vektoru nedokážeme zpětně zprávu vyčíst.

I když z Azure dostáváme synchronně odpověď zpět, a je tedy jasné, ke které zpráve dostávám výsledek, rozhodl jsem se odesílat i jednoznačný identifikátor platformy. To vede k tomu, že pro znalého člověka lze jednotlivé požadavky sledovat i uvnitř MS Azure. Identifikátor sám o sobě vypovídající hodnotu žádnou nemá, ale máme-li k dispozici původní zprávu, jsem ji schopni dohledat.

[[Ukázka vektoru]]

2.8 Konstrukce clusteringu

Microsoft Azure nabízí k přípravě experimentů svoje studio dostupné na adrese `https://studio.azureml.net`.

Ve studiu Azureml je možné vytvářet své projekty, do projektů umístit své experimenty a ty následně vystavit jako webovou službu.

Základem úspěšného experimentu je vytvořit učící model. To je takový model, pro který máme zvolený cílový algoritmus a na předpřipravených datech ho naučíme aby dokázal v našem případě co nejlépe rozdělovat zprávy do clusterů.

2.8.0.1 Předzpracování

Veškeré předzpracování a čištění dat probíhá v mojí aplikaci i přesto jsem základní předzpracování zvolil i do experimentu samotného.

Po načtení vstupních dat dochází odstranění duplicitních řádků. Jako další metoda je využití modulu, který smaže řádky, jimž chybí nějaká data.

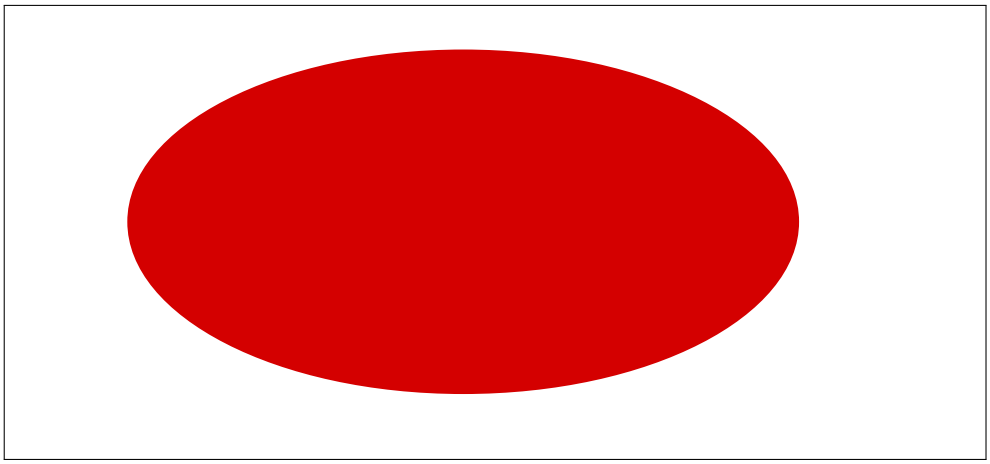
[[Mohl bych použít zároven s klasifikacnim modelem, ale nevim.]]

2.8.0.2 Zpracování

Pro zpracování jsem zvolil K-Means modul, který je připojený na modul pro trénování clusterovacích modulů.

Po natrénování přiřadíme zbytku testovacích dat clustery a může zhlédnout výsledek.

Na obrázku 2.3 je vidět celý vytvořený trénovací experiment.



Obrázek 2.3: Clustering k-means v prostředí MS AZURE ML Studio.

[[Doplnit sem ukázku přiřazení dat a grafy z azure]] [[jak jsem zjistil nejlepší vhodné nastavení]]

2.9 Konstrukce detekce anomálie

Druhou možností, kterou bych rád vyzkoušel je detekce anomálie. To, že chybné požadavky nebo bezpečnostní požadavky se budou výrazněji lišit od běžných zpráv se dá předpokládat.

Princip předzpracování dat v Azureml studiu je stejný jako v při konstrukci modelu pro clustering. Řádky s chybějícími hodnotami a duplikované pro trénování nebudeme používat.

Kromě výše uvedené předzpracující části i zde je část učící a část vyhodnovací.

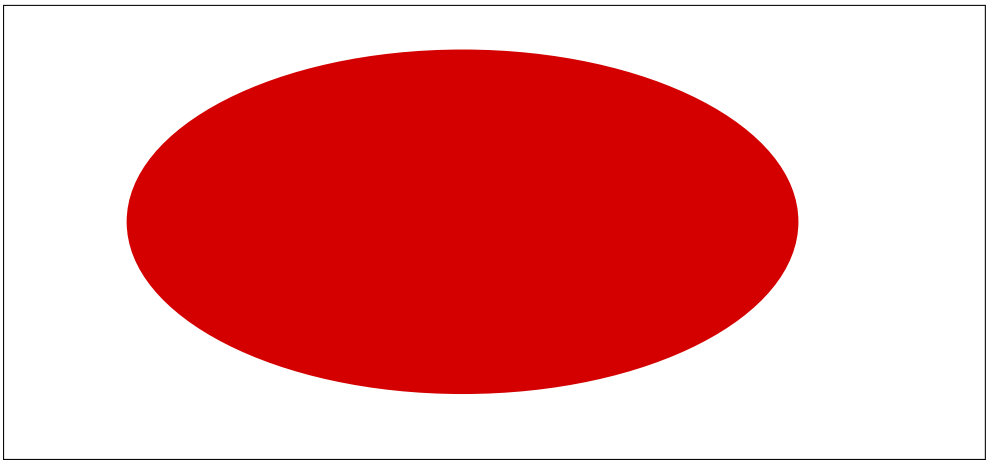
Trénovací model je vidět na obrázku 2.4.

[[jak jsem zjistil nejlepší vhodné nastavení]]

2.10 Prezentace dat

Vzhledem k tomu, že by aplikace měla být schopna určovat bezpečnostní rizika, je třeba nějakým způsobem prezentovat její výstupy. Moninitoring na aplikaci Unify je momentálně postaven na tom, že konkrétní lidé hlídají logy a v případě vyskytu chyb, varování nebo jiné netypické události zjišťují co bylo příčinou.

Rozhodl jsem se tedy, že nejlepší bude grafické znázornění. Kromě údajů o tom, že byl zaznamenán požadavek, který je podezřelý budu grafy využívat i k prezentaci základního monitoringu.



Obrázek 2.4: Detekce anomálií v prostředí MS AZURE ML Studio.

Vzhledem k tomu, že se bude veškerá komunikace ukladát bude vhodné prezentovat například i kolik požadavků na jednotlivé komponentě proběhlo za poslední hodinu a podobně.

Společnost Cetin a.s. [13] ve které bude aplikace testována a jenž je uživatelem integrační platformy používá pro různá grafická znázornění grafy od Google Charts[3].

Tyto grafy jsou napsané v jazyce Javascript. Je tedy možné jejich umístění například na intranetové stránky, kde se vysoce postavení lidé společnosti vyznají lépe než v jednotlivých monitorovacích aplikacích.

Na tomto základě jsem se rozhodl vytvořit REST API [2], jenž budou Google Charts schopny snadno konzumovat a v případné jiné aplikace, které by stály o podobná data budou schopny se jim přizpůsobit.

2.11 Využití dát systémy 3. stran

Do budoucna je potřeba počítat s rozšířením monitoringu a je proto vhodné aplikaci připravit tak, aby její výsledky mohly být využity v aplikacích 3. stran.

Lze předpokládat, že k monirování bezpečnosti provozu budou použity systémy SIEM (Security Information and Event Management) [14]. SIEM funguje na principu, kdy zpracovává co nejvíce údajú, na jejichž základě pak rozeznává neočekávané situace a rizika [15].

Tím, že jsem se rozhodl data ukládat tak, jak uvádím v kapitole 2.5 bude libovolný SIEM po připojení do DB schopen získat jak originální zprávu, tak její normalizovanou verzi popřípadě i výsledek vyhodnocení mé aplikace.

Dále je možnost napojit SIEM i na REST API obdobně jako Google Charts v kapitole 2.10.

Kapitola 3

Realizace

[[Sem vepsat nějaký úvod k této kapitole]]

3.1 Nutné přípravy pro jboss

3.1.1 Připravení modulů

V aplikaci využívám různé java knihovny, abych k nim měl přístup i v aplikačním serveru, je nutné do něj přidat speciální modul.

Jboss umožňuje snadné přidání modulů. Veškeré moduly jsou umístěny v *wildfly/jboss-eap-7.0/modules/system/layers*. Zde jsem vytvořil svůj modul s konkrétními java knihovnami:

- commons-codec-1.10.jar
- json-simple-1.1.1.jar
- mongo-java-driver-3.4.2.jar

3.1.2 Port offset

[[Je možné, že offset ve finále ještě změním.]] Dále bylo nutné pro jboss nastavit portový offset. Protože na serveru není jedinou aplikací, je běžný problém v kolizi portů. Z tohoto důvodu jsem zvolil offset 10000. Webové služby tedy místo portu 8080 běží na portu 18080.

3.1.3 Zapnutí CORS

CORS (Cross-origin resource sharing) neboli *sdílené zdroje odjinud* umožňuje odesílání odpovědí na požadavky z jiné domény [16]. V aplikaci je to potřebné pro rest api, kterého se následně dotazuje Google charts.

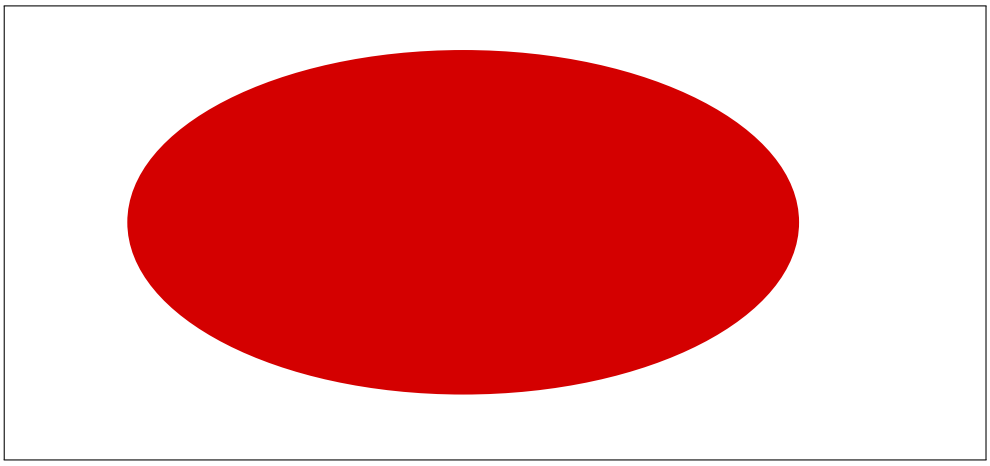
Corse se v Jboss povoluje v konfiguračním souboru pro standalone aplikaci *standalone.xml* pro doménovou *domain.xml*.

3.2 Vytvoření modelu na Azure

[[Ukázka URL]] [[Ukázka Vstupního JSONu a výstupního]] [[V této části bude Prediktivní algoritmus]]

3.2.1 Clustering v Azure

[[Popsat prediktivní experiment]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

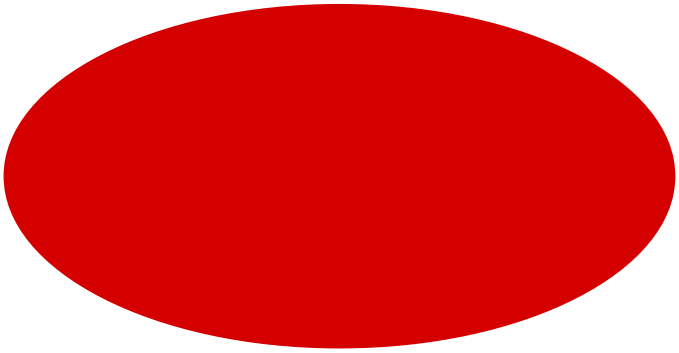


Obrázek 3.1: Prediktivní model clusteringu v Azure.

3.2.2 Detekce anomálií v Azure

[[Popsat prediktivní experiment]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus.

Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

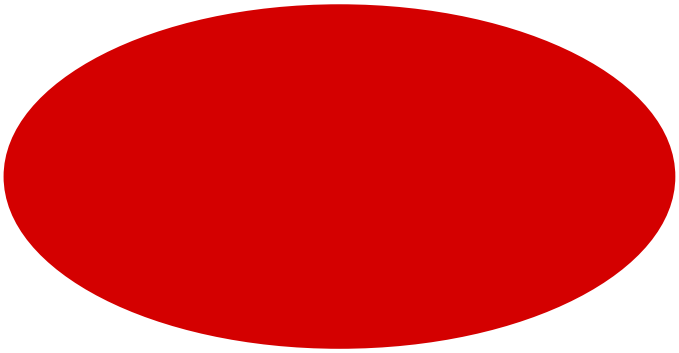


Obrázek 3.2: Prediktivní model v detekci anomálií v Azure.

3.2.3 Webová služba

Po dokončení prediktivního modelu je třeba experiment vystavit tak, abychom ho mohli používat z vlastní sítě.

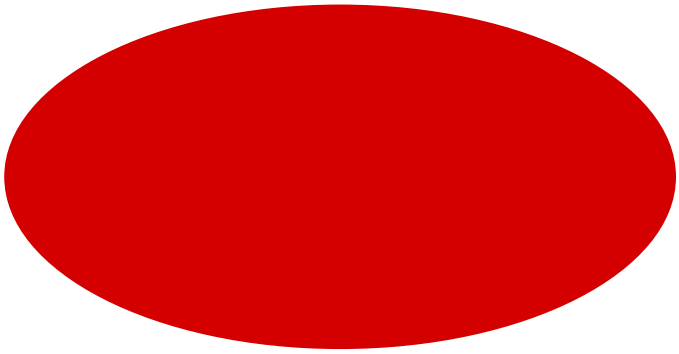
Azure umožňuje takový model spustit jako webovou službu.



Obrázek 3.3: Vytvoření webové služby pomocí stisku tlačítka.

Po vytvoření webové služby získáme takzvaný „API key“. Tento řetězec bude sloužit pro přihlášení se do Azure, při dotazování se na konkrétní službu.

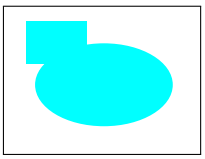
Také je možné službu otestovat. Otevře se nám okno s očekávanýma políčkama (obr. 3.4). Po vyplnění políček se zobrazí odpověď z prediktivního modelu. Tímto způsobem můžeme otestovat funkčnost nebo pár vzorků. Jiná použití by byla velmi časově a zdrojově nevýhodná.



Obrázek 3.4: Zobrazené okno pro otestování prediktivního modelu jako webové služby

3.3 MongoDB

[[Celkově z této sekce jsem rozpačitý]] Pro potřeby aplikace je nutné v co nejrychlejším možném čase ukládat jednotlivé requesty. Při ohromném provoze, který se na integrační platformě vyskytuje to je nutná podmínka pro to, aby bylo možné v reálném čase jednotlivé požadavky zpracovávat.



Obrázek 3.5: Logo MongoDB. [1]

Rozhodl jsem se za tímto účelem využít MongoDB [1], protože se očekává, že bude třeba ukládat v mimořádném případě až 30 záznamů za sekundu, zpravidla bude docházet k více zapisům než čtením.

3.3.1 NoSQL

MongoDB patří do takzvaných NoSQL databází [**?**]. NoSQL v angličtině znamená „Not Only SQL“ [**?**], v překladu „Nikoliv pouze SQL“. Jde o skupinu

nerelačních databází. Takové databáze nejsou primárně postavené na principu tabulek a zpravidla nepoužívají SQL pro práci s daty [**?**].

3.3.2 O MongoDB

MongoDB je licencovaná pod GNU AGPL v3.0 [**?**] licencí. Data jsou ukládána ve formátu BSON. BSON je binárně zakódovaná JSON [**?**].

V MongoDB se vytvářejí kolekce, každá kolekce obsahuje soubory. Soubory mají parametry [**?**]. Soubory a jejich parametry lze v čase libovlně měnit nebo přidávat. Což je výhoda, pokud zjistíme, že aktuální návrh není finální, vyhneme se problémům s migrací do nového schématu.

V rámci souboru je možné definovat čítač, který se využije k tomu, aby automaticky generoval jednoznačný identifikátor k souborům nebo lze využít parametr souboru **__id**. Ten vygeneruje jednoznačnou identifikaci, ze které jsme schopni například získat i čas vložení dokumentu do kolekce.

3.3.3 Využití v práci

3.3.3.1 Kolekce terms

V práci využívám databázi k ukládání všech slov, ze kterých se tvoří vektor, jenž reprezentuje konkrétní požadavek (více v kapitole 2.7). Tím není potřeba je mít v paměti a při připadném výpadku je znova vypočítávat.

V kolekci *terms* ukládám soubory jejichž struktura je automatický identifikátor, slovo pro konstrukci vekotru a timestamp přidání dokumentu do kolekce.

3.3.3.2 Kolekce messages

Další kolekcí je kolekce *messages*. V té jsou uložené veškeré požadavky, které byly přečteny z logů integrační platformy. Protože ještě před uložením do kolekce dochází v Azure k vyhodnocení, je zpráva uložené i s informací, která určuje zdali je požadavek vyhodnocen jako bezpečností riziko nebo není.

Struktura každého souboru je:

- **__id** - automaticky generovaný identifikátor
- **timestamp** - čas uložení souboru
- **original-message** - původní požadavek, tak jak byl převzat z logu integrační platformy
- **normalized-message** - požadavek ve znormalizované podobě
- **platform-id** - jednoznačný identifikátor v rámci integrační platformy
- **assignment** - informace od Azure s výsledkem přiřazení katogire

[[Lépe vysvětlit assignment]] [[Konfigurační kolekce]]

3.3.4 Práce s MongoDB v Javě

V implementaci jsem vytvořil třídu *MongoClientService* (aby bylo možné třídy využívat i v jiných modulech, musí se taková třída skládat z interfacu a jeho implementace, v textu se budu bavit o celku implementace a interfacu dohromady například jako o třídě MongoClientService). Tato třída umožňuje distribuci konkrétní databáZe napříč celou aplikací.

V jednotlivých modulech si vyvoláme instanci konkrétní databáze a nad tou jsme schopni pracovat. Ovladače pro MongoDB nám umožňují jak data v kládat, tak je číst.

3.4 Čtení dat z logů

[[Popis]] Při návrhu zisku jednotlivých požadavků z integrační platformy jsem vycházel z toho, že nový aplikace musí minimálně, či spíše vůbec nezatěžovat Unify [5]. Vzhledem k tomu, že přes integraci proudí veškerý provoz, je sama o sobě dosti vytížená a v případě, že by touto aplikací byl způsoben výpadek došlo by k silnému ztížení veškerých bussiness procesů, což si nelze dovolit.

Unify veškeré požadavky ukládá do logovacích souborů. Některé, převážně rizikové, služby se zároveň ukládají Oracle databáze. Ale vzhledem k tomu, že nejde o všechny dostupné služby rozhodl jsem se toho nevyužít.

Princip získání dat proudicích přes integrační platformu je založen na čtení jednotlivých logovacích souborů. Jako vhodný nástroj jsem vybral Java třídu Tailer z dostupné knihovny org.apache.commons.io [17].

Třída *Tailer*, po implementaci listeneru, se chová stejně jako linuxový příkaz tail [18]. Průběžně kontroluje čtený soubor a každou nově zapsanou řádku zpracovává.

Tímto řešením získáváme data z integrační platformy, aniž bychom jí zatěžovali.

3.5 Předzpracování a odeslání do Azure

[[Možná rozdělit na dvě sekce]]

Protože jsou data odesílána do cloudu, předzpracováváme je lokálně a přímo do Microsfot Azure odesíláme už jen identifikátor zprávy a vypočtený vektor.

Po přečtení zprávy z auditového logu Unify je zpráva předzpracována (2.6) a následně je z ní vytvořen vektor (2.7).

3.5.1 Start aplikace

První start aplikace je komplikovanější v tom, že pro výpočet finálního vektoru ještě nemáme známá vhodná slova, pro která se budou TF a IDF vypočítávat.

Pro případ, kdy je databáze zcela prázdná jsou nejdříve načteny nějaké zprávy (dle konfigurace), z těch jsou vypočetent vhodné termy. V případě, že databáze nějaké zprávy již obsahuje je možné využít je. Nedoporučovaný způsob je vložení slov přímo do databáze. Tato metoda může být vhodná v případě, že například chceme databázi migrovat a nechceme se zdržovat znovu výpočtem.

3.5.2 Získání dat z logu

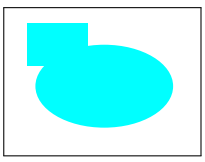
Samotný zisk dat z logu je řešen pomocí Java knihovny Tailer [**?**]. Jednou z věcí, které bylo potřeba vyřešit byla situace, kdy třída Tailer během čekání na nový přírustek logovacího souboru zcela zablokovala program. Situaci jsem vyřešil tím, že procesu, který čte z logu integrační platformy jsem pomocí *ExecutorService* [**?**] umožnil běžet na pozadí aplikace. Tím jsem neblokoval řízení programu.

3.5.3 Předzpracování dat

Celý proces předzpracování zpráv probíhá v implementované třídě *LogListener*. Po získání dat, jako textového řetězce, jsou uložena do struktury třídy *AuditLogMessage* (obr 3.6).

Následuje proces vytvoření vektoru, který je odesílán do MS Azure. Vektor se vytváří dle pravidel uvedených v sekci 2.7 včetně všech procesů předzpracování . Metody pro výpočet TF a IDF jsou implementované ve třídě *WeightsCounterService*.

Vektor samotný je reprezentován jako seznam *Double* čísel.



Obrázek 3.6: Struktura třídy AuditLogMessage.

Pro odeslání dat bylo třeba vytvořit třídu *AzureWebService*. Vytvoření vhodného požadavku na Azure je podmíněno přihlášením se do služby. Proto do hlavičky je přidána *Basic Access authentication* (jednoduché ověření přístup).

Na požadavek ihned dostaneme synchroní odpověď s výsledkem. Výsledek je zpracován a přiložen k datům načteným z logu.

3.6 Uložení dat

[[Popsat v jakém stavu jsou data před ukládáním]] [[Popsat jak se data ukládají]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

3.7 Napojení na google charts

[[Popsat jak vypadá javascript]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[Popsat obecné api, které google charts čeká]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignis-

sim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[Popsat jak je vyřešené rest API]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Kapitola 4

Analýza a vyhodnocení dat

[[Sem vepsat nějaký úvod k této kapitole]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

[[Popsat systém, na který to bylo nasazené]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

4.1 Analýza K-Means

[[Popsat vstupní parametry]] **[[Ukázat výsledky na Prod logu]]** **[[Dojít k závěru]]** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc,

molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

4.2 Analýza Outliner

[[Popsat vstupní parametry]] [[Ukázat výsledky na Prod logu]] [[Dojít k závěru]] Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan

semper.

Kapitola 5

Závěr

Conclusion

Literatura

[1] *MongoDB*. Available from: www.mongodb.com

[2]

[3] *Google Charts*. Available from: https://developers.google.com/chart/

[4] Johnson, C. H. High Level Design Distributed Network Traffic Controller. 02 2005. Available from: https://people.ok.ubc.ca/rlawrenc/research/Students/CJ_05_Design.pdf

[5] *Unify integration platform*. Available from: https://www.physter.com/unify/

[6] Available from: https://www.microsoft.com/cs-cz/

[7] Microsoft, . . *Microsoft Azure*. Available from: https://azure.microsoft.com/cs-cz/

[8] *Microsoft Azure Machine Learning Studio*. Available from: https://studio.azureml.net/

[9] *DEPLOY ANYWHERE WITH RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM*. Available from: https://www.redhat.com/cms/managed-files/mi-deploy-anywhere-jboss-eap-datasheet-inc0405103lw-201605-en.pdf

[10] *Apache Log4j 2*. Available from: https://logging.apache.org/log4j/2.x/

[11] *Class PatternLayout*. Available from: https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html

[12] Sproat, R.; Bedrick, S. CS506/606: Txt Nrmlztn. 2011. Available from: http://www.csee.ogi.edu/~sproatr/Courses/TextNorm/

[13] *CETIN*. Available from: https://www.cetin.cz/

[14] Constantine, C. SIEM and Log Management - Everything you need to know but were afraid to ask, Part 1. 2014. Available from: https://www.alienvault.com/blogs/security-essentials/everything-you-wanted-to-know-about-siem-and-log-management-but-were-afraid

[15] Work?, H. D. S. Written by Colton Bachman. 2016. Available from: https://www.integritysrc.com/blog/313-how-does-siem-work

[16] w3.org. Cross-Origin Resource Sharing. 2014. Available from: https://www.w3.org/TR/cors/#introduction

[17] *Tailer class*. Available from: https://commons.apache.org/proper/commons-io/javadocs/api-2.4/org/apache/commons/io/input/Tailer.html

[18] *Tail*. Available from: https://www.gnu.org/software/coreutils/manual/html_node/tail-invocation.html

Příloha A

Acronyms

GUI Graphical user interface

XML Extensible markup language

Příloha **B**

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `exe` ..... the directory with executables
- `src` ..... the directory of source codes
  - `wbdcm` ..... implementation sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.ps` ..... the thesis text in PS format