

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mojster Jaka

# **Vzorec zaključnega dela**

MAGISTRSKO DELO

MAGISTRSKI ŠTUDIJSKI PROGRAM DRUGE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Peter Klepec  
SOMENTOR: akad. prof. dr. Martin Krpan

Ljubljana, 2016



To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati zaključnega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani [creativecommons.si](http://creativecommons.si) ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda zaključnega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.



## ZAHVALA

*Na tem mestu zapišite, komu se zahvaljujete za izdelavo magistrske naloge. V zahvali se poleg mentorja spodobi omeniti vse, ki so s svojo pomočjo prispevali k nastanku vašega izdelka.*

*Mojster Jaka, 2016*



Vsem rožicam tega sveta.

*"The only reason for time is so that  
everything doesn't happen at once."*

— Albert Einstein





# Kazalo

**Povzetek**

**Abstract**

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Uvod</b>   | <b>1</b>  |
| <b>2</b> | <b>Pregled literature in ozadje</b>                       | <b>3</b>  |
| 2.1      | Farmacevtske dobavne verige . . . . .                     | 4         |
| 2.2      | Upravljanje zaupanja v razpršenih sistemih . . . . .      | 4         |
| 2.3      | Tehnologija veriženja blokov v dobavnih verigah . . . . . | 4         |
| 2.4      | Semantični splet in ontološke metode . . . . .            | 4         |
| 2.5      | Analiza vrzeli in prispevek te naloge . . . . .           | 4         |
| 2.6      | Zaupanje . . . . .  | 4         |
| 2.7      | Ontologija zaupanja . . . . .                             | 4         |
| 2.8      | Pametne pogodbe in pametni orakli . . . . .               | 4         |
| <b>3</b> | <b>Implementacija sistema</b>                             | <b>5</b>  |
| 3.1      | Arhitektura sistema . . . . .                             | 5         |
| 3.2      | Orodja in razvojno okolje . . . . .                       | 11        |
| 3.3      | Ontologija in orodja za sklepanje . . . . .               | 11        |
| 3.4      | Razvoj pametnih pogodb in orakov . . . . .                | 12        |
| 3.5      | Uporabniški vmesnik in API-ji . . . . .                   | 12        |
| <b>4</b> | <b>Evalvacija</b>   | <b>13</b> |
| 4.1      | Varnostna analiza . . . . .                               | 13        |

## KAZALO

|          |  |           |
|----------|--|-----------|
| 4.2      | Zmogljivost in razširljivost . . . . .       | 13        |
| 4.3      | Natančnost ocene zaupanja . . . . .          | 13        |
| 4.4      | Prednosti in omejitve . . . . .              | 13        |
| 4.5      | Primerjava z obstoječimi rešitvami . . . . . | 13        |
| 4.6      | Pridobljena spoznanja . . . . .              | 13        |
| <b>5</b> | <b>Zaključek in nadaljnje delo</b>           | <b>15</b> |
| 5.1      | Povzetek prispevkov . . . . .                | 15        |
| 5.2      | Nadaljnje raziskave . . . . .                | 15        |
| <b>A</b> | <b>Shema ontologije in SWRL pravila</b>      | <b>17</b> |
| <b>B</b> | <b>Izvilleki kode pametnih pogodb</b>        | <b>19</b> |

# Seznam uporabljenih kratic

| kratica | angleško                   | slovensko                             |
|---------|----------------------------|---------------------------------------|
| CA      | classification accuracy    | klasifikacijska točnost               |
| DBMS    | database management system | sistem za upravljanje podatkovnih baz |
| SVM     | support vector machine     | metoda podpornih vektorjev            |
| ...     | ...                        | ...                                   |



# Povzetek

**Naslov:** Vzorec zaključnega dela

V vzorcu je predstavljen postopek priprave magistrskega dela z uporabo okolja L<sup>A</sup>T<sub>E</sub>X. Vaš povzetek mora sicer vsebovati približno 100 besed, ta tukaj je odločno prekratek. Dober povzetek vključuje: (1) kratek opis obravnavanega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

## Ključne besede

*računalnik, računalnik, računalnik*



# Abstract

**Title:** Thesis template

This sample document presents an approach to typesetting your BSc thesis using L<sup>A</sup>T<sub>E</sub>X. A proper abstract should contain around 100 words which makes this one way too short. A good abstract contains: (1) a short description of the tackled problem, (2) a short description of your approach to solving the problem, and (3) (the most successful) result or contribution in your thesis.

## Keywords

*computer, computer, computer*





# Poglavje 1

## Uvod





## Poglavje 2

# Pregled literature in ozadje

### 2.1 Farmacevtske dobavne verige

#### 2.1.1 Izzivi sledenja in proti ponarejanju

#### 2.1.2 Regulativni okvir

### 2.2 Upravljanje zaupanja v razpršenih sistemih

#### 2.2.1 Modeli zaupanja

#### 2.2.2 Zaupanja v IoT in pervasive okolju

### 2.3 Tehnologija veriženja blokov v dobavnih verigah

#### 2.3.1 Javne in dovoljene verige blokov

#### 2.3.2 Pametne pogodbe za nadzor dostopa in sledljivost

### 2.4 Semantični splet in ontološke metode

#### 2.4.1 RDF, OWL in orodja za ontologijo

#### 2.4.2 Ontologija zaupanja

### 2.5 Analiza vrzeli in prispevek te naloge

## Poglavje 3

# Implementacija sistema

### 3.1 Arhitektura sistema

Arhitektura sistema za upravljanje zaupanja v farmacevtski dobavni verigi je sestavljena iz več plasti. Te omogočajo kombinacijo semantičnih podatkov, subjektivne ocene zaupanja in decentraliziranega shranjevanja rezultatov v verigo blokov. Z delitvijo sistema na plasti omogočamo modularnost, večjo preglednost in lažjo implementacijo nadaljnjih razširitev.

Arhitekturo razdelimo na tri plasti, katere bodo podrobneje opisane v naslednjih podpoglavjih:

#### 1. Plast znanja

V tej plasti se nahajajo vse ontološke definicije, primerki akterjev in njihove lastnosti. Sem spada tudi konceptualna zasnova lastnosti zaupanja, kot so licence, temperaturne nepravilnosti, točnost dostave ipd.

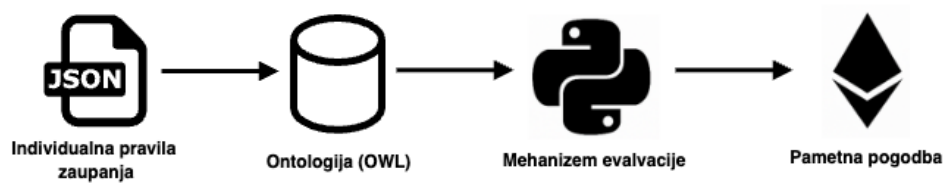
#### 2. Mehanizem za sklepanje zaupanja

Ta plast vključuje sistem za izvajanje ocenjevanja zaupanja. Združuje podatke iz ontologije in pravil zaupanja ter izvede sklepanje za vsak par entitet. Rezultat je preslikava razmerja zaupanja med akterji, ki je pripravljena za zapis v verigo blokov.

#### 3. Graf zaupanja na verigi blokov

Zadnja plast predstavlja pametno pogodbo, ki beleži razmerja zaupanja med entitetami. Shranjevanje rezultatov v verigo blokov omogoča transparentnost, trajnost in decentralizirano preverljivost zaupanja.

Na sliki 3.1 je prikazan arhitekturni pregled sistema.



**Slika 3.1:** Večplastna arhitektura sistema za upravljanje zaupanja

### 3.1.1 Plast znanja

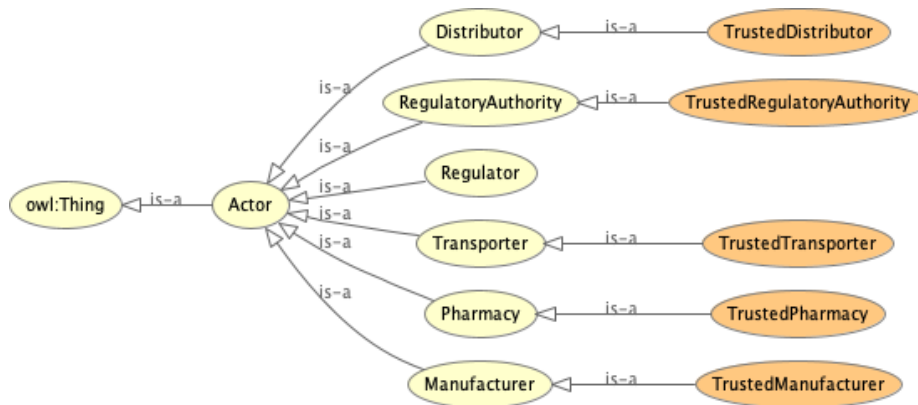
Plast znanja (angl. Knowledge Layer) definira semantično strukturo akterjev, njihove značilnosti in nabor pravil za individualno ocenjevanje zaupanja. V osnovi jo sestavljata dve ključni komponenti: ontologija in pravila zaupanja posameznih akterjev.

**Ontologija** Ontologija v jeziku OWL predstavlja formalno definiran model, ki vključuje vse vrste entitet, ki sodelujejo v farmacevtski dobavni verigi (npr. *Proizvajalec*, *Lekarna*, *Prevoznik*, *Regulator*). Vsaka vrsta entitete ima lahko specifične lastnosti, ki jih sistem uporablja pri sklepanju o zaupanju, kot so na primer:

- `hasDeliveryPunctuality` - točnost dostave,
- `hasTempViolationRate` - stopnja temperaturnih odstopanj,
- `hasLicense` - podatek o veljavni licenci,
- `hasGMP` - skladnost s proizvodnimi standardi GMP [1],
- `hasAuditScore` - rezultat presoje kakovosti.

Ontologija je zapisana v formatu RDF [2] zasnovan na označevalnem jeziku XML [3], ki omogoča enostavno obdelavo s knjižnico `rdflib` [4] v programskem jeziku Python. Na ta način lahko sklepanje temelji na aktualnih podatkih, zapisanih v ontološkem grafu, brez potrebe po relacijski bazi ali ročnem povezovanju.

Na sliki 3.2 vidimo vse akterje v ontologiji predstavljene v programu Protege.



**Slika 3.2:** Akterji ontologije v programu Protege

V ontologiji so definirani tudi posamezni primerki entitet (npr. *Pfizer*, *DHL*, *MediPlus*), skupaj z njihovimi lastnostmi. To omogoča sistemu, da na podlagi resničnih podatkov izvede evalvacijo zaupanja.

**Pravila zaupanja posameznih akterjev** Vsaka entiteta v sistemu lahko definira svoja lastna pravila zaupanja, ki opisujejo pogoje, pod katerimi določena entiteta zaupa drugi. Ta pravila so zapisana v berljivi in strukturirani obliki JSON, kar omogoča enostavno vključevanje v mehanizem sklepanja.

Primer politike pravil zaupanja za entiteto *Pfizer* je prikazan v spodnjem zapisu v formatu JSON:

```

{
  "actor": "http://example.org/trust#Pfizer",
  "trusts": {
    "Transporter": {
      "hasDeliveryPunctuality": { "gte": 0.9 },
      "hasTempViolationRate": { "lte": 0.05 }
    }
  }
}

```



Zapis pove, da entiteta *Pfizer* zaupa samo tistim prevoznikom, ki dosegajo vsaj 90 % pravočasnih dostav (`hasDeliveryPunctuality`  $\geq 0.9$ ) in katerih delež temperaturnih odstopanj pri transportu zdravil ne presega 5 % (`hasTempViolationRate`  $\leq 0.05$ ). Na ta način se pravila zaupanja natančno preslikajo v merljive pogoje, ki temeljijo na lastnostih, definiranih v ontologiji.

Mehanizem sklepanja nato uporabi ta pravila za vsako entiteto, s katero obravnavani akter vzpostavi zaupanje ter preveri, ali so pogoji izpolnjeni.

Ker so pravila zapisana v JSON obliki, je možno preprosto razviti uporabniški vmesnik, kjer lahko predstavnik podjetja (npr. administrator lekarne ali regulatorja) preko obrazca določil svoje kriterije za zaupanje. Vmesnik bi generiral JSON strukturo na podlagi definiranih lastnosti v ontologiji. To nam omogoča visoko stopnjo prilagodljivosti in enostavne uporabe brez tehničnega predznanja.

JSON pravila služijo kot most med semantičnim modelom in subjektivno presojo vsakega akterja. Tako omogočamo personalizirano ocenjevanje zaupanja.

### 3.1.2 Mehanizem za sklepanje zaupanja (angl. Trust Resolution Engine)

Mehanizem za sklepanje zaupanja predstavlja osrednjo logično plast sistema, kjer se združujejo podatki iz ontologije in pravila zaupanja posameznih akterjev. Tukaj se izvede proces ocenjevanja, ki določi, ali določena entiteta zaupa drugi glede na pogoje, zapisane v pravilih zaupanja posameznega akterja.

**Delovanje mehanizma** Python modul, razvit za potrebe sistema, izvaja naslednje korake:

1. Naloži ontologijo (OWL/RDF) in iz nje izlušči podatke o vseh akterjih ter njihovih lastnostih.

2. Prebere pravila zaupanja vsakega akterja, ki so zapisana v JSON formatu.
3. Za vsak par (ocenjevalec in tisti, ki je ocenjen) preveri, ali vrednosti atributov ocenjenega akterja ustrezajo pravilom ocenjevalca.
4. Rezultate ocenjevanja shrani v zbirko rezultatov (CSV datoteka) in jih pripravi za zapis v verigo blokov.

**Primer ocenjevanja** Če ima npr. akter *Pfizer* pravilo, da zaupa le prevoznikom z vsaj 90 % točnih dostav in manj kot 5 % temperaturnimi nepravilnostmi, bo mehanizem preveril te pogoje za vsakega prevoznika, zapisanega v ontologiji. Rezultat ocenjevanja je binarna odločitev (`true/false`), ki pove, ali akter izpolnjuje kriterije zaupanja.

### 3.1.3 Graf zaupanja na verigi blokov (angl. Blockchain Trust Registry)

Zadnja plast arhitekture je namenjena trajnemu in preverljivemu beleženju rezultatov ocenjevanja zaupanja. To dosežemo z uporabo pametne pogodbe na verigi blokov, ki deluje kot decentraliziran register zaupanja (angl. *trust registry*).

**Struktura pametne pogodbe** Pametna pogodba vsebuje podatkovno strukturo, ki beleži relacije zaupanja med akterji v obliki:

$$(akter1, akter2) \mapsto status\_zaupanja$$

kjer je `status_zaupanja` logična vrednost (`true/false`), ki označuje, ali akter 1 zaupa akterju 2.

**Funkcionalnosti** Pametna pogodba omogoča naslednje osnovne funkcije:

- `setTrustStatus()` — zapis oziroma posodobitev rezultata zaupanja na verigi blokov.

- `isTrusted()` — preverjanje, ali določen akter zaupa drugemu.
- `event TrustUpdated` — beleženje dogodkov o spremembi zaupanja za transparentnost in enostavno spremljanje.

## 3.2 Orodja in razvojno okolje

## 3.3 Ontologija in orodja za sklepanje

### 3.3.1 Ontologija zaupanja

Ontologija zaupanja je osnovni gradnik sistema, saj omogoča formaliziran opis vseh akterjev v farmacevtski dobavni verigi ter njihovih lastnosti. Zasnovana je v jeziku OWL. Glavni namen ontologije je, da lahko vse akterje opišemo na enoten način in jih obogatimo s podatki, ki so pomembni za ocenjevanje zaupanja.

#### Razredi akterjev

V ontologiji so definirani osnovni razredi, ki predstavljajo tipe akterjev. To so proizvajalec, distributer, lekarna, prevoznik in regulator. Vsi ti razredi so podrazredi splošnega razreda `Actor`, kar pomeni, da jih lahko obravnavamo skupaj, ko govorimo o celotni verigi. Vsak tip akterja ima svoje značilne lastnosti. Na primer, proizvajalci imajo lastnost `hasGMP`, ki pove, ali imajo veljavno GMP skladnost, prevozniki imajo lastnosti `hasDeliveryPunctuality` in `hasTempViolationRate`, lekarne pa lastnost `hasPrescriptionComplianceRate`.

#### Lastnosti zaupanja

Lastnosti opisujejo merljive kriterije, ki se uporabljajo pri presoji zaupanja. Nekaj primerov: - `hasGMP` opisuje ali je proizvajalec skladen z dobrimi proizvodnimi praksami, - `hasAuditScore` predstavlja oceno presoje kakovosti, - `hasLicense` označuje, ali ima akter veljavno licenco, - `hasDeliveryPunctuality`

izraža delež pravočasnih dostav, - `hasTempViolationRate` pove, kolikšen delež transportov je imel temperaturna odstopanja, - `hasPrescriptionComplianceRate` opisuje skladnost lekarne pri izdaji zdravil na recept, - `hasIssuedCertifications` je število certifikatov, ki jih je regulator izdal, - `hasJurisdictionLevel` pa določa raven pristojnosti regulatorja (npr. lokalna, nacionalna, globalna).

### **Razredi zaupanja**

Za vsak tip akterja so v ontologiji definirani tudi posebni razredi, ki predstavljajo zaupanja vredne entitete. To so na primer `TrustedManufacturer`, `TrustedDistributor`, `TrustedPharmacy`, `TrustedTransporter` in `TrustedRegulatoryAuthority`. Ti razredi so definirani z ekvivalenčnimi pogoji, kar pomeni, da reasoner lahko samodejno uvrsti entiteto v določen zaupanja vreden razred, če so pogoji izpolnjeni. Na primer, proizvajalec, ki ima GMP certifikat in dovolj visoko oceno presoje kakovosti, bo klasificiran kot `TrustedManufacturer`. Podobno velja za druge tipe akterjev.

### **Primerki akterjev**

V ontologijo so dodani tudi konkretni primerki, ki predstavljajo dejanske akterje v sistemu. Tako imamo na primer proizvajalca Pfizer in Novartis, prevoznika DHL, lekarno MediPlus, distributerja EuroLogistics ter regulatorja EMA. Vsak od teh primerkov ima določene lastnosti, kot so na primer ocena presoje, točnost dostave ali število izdanih certifikatov. Te vrednosti se potem uporabijo pri sklepanju o zaupanju.

## **3.4 Razvoj pametnih pogodb in orakov**

## **3.5 Uporabniški vmesnik in API-ji**

## Poglavje 4

### Evalvacija

4.1 Varnostna analiza

4.2 Zmogljivost in razširljivost

4.3 Natančnost ocene zaupanja

4.4 Prednosti in omejitve

4.5 Primerjava z obstoječimi rešitvami

4.6 Pridobljena spoznanja



## Poglavje 5

### Zaključek in nadaljnje delo

#### 5.1 Povzetek prispevkov

#### 5.2 Nadaljnje raziskave





## Dodatek A

### Shema ontologije in SWRL pravila



## **Dodatek B**

### **Izvlečki kode pametnih pogodb**



# Literatura

- [1] J. D. Nally, Good manufacturing practices for pharmaceuticals, CRC Press, 2016.
- [2] R. Cyganiak, D. Wood, M. Lanthaler, Rdf 1.2 concepts and abstract syntax, World Wide Web Consortium (W3C) Recommendation (2024).  
URL <https://www.w3.org/TR/rdf12-concepts/>
- [3] D. Beckett, B. McBride, Rdf/xml syntax specification (revised), World Wide Web Consortium (W3C) Recommendation (2004).  
URL <https://www.w3.org/TR/rdf-syntax-grammar/>
- [4] C. Boettiger, rdflib: A high level wrapper around the redland package for common rdf applications (2018). doi:10.5281/zenodo.1098478.  
URL <https://doi.org/10.5281/zenodo.1098478>