

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žan Pižmoht

**Sistem za zagotavljanje zaupanja v  
dobavni verigi zdravil z uporabo  
verige blokov**

MAGISTRSKO DELO  
MAGISTRSKI ŠTUDIJSKI PROGRAM DRUGE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Vlado Stankovski  
SOMENTOR: doc. dr. Petar Kochovski

Ljubljana, 2025



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 sestavnine dela kot tudi rezultati zaključnega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujojo, 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 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.*

*Žan Pižmoht, 2025*



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	Zaupanje . . . . .	3
2.2	Upravljanje zaupanja v razpršenih sistemih . . . . .	6
2.3	Farmacevtske dobavne verige . . . . .	9
2.4	Tehnologija veriženja blokov v dobavnih verigah . . . . .	12
2.5	Semantični splet in ontološke metode . . . . .	16
2.6	Analiza vrzeli in prispevek te naloge . . . . .	18
<b>3</b>	<b>Implementacija sistema</b>	<b>21</b>
3.1	Uporabljene tehnologije . . . . .	21
3.2	Arhitektura sistema . . . . .	23
3.3	Orodja in razvojno okolje . . . . .	32
3.4	Ontologija in orodja za sklepanje . . . . .	32
3.5	Razvoj pametnih pogodb in oraklov . . . . .	34
3.6	Uporabniški vmesnik in API-ji . . . . .	34
<b>4</b>	<b>Evalvacija</b>	<b>35</b>
4.1	Varnostna analiza . . . . .	35
4.2	Zmogljivost in razširljivost . . . . .	35

## *KAZALO*

4.3	Natančnost ocene zaupanja . . . . .	35
4.4	Prednosti in omejitve . . . . .	35
4.5	Primerjava z obstoječimi reštvami . . . . .	35
4.6	Pridobljena spoznanja . . . . .	35
<b>5</b>	<b>Zaključek in nadaljnje delo</b>	<b>37</b>
5.1	Povzetek prispevkov . . . . .	37
5.2	Nadaljnje raziskave . . . . .	37
<b>A</b>	<b>Shema ontologije in SWRL pravila</b>	<b>39</b>
<b>B</b>	<b>Izvlečki kode pametnih pogodb</b>	<b>41</b>

# Seznam uporabljenih kratic

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



# Povzetek

**Naslov:** Sistem za zagotavljanje zaupanja v dobavni verigi zdravil z uporabo verige blokov

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 obravnавnega problema, (2) kratek opis vašega pristopa za reševanje tega problema in (3) (najbolj uspešen) rezultat ali prispevek magistrske naloge.

## Ključne besede

*zaupanje, tehnologija veriženja blokov, dobavna veriga zdravil*



# Abstract

**Title:** Trust System in Pharmaceutical Supply Chain Using Blockchain

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

*trust, blockchain, pharmaceutical supply chain*



# Poglavlje 1

## Uvod



# Poglavlje 2

## Pregled literature in ozadje

To poglavje združuje temeljne koncepte, na katerih temelji sistem, razvit v okviru magistrske naloge. Začnemo z razumevanjem zaupanja in njegovih lastnosti ter s pregledom pristopov, ki se v razpršenih sistemih uporablja za njegovo modeliranje in ocenjevanje. V nadaljevanju se posvetimo farmacevtskim dobavnim verigam, kjer izpostavimo ključne težave, ki se pojavljajo pri zagotavljanju sledljivosti, varnosti in sodelovanja med akterji. Tu omenimo obstoječe rešitve, ki se že uporablja v praksi in predstavimo probleme, ki jih te rešitve imajo. Sledi pregled tehnologije veriženja blokov, pametnih pogodb in orakljev, ki danes predstavljajo pomemben del modernih decentraliziranih arhitektur. Poglavlje zaključimo s predstavitvijo semantičnega spletka in ontoloških metod, ki so potrebne za formalno opisovanje akterjev in njihovih lastnosti. Ta poglavja skupaj strnemo v množico, v kateri lahko prepoznamo vrzeli obstoječih pristopov in pripravimo teren za rešitev, ki jo predstavimo v nadaljevanju naloge.

### 2.1 Zaupanje

Pojem zaupanja je temeljni gradnik pri zasnovi našega sistema za farmacevtsko dobavno verigo. Zaupanje je kompleksen, večdimenzionalen in interdisciplinaren koncept, ki ga obravnavajo različne vede. Obravnavata se v sociologiji,

psihologiji, ekonomiji, računalništvu in še ostalih vedah. Prav zaradi te raznolikosti v literaturi ne obstaja enotna definicija, temveč več pristopov, ki poudarjajo različne vidike [1, 2].

### **2.1.1 Definicije zaupanja**

V družboslovнем kontekstu je zaupanje pogosto opredeljeno kot stanje pozitivnih pričakovanj glede dejanj druge osebe v okoliščinah, kjer obstaja določena stopnja tveganja [3]. Gambetta [4] zaupanje definira kot subjektivno verjetnost, da bo agent opravil določeno dejanje, še preden je to mogoče preveriti. Mayer, Davis in Schoorman [5] pa ga opredelijo kot pripravljenost ene stranke, da se izpostavi ranljivosti glede na dejanja druge stranke, ob pričakovanju, da bo ta delovala v skladu s pričakovanji.

V digitalnem okolju se koncept zaupanja prenaša iz družbenega v tehnični kontekst. Denning [6] trdi, da je zaupanje v sistem lastnost, ki jo je mogoče formalno določiti in jo modelirati. Sistem je torej zaupanja vreden, če mu njegovi uporabniki zaupajo glede na opaženo skladnost z vnaprej določenimi standardi. Grandison in Sloman [2] zaupanje obravnavata kot kvalificirano prepričanje zaupnika o kompetentnosti, integriteti, varnosti in zanesljivosti zaupanja vrednega subjekta.

### **2.1.2 Lastnosti zaupanja**

Zaupanje je po svoji naravi večdimenzionalen pojem, ki ga ni mogoče zajeti z eno samo definicijo. V literaturi se ponavlja, da zaupanje vključuje čustvene in vedenjske komponente, ki se med seboj prepletajo in vplivajo na odločanje posameznika ali sistema [7, 8]. V družbenem kontekstu gre za pripravljenost posameznika, da se izpostavi tveganju na podlagi prepričanja, da bo druga stran delovala predvidljivo in skladno s pričakovanji. V informacijskih sistemih pa to pomeni sposobnost sistema, da sprejema odločitve o sodelovanju na podlagi preteklih izkušenj in ocnjene zanesljivosti entitet.

Zaupanje obstaja le v razmerju med dvema ali več entitetami, kjer ena

zaupa drugi v določeni situaciji ali kontekstu. Takšno razmerje je dinamično, saj se lahko stopnja zaupanja sčasoma povečuje ali zmanjšuje. Poleg tega je zaupanje asimetrično: dejstvo, da entiteta A zaupa entiteti B, še ne pomeni, da bo tudi B zaupala A [5].

Ena od osrednjih značilnosti zaupanja je njegova povezanost s tveganjem in negotovostjo. Zaupanje je smiselno le v okoliščinah, kjer obstaja možnost, da drugi akter ne bo ravnal skladno s pričakovanji [3]. Če bi imeli popoln nadzor ali popolno informacijo, zaupanje sploh ne bi bilo potrebno. V kontekstu varnostnih in distribucijskih sistemov zaupanje dopoljuje formalne zaščitne mehanizme, kot so avtentikacija in šifriranje, saj omogoča oceno vedenja akterjev tam, kjer tehnični ukrepi ne zadostujejo [6].

Zaupanje je tudi subjektivno in kontekstualno. Različni akterji lahko enake vedenjske vzorce interpretirajo različno, odvisno od svojih ciljev, izkušenj, regulatornih zahtev, prepričanj itn.

### 2.1.3 Atributi zaupanja

Da bi bilo zaupanje uporabno v digitalnem okolju, ga je potrebno izraziti z merljivimi atributi, ki opisujejo lastnosti ali vedenje zaupanja vrednega subjekta. Ti atributi omogočajo, da se konceptualna ideja zaupanja preslika v formalne modele. V literaturi se pojavljajo različne skupine atributov.

Klasični modeli, kot sta Mayer, Davis in Schoorman [5] ter McKnight in Chervany [7], opredeljujejo tri temeljne dimenzije zaupanja: kompetentnost, integriteto in dobrohotnost. Kompetentnost se nanaša na sposobnost entitete, da učinkovito izvede nalogo, integriteta na njeno zavezanost etičnim in profesionalnim načelom, dobrohotnost pa na odsotnost namere povzročiti škodo. Kasneje so bili tem trem dodani še predvidljivost in zanesljivost, ki poudarjata konsistentnost vedenja skozi čas.

V digitalnih in porazdeljenih okoljih so se pojavili dodatni atributi, ki izhajajo iz tehničnih vidikov zaupanja. Grandison in Sloman [2] zaupanje opišeta kot prepričanje o kompetentnosti, varnosti in verodostojnosti drugega subjekta. Denning [6] poudarja, da je zaupanja vrednost sistema mogoče oce-

njevati tudi z vidika varnosti, skladnosti s pravili ter preverljivosti delovanja. Ti atributi omogočajo formalno ocenjevanje zaupanja med akterji.

V farmacevtski dobavni verigi imajo atributi zaupanja izrazito regulatorno in sledilno komponento. Uddin in sodelavci [9] poudarjajo, da so pri zagotavljanju zaupanja ključni kazalniki sledljivost, pristnost in celovitost podatkov. Kayhan in sodelavci [10] izpostavljajo še pomen transparentnosti, nespremenljivosti in izvorne sledljivosti (angl. *provenance*), ki jih omogoča tehnologija veriženja blokov.

Predstavljeni atributi ne predstavljajo univerzalne množice meril, temveč zbirko značilnosti, ki jih lahko posamezen akter uporabi pri lastni presoji zaupanja. V sistemu, ki je predstavljen v tej nalogi, se atributi uporabljajo predvsem kot podatkovne lastnosti (angl. *data properties*) v ontologiji. Te omogočajo, da akterji formalno zapišejo svoje kriterije za ocenjevanje drugih entitet. S tem v sistemu omogočamo standardizirano predstavitev vhodnih metrik, ki jih posamezni akterji uporabljajo pri svojem algoritmu za izračunavanje zaupanja.

## 2.2 Upravljanje zaupanja v razpršenih sistemih

Huaizhi Li in Mukesh Singhal v članku *Trust Management in Distributed Systems* [11] opredelita upravljanje zaupanja kot proces zbiranja informacij, potrebnih za vzpostavitev zaupanja med entitetami, ter dinamičnega spremeljanja in prilagajanja obstoječih razmerij zaupanja. Avtorja poudarjata, da v razpršenih okoljih kot so internet, sistemi enakovrednih entitet (angl. *peer-to-peer*) in mobilna omrežja ad hoc pogosto sodelujejo entitete, ki se med seboj ne poznajo. Zato zaupanje postane ključni mehanizem za zmanjšanje tveganja in zagotavljanje sodelovanja. Namen sistema za upravljanje zaupanja je ohranjati ažurne in skladne informacije o zaupanju med akterji v omrežju.

### 2.2.1 Dokazni modeli zaupanja

Dokazni modeli temeljijo na preverljivih dokazih, kot so digitalna potrdila, javni ključi ali kriptografski podpisi. Ti pristopi zagotavljajo preverjanje identitete in integritete entitet, ne pa tudi njihove dejanske zanesljivosti. Najpogostejsa primera sta hierarhični sistem X.509 [12] in decentralizirani sistem PGP [13], kjer se zaupanje posreduje prek verige certifikatov oziroma prek subjektivnih ocen uporabnikov. Tak model se uporablja predvsem pri inicializaciji zaupanja in zagotavlja osnovno preverjanje pristnosti brez ocenjevanja vedenja.

### 2.2.2 Priporočilni modeli zaupanja

Priporočilni modeli gradijo zaupanje na osnovi izkušenj in posredovanih ocen drugih entitet. Vsaka entiteta ocenjuje druge glede na pretekle interakcije, te ocene pa se lahko delijo naprej kot priporočila. Model uporablja pogojno tranzitivnost zaupanja – entiteta A lahko zaupa entiteti C, če A zaupa B kot priporočitelju in lahko ovrednoti njegovo oceno.

Pri tem se uporablja zvezna vrednost zaupanja  $tv_T$  med 0 in 1, izračunana iz vrednosti priporočil po poti  $A \rightarrow B \rightarrow C \rightarrow D$ . Za končno vrednost se uporabi zmnožek delnih zaupanj na poti:

$$tv_T = \prod_{i=1}^n \frac{rtv(i)}{4} \times tv(T)$$

kjer  $rtv(i)$  predstavlja stopnjo zaupanja v posameznega posrednika,  $tv(T)$  pa končno vrednost zaupanja v ciljno entitetu. Če obstaja več poti med akterjema, se končna vrednost določi kot povprečje rezultatov posameznih poti.

### 2.2.3 Porazdeljeno ocenjevanje zaupanja

V omrežjih enakovrednih akterjev (angl. *peer-to-peer*) se zaupanje ocenjuje decentralizirano. Vsako vozlišče vodi lastno evidenco interakcij in vrednoti

druge glede na uspešnost sodelovanja. Xiong in Liu predlagata metriko, ki temelji na razmerju med pozitivnimi in negativnimi interakcijami:

$$T(u, t) = \frac{\sum_{v \in P, v \neq u} S(u, v, t) Cr(v, t)}{\sum_{v \in P, v \neq u} I(u, v, t)}$$

kjer  $S(u, v, t)$  označuje zadovoljstvo uporabnika  $u$  z  $v$  do časa  $t$ ,  $Cr(v, t)$  je korekcijski faktor za filtriranje povratnih informacij,  $I(u, v, t)$  pa število interakcij med  $u$  in  $v$ . Rezultat  $T(u, t)$  je vedno v intervalu  $(0, 1)$  in predstavlja trenutno stopnjo zaupanja v entitetu. Sistem lahko določi pragove  $C_1$  in  $C_2$ , pri čemer entiteta velja za zaupanja vredno, če velja  $I(u, t) > C_1$  in  $T(u, t) > C_2$ .

Takšni sistemi so učinkoviti pri akumuliraju ocen, vendar občutljivi za napačne ali zlonamerne povratne informacije. Zato številne rešitve uvajajo uteževanje priporočil glede na zanesljivost virov ali časovno starost podatkov.

### 2.2.4 Dinamično posodabljanje zaupanja

Ker se vedenje akterjev sčasoma spreminja, se mora zaupanje prilagajati novim podatkom. Posodobitev vrednosti se pogosto izvede z amortizacijskimi ali diskontnimi faktorji, ki dajejo večjo težo nedavnim interakcijam. Primer takega mehanizma je izračun nove vrednosti zaupanja po zaključeni interakciji:

$$T_{\text{new}} = \frac{r + N T_{\text{old}} e^{-\beta \Delta t}}{1 + N e^{-\beta \Delta t}}$$

kjer  $T_{\text{old}}$  predstavlja prejšnjo vrednost zaupanja,  $r$  novo oceno po transakciji,  $N$  število izvedenih interakcij,  $\beta$  faktor dušenja,  $\Delta t$  pa čas od zadnje posodobitve. Na ta način sistem zagotavlja, da nove izkušnje hitreje vplivajo na oceno, stare pa postopoma izgubijo težo.

### 2.2.5 Povezava z našo rešitvijo

Predstavljeni modeli zaupanja predstavljajo osnovo za delovanje razvitega sistema. Vsak akter v dobavni verigi lahko uporabi svoj način izračuna zaupanja, ki temelji na dokazih, priporočilih ali dinamičnih metrikah. Sistem ne

vsiljuje enotnega algoritma, temveč omogoča, da akter sam oblikuje pravila na podlagi podatkov iz ontologije. Rezultati teh izračunov se shranjujejo v verigo blokov, ki deluje kot skupen in trajen zapis ocen. S tem smo združili različne pristope v enoten sistem, kjer je zaupanje merljivo in preverljivo.

## 2.3 Farmacevtske dobavne verige

### 2.3.1 Značilnosti in pomen sledljivosti

Farmacevtska dobavna veriga je ena najbolj reguliranih in kompleksnih industrijskih verig. Vključuje proizvajalce, distributerje, prevoznike, lekarne, regulatorne agencije in končne uporabnike [14]. Zaradi velikega števila vmesnih akterjev in globalnega obsega poslovanja je nadzor nad izvorom in kakovostjo zdravil zahtevna naloga.

Sledenje zdravilom omogoča identifikacijo posamezne serije skozi celoten življenjski cikel izdelka, od proizvodnje do izdaje pacientu [10]. Natančno beleženje gibanja zdravil preprečuje vstop ponaredkov v sistem in omogoča hitro ukrepanje ob odpoklicih ali neskladjih. Sledljivost je zato neposredno povezana z varnostjo pacientov in zaupnostjo zdravstvenega sistema [15].

Za zagotavljanje sledljivosti so regulatorji uvedli stroge predpise. Evropska unija je sprejela Direktivo o ponarejenih zdravilih (2011/62/EU) [16] in Delegirano uredbo (EU) 2016/161 [17], ki zahtevata enolično označevanje embalaže z dvodimenzionalno kodo in uporabo evropskega sistema EMVS. V ZDA podobno vlogo opravlja zakon DSCSA, ki uvaja popolnoma digitalno sledenje zdravil do leta 2025 [18].

Ti ukrepi predstavljajo pomembne korake k večji sledljivosti, vendar še vedno temeljijo na centraliziranih bazah podatkov, kjer posamezni akterji nimajo popolnega vpogleda v celotno verigo.

### 2.3.2 Izzivi in tveganja

Kljub regulacijam farmacevtske dobavne verige ostajajo izpostavljeni številnim tveganjem. Največji problem predstavljajo ponarejena in neustrezna zdravila, katerih število v zadnjih letih narašča. Po podatkih Svetovne zdravstvene organizacije je bilo med letoma 2017 in 2021 zabeleženih 877 incidentov ponarejenih ali neustreznih medicinskih izdelkov, kar pomeni povprečno letno rast za 36 % [19]. Ti izdelki so bili zaznani v vseh regijah sveta, tudi v državah z razvitetimi regulativnimi sistemi. Ponarejena zdravila so lahko neučinkovita ali nevarna, njihova prisotnost pa neposredno ogroža zdravje pacientov in zmanjšuje zaupanje javnosti v zdravstvene sisteme.

Drugo tveganje izhaja iz razdrobljenosti informacijskih sistemov. Podatki o serijah, skladiščih in pošiljkah se pogosto hranijo v ločenih bazah, ki med seboj niso povezane [9]. To povzroča zamude, napake in oteženo preverjanje izvora zdravila. Centralizirane rešitve, kot je EMVS, sicer izboljšajo preverjanje avtentičnosti, vendar ne rešujejo težave zaupanja med različnimi organizacijami, ki podatke ustvarjajo in posredujejo.

Poleg ponarejanja in pomanjkljive sledljivosti se pojavljajo tudi tveganja, povezana z motnjami v dobavi zdravil. V kriznih obdobjih, kot so pandemije ali politične napetosti, se zaradi pomanjkanja podatkov o zalogah in zanesljivosti partnerjev pogosto pojavijo nepričakovane prekinitve dobav. Takšni dogodki razkrivajo, kako pomembno je zaupanje med organizacijami in pravočasna izmenjava podatkov.

Kljub tehnološkim in pravnim izboljšavam obstoječi modeli še vedno ne zagotavljajo celovitega vpogleda v dogajanje znotraj verige. Potrebne so rešitve, ki združujejo varno upravljanje podatkov, decentralizirano preverjanje in semantično usklajevanje informacij med akterji.

### 2.3.3 Regulativni okvir in trenutne rešitve

Evropska zakonodaja temelji na Direktivi o ponarejenih zdravilih (2011/62/EU) [16] ter Delegirani uredbi (EU) 2016/161 [17]. Skupaj uvajata obvezne var-

nostne elemente na embalaži, dvodimenzionalno kodo ter sistem EMVS (European Medicines Verification System). EMVS omogoča preverjanje serijske številke ob izdaji zdravila, sistemi NMVS pa sodelujejo prek skupnega evropskega vozlišča. Sistem zaznava poskuse ponarejanja v fazi distribucije, ne omogoča pa vpogleda v celotno zgodovino serije, saj temelji na centralizirani arhitekturi [10].

V ZDA sledljivost ureja zakon DSCSA (Drug Supply Chain Security Act), ki je del zakona DQSA iz leta 2013 [18]. DSCSA uvaja elektronsko izmenjavo treh vrst podatkov: transakcijske informacije, zgodovine transakcij in izjave o skladnosti. Sistem zahteva popolnoma interoperabilno sledljivost do leta 2025. Kljub napredku ameriški model ostaja pretežno centraliziran, podatki pa se prenašajo predvsem med neposrednimi poslovnimi partnerji [9].

Na mednarodni ravni Svetovna zdravstvena organizacija (WHO) pripravlja smernice za dobre distribucijske prakse in spremi pojav ponarejenih in neustreznih izdelkov.

Za izpolnjevanje regulativnih zahtev podjetja uporabljam različne informacijske platforme, med katerimi so najpogosteje centralizirane rešitve, kot so že omenjeni EMVS, TraceLink in SAP ATTP. Te rešitve izboljšujejo sledljivost serij in omogočajo avtomatizirano poročanje regulatorjem, vendar ne omogočajo skupnega vpogleda v podatke ali preverjanja zanesljivosti akterjev po celotni verigi [10]. Podatkovni tokovi ostajajo razdrobljeni, zanašajo pa se tudi na zaupanja vredne posrednike.

V zadnjem desetletju se pojavlja projekti, ki preučujejo uporabo tehnologije veriženja blokov v farmacevtskih dobavnih verigah. Projekta Medi-Ledger v ZDA in PharmaLedger v Evropi uporabljata porazdeljene zapise za izboljšanje integritete in sledljivosti podatkov [9]. Takšni pristopi dokazujo, da blockchain omogoča nespremenljivost zapisov in enostavnejše preverjanje informacij, vendar večinoma rešujejo le problem integritete podatkov.

Kljub temu večina rešitev ostaja usmerjena v zaupanje v podatke, medtem ko je upravljanje zaupanja med posameznimi organizacijami pogosto implicitno, vezano na statične pravice dostopa ali vnaprej dogovorjene po-

godbeno-pravne odnose. Manj je pristopov, kjer bi lahko vsak akter izrecno definiral svoje kriterije zaupanja in iz njih izpeljal odločitve o sodelovanju. Prav to vrzel bo zapolnil sistem, predstavljen v tej nalogi, saj kombinira semantični model dobavne verige, subjektivne politike zaupanja in zapis rezultatov v verigo blokov.

## 2.4 Tehnologija veriženja blokov v dobavnih verigah

Tehnologija veriženja blokov je bila prvič predstavljena leta 2008 v sklopu kriptovalute Bitcoin [20]. Osnovna ideja je zasnova podatkovne strukture, ki povezuje zapise v verigo blokov. Vsak blok vsebuje podatke, kriptografski povzetek prejšnjega bloka in časovno oznako. Takšna vezava med bloki preprečuje naknadne spremembe podatkov, saj bi sprememba v enem bloku povzročila neujemanje celotne verige [21].

Veriga blokov deluje v porazdeljenem omrežju, kjer vsako vozlišče hrani kopijo celotne verige. Soglasje o pravilnem zaporedju blokov se doseže s konsenznim mehanizmom. V javnih verigah se pogosto uporablja t. i. Proof of Work ali Proof of Stake, medtem ko dovoljena omrežja uporabljajo lažje mehanizme, ki temeljijo na znanih udeležencih. Zaradi porazdeljene narave sistem ne potrebuje centralne avtoritete, kar omogoča zanesljivo beleženje podatkov v okolju z različnimi deležniki.

Poleg javnih omrežij, kot je Ethereum, so v praksi pogosta dovoljena omrežja, kjer je dostop omejen na pooblašcene organizacije. Primer takšnega sistema je Hyperledger Fabric [22]. Ta pristop omogoča višjo prepustnost, večjo zasebnost podatkov in nadzor nad identitetami v omrežju. V dobavnih verigah so takšne lastnosti ključne, saj morajo podjetja varovati poslovne podatke, hkrati pa zagotavljati zanesljivo izmenjavo informacij.

Verige blokov omogočajo nespremenljiv zapis dogodkov, preverljivo zgodovino podatkov in enoten pogled na transakcije. To so pomembne zahteve farmacevtskih dobavnih verig, saj želimo omogočiti transparenten pogled na

transakcije v takšnem sistemu.

### **2.4.1 Pametne pogodbe**

Pametne pogodbe predstavljajo enega ključnih konceptov sodobnih verig blokov. Gre za programe, ki se shranijo in izvajajo neposredno v verigi blokov ter se sprožijo, ko so izpolnjeni določeni pogoji. Ethereum je prvi uvedel splošno platformo za izvajanje pametnih pogodb prek navideznega stroja EVM (Ethereum Virtual Machine), ki omogoča deterministično izvajanje kode na vseh vozliščih omrežja [23].

Pametne pogodbe omogočajo avtomatizacijo pravil in procesov, pri čemer ni potrebno zaupati enemu samemu posredniku. Njihovo delovanje je predvidljivo, saj se vedno izvršijo natančno tako, kot je določeno v kodi. Zaradi teh lastnosti se pogosto uporabljajo v scenarijih, kjer je treba uveljaviti dogovorjena pravila med neodvisnimi akterji. V dobavnih verigah se uporabljajo za potrjevanje dogodkov, registracijo premikov blaga ter preverjanje skladnosti podatkov.

V modelih, ki temeljijo na verigi blokov, pametne pogodbe služijo tudi kot enotna plast poslovne logike. S tem omogočajo, da vsi akterji delujejo na osnovi istega sklopa pravil. To je posebej pomembno v okoljih, kjer se morajo organizacije dogovarjati o interpretaciji podatkov ali izvajjanju procesov. Pametne pogodbe z nespremenljivo kodo zagotovijo, da se pravila ne spreminjajo brez soglasja deležnikov.

V predstavljenem sistemu ima pametna pogodba osrednjo vlogo, saj hrani rezultate ocenjevanja zaupanja in omogoča preverjanje vzpostavljenih razmerij med akterji. Pogodba vsebuje funkcije za zapisovanje rezultatov, njihovo posodabljanje in preverjanje stanja zaupanja. Tako tvori register zaupanja, ki je dostopen vsem udeležencem verige.

### 2.4.2 Orakli v pametnih pogodbah

Pametne pogodbe lahko zanesljivo izvajajo le pravila nad podatki, ki so že zapisani na verigi blokov. Ker so verige blokov zasnovane kot zaprt, determinističen sistem, nimajo neposrednega dostopa do zunanjega sveta. Da bi pametne pogodbe lahko reagirale na dogodke iz realnega okolja (cene na trgu, vremenske razmere, meritve senzorjev, podatke informacijskih sistemov), potrebujemo posrednika, ki te podatke pridobi in jih v preverljivi obliki posreduje na verigo. Tak posrednik se imenuje orakelj [24].

Caldarelli oraklje opisuje kot celoten ekosistem, ki povezuje zunanje vire podatkov z decentralizirano aplikacijo [25]. Tipično ga sestavlja trije deli: vir podatkov, komunikacijski kanal in pametna pogodba. Vir podatkov je lahko spletni vmesnik API, podatkovna baza, senzor v internetu stvari ali celo človek, ki potrdi nek dogodek.

Komunikacijski kanal predstavlja vozlišče oziroma niz vozlišč, ki podatke zbrane iz vira prevedejo v obliko transakcije in jih pošljejo pametni pogodbi. Pametna pogodba na drugi strani vsebuje pravila, s katerimi sprejme ali zavrne posredovane podatke in na njihovi podlagi sproži ustrezno logiko.

Ker oraklji ponovno uvajajo zaupanje v zunanje subjekte, se je v literaturi uveljavil pojmom *problem oraklja* (angl. oracle problem) [24]. Blockchain sam po sebi nudi lastnosti, kot so nespremenljivost, deterministično izvajanje in odpornost proti cenzuri, vendar te lastnosti ne veljajo avtomatično za podatke, ki prihajajo preko oraklja. Če vir podatkov ni zanesljiv ali je komunikacijski kanal centraliziran, lahko napačni podatki povzročijo napačno izvedbo pametne pogodbe, ne da bi to bilo na verigi neposredno razvidno. Zato je v skupnosti poudarjena potreba po jasnem modelu zaupanja za vsak orakelj. Ta mora opisati, kakšen je motiv oraklja, da deluje pošteno, in kako sistem zazna ali kaznuje odstopanja [24].

V praksi se pojavlja več arhitektur, ki omogočajo delovanje orakljev. Najpreprostejši so centralizirani oraklji, kjer ena organizacija zbira podatke in jih pošilja pametnim pogodbam. Tak pristop je enostaven za implementacijo, vendar ponovno uvaja enotno točko zaupanja in napake. Sledijo jih poraz-

## 2.4. TEHNOLOGIJA VERIŽENJA BLOKOV V DOBAVNIH VERIGAH<sup>15</sup>

deljeni oziroma decentralizirani oraklji, kjer več neodvisnih vozlišč pridobiva in agregira podatke ter tako zmanjšuje tveganje manipulacije. Primer takšne zasnove je sistem ASTRAEA, ki uporabi glasovalno igro z vložki (angl. stake) in ločene vloge udeležencev (oddajatelji trditev, glasovalci, certificiranci), da z ekonomsko motivacijo spodbudi udeležence k poštenemu poročanju o resničnosti trditev [26]. Podobno pristopajo tudi druge rešitve, kjer se omrežja orakljev, kot je Chainlink, uporablajo za zbiranje podatkov iz več API-jev in senzorjev ter za izračun dodatnih metrik (npr. reputacije) še pred zapisom v verigo blokov [27].

V okviru te naloge oraklje obravnavamo kot ločeno plast med semantičnim delom sistema in verigo blokov. Podobno kot pri Chainlinku smo zasnovali decentralizirano omrežje pametnih orakljev, kjer teče koda. Ta iz ontologije in pravil zaupanja prebere podatke, izvede izračun zaupanja ter rezultat posredujejo pametni pogodbi. S tem sledimo pristopu, ki ga Kochovski in sodelavci imenujejo *pametni oraklji brez zaupanja* (angl. trustless smart oracles), kjer so oraklji zasnovani tako, da so preverljivi in zamenljivi, veriga blokov pa služi kot nespremenljiv dnevnik rezultatov, ki jih izračunajo oraklji [27]. Na ta način odpravimo potrebo po centraliziranem zaupnem posredniku, hkrati pa omogočimo, da se različne implementacije orakljev oz. različni izračuni zaupanja enotno zapišejo v register zaupanja na verigi blokov. Podrobnejši opis delovanja decentraliziranega sistema orakljev si bomo pogledali v poglavju 3.

### **2.4.3 Decentralizirane identitete in preverljiva dokazila**

Decentralizirane identitete (DID) predstavljajo mehanizem za dosledno in preverljivo identifikacijo akterjev v decentraliziranih okoljih. Standard W3C DID opisuje identifikatorje, ki niso vezani na centraliziranega ponudnika, temveč so pod nadzorom lastnika identitete [28]. DID je globalno enoličen identifikator, na primer `did:ethr:0x1234...`, pri čemer se podatki o tem, kako identiteto preveriti, hranijo v tako imenovanem DID dokumentu. Ta

dokument vsebuje javne ključe, metode preverjanja in dodatne atribute, ki jih sistem lahko uporabi pri avtentikaciji.

Preverljiva dokazila (angl. Verifiable Credentials - VC) so standardizirana digitalna dokazila, ki jih neka entiteta izda drugi entiteti. Model W3C sledi arhitekturi izdajatelj–imetnik–preveritelj. Izdajatelj (Issuer) ustvari in podpiše dokazilo, imetnik (Holder) ga hrani, preveritelj (Verifier) pa ga validira s pomočjo kriptografskih podpisov [29]. VC omogočajo prenos lastnosti ali certifikatov na način, ki je preverljiv, varen in ne potrebuje zaupanja v posrednika. V literaturi so prepoznani kot jedrni gradnik digitalnih identitet, ki združujejo suverenost uporabnika, interoperabilnost ter preverljivost podatkov [30].

V predstavljenem sistemu ima vsak akter v ontologiji lasten DID. DID služi kot osnovni identifikator, na katerega se vežejo atributi in podatki v ontološkem grafu. Na ta način sistem ohrani enotno identiteto akterjev med vsemi plastmi arhitekture.

Preverljiva dokazila uporabljam za opis lastnosti, ki so pomembne za ocenjevanje zaupanja. Regulator, kot je EMA, lahko na primer izda VC, ki potrjuje veljavnost licence, certifikat GMP ali rezultat presoje kakovosti. Ta dokazila so digitalno podpisana in povezana z DID akterja. Orakelj lahko med izračunom zaupanja prebere VC, preveri podpis izdajatelja ter iz podatkov izlušči metrike, ki jih akter uporablja v svoji politiki zaupanja. Če akter v svojih pravilih določi, da mora imeti partner veljavno licenco ali ustrezno oceno presoje kakovosti, lahko te kriterije pridobi neposredno iz VC.

## 2.5 Semantični splet in ontološke metode

Semantični splet predstavlja razširitev svetovnega spleta, kjer podatki niso opisani le sintaktično, temveč tudi semantično. To pomeni, da so podatki strukturirani na način, ki omogoča strojno razumevanje in logično sklepanje. Temeljna ideja je, da lahko različni sistemi izmenjujejo podatke v enotnih formatih ter iz njih izpeljujejo nova dejstva, ne da bi se morali dogоворiti o

skupni implementaciji ali aplikaciji [31].

RDF (Resource Description Framework) je osnovni podatkovni model semantičnega spletja. Podatke predstavlja v obliki trojk subjekt–predikat–objekt, kar omogoča enostavno združevanje in razširjanje podatkovnih grafov [32]. RDF ne predpisuje sheme, zato se lahko prilagodi različnim domenam. Zapis se pogosto izvaja v obliki Turtle, RDF/XML ali JSON-LD.

Nad RDF je zasnovan OWL (Web Ontology Language), ki omogoča formalno opisovanje razredov, lastnosti in odnosov med pojmi. OWL temelji na deskripcijski logiki, zaradi česar omogoča avtomatsko sklepanje in preverjanje konsistence ontologije [33]. Z uporabo reasonerjev lahko sistem samodejno razvršča primerke v ustrezne razrede, zazna protislovja in izpelje nova dejstva, ki niso bila eksplisitno zapisana.

Ontologije so osrednji gradnik semantičnega spletja. Predstavljajo formalne modele znanja, kjer so pojmi, lastnosti in hierarhije jasno definirani. Tak pristop je posebej uporaben v kompleksnih domenah, kjer nastopa veliko različnih akterjev in množica podatkovnih razmerij. Ontologije omogočajo enotno semantično podlago, ki ni vezana na implementacijo posameznega sistema, temveč na konceptualni model podatkov.

V predstavljenem sistemu ontologija služi kot centralni model znanja za vse akterje farmacevtske dobavne verige. RDF omogoča predstavitev akterjev in njihovih lastnosti kot povezav v grafu, OWL pa dodaja formalne razrede in omejitve, ki omogočajo avtomatsko sklepanje o lastnostih ali vlogah akterjev. Reasoner iz RDF–OWL grafa izlušči podatke, ki jih orakelj nato uporabi pri izračunu zaupanja. Z uporabo semantičnih tehnologij je mogoče podatke strukturirati na način, ki je razširljiv, interoperabilen in nedvisen od posamezne aplikacije. Zato je ontološka plast primerna kot osnova za izmenjavo informacij med različnimi organizacijami in se dobro povezuje s preostalo arhitekturo sistema.

## 2.6 Analiza vrzeli in prispevek te naloge

V tem poglavju smo predstavili ključne gradnike, ki so pomembni za razumevanje razvitega sistema. Najprej smo opredelili pojem zaupanja in pokazali, da gre za večdimenzionalen koncept, ki ga ni mogoče zajeti z eno samo definicijo. Različni modeli poudarjajo kompetentnost, integriteto, dobrohotnost in zanesljivost, v digitalnem okolju pa se temu pridružijo še vidiki varnosti, verodostojnosti in skladnosti z zahtevami sistema. Nato smo opisali modele upravljanja zaupanja v razpršenih sistemih ter pokazali, da se ti modeli večinoma osredotočajo na izračun številčne vrednosti zaupanja na podlagi dokazov, priporočil ali zgodovine interakcij.

V nadaljevanju smo predstavili posebnosti farmacevtskih dobavnih verig. Te verige so močno regulirane in vključujejo veliko različnih akterjev. Regulativa, kot sta evropska direktiva o ponarejenih zdravilih in zakon DSCSA v ZDA, uvaja sledljivost serij in elektronsko izmenjavo podatkov. Kljub temu ostajajo pomembne vrzeli. Obstojeci sistemi večinoma temeljijo na centraliziranih bazah podatkov, kjer posamezna podjetja nimajo celovitega vpogleda v verigo. Ponarejena ali neustrezna zdravila se še vedno pojavljajo, podatkovni tokovi pa so razdrobljeni in vezani na zaupanja vredne posrednike. Trenutne rešitve tako izboljšujejo sledljivost, ne rešujejo pa dovolj dobro vprašanja, komu lahko posamezna organizacija zaupa pri sodelovanju.

Tehnologija veriženja blokov in pametne pogodbe ponujajo nov način upravljanja podatkov v takih okoljih. Verige blokov omogočajo nespremenljiv zapis dogodkov in enoten pogled na podatke v porazdeljenem omrežju. Pametne pogodbe omogočajo izvajanje skupnih pravil, ki veljajo za vse udeležence. Oraklji rešujejo problem povezave med verigo blokov in zunanjimi viri podatkov, vendar ponovno odpirajo vprašanje, komu zaupati pri posredovanju podatkov. Semantični splet in ontologije pa omogočajo formalno predstavitev znanja o akterjih in njihovih lastnostih ter avtomatsko sklepanje nad podatki. V literaturi se ti pristopi le redko združijo v enotno arhitekturo, kjer bi imeli akterji možnost, da na podlagi skupne ontologije definirajo svoje kriterije zaupanja, rezultat teh ocen pa se nato trajno zapise v verigo blokov.

Na tej točki se pokaže osrednja vrzel. Obstojče rešitve za sledljivost zdravil se osredotočajo na varnost in integriteto podatkov, upravljanje zaupanja med organizacijami pa je običajno implicitno. Temelji na statičnih pravicah dostopa, pogodbah in ročnih pregledih partnerjev. Po drugi strani številni modeli zaupanja in ontološki okviri ne upoštevajo posebnosti farmacevtske domene in niso povezani z verigo blokov. Manj je pristopov, kjer bi lahko vsak akter izrecno zapisal svojo politiko zaupanja, jo vezal na skupni semantični model ter rezultate ocenjevanja delil na način, ki ga lahko drugi akterji neodvisno preverijo.

Prispevek te naloge je predlog sistema, ki te elemente poveže v enoten okvir. Ontologija farmacevtske dobavne verige predstavlja skupno semantično plast, v kateri so akterji in njihove lastnosti opisani na strukturirani in razširljiv način. Vsak akter lahko nad temi podatki definira lastna pravila zaupanja v berljivi obliki JSON in tako določi, katere metrike so zanj pomembne. Mehanizem za sklepanje iz ontologije prebere lastnosti akterjev, uporabi lokalna pravila in za vsak par udeležencev izračuna rezultat zaupanja. Decentralizirani oraklji nato te rezultate prenesejo v pametno pogodbo, ki na verigi blokov hrani register relacij zaupanja. Sistem omogoča tudi vezavo na decentralizirane identitete in preverljiva dokazila, kadar akterji želijo izvor podatkov podpreti z digitalnimi certifikati.

Tak pristop neposredno naslavlja vrzeli, opisane v tem poglavju. Namesto enotne, vnaprej določene metrike zaupanja omogoča, da vsak akter oblikuje lastno politiko na osnovi skupnega modela podatkov. Rezultati teh različnih politik so kljub temu zbrani v enoten register, ki ga lahko delijo vsi udeleženci verige. S tem sistem podpira različne poglede na zanesljivost partnerjev, hkrati pa ohranja konsistentno predstavitev identitet in podatkovnih lastnosti.

Čeprav je sistem razvit na primeru farmacevtske dobavne verige, je zasnova splošna. Ontologijo in nabor atributov bi bilo mogoče prilagoditi drugim domenam, na primer pametnim mestom ali omrežjem interneta stvari, kjer številni akterji izmenjujejo podatke in storitve. Enak vzorec se ponovi

tudi tam: skupni semantični model entitet, subjektivne politike zaupanja, modul za sklepanje in register rezultatov na verigi blokov.

V nadaljevanju naloge bomo opisali, kako je tak sistem konkretno implementiran ter kako se obnaša v izbranih scenarijih uporabe.

# Poglavlje 3

## Implementacija sistema

### 3.1 Uporabljene tehnologije

Pri razvoju sistema smo uporabili različna orodja in platforme, ki skupaj omogočajo obdelavo semantičnih podatkov, izvajanje izračuna zaupanja ter zapis rezultatov v verigo blokov. V tej sekciji podajamo pregled najpomembnejših tehnologij in njihovih vlog v sistemu.

**Apache Jena Fuseki** Semantični podatki o akterjih in njihovih lastnostih so shranjeni v strežniku Apache Jena Fuseki, ki predstavlja RDF podatkovno bazo z dostopom prek SPARQL vmesnika. Fuseki omogoča centralizirano hrambo ontologije in primerkov ter enoten način pridobivanja podatkov za vse oraklje. Zaradi stabilnosti, enostavne uporabe in dobre združljivosti s standardi RDF je bila izbira naravna. Sistem teče v Docker vsebniku, kar omogoča hitro postavitev in ponovljivost okolja.

**Ontologija v OWL/RDF** Ontologija farmacevtske dobavne verige je zapisana v jeziku OWL, ki definira razrede, lastnosti in odnose med akterji. Podatki so organizirani v RDF trojkah, kar omogoča interoperabilnost z različnimi orodji semantičnega spletja. Ontologija predstavlja osrednjo plast sistema, saj definira strukturo znanja, ki jo oraklji uporablajo pri sklepa-

nju. Celoten model je fleksibilen, zato ga je mogoče prilagoditi tudi drugim domenam.

**JSON pravila zaupanja** Vsak akter lahko svoje kriterije zaupanja definira v JSON obliki. JSON je izbran zaradi enostavne berljivosti in neposredne integracije s Pythonom. Pravila določajo pragove, ki jih mora akter izpolniti, na primer minimalno točnost dostave ali največjo dovoljeno stopnjo temperaturnih odstopanj. Python orakelj ta pravila interpretira in preveri, ali podatki iz ontologije ustrezajo zahtevam ocenjevalca.

**Ethereum, Solidity in pametna pogodba** Za zapis rezultatov izračunanega zaupanja smo uporabili verigo blokov Ethereum. Pametna pogodba TrustGraph, napisana v jeziku Solidity, hrani relacije zaupanja med akterji. Ethereum je bil izbran zaradi široke podpore za orodja, stabilnega ekosistema in zmožnosti izvajanja pametnih pogodb brez zaupanja v enega posrednika. Razvoj in testiranje sta izvedena s pomočjo orodja Foundry, ki nudi lokalno testno omrežje, kompilator ter hitre skripte za izvajanje transakcij.

Pametna pogodba omogoča zapis rezultatov izračuna, preverjanje stanja zaupanja in beleženje dogodkov. Deluje kot nespremenljiv register, ki je dostopen vsem akterjem sistema.

**Decentralizirane identitete in preverljiva dokazila** Vsak akter v ontologiji ima dodeljen decentraliziran identifikator (DID). DID služi kot enoličen identifikator, ki povezuje podatke v ontologiji s predstavnikom akterja v decentraliziranem okolju. Poleg tega sistem podpira uporabo preverljivih dokazil (VC), ki jih lahko izdajajo regulatorji, kot je EMA. Ta dokazila vsebujejo metrike, kot so veljavnost licence, GMP skladnost ali rezultati presoje kakovosti. Orakelj lahko, kadar akter to določi v pravilih, med izračunom zaupanja prebere VC in uporabi podatke kot dodatne kriterije.

**Python in osnovne knjižnice** Jedro mehanizma za izračun zaupanja je razvito v programskega jezika Python. Uporabljam ga predvsem zaradi

enostavne integracije z ontologijo, preprostega dela z JSON datotekami ter dobre podpore za povezovanje z verigo blokov prek knjižnice `web3.py` [34].

**Pametni oraklji** Oraklji predstavljajo most med semantično plastjo in verigo blokov. Vsak orakelj zažene Python modul, ki prebere ontologijo, uporabi lokalna pravila akterja in izračuna vrednosti zaupanja. Nato preko `web3.py` pošlje transakcijo pametni pogodbi. Oraklji so zasnovani modularno, zato jih lahko organizacije poganjajo samostojno, kar omogoča decentralizirano izvajanje izračuna.

**Docker** Vse komponente sistema so zajete v vsebnikih Docker. To nam je omogočilo hitro postavitev razvojnega okolja, enostavno upravljanje odvisnosti in ponovljivost testov. Vsak orakelj, strežnik Fuseki, in lokalna Ethereum veriga tečejo v ločenih vsebnikih. Docker je uporabljen izključno za potrebe testiranja med razvojem.

## 3.2 Arhitektura sistema

Arhitektura razvitega sistema temelji na večplastnem pristopu, ki ločuje podatkovni model, mehanizme izračuna in zapis rezultatov zaupanja. Takšna zasnova omogoča jasne meje med posameznimi odgovornostmi, kar olajša razvoj, testiranje in morebitno razširjanje sistema na druge domene. V okviru magistrske naloge smo zasnovali arhitekturo, ki je dovolj splošna, da jo je mogoče uporabiti v različnih razpršenih okoljih, od farmacevtskih dobavnih verig do pametnih mest.

Osnovna ideja arhitekture je bila, da združimo tri ključne tehnologije: semantični splet, pametne oraklje in verigo blokov. Semantični splet omogoča opis akterjev in njihovih lastnosti v ontološkem grafu. Oraklji predstavljajo računsko plast, kjer posamezni akter izvede oceno zaupanja po svojih pravilih. Veriga blokov pa služi kot zanesljiva in nespremenljiva plast, kjer se hranijo rezultati teh ocen.

Sistem se razlikuje od ostalih rešitev, ki se osredotočajo zgolj na sledljivost ali upravljanje podatkov. Omogoča, da vsak akter izrecno definira svoje kriterije zaupanja in jih izvede nad skupnim modelom znanja. Rezultati teh izračunov se zapišejo v decentraliziran register, ki je enak za vse udeležence.

V nadaljevanju poglavja predstavimo zasnovno arhitekturo in posamezne plasti, prikažemo njihov medsebojni tok podatkov ter utemeljimo vlogo uporabljenih tehnologij. Tak pristop daje sistemu modularnost, zamenljivost izračunskih enot in razširljivost na druge aplikacijske domene.

### 3.2.1 Pregled večplastne zasnove

Arhitektura sistema temelji na treh jasno ločenih plasteh, ki skupaj tvorijo celoten potek od zajema podatkov do zapisa rezultatov zaupanja na verigo blokov. Takšna razdelitev omogoča boljše razumevanje posameznih vlog in enostavnejše nadgrajevanje sistema, saj lahko vsako plast obravnavamo neodvisno od drugih.

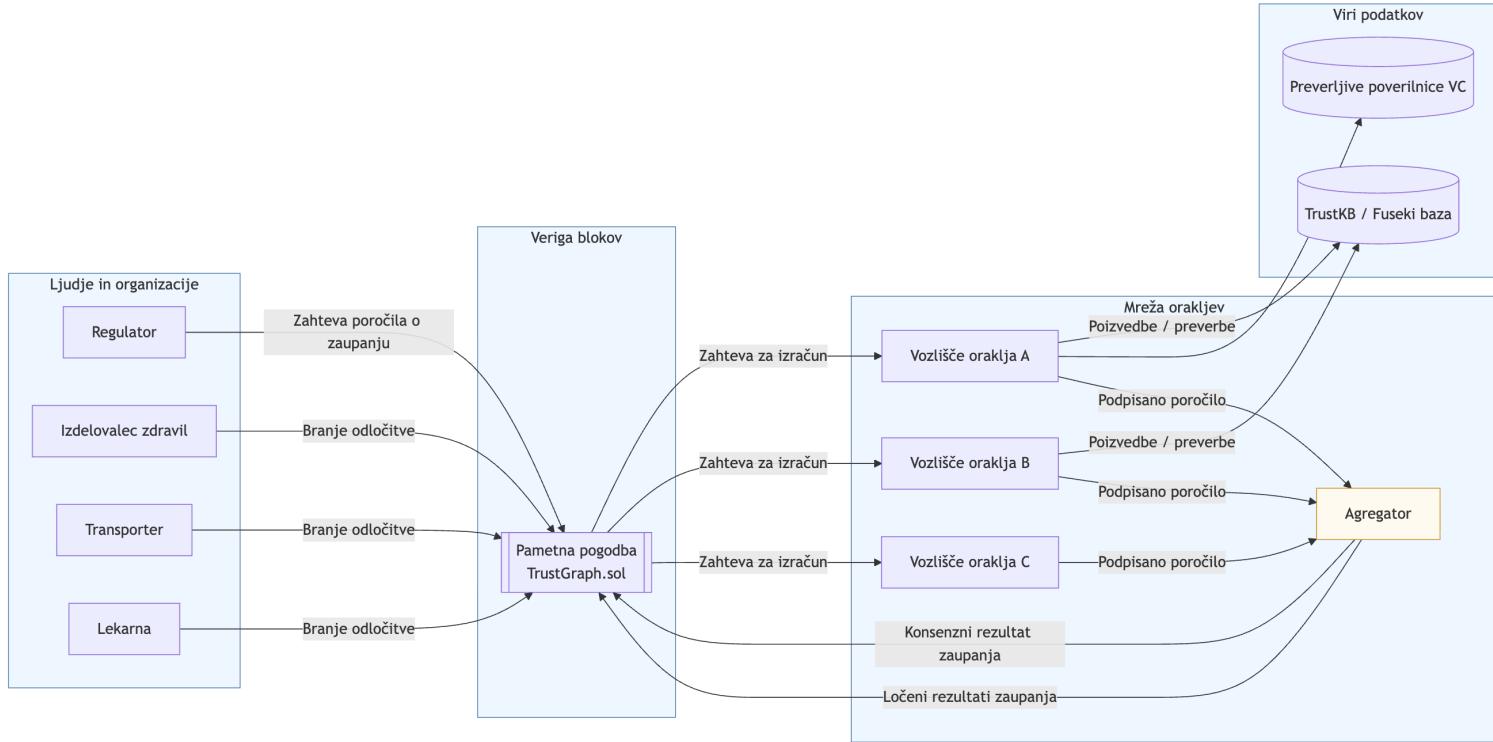
Na sliki 3.1 je prikazan kontekstni pogled celotnega sistema. Diagram združuje tri glavne sklope: akterje iz dobavne verige na levi, infrastrukturo orakljev in podatkovnih virov na desni ter verigo blokov v sredini. Akterji, kot so regulator, proizvajalec zdravil, prevoznik in lekarna, komunicirajo izključno s pametno pogodbo `TrustGraph.sol`. Ta beleži rezultate zaupanja in upravlja zahteve za ponovno oceno posameznih entitet.

Ko regulator ali drug akter sproži zahtevo za izračun zaupanja, pametna pogodba pošlje zahtevo omrežju orakljev. Vsak orakelj nato izvede svoj del naloge: iz ontologije in baze TrustKB pridobi podatke, po potrebi preveri pripadajoča preverljiva dokazila, nato pa izračuna rezultat zaupanja. Oraklji rezultate podpišejo in posredujejo agregatorju, ki pripravi končni konsenzni izračun in ga zapiše nazaj v pametno pogodbo.

V skrajnjem desnem delu diagrama je prikazana plast znanja. Ta vključuje ontologijo (TrustKB), kjer so zbrani koncepti, njihove lastnosti in meritve, ter skladišče dokazil VC, ki zajema licence, certifikate in druge verodostojne podatke o akterjih. Oraklji pri izračunu združujejo oba vira: semantične

lastnosti iz ontologije in verodostojne podatke iz dokazil.

Sistem tako deluje skozi jasno razmejene sklope, kar je razvidno iz celotnega toka na diagramu – od zahteve, preko izračuna, do zapisa končnega rezultata zaupanja na verigi blokov.



Slika 3.1: Kontekstni pogled sistema za upravljanje zaupanja.

V nadaljevanju so podrobneje opisane tri plasti, ki sestavljajo arhitekturo:

1. **Plast znanja (Knowledge Layer)** V tej plasti se nahaja ontologija dobavne verige in opisne lastnosti akterjev. Ontologija predstavlja skupni referenčni model, ki omogoča enotno interpretacijo podatkov o akterjih. Poleg ontologije se tukaj nahaja tudi skladišče preverljivih dokazil (VC), ki dodaja podatke o licencah, certifikatih in presojah kakovosti.
2. **Plast izračuna zaupanja (Trust Resolution Layer)** Oraklji v tej plasti izvajajo izračune zaupanja. Podatke pridobijo iz ontologije in preverljivih dokazil ter jih povežejo s pravili, ki jih določi posamezen akter. Pravila so zapisana v JSON obliku in opisujejo pogoje, ki jih mora ocenjovana entiteta izpolnjevati. Oraklji izvedejo izračun, rezultat podpišejo in ga posredujejo agregatorju.
3. **Plast registra zaupanja na verigi blokov (Blockchain Trust Registry Layer)** Ta plast vključuje pametno pogodbo, ki hrani rezultate ocenjevanja zaupanja. Pogodba deluje kot decentraliziran register, ki omogoča vpogled v trenutna in pretekla razmerja zaupanja med akterji. Vanjo se vpisujejo tako posamezni odgovori orakljev kot konsenzni rezultat evalvacije.

V naslednjih poglavjih so opisane tehnične podrobnosti posamezne plasti ter njihove medsebojne povezave.

### **Plast znanja**

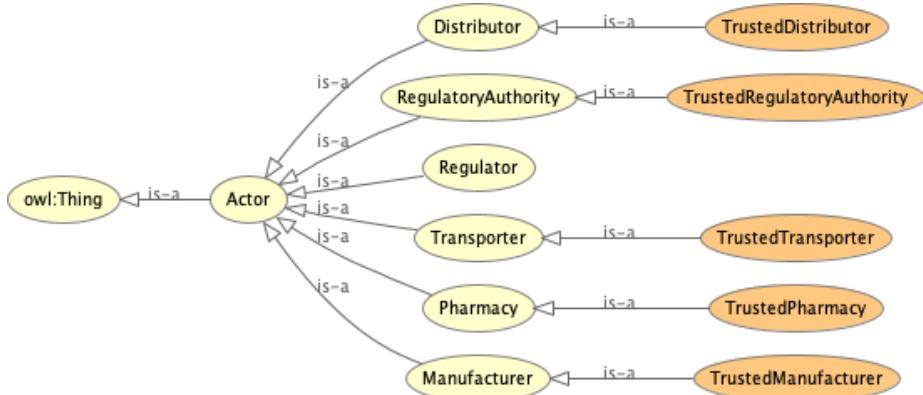
Plast znanja (angl. Knowledge Layer) definira semantično strukturo akterjev, njihove značilnosti in nabor pravil za individualno ocenjevanje zaupanja. V osnovi jo sestavlja dve ključni komponenti: ontologija in pravila zaupanja posameznih akterjev.

**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 [35],
- `hasAuditScore` - rezultat presoje kakovosti.

Ontologija je zapisana v formatu RDF [32] zasnovan na označevalnem jeziku XML [36], ki omogoča enostavno obdelavo s knjižnico `rdflib` [37] v programskej 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 ( $\text{hasDeliveryPunctuality} \geq 0.9$ ) in katerih delež temperaturnih odstopanj pri transportu zdravil ne presega 5 % ( $\text{hasTempViolationRate} \leq 0.05$ ). 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 le-

karne 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.

### 3.2.2 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 zbirkko 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 % temperaturnih nepravilnostih, bo mehanizem preveril te pogoje za vsakega prevoznika, zapisanega v

ontologiji. Rezultat ocenjevanja je binarna vrednost (`true/false`), ki pove, ali akter izpolnjuje kriterije zaupanja.

### 3.2.3 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.

### 3.3 Orodja in razvojno okolje

### 3.4 Ontologija in orodja za sklepanje

#### 3.4.1 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`, itn. Ti razredi so definirani z ekvivalentnimi 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.

#### 3.4.2 Primer toka podatkov in logike

Za ponazoritev delovanja si oglejmo primer, kjer proizvajalec Pfizer ocenjuje prevoznika DHL:

1. **Ontologija (OWL):** v ontologiji so zabeležene lastnosti prevoznika DHL, npr. `hasDeliveryPunctuality = 0.95` in `hasTempViolationRate = 0.02`.
2. **Pravila zaupanja:** Pfizerjeva politika zaupanja za razred `Transporter` zahteva `hasDeliveryPunctuality ≥ 0.90` in `hasTempViolationRate ≤ 0.05`.
3. **Sklepanje (Python):** mehanizem prebere lastnosti DHL iz ontologije in jih primerja s Pfizerjevimi pogoji. Ker sta oba pogoja izpolnjena, rezultat sklepanja je `true`.

4. **Zapis v verigo blokov:** rezultat (`Pfizer` zaupa `DHL`) se zapiše v pametno pogodbo. Funkcija `isTrusted(Pfizer, DHL)` nato vrne `true`.

Na ta način sistem poveže semantične podatke, subjektivna pravila in trajne zapise rezultatov v verigi blokov v enoten potek.

### 3.4.3 Način posodabljanja blockchain registra

Zapis rezultatov v pametno pogodbo se lahko sproži na tri načine:

- **Ročno:** prek CLI ali uporabniškega vmesnika, primerno za prezentacije in testne scenarije.
- **Dogodkovno:** ob spremembi entitete, pravil ali metrik v ontologiji (npr. nova licenca, posodobljen audit).
- **Periodično:** v rednih intervalih (npr. dnevno), pri čemer se zapis izvrši le ob spremembi rezultata glede na prejšnje stanje.

## 3.5 Razvoj pametnih pogodb in oraklov

## 3.6 Uporabniški vmesnik in API-ji

# **Poglavlje 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**



# Poglavlje 5

## Zaključek in nadaljne delo

### 5.1 Povzetek prispevkov

### 5.2 Nadaljne raziskave



## **Dodatek A**

### **Shema ontologije in SWRL pravila**



## Dodatek B

### Izvlečki kode pametnih pogodb



# Literatura

- [1] S. Holtmanns, Trust modeling and management: From social trust to digital trust, in: Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions, IGI Global, 2008. doi:10.4018/978-1-59904-804-8.ch013.
- [2] T. Grandison, M. Sloman, A survey of trust in internet applications, IEEE Communications Surveys & Tutorials 3 (4) (2000) 2–16.
- [3] S. D. Boon, J. G. Holmes, The dynamics of interpersonal trust: Resolving uncertainty in the face of risk, Cooperation and Prosocial Behavior (1991) 167–182.
- [4] D. Gambetta, Trust: Making and breaking cooperative relations, Basil Blackwell, 1988.
- [5] R. C. Mayer, J. H. Davis, F. D. Schoorman, An integrative model of organizational trust, Academy of Management Review 20 (3) (1995) 709–734.
- [6] D. E. Denning, A new paradigm for trusted systems, Proceedings of the 1993 New Security Paradigms Workshop (1993) 36–41.
- [7] D. H. McKnight, N. L. Chervany, Trust and distrust definitions: One bite at a time, Trust in Cyber-societies 2246 (2001) 27–54.
- [8] S. Karthik, R. Shankar, N. Arunkumar, Ontology-based trust model for pervasive computing environments, Journal of Ambient Intel-

- ligence and Humanized Computing 8 (2017) 557–568. doi:10.1007/s12652-016-0442-y.
- [9] M. Uddin, K. Salah, R. Jayaraman, S. Pesic, S. Ellahham, Blockchain for drug traceability: Architectures and open challenges, Health informatics journal 27 (2) (2021) 14604582211011228.
  - [10] H. Kayhan, Ensuring trust in pharmaceutical supply chains by data protection by design approach to blockchains, Blockchain in Healthcare Today 5 (2022) 10–30953.
  - [11] H. Li, M. Singhal, Trust management in distributed systems, Computer 40 (2) (2007) 45–53. doi:10.1109/MC.2007.76.
  - [12] R. Housley, W. Ford, W. Polk, D. Solo, Rfc2459: Internet x. 509 public key infrastructure certificate and crl profile (1999).
  - [13] M. Elkins, D. Del Torto, R. Levien, T. Roessler, Mime security with openpgp, Tech. rep. (2001).
  - [14] S. K. Panda, S. C. Satapathy, Drug traceability and transparency in medical supply chain using blockchain for easing the process and creating trust between stakeholders and consumers, Personal and Ubiquitous Computing 28 (1) (2024) 75–91.
  - [15] W. H. Organization, et al., A study on the public health and socioeconomic impact of substandard and falsified medical products: executive summary (2017).
  - [16] Council of European Union, Directive 2011/62/eu of the european parliament and of the council of 8 june 2011 amending directive 2001/83/ec on the community code relating to medicinal products for human use, as regards the prevention of the entry into the legal supply chain of falsified medicinal products text with eea relevance (2011).  
URL <https://eur-lex.europa.eu/eli/dir/2011/62/oj/eng>

- [17] Council of European Union, Commission delegated regulation (eu) 2016/161 of 2 october 2015 supplementing directive 2001/83/ec of the european parliament and of the council by laying down detailed rules for the safety features appearing on the packaging of medicinal products for human use (text with eea relevance) (2015).  
URL [https://eur-lex.europa.eu/eli/reg\\_del/2016/161/oj/eng](https://eur-lex.europa.eu/eli/reg_del/2016/161/oj/eng)
- [18] U.S. Food and Drug Administration, Drug supply chain security act (dscsa) (2017).  
URL <https://www.fda.gov/Drugs/DrugSafety/DrugIntegrityandSupplyChainSecurity/DrugSupplyChainSecurityAct/ucm382022.htm>
- [19] W. H. Organization, et al., WHO global surveillance and monitoring system for substandard and falsified medical products (2017).
- [20] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Available at SSRN 3440802 (2008).
- [21] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, et al., Blockchain technology: Beyond bitcoin, *Applied innovation* 2 (6-10) (2016) 71.
- [22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the thirteenth EuroSys conference, 2018, pp. 1–15.
- [23] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, *Ethereum project yellow paper* 151 (2014) (2014) 1–32.
- [24] G. Caldarelli, Understanding the blockchain oracle problem: A call for action, *Information* 11 (11) (2020) 509.

- [25] G. Caldarelli, Overview of blockchain oracle research, *Future Internet* 14 (6) (2022) 175.
- [26] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, A. Kastania, Astraea: A decentralized blockchain oracle, in: 2018 IEEE international conference on internet of things (IThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), IEEE, 2018, pp. 1145–1152.
- [27] P. Kochovski, S. Gec, V. Stankovski, M. Bajec, P. D. Drobintsev, Trust management in a blockchain based fog computing platform with trustless smart oracles, *Future Generation Computer Systems* 101 (2019) 747–759.
- [28] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, M. Sabadello, J. Holt, Decentralized identifiers (dids) v1. 0, Draft Community Group Report (2020).
- [29] M. Sporny, D. Longley, D. Chadwick, et al., Verifiable credentials data model 2.0, W3C Recommendation (2025).  
URL <https://www.w3.org/TR/vc-data-model-2.0/>
- [30] C. Mazzocca, A. Acar, S. Uluagac, R. Montanari, P. Bellavista, M. Conti, A survey on decentralized identifiers and verifiable credentials, *IEEE Communications Surveys & Tutorials* (2025).
- [31] O. Lassila, J. Hendler, T. Berners-Lee, The semantic web, *Scientific american* 284 (5) (2001) 34–43.
- [32] R. Cyganiak, P. Reynolds, Rdf 1.1 concepts and abstract syntax, W3C Recommendation, (pridobljeno 10. 11. 2025) (2014).  
URL <https://www.w3.org/TR/rdf11-concepts/>
- [33] W. W. W. Consortium, et al., Owl 2 web ontology language document overview (2012).

- [34] Web3 python library, <https://web3py.readthedocs.io/en/stable/>.
- [35] J. D. Nally, Good manufacturing practices for pharmaceuticals, CRC Press, 2016.
- [36] 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/>
- [37] 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>