UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
NOVOM SADU

Blokčejn baziran model za praćenje usklađenosti softvera u industrijskim upravljačkim sistemima sa zahtevima za bezbedan razvoj softvera

DOKTORSKA DISERTACIJA

Mentori:
prof. dr Aleksandar Erdeljan
prof. dr Goran Sladić

Kandidat:
Jelena Marjanović

Novi Sad, 2023. godina

УНИВЕРЗИТЕТ У НОВОМ САДУ **ОБРАЗАЦ – 5а**

НАВЕСТИ НАЗИВ ФАКУЛТЕТА ИЛИ ЦЕНТРА

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА[1]

Врста рада:	Докторска дисертација
Име и презиме аутора:	Јелена Марјановић (рођена Станковски)
Ментор (титула, име, презиме, звање, институција)	проф. др Александар Ердељан, редовни професор, Факултет техничких наука проф. др Горан Сладић, редовни професор, Факултет техничких наука
Наслов рада:	Блокчејн базиран модел за праћење усклађености софтвера у индустријским управљачким системима са захтевима за безбедан развој софтвера
Језик публикације (писмо):	Српски латиница
Физички опис рада:	Унети број: Страница 109 Поглавља 6 Референци 167 Табела 10 Слика 23 Графикона / Прилога /
Научна област:	Електротехничко и рачунарско инжењерство
Ужа научна област (научна дисциплина):	Примењено софтверско инжењерство
Кључне речи / предметна одредница:	Блокчејн, индустријски управљачки системи, захтеви, безбедан развој софтвера
Резиме на језику рада:	Дисертација се бави истраживањем је примена *Hyperledger Fabric* блокчејн решења за праћење усклађености софтвера са безбедносним захтевима у индустријским управљачким системима. Дефинисан је модел који обухвата учеснике, случајеве коришћења и принцип безбедности података. Валидација модела спроведена је кроз анализу безбедносне праксе Управљање безбедношћу, део стандарда IEC 62443-4-1, који обухвата 13 захтева. Модел омогућава транспарентност, непорељивост, следљивост и доступност информација, битне особине за индустријске управљачке системе у критичним инфраструктурама. Поверљивост

[1] Аутор докторске дисертације потписао је и приложио следеће Обрасце:
5б – Изјава о ауторству;
5в – Изјава о истоветности штампане и електронске верзије и о личним подацима;
5г – Изјава о коришћењу.
Ове Изјаве се чувају на факултету у штампаном и електронском облику и не кориче се са тезом.

	информација обезбеђена је употребом приватне блокчејн мреже попут *Hyperledger Fabric*. Даље, дефинисани су дијаграми случајева коришћења и организације неопходни за функционалност система. Коришћен је IPFS за складиштење докумената, а затим је постављено решење за *Hyperledger Fabric* блокчејну мрежу. Овај приступ пружа увид у усклађеност софтвера, посебно у критичним секторима, обезбеђујући сигурност података и ефикасну имплементацију решења.
Датум прихватања теме од стране надлежног већа:	27.4.2023.
Датум одбране: (Попуњава одговарајућа служба)	
Чланови комисије: (титула, име, презиме, звање, институција)	Председник: др Дарко Чапко, редовни професор, Факултет техничких наука, Нови Сад Члан: др Иван Вулић, доцент, Војна акамедија, Београд Члан: др Имре Лендак, ванредни професор, Факултет техничких наука, Нови Сад Члан: др Милан Гаврић, доцент, Факултет техничких наука, Нови Сад Ментори: др Александар Ердељан, редовни професор, Факултет техничких наука, Нови Сад др Горан Сладић, редовни професор, Факултет техничких наука, Нови Сад
Напомена:	

UNIVERSITY OF NOVI SAD

FACULTY OR CENTER

KEY WORD DOCUMENTATION[2]

Document type:	Doctoral dissertation
Author:	Jelena Marjanović (née Stankovski)
Supervisor (title, first name, last name, position, institution)	Ph.D. Aleksandar Erdeljan, Full Professor, Faculty of Technical Sciences, Novi Sad Ph.D. Goran Sladić, Full Professor, Faculty of Technical Sciences, Novi Sad
Thesis title:	Blockchain-based model for tracking software requirement compliance in industrial control systems with secure software development lifecycle
Language of text (script):	Serbian language latin script
Physical description:	Number of: Pages 109 Chapters 6 References 167 Tables 10 Illustrations 23 Graphs / Appendices /
Scientific field:	Electrical and computer engineering
Scientific subfield (scientific discipline):	Power Software Engineering
Subject, Key words:	Blockchain, industrial control systems, requirements, secure development lifecycle
Abstract in English language:	This thesis investigates the application of the Hyperledger Fabric blockchain solution for monitoring software compliance with security requirements in industrial control systems. A model is defined that includes participants, use cases and the principle of data security. Validation of the model was carried out through the analysis of the safety practice Security management, part of the standard IEC 62443-4-1, which includes 13 requirements. The model enables transparency, non-repudiation, traceability and availability of information, essential features for industrial management systems in critical infrastructures. Information confidentiality is ensured by using a private blockchain network like Hyperledger Fabric. Furthermore, use case diagrams and organization necessary for system functionality are defined. IPFS was used to store

[2] The author of doctoral dissertation has signed the following Statements:
5б – Statement on the authority,
5в – Statement that the printed and e-version of doctoral dissertation are identical and about personal data,
5г – Statement on copyright licenses.
The paper and e-versions of Statements are held at he faculty and are not included into the printed thesis.

	documents, and then the solution was deployed on the Hyperledger Fabric blockchain network. This comprehensive approach provides insight into software compliance, particularly in critical sectors, ensuring data security and effective solution implementation.
Accepted on Scientific Board on:	27.4.2023.
Defended: (Filled by the faculty service)	
Thesis Defend Board: (title, first name, last name, position, institution)	President: Ph.D. Darko Čapko, Full Professor, Faculty of Technical Sciences, Novi Sad Member: Ph.D. Ivan Vulić, Assistant Professor, Military academy, Belgrade Member: Ph.D. Imre Lendak Associate Professor, Faculty of Technical Sciences, Novi Sad Member: Ph.D. Milan Gavrić, Assistant Professor, Faculty of Technical Sciences, Novi Sad Supervisors: Ph.D. Aleksandar Erdeljan, Full Professor, Faculty of Technical Sciences, Novi Sad Ph.D. Goran Sladić, Full Professor, Faculty of Technical Sciences, Novi Sad
Note:	

Spisak slika

Spisak tabela

Spisak listinga

Spisak skraćenica

Skraćenica	Značenje
APT	Advanced Persistant Threat
ASIC	Application-Specific Integrated Circuits
BCNA	Blockchain Administrator
BFT	Byzantine Fault Tolerance
CA	Certificate Authority
CIA	Confidentiality, Integrity, Availability
CLASP	Comprehensive, Lightweight Application Security Process)
CMMI-DEV	Capability Maturity Model Integration for Developemt
COBIT	Control Objectives for Information and Related Technologies
CPU	Central Processing Unit
CSF,	Cybersecurity Framework
CVSS	Common Vulnerability Scoring System
DAST	Dynamic Analysis Securiry Testing
DAO	Decentralized Autonomous Organization
DCS	Distributed Control Systems
EPM	Empresas Públicas de Medellín
ETH	Ethereum
EU	Evropska Unija
EVM	Ethereum Virtual Machine
FBFT	Federated Byzantine Fault Tolerance
FBI	Federal Birou of Investigation
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Modernization Act
FPGA	Field-Programmable Gate Array
GAISP	Generally Accepted Information Security Principles
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
ICS	Industrial Control System
IED	Intelligent electronic device
ISA	International Society of Automation
ISMS	Information security management systems
ISO	International Organization for Standardization
IT	Information Technology
MAC	Message Authentication Codes
NIST	National Institute of Standards and Technology

NSA	National Security Agency
OT	Operational Technology
PAC	Programmable Automation Controllers
PBFT	Practical Byzantine Fault Tolerance
PCI-DSS	Payment Card Industry Data Security Standard
PLC	Programmable Logic Controllers
RACI	Responsible, Accountable, Consulted, Informed
RTU	Remote Terminal Unit
SAD	Sjedinjene Američke Države
SAST	Static Analysis Security Testing
SC	Smart contract
SCADA	Supervisory Control and Data Acquisition
SHA	Secure Hash Algorithm
SM	Security Management
WSL2	Windows Subsystem for Linux

Sažetak

Kao krajnji potrošači električne energije, gotovo nikada ne razmišljamo o široj slici električne energije, kako nastaje, kako dolazi do nas, koji sistemi nam omogućavaju neometanu upotrebu električne energije, da li su podaci obezbeđeni ili su dopali u ruke zlonamernih korisnika, što za posledicu može imati prekid u isporuci električne energije. Dokle god su naši uređaji upotrebljivi i život se odvija neometano, mogućnost nedostupnosti električne energije deluje nezamislivo. Krajnjim potrošačima električne energije, mogućnost zlonamernog napada se ogleda u prisustvu, odnosno odsustvu dostupnosti električne energije, dok sa druge strane, inženjerima koji rade u domenu kritičnih infrastruktura, razlozi zbog kojih je došlo do prekida isporuke električnom energijom su daleko mnogobrojniji. Takođe nezamislivo može zvučati nedostupnost interneta ili uređaj koji nije na neki način povezan sa internetom, budući da se pored pametnih uređaja, na internet povezuju i električne mreže i industrijski sistemi [1]. Broj zlonamernih, sajber napada na kritične infrastrukture, kao što je energetski sektor, je u porastu, budući da takvi sistemi zbog svoje složenosti predstavljaju izazove i za najmoćnije i najveštije hakere sveta [2]. Kako se više ne postavlja pitanje da li će doći do sajber napada, već se postavlja pitanje kada će doći do takvog napada, na inženjerima kritičnih infrastruktura i svima ostalima uključenih u taj sektor je od izuzetnog značaja implementacija bezbednosnih kontrola, koje će minimizirati šanse za sajber napad, kao i smanjiti uticaj eventualnog napada. Brojni standardi, regulative, okviri i zakoni su definisani sa ciljem podizanja bezbednosti industrijskih upravljačkih sistema, koji su deo kritičnih infrastruktura. Svaki od tih standarda, regulativa, okvira i zakona donosi određeni broj zahteva sa kojima je neophodno biti usklađen. Broj zahteva sa kojima je neophodno biti usklađen se može meriti i u stotinama, što predstavlja izazov za sve uključene u proces održavanja usklađenosti sistema sa definisanim zahtevima. Korišćenje dobrih praksi u okviru procesa inženjerstva zahteva omogućava adekvatno praćenje, analizu i procenu neophodnih resursa za ispunjenje zahteva. Zahtevi vezani za bezbednost industrijskih upravljačkih sistema i usklađenost sistema sa zahtevima predstavljaju poverljive informacije, budući da mogu otkriti nedostatke koje zlonamerni korisnici mogu iskoristiti kako bi onemogućili neometani rad sistema u okviru kritičnih infrastruktura. Ukoliko je potrebno osigurati da pored dokaza o usklađenosti softvera sa zahtevima postoji sistem koji pruža otpornost, omogućava transparentnost, neporecivost i mogućnost praćenja, tada se praćenje usklađenosti sa zahtevima može sprovesti putem blokčejn mreže. Blokčejn tehnologija je prvenstveno bila korišćena u finansijskom sektoru, međutim, zbog svih osobina, kao što su neporecivost, otpornost i sledljivost, broj sektora koji pronalaze benefite korišćenja te tehnologije, je u porastu. S obzirom da se podaci kojima rukuju industrijski upravljački sistemi u kritičnim infrastrukturama smatraju poverljivim, usklađenost takvih sistema sa zahtevima je neophodno štititi na odgovarajući način, što privatne blokčejn mreže

omogućavaju, budući da su informacije dostupne isključivo korisnicima koji za to imaju potrebu.

U okviru ove disertacije, predložen je blokčejn baziran model za praćenje usklađenosti zahteva za razvoj softvera koji se nalaze u industrijskim upravljačkim sistemima, sa posebnim akcentom na zahteve koji su definisani kao neophodni za bezbedan razvoj softvera. Takođe, definisani su i učesnici u procesu praćenja zahteva i njihovi slučajevi korišćenja. Definisani model je validiran na realnom primeru 13 zahteva iz IEC 62443-4-1 standarda, koristeći principe rada blokčejn tehnologija, konkretno Hyperledger Fabric rešenja čime je pristup informacijama omogućen samo prethodno registrovanim i autorizovanim korisnicima.

Abstract

As end consumers of electricity, we almost never think about the wider picture of electricity, how it is created, how it reaches us, which systems allow us to use electricity without interruption, whether the data is secured or has fallen into the hands of malicious users, which can result in have an interruption in the supply of electricity. As long as our devices are usable and life goes on smoothly, the possibility of electricity being unavailable seems unimaginable. For the end consumers of electricity, the possibility of a malicious attack is reflected in the presence or absence of the availability of electricity, while on the other hand, for engineers who work in the field of critical infrastructures, the reasons for the interruption of electricity supply are far more numerous. The unavailability of the Internet or a device that is not somehow connected to the Internet may also sound unimaginable, since in addition to smart devices, power grids and industrial systems are also connected to the Internet [1]. The number of malicious, cyber attacks on critical infrastructures, such as the energy sector, is on the rise, as such systems, due to their complexity, pose challenges to even the world's most powerful and skilled hackers [2]. As the question is no longer whether there will be a cyber-attack, rather the question is when such an attack will occur, it is extremely important for critical infrastructure engineers and everyone else involved in that sector to implement security controls that will minimize the chances of a cyber. attack, as well as reduce the impact of a possible attack. Numerous standards, regulations, frameworks and laws have been defined with the aim of increasing the security of industrial control systems, which are part of critical infrastructures. Each of those standards, regulations, frameworks and laws brings a certain number of requirements with which it is necessary to comply. The number of requirements with which it is necessary to be compliant can be measured in the hundreds, which represents a challenge for everyone involved in the process of maintaining compliance of the system with the defined requirements. The use of good practices within the requirements engineering process enables adequate monitoring, analysis and assessment of the necessary resources for the fulfillment of requirements. The requirements related to the security of industrial control systems and the system's compliance with the requirements represent confidential information, since they can reveal flaws that malicious users can exploit to prevent the smooth operation of systems within critical infrastructures. If it is necessary to ensure that in addition to proof of compliance of the software with the requirements, there is a system that provides resilience, enables transparency, non-repudiation and traceability, then the monitoring of compliance with the requirements can be carried out through the blockchain network. Blockchain technology was primarily used in the financial sector, however, due to all its features, such as non-repudiation, resilience and traceability, the number of sectors that find the benefits of using this technology is increasing. Given that the data handled by industrial control systems in critical infrastructures are considered confidential, the compliance of such systems with the requirements must be

protected in an appropriate way, which private blockchain networks make possible, since the information is available only to users who need it.

Within this dissertation, a blockchain-based model for compliance monitoring of software development requirements found in industrial control systems is proposed, with a special emphasis on requirements defined as necessary for secure software development. Also, the participants in the request monitoring process and their use cases are defined. The defined model was validated on a real example of 13 requirements from the IEC 62443-4-1 standard, using the working principles of blockchain technologies, specifically the Hyperledger Fabric solution, which allows access to information only to previously registered and authorized users.

1.Uvod

Svedoci smo neverovatne digitalizacije i napretka računarstva i informacionih tehnologija. Ukoliko se nalazimo u urbanoj sredini, mala je šansa da nam u našoj neposrednoj blizini ne stoji neki pametni uređaj koji nam olakšava život, što nije bila situaciju pre nekoliko decenija. To može biti pametni sat na ruci koji nas podseća da smanjimo stres, električni automobili koji pomažu smanjenju emisije štetnih gasova, telefon koji nam predstavlja svet u malom ili neki uređaj u okviru pametne kuće koji nam omogućava da daljinski podesimo temperaturu kuće pre nego što se vratimo. Svi ti uređaji na koje se toliko oslanjamo na kraju dana imaju jednu zajedničku vezu: električna energija. Telefon punimo gotovo svaki dan, električni automobil punimo na svakih 400 pređenih kilometara, telefon nekada punimo i svako veče, dok uređaji u okviru pametne kuće zahtevaju struju kako bi radili, kao i internet konekciju, gde nam je ruter ponovo priključen u struju. Dostupnost i pouzdanost električne energije se ne dovodi u pitanje sve do momenta nestanka struje, gde je u tom trenutku elektro distribucija prvi broj koji se poziva. Dok se u Evropi nisu javljali duži nestanci struje koji su ostavljali veliki broj ljudi bez struje, u Sjedinjenim Američkim Državama (SAD) se desio jedan od najgorih scenarija. Tokom parališuće oluje na zimu 2021. godine, više od 4,5 miliona domova u američkoj saveznoj državi Teksas je ostalo bez struje. Prekid u isporuci električne energije u nekim delovima trajao i po nekoliko dana, što je na kraju rezultiralo sa 57 izgubljenih života i gubitkom od 195 milijardi dolara [3]. Svi ovi elementi pokazuju zašto je energetski sektor definisan kao kritična infrastruktura, zajedno sa 15 drugih sektora. Kada se pogleda pojašnjenje CISA (eng. *Cybersecurity and Infrastructure Security Agency*), kritične infrastrukture predstavljaju sektore čiji elementi, sistemi i mreže, bilo virtuelne ili fizičke, koji su od tolike važnosti za SAD da se njihovo onesposobljavanje ili uništavanje oslikava na bezbednost, ekonomsku stabilnost, javno zdravlje i bezbednost ili neku od kombinacija [4]. Identifikacija energetskog sektora kao kritične infrastrukture omogućava bolje razumevanje i prioritetno reagovanje u slučaju nepogoda, čime se odražava i potreba za sveobuhvatnom zaštitom ključnih sistema. Kombinacija povezanosti, složenosti i uticaja energetskog sektora u društvu, predstavlja osnovni stub modernog života. Prepoznavanje i adekvatna odbrana energetskog sektora, kao i drugih kritičnih infrastruktura, predstavlja korak ka očuvanju sigurnosti i stabilnosti infrastrukture. Nestanak struje u Teksasu je to pokazao, jer je onesposobilo saveznu državu da funkcioniše normalno, što je izazvalo ogromne ekonomske gubitke, kao i ljudske gubitke.

Prirodne nepogode nažalost nisu jedina pretnja stabilnosti energetskog sektora. Stabilnost energetskog sektora je i pod stalnom pretnjom zbog sve većih ekoloških izazova, kao što su sve češći ekstremni vremenski uslovi. Takođe, prelaskom na održive izvore energije, energetska infrastruktura je pod pritiskom da se omoguće i ti novi zahtevi, što predstavlja izazov koji nije postojao pre samo par decenija. Međutim, pored navedenih pretnji, postoje i veštački izazovi koji mogu naneti ozbiljnu štetu energetskoj infrastrukturi i kreirati scenario sličan onome u Teksasu. Industrijski upravljački sistemi, koji se koriste za kontrolu, nadzor i

upravljanje kritičnih infrastruktura, predstavljaju izazov za hakere, spremne za izvršavanje sajber napada, budući da napad na sistem može da ostavi drastične posledice. Napadi na industrijske upravljačke sistem nisu samo teorijska mogućnost za koju se treba pripremati, budući da su se u prošlosti već dešavali. Verovatno najpoznatiji napad koji se desio na industrijski upravljački sistem poznat je pod nazivom Stuxnet, koji je otkriven 2010. godine i odgovoran je za štetu nastalu u nuklearnom postrojenju u Iranu [5]. Važnost zaštite industrijskih upravljačkih sistema je izuzetno velika i taj značaj se jasno uočava u budžetima i regulativama SAD i Evropske Unije (EU), koje su posvećene unapređenju spremnosti industrijskih upravljačkih sistema. Njihov cilj je smanjenje rizika od eventualnih sajber napada i stvaranje sigurnijeg okruženja za kritične infrastrukture. SAD, kao i Evropska Unija, prepoznaju da je zaštita industrijskih upravljačkih sistema od izuzetne važnosti za očuvanje sigurnosti kritičnih infrastruktura. Zbog toga su izdvojili značajne resurse u svojim budžetima kako bi poboljšali kritične infrastrukture, sa ciljem smanjenja rizika od potencijalnih sajber napada. Osim finansijskih sredstava, SAD i EU su uspostavile stroge regulative koje se odnose na zaštitu kritičnih infrastruktura. Kroz ove regulative, i SAD i EU pokušavaju da povećaju nivo pripravnosti sistema i minimizuju rizik od štetnih posledica koje mogu proisteći iz sajber napada. Uspostavljanje takvih regulativa predstavlja jasnu indikaciju da su SAD i EU svesne rizika koji proističu iz potencijalnih sajber napada na industrijske upravljačke sisteme. Njihov fokus je na stvaranju otpornih sistema koji mogu izdržati napade i zaštititi kritične infrastrukture od eventualnih negativnih posledica. Stoga je važno je naglasiti da je bezbednost industrijskih upravljačkih sistema postala prioritet za SAD i EU. Njihovi budžeti i regulative jasno ukazuju na njihovu opredeljenost da se uhvate u koštac s izazovima sajber sigurnosti i da ojačaju spremnost industrijskih sektora na potencijalne sajber napade. Ova posvećenost ima za cilj održavanje stabilnosti i pouzdanosti kritičnih infrastruktura, čime se osigurava bezbedno i pouzdano funkcionisanje industrijskih sistema u digitalnom dobu.

U nastavku poglavlja biće detaljnije prikazana motivacija za ovu doktorsku disertaciju, sama definicija problema, hipoteze kao i struktura doktorske disertacije.

1.1. Motivacija i definisanje problema

Učesnici u procesu razvoja softverskih proizvoda, počevši od domen eksperta, preko softverskih inženjera do inženjera kontrole kvaliteta softvera, postaju svesniji posledica koje potencijalna greška u razvoju može da proizvede. Takve greške mogu uvesti bezbednosne ranjivosti, čijim se eksploatisanjem može direktno uticati na gubitak poverljivosti i integriteta podataka ili dostupnosti sistema. Ranjivosti, odnosno slabosti sistema koje mogu biti eksploatisane, imaju veći uticaj ako se pojave u industrijskim upravljačkim sistemima (eng. *Industrial Control Systems*, ICS), jer neposredno utiču na ponašanje procesa u realnom vremenu [6], gde osnovni uzrok tih ranjivosti mora biti obrađen [7]. Jedan primer industrijskog upravljačkog sistema koji se koristi u kritičnoj infrastrukturi je nadzorna kontrola i akvizicija podataka (eng. *Supervisory Control and Data Acquisition,* SCADA). Nadzorno upravljački

sistem u kritičnim infrastrukturama predstavlja složen distribuirani softverski sistem sa geografski dislociranim komponentama i uređajima, koji uključuje veliki broj uređaja i korisnika, čime je povećana izloženost sistema potencijalnim napadima. Dostupnost kritične infrastrukture, kojom upravljaju industrijski upravljački sistemi, je od izuzetne važnosti za funkcionisanje ljudi, stoga je potreban visok nivo bezbednosti kako bi se sprečile ranjivosti takvih sistema. Kao primer, može se analizirati greška u softveru za upravljanje energijom, koja je dovela do nestanka struje na severoistoku SAD 2004. godine, što je moglo biti sprečeno da je revizija koda obavljena u fazi implementacije [8]. Uticaj sajber napada koji su se desili na industrijske upravljačke sisteme, počevši od prvog javno poznatog SCADA sajber napada, napad na „Transsibirski cevovod" [9], do najnovijeg napada malverom na „Kolonijalni cevovod" [10] je sve veći, što je razmatrano u nekoliko istraživanja [11], [12], [13]. Sajber napadi u globalnom lancu snabdevanja naftom su ranije prepoznati kao potencijalni problem, pa su autori u istraživanju [14] analizirali sajber pretnje i predložili hitne protivmere. Prilikom projektovanja takvog sistema mogu se koristiti modeli i algoritmi za optimizaciju izdržljivosti kritičnog sistema održavanjem redundantnosti komponenti, koji bi se aktivirao kada je sistem pod napadima [15]. Ranjivosti u industrijskim upravljačkim sistemima su pokazale da se položaj sajber-bezbednosti mora poboljšati i da se treba pozabaviti osnovnim uzrokom ranjivosti industrijskih upravljačkih sistema [7].

Pronalaženje osnovnog uzroka problema koji su se desili je način da se minimiziraju buduće greške, dok je sprovođenje bezbednosnih provera od ranog razvoja, takođe način da se greške otkriju ranije u razvoju softvera [16]. Takav pristup se može dopuniti primenom relevantnih industrijskih praksi za bezbedan razvoj proizvoda. Standard IEC 62443-4-1 [17], pod nazivom Zahtevi životnog ciklusa bezbednog razvoja proizvoda (eng. *Secure product development lifecycle requirements*), pomaže industrijskim sistemima automatizacije i kontrole da poprave svoje stanje i zrelost bezbednosnih kontrola, primenom najboljih bezbednosnih praksi u svakom aspektu životnog ciklusa razvoja proizvoda. Standard IEC 62443-4-1 je podeljen u osam celina (tzv. praksi), koje se bave definisanjem bezbednosnih zahteva, sigurnim dizajnom, bezbednom implementacijom, verifikacijom i validacijom, upravljanjem defektima, upravljanjem ispravkama i prestankom životnog veka proizvoda [17]. Kako je standard IEC 62443-4-1 napisan u obliku 47 zahteva, proces inženjerstva zahteva je od velikog značaja usled jasno definisanih faza kroz koje je neophodno da svaki zahtev prođe. Inženjerstvo zahteva obuhvata proces koji prati životni ciklus zahteva [18]. Proces inženjerstva zahteva se koristi u različitim oblastima, jer omogućava prolazak zahteva kroz nekoliko faza koje mogu da se prate, što je analizirano u radovima [18], [19], [20], [21]. Stručne smernice u okviru procesa inženjerstva zahteva su analizirane i poboljšane u okviru [18], [19], [20], [21], ali su takođe analizirane i za potrebe startap (eng. *start-up*) kompanija [22] i prilagođene su za sajber fizičke sisteme [23], [24].

Kako inženjerstvo zahteva obuhvata prikupljanje, dokumentovanje, validaciju i verifikaciju, i na kraju planiranje zahteva [18], potrebno je razmotriti dostupne tehnologije koje

omogućavaju praćenje životnog ciklusa zahteva. Jedan od načina da se dokumenti održavaju verzionisanim, dok su u isto vreme otporni na neovlašćeno menjanje koje garantuje da informacije koje su napisane nisu izmenjene, jeste da se koriste funkcije koje blokčejn tehnologija pruža. Dok neka rešenja zahtevaju da informacije uskladištene na blokčejnu budu javno dostupne i zahteva se da su informacije javno verifikovane od strane učesnika blokčejn mreže, druga rešenja ostavljaju informacije dostupne samo prethodno autorizovanim stranama. Pregled blokčejn klasifikacije uradili su Golosova i ostali [25], gde je data razlika između javnog, privatnog, dozvoljenog i blokčejn bez dozvole. Privatne blokčejn mreže su pogodne za sisteme koji treba da koriste blokčejn tehnologiju, pri čemu nemaju svi korisnici pravo unosa podataka na blokčejn, kao ni pravo pregleda podataka na blokčejnu. Hyperledger Fabric je distribuirana knjiga, koja se koristi za kreiranje blokčejn rešenja koja zahtevaju privatnu blokčejn mrežu sa odobrenjem, tj. mrežu koju kreira i održava prethodno ovlašćeni skup članova. Hyperledger Fabric omogućava kreiranje pametnih ugovora, transakcija, konzorcijuma kao što imaju i druge implementacije blokčejna, ali je takođe uveo pojmove kao što su organizacija, usluga naručivanja i kanal [26].

Potreba za bezbednosnim praksama životnog ciklusa razvoja softvera je identifikovana od strane akademske zajednice i industrije. Pitanje primenljivosti praksi za bezbedan razvoj softvera i opravdanosti upotrebe ljudskih resursa, zajedno sa dodatnim troškovima za koje se veruje da bezbednosne prakse pridodaju prilikom razvoja razmatrano je u [27], gde je predloženo nekoliko modela za procenu troškova koji uzimaju u obzir bezbednost i zaključeno je da postojeći modeli nisu bili propisno validirani. Za argument, da bezbednosne prakse uvode preterane troškove u pogledu vremena i novca, autori [28] su pokazali da se Microsoft-ov životni ciklus bezbednog razvoja može koristiti čak i u malom timu, ali tvrde da je potrebno sprovesti dodatne odgovarajuće analize troškova. Druga grupa autora [29] pretpostavlja da će implementacija bezbednosti uvesti dodatne troškove u smislu vremena i dodatnih ljudskih resursa. Postoje određeni izazovi kada su bezbednosne prakse izostavljene iz razvoja softvera i treba ih uvesti naknadno, posebno u agilnom razvoju koji se oslanja na Skram (eng. *Scrum*) metodologiju. Takav razvoj se zasniva na brzom razvoju funkcija i obično traje manje od 30 dana. Autori u istraživanju [29] tvrde da tako kratak period ne ostavlja vremena da se bezbednosne prakse implementiraju u punom obimu i predlažu bezbednosni Skram proces, koji omogućava uvođenje „agilnih“ bezbednosnih aktivnosti u proces. Izmenjeni proces je ocenio tim programera, opisujući takav proces kao „srednje“ agilan i „srednje“ isplativ. Kao što su autori naveli, takav proces je uveo dodatne troškove u smislu vremena, ali analiza ne primenjivanja bezbednih praksi, koja bi mogla da dovede do bezbednosnih problema kao što su proboj i distribuirani napad uskraćivanja usluge (eng. *Distributed denial of service*, DDoS), nije razmatrana. Takođe, stručnjaci za bezbednost u kompanijama imaju različite uloge, od bezbednosnih inženjera, konsultanata ili revizora. Rad koji su obavili revizori bezbednosti predstavili su autori u istraživanju [30] u formi intervjua, dajući uvid u bezbednosne prakse, kao što su statička analiza programskog koda i testovi penetracije. Kako su autori zaključili, za

poboljšanje bezbednosti aplikacija potrebna je kombinacija organizacionih procesa, obuke programera i adekvatnih alata.

Osnovna motivacija za istraživanje u ovom radu jeste da se omogući industrijskim upravljačkim sistemima, koji žele da poboljšaju svoje bezbednosno stanje i zrelost, da imaju model zasnovan na privatnom blokčejnu, tako da mogu da prate i upravljaju zahtevima za bezbedan životni ciklus razvoja softvera. Takvo praćenje bi omogućilo industrijskim upravljačkim sistemima da promovišu saradnju i poverenje među različitim stranama, dok su u isto vreme informacije sačuvane na blokčejnu dostupne samo prethodno autorizovanim korisnicima. Podaci industrijskih upravljačkih sistema se mogu klasifikovati kao osetljivi, što zahteva bezbednosne kontrole kako bi informacije bile dostupne isključivo ljudima i sistemima koji za to imaju odobrenje. Uzimajući u obzir osetljivost podataka, jedna od bezbednosnih kontrola jeste da se podaci vezani za industrijske upravljačke sisteme postavljaju na privatne blokčejn mreže, budući da privatne blokčejn mreže ne omogućavaju javni uvid u podatke koji su skladišteni na mreži, ali ujedno nudi vid transparentnosti skupa akcija i podataka samo ograničenoj grupi entiteta.

1.2. Hipoteze i ciljevi

Na osnovu prethodnog odeljka, mogu se definisati hipoteze na kojima se temelji istraživanje:

- **H1**: Moguće je definisati model za praćenje usklađenosti sa zahtevima za bezbedan razvoj softvera, a koji se odnose na industrijske upravljačke sisteme.
- **H2**: Moguće je definisati učesnike u procesu praćenja zahteva, njihove slučajeve korišćenja, uzimajući u obzir osetljivost podataka sa kojima učesnici rukuju i prateći princip da su podaci obezbeđeni samo ovlašćenim pojedincima, neophodnim za obavljanje svojih dužnosti.
- **H3**: Moguće je validirati model koristeći principe rada blokčejn tehnologija, konkretno Hyperledger Fabric rešenja koje pruža pristup informacijama prethodno registrovanim i autorizovanim korisnicima.

Iz prethodno definisanih hipoteza izvode se primarni ciljevi ove teze pri čemu očekivani rezultati uključuju sledeće:

- Definisanje modela, učesnika i slučajeva korišćenja za praćenje usklađenosti sa zahtevima za bezbedan razvoj softvera u industrijskim upravljačkim sistemima.
- Validacija modela korišćenjem dostupnih funkcija Hyperledger Fabric implementacije blokčejn tehnologije.

1.3. Struktura doktorske disertacije

U okviru uvodnog Poglavlja 1, prikazani su motivi za istraživanje i pisanje ove doktorske disertacije. Takođe, opisan je i problem koji je uočen i želi da se reši, kao i nedostaci postojećih rešenja. Na kraju, definisane su hipoteze i ciljevi ove doktorske disertacije, koje će biti referencirane u kasnijim delovima disertacije.

U okviru Poglavlja 2, diskutovano je aktuelno stanje u oblasti, pregled trenutne literature u oblasti industrijskih upravljačkih sistema, kritične infrastrukture, bezbednosti industrijskih upravljačkih sistema, primena blokčejn tehnologije u industrijskim upravljačkim sistemima kao i inženjerstvo zahteva i upotreba blokčejn tehnologije u inženjerstvu zahteva.

U okviru Poglavlja 3, prikazane su blokčejn implementacije, kriptografija koja se nalazi iza blokčejn mreža, kao i bezbednosni aspekti blokčejn mreža i dosadašnji sajber napadi na blokčejn mreže.

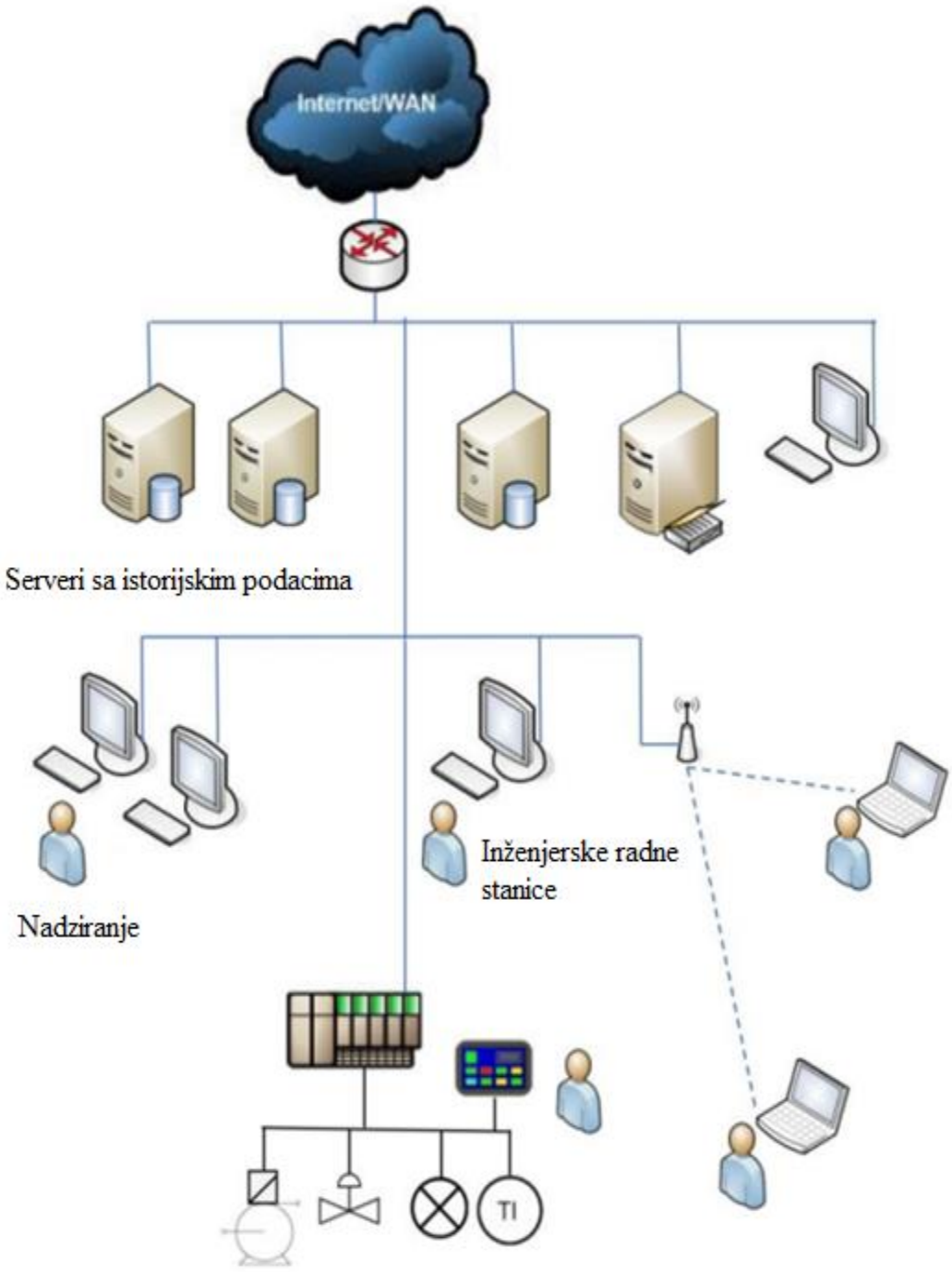
U okviru Poglavlja 4, opisane su metodologije za razvoj softvera, standardi i prakse koje se mogu koristiti za bezbedan razvoj softvera, čiji cilj je da bezbednost bude sastavni deo razvoja softvera, što je posebno bitno za razvoj softvera za industrijske upravljačke sisteme. Takođe, definisan je model za praćenje zahteva za bezbedan razvoj softvera, u okviru kojeg su identifikovani tokovi podataka, uloge u samom modelu i definisane su odgovornosti identifikovanih uloga.

U okviru Poglavlja 5, prikazana je validacija predloženog modela, na zahtevima koji su deo prakse za Upravljanje bezbednošću, u okviru standarda IEC 62443-4-1. Validacija je urađena za svaki korak koji je predstavljan modelom, kao i postavljanje rešenja na Hyperledger Fabric okruženje.

U okviru Poglavlja 6, prikazan je zaključak ove doktorske disertacije, sa mogućim pravcima daljeg istraživanja.

2.Aktuelno stanje u oblasti

Kako bi se bolje razumelo šta su industrijski upravljački sistemi, može se pogledati definicija okviru NIST standarda pod nazivom „*Guide to Industrial Control Systems (ICS) Security*", gde: „industrijski upravljački sistemi predstavljaju opšti opis koji obuhvata više tipova upravljačkih sistema, uključujući sistem za nadzor i prikupljanja podataka (eng. *Supervisory control and data acquisition*, SCADA), distribuirani upravljački sistemi (eng. *Distributed Control Systems*, DCS), i drugi sistemi koji se mogu naći u industrijskom sektoru i kritičnim infrastrukturama, kao što su programabilni logički kontroleri (eng. *Programmable Logic Controllers*, PLC)" [31]. Kada se pogledaju i drugi opisi, pored navedenih elemenata, industrijski upravljački sistemi uključuju i programabilne automatske kontrolere (eng. *Programmable Automation Controllers*, PAC), udaljenje terminalne jedinice (eng. *Remote Terminal Unit*, RTU), pametne elektronske uređaje (eng. *intelligent electronic device*, IED) i senzore [32]. U okviru rada [33], prikazane su tipične arhitekture industrijski upravljačkih sistema, što je prikazano i na Slici 1. Kroz isti rad [33], dato je i pojašnjenje razlike između IT (eng. *Information Technology*) i OT (eng. *Operational Technology*) sistema, gde je glavna distinkcija to što OT sistem bude izolovan od ostatka IT sistema i bude povezan sa fizičkim uređajima na terenu.



Slika 1 Arhitektura industrijskog upravljačkog sistema [33]

Industrijski upravljački sistemi se koriste u velikom broju industrija, uključujući i kritične infrastrukture, koje uključuju manufakturu, isporuku i saobraćaj [31]. Ukoliko bi došlo do onesposobljavanje ili uništavanje kritičnih infrastruktura, takvi događaji bi se oslikali na bezbednost, ekonomsku stabilnost, javno zdravlje i bezbednost ili neku od kombinacija [4]. Sjedinjene Američke Države su identifikovale sledećih 16 sektora kao kritične infrastrukture: hemijski sektor, sektor komercijalnih objekata, sektor za komunikacije, sektor kritične

proizvodnje, sektor brana, sektor odbrambene industrije, sektor hitne pomoći, energetski sektor, sektor finansijskih usluga, sektor za hranu i poljoprivredu, sektor državnih objekata, sektor zdravstva i javnog zdravlja, sektor informacionih tehnologija, sektor nuklearnih reaktora, materijala i otpada, sektor transportnih sistema i sektor voda i otpadnih voda [4]. Sama definicija kritičnih infrastruktura obuhvata i aspekt bezbednosti, što se može posmatrati i kao fizička i sajber bezbednost. Kada se pogleda trend koji postoji u nauci [34], vidi se rast broja istraživanja vezanih za kritične infrastrukture i bezbednost. Kako su industrijski upravljački sistemi sastavni deo kritičnih infrastruktura, neophodno je sagledati i aspekt bezbednosti takvih sistema. Uzimajući u obzir da bilo kakve ranjivosti u tim sistemima mogu imati ozbiljne posledice po bezbednost ljudi i ekonomsku stabilnost države, javlja se potreba za podizanjem nivoa njihove zaštite i nivoa svesti o eventualnim posledicama ne rešavanja identifikovanih problema. S obzirom na njihovu ključnu ulogu, svaka ranjivost u tim sistemima može dovesti do problema u radu infrastrukture i izazvati ozbiljne posledice po društvo i ekonomiju, što je opisano u ranijim primerima. Bezbednost industrijskih upravljačkih sistema je od vitalnog značaja za zaštitu ljudskih života i imovine. Potencijalne ranjivosti u tim sistemima mogu biti iskorišćene od strane napadača kako bi se izvršili sajber napadi, što može rezultirati poremećajem rada kritičnih infrastruktura, gubitkom resursa i čak ugrožavanjem fizičke sigurnosti. Stoga, da bi se održala stabilnost i sigurnost, neophodno je usmeriti pažnju na analizu i otklanjanje ranjivosti industrijskih upravljačkih sistema.

Bezbednost industrijskih upravljačkih sistema možemo posmatrati dvojako: fizička bezbednost sistema ne sme biti zanemarena jer može predstavljati moguću tačku napada. Kada se priča o fizičkoj bezbednosti nekog sistema, tu se podrazumevaju kontrole koje možemo fizički uočiti, kao što su ograde, čuvari, psi, vrata zaštićena specijalizovanim zaštitama ulaska, alarmi, itd. U Kaliforniji, 2014. godine došlo je ispaljivanja više hitaca iz poluautomatskog oružja, koje je proizvelo štete na 17 transformatora. Incident se završio bez većih smetnji u isporuci električne energije [35]. Slična situacija je viđena u maju 2022. godine, kada je pucano na trafostanice, ostavljajući preko 30 hiljada potrošača bez napajanja električnom energijom [36]. U okviru NIST standarda [31], postoji poglavlje posvećeno fizičkoj i zaštiti životne sredine, dok su sve predložene kontrole, ukupno 23, obrađene u zasebnom NIST standardu pod nazivom *Security and Privacy Controls for Information Systems and Organizations* [37]. Standard je usmeren i na kontrole koje mogu primeniti organizacije, sa ciljem smanjivanja rizika. Kako insajderske pretnje mogu takođe predstavljati polaznu tačku za napad, potrebno je postaviti kontrole koje će sprečiti, ili makar smanjiti šansu za takvu vrstu napada.

Pored standarda [37], u širokoj primeni je serija standarda ISO 27001 pod nazivom Sistem Menadžmenta Zaštite i Bezbednosti Informacija (eng. *Information security management systems,* ISMS). Serija standarda ISO 27001 kreirana je od strane Međunarodne organizacije za standardizaciju (eng. *International Organization for Standardization*, ISO) i Međunarodne elektrotehničke komisije (eng. *International Electrotechnical Commision*, IEC) i obuhvata 3 standarda, pod nazivima: ISO/IEC 27000 Informaciona tehnologija – Tehnike bezbednosti –

Sistemi menadžmenta bezbednošću informacija – Pregled i rečnik, ISO/IEC 27001 Informacione tehnologije – Tehnike bezbednosti – Sistemi menadžmenta bezbednošću informacija – Zahtevi, ISO/IEC 27002 Informaciona tehnologija – Tehnike sigurnosti – Pravila prakse za upravljanje sigurnošću informacija [38]. Zahtevi opisani u okviru standarda ISO/IEC 27001 su primenljivi na širok skup industrija, kao što su javne organizacije [39] i vladine organizacije zadužene za obezbeđivanje glasanja [40], budući da se predstavljene kontrole implementiraju na nivou organizacije. Svakako, standard je primenljiv i na energetski sektor, što je opisano u radu [41], gde je organizacija koja se nalazi u industriji nafte i gasa prikazala unapređenje svog sistema, primenom datog standarda.

Uz identifikovanje fizičkih kontrola koje je neophodno primeniti kako bi industrijski upravljački sistemi bili bezbedniji, potrebno je vršiti testiranja kako bi se proverila uspešnost samih kontrola. U okviru [42], prikazane su varijacije testnih okruženja za industrijske upravljačke sisteme, što uključuje fizičko, hibridno i virtuelno okruženje. Virtuelno okruženje, kao najjednostavnije za postavljanje, ne može u dovoljnoj meri da oslika stvarne moguće fizičke napade, čime se priprema za stvarne napade ne može sprovesti u realnim uslovima. Ideja za kreiranje fizičkih okruženja za testiranje nije neophodna samo za industrijske upravljačke sisteme u energetskom sektoru, već i u drugim sektora koji zavise od takvih sistema, što je autorima [43] pomoglo da kroz 4 različita napada, ukažu na moguća unapređenja sistema za pomorsku odbranu. Slične ideje fizičkih okruženja za testiranje industrijskih upravljačkih sistema prikazane su i u radovima [44] i [45].

Fizička bezbednost industrijskih upravljačkih sistema se ne sme zanemariti ili umanjiti mogućnost napada na kritične infrastrukture kroz taj pravac, međutim, mnogo veća pažnja pridaje se računarskoj, odnosno sajber bezbednosti sistema. U pregledu sajber napada koji su se desili na industrijske upravljačke sisteme [46], vidimo i prvi zabeleženi napad, koji se desio 1903. godine. Te godine je G. Markoni prezentovao bezbedno bežično slanje poruke sa radio stanice koja je bila udaljena oko 500km od Londona. Međutim, pred sam početak prezentacije, emitovana je poruka koja nije došla od izlagača, čime je pokazana slabost prikazane komunikacije, budući da je bilo moguće napasti sistem, što je kasnije i sam Markoni potvrdio. Autori u [46], napravili su pregled napada koji su se desili do 2018. godine, pojašnjavajući i najpoznatije napade na industrijske upravljačke sisteme, nazvani Stuxnet, NotPetya, TRITON/Trisis/HatMan i drugi. Analiza napada koji su se desili na industrijske upravljačke sisteme prikazan je i u radu [47], gde je pored prethodno pomenutih napada prikazano i još nekoliko napada koji su se desili do 2020. godine. Autori su u radu [47], prikazali i tabela sa nazivom napada, godinom kada se napad desio, ulazno mesto napadača, vrstu napadača, sektor i uticaj koji je napad ostavio. Iako su svi pobrojani napadi opisani, tri napada su identifikovana kao ključna, budući da autori smatraju da su najpoznatiji i imaju najviše dostupnih informacija. Jedan od napada je Stuxnet napad, koji je zbog svoje kompleksnosti i pokrivenosti u medijima i nauci, pojašnjen u velikom broju članaka i radova [48], [49], [50], [51] [52], [53]. Meta napada je bio Iranski nuklearni program, čiji napadači nisu do kraja sa sigurnošću utvrđeni i taj napad

se smatra jednim od najsofisticiranijih u sferi sajber ratovanja. Iako Stuxnet i Triton/Trisis napad imaju zajedničke osobine, s obzirom da su oba napada realizacija malvera, Triton napad je imao drugačiji ulaz u sistem. Malver Triton/Trisis je ušao preko radne stanice i bio usmeren ka jednom postrojenju nafte i gasa na Srednjem Istoku [47]. Na osnovu podataka dostupnih u [54] i [55], napravljena je Tabela 1 u kojoj su sumirani neki od sajber napada na industrijske upravljačke sisteme koji su se dogodili u 2022. godini.

Tabela 1 Sajber napadi na industrijske upravljačke sisteme u 2022. godini

Naziv	**Tip napada**	**Industrija**	**Država**	**Opis**
Napad na belorusku železnicu	Ransomver	Transport	Belorusija	Napad na belorusku železnicu, rezultovao je šifrovanjem velikog broja sistema.
Napad na Rosneft Deutschland GmbH	Bez informacija	Nafta	Nemačka	Iako su hakeri uspeli da preuzmu 20 gigabajta podataka, cevovodi i rafinerije nisu bile pogođene sajber napadom i nastavile su neometano da rade.
Conti ransomver	Ransomver	Vetrenjače	Nemačka	Nakon rane detekcije sajber napada, Nordex SE kompanija, jedan od najvećih proizvođača vetrenjača, bila je prinuđena da ugasi svoje IT sisteme kako bi sprečila dalje širenje napada.
Revil napad	Rasnsomver	Nafta	Indija	Ransomver napad afektovao je IT sisteme u kompaniji *Oil India,* kada su napadači tražili otkupninu od 7,5 miliona dolara. Sama kompanija je izjavila da ovim sajber napadom nisu bili afektovani sistemi proizvodnje i bušenja nafte.
Napadi na Creos i Enovos	Malver	Energija	Luksemburg	Kompanija sa sedištem u Luksemburgu, Encevo, saopštila je da su dve ćerke firme bile mete sajber napada na leto 2022. godine, kada su napadači uspeli da izvuku podatke i učine podatke nedostupnim, onemogućava-

				jući pristup sajtovima za kupce tih firmi.
Napad na DESFA	Rasnsomver	Prirodni gas	Grčka	Ransomver grupa pod nazivom *Ragnar Locker* preuzela je odgovornost za napad, u okviru kojeg je došlo do curenja osetljivih kompanijskih podataka. Takođe, grupa je tvrdila da je pronašla određeni broj bezbednosnih propustsa na DESFA sistemima, ali se kompanija nije oglašavala po tom pitanju.
Napad na Eni	Ransomver	Nafta	Italija	Tehnički detalji napada nisu poznati. Sam napad je afektovao kompanijsku mrežu, ali je po saopštenju kompanije brzo detektovan i prijavljen vlastima.
Napad na Tata Power	Ransomver	Energija	Indija	Ransomver grupa napadača pod nazivom *Hive,* preuzela je odgovornost za napad, pokazujući podatke o kompaniji na tamnoj mreži (eng. *dark web*), a vezani su za zaposlene, ugovore i dobavljače.
Napad na EPM	Ransomver	Energija	Kolumbija	Napad na *Empresas Públicas de Medellín* (EPM) doveo je do otežanog rada zaposlenih, ali i verovatno velike količine podataka koji su ukradeni, budući da je daljom analizom identifikovano nekoliko foldera koji su imali prefiks EPM, na nezaštićenom serveru koji se koristi za analizu.
Napad na Enercity	Bez informacija	Energija	Nemačka	Jedan od najvećih opštinskih snabdevača energijom je bio meta sajber napada. Tokom samog napada, nije došlo do

				prekida isporuke električne energije, budući da sama kritična infrastruktura nije bila napadnuta i nije došlo do curenja ličnih podataka klijenata, već se napad završio sa nedostupnošću veb sajta kompanije.
Napad na Eesti Energia	Distribuirani napad uskraćivanja usluge (DDoS)	Energija	Estonija	Napad je obuhvatio nekoliko veb sajtova i aplikacija, koje pripadaju nacionalnom proizvođaču električne energije u Estoniji, ali napadači nisu uspeli da ukradu podatke.

U Tabeli 1 prikazani su i neki od mogućih tipova napada, kao što ransomver i malver, ali u tabeli nisu prikazani tipovi napadača koji su taj napad i izvršili. Za neke od napada se smatra da je delo neke od APT (eng. *Advanced Persistent Threat*) grupa, koje se smatraju za izuzetno vešte i tehnički sposobne hakere. Ukoliko sama grupa ne preuzme odgovornost za neki napad, može se desiti i da se sami napadači nikada ne otkriju. Prilikom pravljenja plana zaštite informacija sistema, neophodno je analizirati tip napadača koji može izvršiti napad. Ukoliko se radi o neosetljivim informacijama, odnosno informacijama koje ne mogu naškoditi privatnosti ljudi, bezbednosti ljudi i sistema, kontrole koje je potrebno primeniti ne moraju biti istog nivoa kao kada je neophodno obezbediti nuklearno postrojenje, gde napad može ostaviti katastrofalne posledice. U te svrhe, moguće je iskoristiti opise sajber napadača koji su opisani u [56], gde postoji jasna razlika između napadača koji mogu zarad samopromocije da naprave manju štetu koristeći javno dostupne alate i informacije i napadača koji su sponzorisani od strane zainteresovanih država, sa ciljem nelegalnog pristupa vladinim organizacijama, sistemima i kritičnim infrastrukturama. Sa druge strane, autori [57] su napadače na organizacije podelili u 5 kategorija, spram tipa štete koju žele da načine. Ti tipovi štete su kategorisani kao fizička ili digitalna šteta, ekonomska šteta, psihološka šteta, reputaciona šteta i socijalna šteta. Svaki od tipova štete ima i definisane motive koji bi napadače svrstali u datu grupu. Takav korak kategorizacije predstavlja početnu tačku, sa ciljem boljeg razumevanja potencijalne štete i motiva, dok autori [57] ističu da je sledeći korak pravljenje modela koji je okrenut ka digitalnim resursima.

Različite statistike govore o broju sajber napada koji se dešavaju konstantno u svetu, gde jedan izvor navodi da je to svakih 39 sekundi [58], a drugi da se sajber napad desi svake 44 sekunde u svetu [59]. Treba imati na umu i da je ovo statistika za broj pokušanih napada, ali ne i uspešnih, kao i da broj uspešnih napada ne raste nužno brzinom kojom raste broj samih pokušaja. Svi ovi napadi i statistike idu u prilog sve jačoj proveri postojećih bezbednosnih

kontrola, kao i zahtevima da se poštuju različiti bezbednosni standardi, koji su predloženi za svaku industriju pojedinačno. Kada se pogledaju zahtevi u okviru bezbednosnih standarda, usmerenih na kritične infrastrukture, možemo ih posmatrati kao zahteve koji pomažu samim organizacijama da podignu svoju bezbednost na viši nivo, ali postoje i standardi koji svojim zahtevima usmeravaju proizvođače softvera za upravljanje industrijskim upravljačkim sistemima da implementiraju bezbednost kroz čitav životni ciklus razvoja softvera. Jedan od standarda koji svojim kontrolama i zahtevima utiče na podizanje bezbednosti organizacije je ISO27001, koji je primenljiv ne samo u kritičnim infrastrukturama već i u drugim oblastima, kao što su bankarski sektor, zdravstvo itd. Standard ISO 27001 je standard za pravljenje sistema menadžmenta bezbednošću informacija, čijom implementacijom organizacije imaju obavezu sprovođenja treninga za zaposlene, održavanje liste postojećih sistema, klasifikacija informacija, pravo pristupa osetljivim informacijama, kao i redovna evaluacija postojećih kontrola. Standard može biti primenjen i na organizacije koje pružaju *cloud* usluge, gde su autori u [60] prepoznali da se zahtevi, iako predstavljeni kao opšti, ne mogu u potpunosti primeniti na organizacije koje su prešle na cloud oblik poslovanje. Takođe, autori u [60] prave razliku između privatnih i javnih cloud rešenja, budući da se kontrole iz ISO 27001 mogu tumačiti na različite načine u odnosu na tip okruženja. Kao dalji pravci istraživanja, autori u [60] predloželi su standarde COBIT, GAISP, SSE-CMM, FISMA, ISNI/ISA 99 i NIST. Prvi predloženi okvir, COBIT (eng. *Control Objectives for Information and Related Technologies),* usmeren je na upravljanje informacionim tehnologijama. Prva verzija nastala je 1996. godine dok je poslednja verzija, COBIT 5, završena 2019. godine i uvažila je veće izmene koje su organizacije imale priliku da predlože, a detaljno su pojašnjene u okviru rada [61]. Iako standardi i regulative mogu predstavljati odličan izvor informacija i dobrih praksi, potrebno je obratiti pažnju koji od standarda je obavezan za specifičnu industriju, jer neusaglašenosti sa standardom ili regulativom mogu dovesti do sankacionisanja, kao što je na primer novčana kazna, a koji standardi i regulative se mogu primeniti bez potrebe za zvaničnom potvrdom. Kada je u pitanju pomenuti okvir FISMA (eng. *Federal Information Security Modernization Act)* iz rada [61], treba imati na umu da je taj okvir obavezan za sve federalne agencije koji se nalaze u Sjedinjenim Američkim Državama. Poređenje FISMA okvira sa standardom ISO 27000, kao relativno sličnim okvirima, obrađeno je u okviru rada [62]. Pored poređenja FISMA i ISO 27000 standarda, u istom radu nalazi se i poređenje sa PCI-DSS standardom kao i HIPAA. Vizuelna reprezentacija preklapanja pokazuje koliko je FISMA okvir širi od ostalih spomenutih standarda [62], ali je neophodno sagledati i oblasti u kojima se dati okviri i standardi mogu primeniti. Bez obzira na postojanje zajedničkih kontrola, PCI-DSS (eng. *Payment Card Industry Data Security Standard*) standard [63] je usmeren na bankarski sektor i diktira bezbednosne kontrole koje je neophodno ispuniti u slučaju čuvanja, kreiranja i upravljanja transakcijama. Sa druge strane, HIPAA (eng. *Health Insurance Portability and Accountability Act*) je isključivo vezan za zdravstveni sektor i zaštitu privatnosti pacijenata [64].

Broj standarda, okvira i regulativa može postati prevelik kako bi se ispratili svi zahtevi koje je potrebno zadovoljiti, što je problem koji može biti rešen praćenjem procesa koji su dostupni

kao deo inženjerstva zahteva. Oslanjajući se na razne standarde, kao što su IEC 615081 standard, CENELEC 50126, 50128 i 50129 i ISO 26262 standard, autor [65] je predočio definiciju zahteva, gde je zahtev definisan kao „izjava koja ukazuje na potrebu i/ili ograničenje i može biti napisana u matematičkom, prirodnom jeziku itd.". Inženjerstvo zahteva za autore predstavlja prikupljanje, razvoj i upravljanje zahtevima, koje je podeljeno u više podfaza, od studije izvodljivosti do dokumentovanja [66]. Cheng i Atlee u [67], su drugačije predstavili faze inženjerstva zahteva, pokazujući da postoje 5 faza, odnosno izvlačenje (eng. *elicitation*), modelovanje, analiza zahteva, validacija i verifikacije i na kraju upravljanje zahtevima. Za svaku od faza definisane su metodologije, strategije i alati koji mogu doprineti uspešnosti te faze. Autori su takođe prepoznali i probleme prilikom poslednje faze, odnosno faze upravljanje zahtevima, što je pogotovo istaknuto u slučaju kada postoji veliki broj zahteva koji su u različitim fazama razvoja za različite proizvode. Ukoliko se fokusiramo na problem praćenja zahteva, autori [68] su identifikovali koje bitne, delimično bitne i potencijalno zanemarive osobine treba da ima neki alat za upravljanje zahtevima. Uprkos velikom broju dostupnih metodologija za praćenje zahteva, problemi prilikom analize, prikupljanja ili modelovanja se i dalje mogu uočiti u praksi. Različite vrste metodologija koje mogu biti primenjene u okviru inženjerstva zahteva, autor [69] je u predočio kroz različite metafore, koje ima cilj da lociraju i identifikuju problem. Iako metafore omogućavaju različit pogled na probleme koji se javljaju u inženjerstvu zahteva, postoje određene sličnosti za svaku od metafora. Ukoliko se posmatraju tri elementa, okruženje, specifikacija i zahtevi, autor [69] je uočio zavisnost koja važi: „za svako okruženje kojem je potreban zahtev, potrebno je pronaći specifikaciju kao rešenje koje će omogućiti tu vezu".

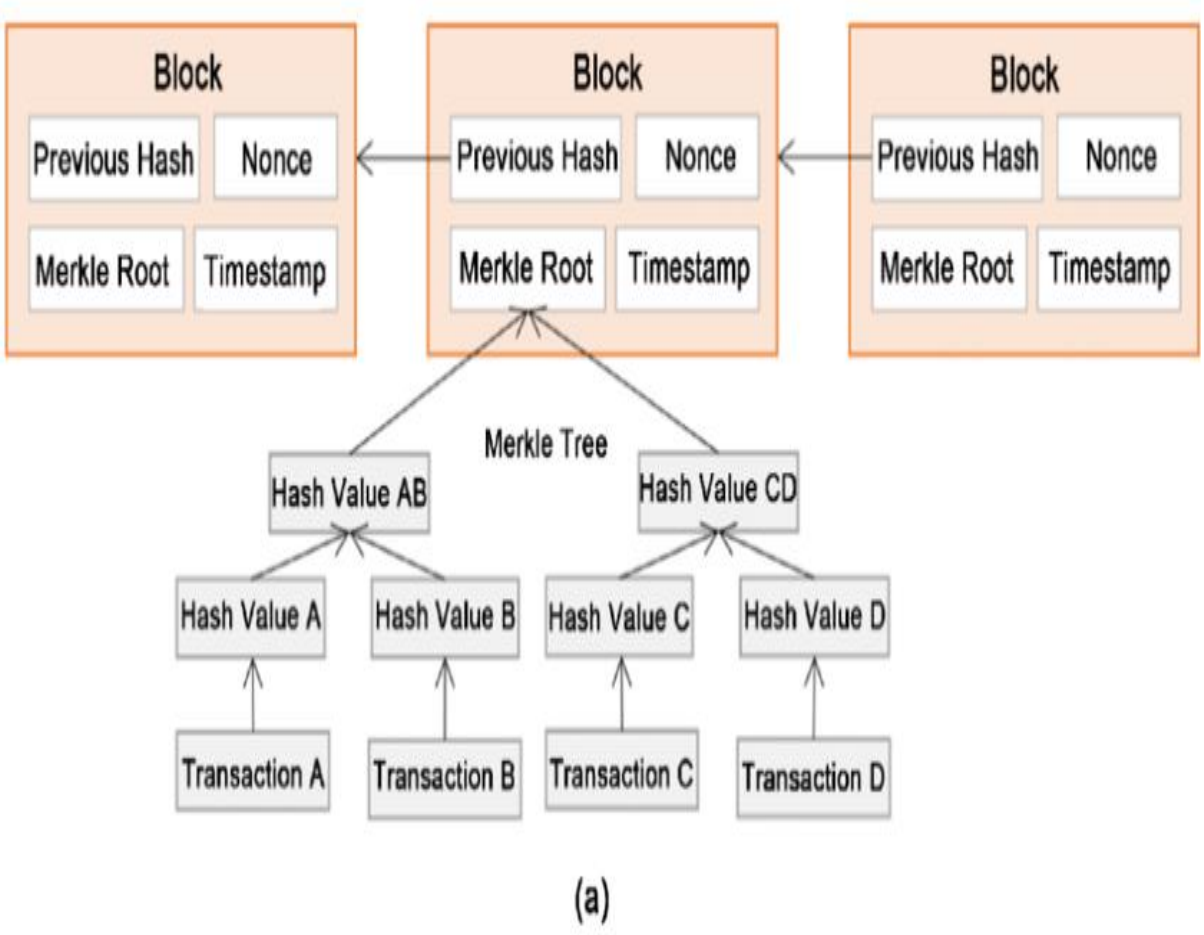
Ako bismo pored praćenja životnog ciklusa zahteva želeli da se zahtevi ili prateći dokumenti održavaju verzionisanim, dok su u isto vreme rezistentni na neovlašćeno menjanje, mogli bismo da iskoristimo osobine koje pruža blokčejn tehnologija. Blokčejn tehnologija je prvo pažnju privukla u finansijskom svetu kada je digitalni novac pod nazivom Bitkoin predstavljen 2009. godine. Sama tehnologija je međutim mnogo šira od finansija i našla je primenu u zdravstvu [70], gde su autori prikupili istraživanja i trenutne primene blokčejn tehnologije u različitim podoblastima zdravstva, kao što su upravljanje zdravstvenim kartonima, lanci snabdevanja lekovima, istraživanje i edukacija, kao i analitike zdravstvenih podataka. Uprkos velikom broju mogućih primena, autori su identifikovali i određene oblasti koje zahtevaju dodatno istraživanje kako bi se u potpunosti omogućila primene blokčejn tehnologije u zdravstvu, gde su neki od problema skalabilnost, interoperabilnost, bezbednost i privatnost. Pored zdravstva, uočena je i moguća primena blokčejn tehnologije u okviru sektora javne uprave, konkretno prilikom upotrebe usluga notara [71], [72], gde je ideja da se ukine notar kao centralni autoritet. Takođe, jedna od mogućih oblasti primene blokčejn tehnologije prikazan je na primeru katastra [73], gde su autori predstavili unapređenje trenutnog načina vođenja informacionog sistema, time što je novo rešenje javno dostupno na javnoj blokčejn mreži sa dozvolom. Svakako neizostavni način upotrebe korišćenje blokčejn tehnologije je upotreba u oblasti lanaca snabdevanja [74]. Pored navedenih, Demi i ostali su u okviru više

radova [75], [76], [77], [78] istraživali i pokazali prednosti korišćenja blokčejn tehnologije u praćenju zahteva. Prednosti koje donosi blokčejn tehnologija, kao što su otpornost, transparentnost, neporecivost se mogu iskoristiti za potrebe upravljanja zahtevima, budući da zahteve neophodno zavoditi, ažurirati i implementirati, dok se vodi računa o sledljivost. Kada je u pitanju otpornosti, ona se može posmatrati u svojstvu dostupnosti podataka. Kako je blokčejn mreža jedan decentralizovani i distribuirani sistem, problem kao što je jedna tačka kvara (eng. *single point of failure*) je izbegnut, budući da je mreža rasprostranjena na veći broj čvorova, te nedostupnost jednog čvora ne znači i nedostupnost cele mreže. Sa stanovišta transparentnosti, blokčejn tehnologija omogućava čitavu istoriju transakcija koje su zabeležene na toj mreži. Treba imati na umu ta takva istorija transakcija ne mora da znači da je ona dostupna i svim korisnicima mreže, što je pogotovo slučaj sa privatnim blokčejn mrežama, koje definišu prava pristupa određenom skupu korisnika. Kao bitna osobina blokčejn tehnologije je svakako neporecivost, čime se onemogućava korisnik koji želi da skloni odgovornost sa svojih aktivnosti, u čijem slučaju se ostali korisnici mogu osloniti i na osobinu sledljivosti. Pored sledljivosti, neophodno je vodi računa i o bezbednosti informacija koje se postavljaju na blokčejn mrežu. Razliku u tipovima blokčejn mreža napravili su Golosova i ostali [25], gde su prikazane razlike i prednosti privatne i javne blokčejn mreže, o kojima će kasnije biti više reči. Ukoliko je potrebno skladištiti informacije kojima pristup ima samo unapred određeni skup učesnika, privatna blokčejn mreža predstavlja pogodnije rešenje, budući da javna blokčejn mreža podrazumeva da su informacije dostupne svima, bez ikakvih ograničenja.

Objedinjavanjem prethodnog diskutovanih tema, industrijski upravljački sistemi, kritične infrastrukture, njihova bezbednost, napadači i prethodni napadi, inženjerstvo zahteva i blokčejn tehnologija, dolazi se do motivacije za ovo istraživanje. Omogućavajući industrijskim upravljačkim sistemima, koji žele da poboljšaju svoje bezbednosno stanje i zrelost, predložen je model zasnovan na privatnom blokčejnu, čime se omogućava praćenje i upravljanje zahtevima, koji su usmereni na bezbedan životni ciklus razvoja softvera. Kako su podaci industrijskih upravljačkih sistema osetljivi, odnosno zahtevaju dodatne bezbednosne kontrole čime bi informacije bile dostupne isključivo ljudima i sistemima koji za to imaju odobrenje, neophodno je jasno definisati kontrole pristupa sa ciljem zaštite informacija. S tim na umu, jedna od bezbednosnih kontrola jeste da se podaci vezani za industrijske upravljačke sisteme postavljaju na privatne blokčejn mreže. Privatne blokčejn mreže predstavljaju jedan od načina korišćenja blokčejn tehnologije, na način da se ne omogućava javni uvid u podatke koji su skladišteni na mreži. Time je onemogućen pristup podacima koji se mogu smatrati osetljivim osobama koje nemaju pravo da imaju uvid ili pravo izmene tih podataka. Ovakva osobina je od izuzetne važnosti za kritične infrastrukturne sisteme, budući da se informacije koje takvi sistemi koriste klasifikuju kao osetljivi ili poverljivi. Ukoliko bi informacije koje koriste kritične infrastrukture bile javno dostupne, mogle bi da predstavljaju način za sajber napade, čime se ugrožava bezbednost i privatnost ljudi.

3.Blokčejn mreže

Blokčejn tehnologija se zasniva na kriptografskim algoritmima koji postoje duži perdiod, ali su tek u poslednjoj deceniji iskorišćeni u okviru jedne decentralizovane tehnologije, zasnivane na lancu blokova, odnosno istoriji svih transakcija koje su se dogodile. Ta istorija je organizovana u blokove, te otuda ime *blockchain* (eng. *block* – blok, *chain* – lanac). Na Slici 2 je prikazan lanac blokova koji sadrži tri bloka. U okviru svakog bloka se nalazi heš na prethodni blok, slučajna vrednost, koren Merkle stabla i vreme. Svaki novi blok koji nastaje, uvezuje se u listu, tako da ima vezu sa prethodnim blokom u vidu heša [79]. Ono što čini blokčejn otporan na manipulisanje (eng. *tamperproof*) su kriptografski otisak koji čini svaki blok u lancu jedinstvenim, kao i konsenzus protokol, u okviru kojeg se 51% čvorova mora složiti po pitanju redosleda i sadržaja blokova.



Slika 2 Lanac blokova [79]

Blokčejn mrežu karatketriše i *peer-to-peer* arhitektura, gde je definicija takve arhitekture data u okviru [80]: „*distribuirana mrežna arhitektura se može nazivati arhitekturom ravnopravnih računara ukoliko učesnici u mreži dele deo svojih hardverskih resursa (procesnu*

moć, kapacitet za skladištenje, štampače itd.). Takvi deljeni resursi su potrebni kako bi se omogućili servisi i sadržaj koje nudi mreža (npr. deljenje fajlova). Resursima mogu direktno pristupiti učesnici, bez potrebe za posrednikom". Kada se definicija pogleda u svetlu blokčejn mreža, arhitektura ravnopravnih računara omogućava blokčejn mreži da ima stalnu dostupnost, budući da nedostupnost jednog računara ne znači i prekid dostupnosti čitave mreže, ali znači i da svaki učesnik u mreži ima trenutno najnovije stanje mreže kod sebe.

Potrebno je napomenuti da blokčejn kao tehnologija i Bitkoin nisu sinonimi. Blokčejn tehnologija je popularnosti stekla nastankom Bitkoina, koji predstavlja decentralizovani način razmene novca. Zapaženost Bitkoina je dobijena zbog arhitekture rešenja, koja isključuje banku kao posrednika svih transakcija. Pored Bitkoina, veliki značaj imaju i Etereum (eng. *Ethereum*) i Hyperledger Fabric implementacije blokčejn tehnologije. Etereum je stekao ugled omogućivši korisnicima da kreiraju svoje aplikacije, napisane u obliku pametnih ugovora (eng. *smart contract*). Hyperledger Fabric takođe omogućava pisanje pametnih ugovora, ali za razliku od Etereuma, koji se okrenuo ka javnim blokčejn implementacijama, Hyperledger Fabric se okrenuo korporacijama, omogućavajući im privatne blokčejn mreže, odnosno mreže kojima mogu da pristupe samo prethodno autorizovani korisnici. Detalji implementacije, kao i razlike između Etereum i Hyperledger Fabric mreže su dati u narednim odeljcima.

U okviru ovog poglavlja će biti i osvrt na bezbednosne mehanizme koji su ugrađeni u blokčejn mreže, ali i na sajber napade koji su se do sada desili.

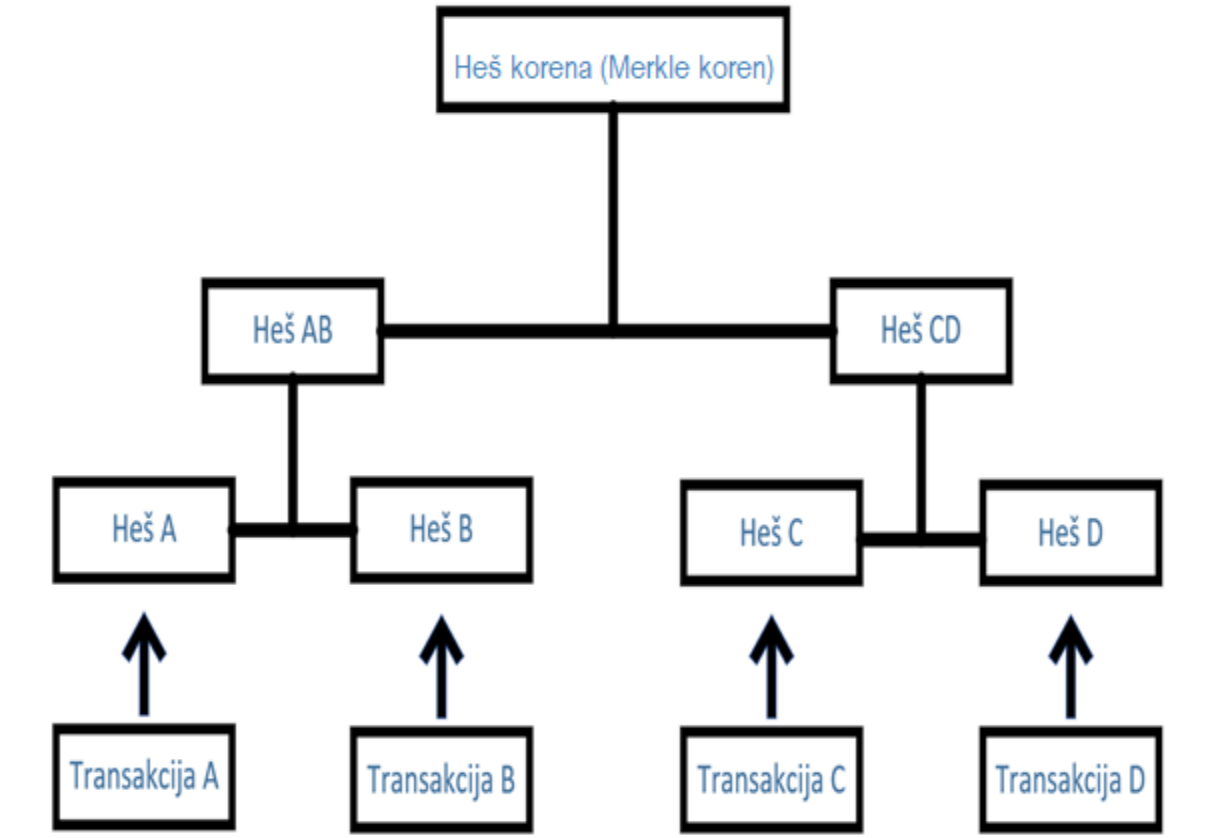
3.1. Kriptografski principi u blokčejn mreži

Osnova blokčejn tehnologije leži na kriptografskim principima koji omogućavaju korisnicima mreže poverljivost (u slučaju privatnih blokčejn mreža što će biti pojašnjeno kasnije u poglavlju) i integritet. Integritet, kao jedan CIA trijade (eng. *Confidentiality, Integrity, Availability),* je omogućen heš funkcijama, digitalnim potpisima i kodom za proveru identiteta poruke (eng. *Message Authentication Codes*, MAC) koje omogućavaju svakom bloku u lancu da pokaže da je nepromenjen. Kada je u pitanju dostupnost, ona je omogućena distribuiranom arhitekturom koju poseduje blokčejn mreža. Distribuirana arhitektura isključuje jedinstveno mesto na kojem se nalaze sve potrebne informacije, već su sve informacije dostupne na različitim lokacijama, odnosno dostupne su na svi čvorovima koji učestvuju u kreiranju blokova. U okviru ovog odeljka, biće opisane kriptografske tehnike i protokoli koji se koriste u okviru blokčejn tehnologije.

3.1.1. Merkle stablo

Merkle stablo je osnovni deo blockchain tehnologije [81]. Merkle stablo je binarno stablo, u kojem su listovi heširane vrednosti transakcija, a svaki čvor koji nije list je heš vrednost svoje dece. Koren Merkle stabla se naziva heš koren ili Merkle koren. Na Slici 3 je prikazano kreiranje Merkle korena, koje počinje od dole. U slučaju neparnog broja listova u Merkle

stablu, jedna od transakcija će biti duplirana, čime je ovo stablo ujedno i balansirano. Transakcije A, B, C i D su listovi Merkle stabla i za njih se radi heš funkcija (u slučaju Ethereum blokčejna radi se Keček256 [82]) i tada nastaju Hash A, B, C i D. Sledeći korak predstavlja pravljenje heša od Hash A i Hash B, odnosno Hash C i Hash D, gde nastaju HashAB i HashCD, respektivno. Poslednji korak u ovom primeru predstavlja heširanje HashAB i HashCD čime nastaje Merkle koren. Merkle stablo sadrži sve podatke o transakcijama i nalazi se u zaglavlju bloka. Na ovaj način se zadržava redosled transakcije, jer u slučaju bilo kakve izmene neke transakcije, heš te transakcije i svaki heš iznad u stablu će biti izmenjen, kao i sam koren na kraju. Koristeći Merkle stablo, može se jednostavno testirati da li je neka transakcija uključena u blok ili ne. Bitkoin za heširanje podataka u Merkle stablu primenjuje dva puta SHA256 heš algoritam, što se naziva i dupli-SHA256. Ono što Merkle stablo čini veoma efikasnom strukturom je što se provera pripadnosti nekog elementa stablu može dobiti sa najviše $2 * log2(N)$ kalkulacija, gde N predstavlja broj elemenata u stablu [83].



Slika 3 Merkle stablo

Što se tiče efikasnosti Merkle stabla, u Tabeli 2 je prikazan rast broja transakcija i veličina putanje koja se koristi za proveru transakcije.

Tabela 2 Efikasnost Merkle stabla

Broj transakcija	**Veličina bloka**	**Veličina putanje (u heševima)**	**Veličina putanje (u bajtima)**
16	4 kilobajta	4	128
512	128 kilobajta	9	288
2048	512 kilobajta	11	352
65535	16 megabajta	16	512

Iz Tabele 2 se vidi kako veličina bloka brzo raste od 4KB do 16 MB, dok Merkle putanja potrebna za dokaz uključivanja u transakciju raste mnogo sporije, od 128 bajtova do 512 bajtova [83].

3.1.2. Merkle Patricia stablo

Ethereum blokčejn ne koristi samo jedno stablo, već tri, i to za stanje, transakcije i račune. Pod terminom stanje (eng. *state*) se podrazumeva skup svih podataka koji definišu trenutno stanje sistema, dok računi predstavljaju adrese dugačke 40 heksadecimalnih karaktera, neophodno za praćenje novčanih sredstava korisnika ili pametnih ugovora, o kojima će biti reči kasnije. Takođe, stabla su unapređena, tako da Ethereum koristi Merkle Patricia stablo. Patricia (Practical Algorithm To Retrieve information Coded in Alphanumeric) je algoritam koji pruža fleksibilno čuvanje, indeksiranje i čitanje informacija iz velikih fajlova [84]. Merkle stablo je pogodno za transakcije jer se one unutar bloka ne smeju menjati. Ovakva struktura ne odgovara Ethereum-u koji želi da prati i stanje. Stablo stanja predstavlja mapiranje između ključa i vrednosti, gde su ključevi adrese računa, a vrednosti su osobine tog računa, kao što su iznos, slučajna vrednost i kôd (u slučaju pametnih ugovora, koji će biti kasnije objašnjeni). Budući da se iznosi na računima menjaju, i takvo stablo zahteva brze izmene. Pored toga, za stablo stanja je potrebna struktura podataka koja može relativno brzo da preračuna koren stabla nakon unosa, izmene ili brisanja, što se može desiti kada se doda novi račun ili se promeni iznos na računu. Takođe, takva struktura bi trebalo da ima ograničenu dubinu, čime bi se sprečio *Denial-Of-Service* napad, u okviru kojeg bi napadač pravio takve transakcije da budu što je dublje moguće, čime bi izmena bila veoma spora. Merkle Patricia stablo je u osnovi radiks stablo [85], odnosno stablo koje se koristi da bi se čuvala mapiranje ključ-vrednost, gde je ključ putanja kroz stablo do tog čvora, kako bi se došlo do odgovarajuće vrednosti. U stablu su čvorovi referencirani pomoću heš funkcija, tako da je koren stabla kriptografski otisak te strukture podataka. Ethereum uvod četiri tipa čvorova, kako bi se poboljšala efikasnost. Uvedeni čvorovi su:

- Prazan čvor
- Čvor list: čvor sa ključem i vrednosti

- Čvor grane: lista dužine 17, gde prvih 16 elemenata odgovara 16 heksadecimalnih karaktera koji mogu biti u ključu, a poslednji element predstavlja vrednost koja pokazuje da li postoji par ključ-vrednost gde ključ završava na čvoru grane
- Čvorovi produžetaka: ključ-vrednost čvor gde je vrednost heš vrednost nekog drugog čvora.

U okviru transakcionog stabla, takođe postoji mapiranje između ključa vrednosti, s tim da u ovom slučaju vrednost odgovora identifikatoru transakcije.

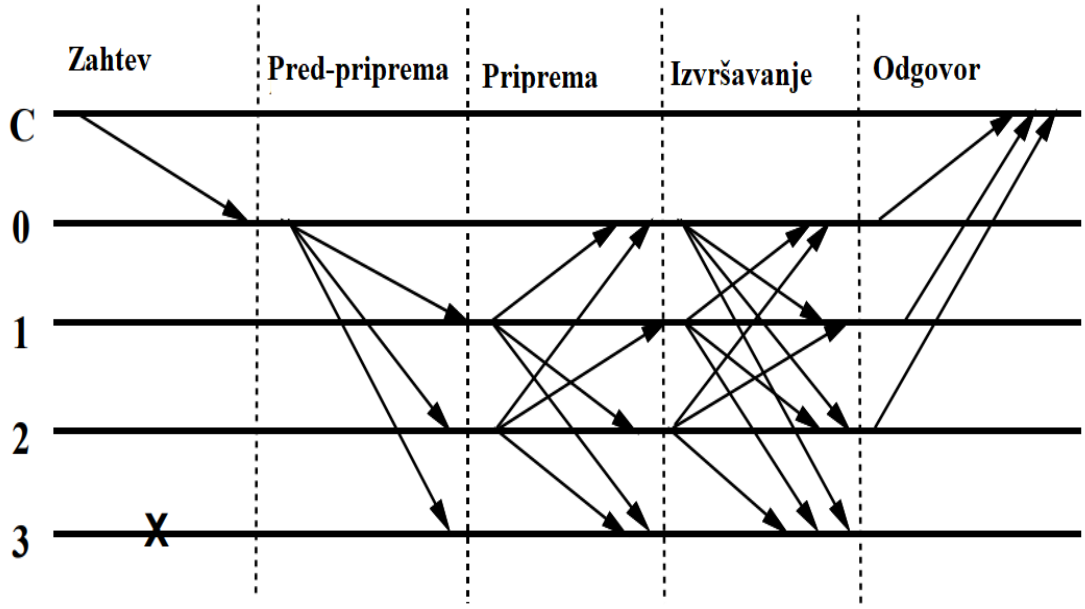
3.1.3. Problem vizantijskih generala

Problem vizantijskih generala opisali su L.Lamport, R.Šostak i M.Pise 1982.godine [86]. Problem govori o postizanju dogovora između određenog broja generala, među kojima su i izdajnici kojima je cilj da sabotiraju napad. Na ovaj način, objašnjen je problem koji se može desiti u okviru distribuiranog sistema, gde neki od čvorova pokušavaju da iskvare sistem, odnosno da ga preuzmu. Otpornost na vizantijsku grešku je karakteristika koja opisuje sistem koji je otporan na ispade iz klase problema vizantijskih generala. Proof-of-Work, koji će kasnije biti opisan, je probabilističko rešenje problema vizantijskih generala i upotrebljava se u Bitkoinu.

U distribuiranim sistemima, Byzantine Fault Tolerance (BFT) može biti dobar metod za rešavanje grešaka u prenosu. Do 1999. Practical Byzantine Fault Tolerance (PBFT) sistem je bio predložen i složenost algoritma je spuštena na polinomski nivo, čime se dobilo značajno poboljšanje po pitanju efikasnosti. Proces PBFT je prikazan na Slici 3. On se sastoji iz 5 koraka:

1. Zahtev (eng. *request*): klijent šalje zahtev glavnom serverskom čvoru, a glavni čvor daje tom zahtevu vremensku značku (eng. *timestamp*).
2. Pred-priprema (eng. *pre-prepare*): glavni serverski čvor beleži sve zahteve i dodeljuje im redni broj, po redosledu slanja. Zatim glavni čvor šalje pred-pripremnu poruku ostalim serverskim čvorovima. Tada serverski čvorovi odlučuju da li da prihvataju zahtev ili ne.
3. Priprema (eng. *prepare*): ukoliko je serverski čvor odlučio da prihvati zahtev, on šalje pripremnu poruku svim serverskim čvorovima i prima pripremnu poruku od drugih čvorova. Nakon što je prihvatio $2f+1$ poruku (gde je f najveći broj odgovora koji mogu biti netačni), ukoliko je većina čvorova odabrala da prihvati zahtev, ulazi se u izvršnu fazu.
4. Izvršavanje (eng. *commit*): svaki čvor u izvršnoj fazi šalje izvršnu poruku svim ostalim čvorovima. Istovremeno, ukoliko serverski čvor primi $2f+1$ izvršnu poruku, može doći do zaključka da je većina čvorova postigla konsenzus za prihvatanje zahteva. U tom slučaju, čvor izvršava instrukcije koje su poslate u okviru zahteva.

5. Odgovor (eng. *reply*): serverski čvor šalje odgovor klijentu. Ukoliko klijentski čvor ne dobije odgovor usled usporenja u mreži, zahtev se ponovo šalje serverskom čvoru. Ukoliko je zahtev izvršen, serverski čvorovi će samo ponovo poslati odgovor [87].



Slika 4 Koraci u PBFT

3.1.4. Konsenzus algoritmi

Konsenzus algoritam, u računarstvu, predstavlja uspešan dogovor o nekom podatku u okviru distribuiranog procesa ili distribuiranog sistema. Takvi algoritmi su potrebni u sistemima gde nisu svi čvorovi pouzdani, odnosno neki čvorovi imaju za cilj da sabotiraju čitav sistem. Poređenje konsenzus algoritama je opisano u [88]. U narednim odeljcima biće opisani konsenzus algoritmi koji se najčešće sreću u praksi.

3.1.4.1. Proof-of-work (PoW)

Proof-of-work (PoW) je konsenzus algoritam koji koristi Bitkoin. Osnovna ideja je da se za izvršen rad nagradi čvor koji je uložio resurse za verifikovanje transakcije i kreiranje bloka. Na osnovu informacija o prethodnom bloku, različiti čvorovi računaju specifičan matematički problem, koji je zahtevan sa strane računarski resursa. Prvi čvor koji nađe rešenje matematičkog problema, kreira blok i kao nagradu dobija određeni broj Bitkoina. Koraci za izračunavanje obuhvataju:

1. Odrediti težinu: nakon kreiranja svakih 2016 blokova, Bitkoin algoritam rudarenja će dinamički postaviti težinu na osnovu heš *rate*-a cele mreže.

2. Sakupljanje transakcija: potrebno je pokupiti sve transakcije u mreži koje su na čekanju, od kreiranja poslednjeg bloka. Zatim se računa Merkle stablo za svaku transakciju i popunjava se broj bloka, 256 bitna heš vrednost prethodnog bloka, ciljana heš vrednost, slučajna vrednost i ostale informacije.

3. Izračunavanje: proći kroz vrednosti za slučajnu vrednost (eng. *nonce*) od 0 do 2^{32} i izračunati duplu SHA256 heš vrednost iz drugog koraka. Ukoliko je heš vrednost manja ili jednaka od ciljane vrednosti, blok se šalje svim ostalim učesnicima u mreži.

4. Novi početak: ukoliko čvor nije uspeo da izračuna heš vrednost u određenom periodu, ponavlja se drugi korak. Ukoliko je neki drugi čvor završio računanje, počinje sa korakom 1.

Novo kreirani blokovi se uvezuju u blok ispred. Dužina lanca proporcionalna je količini opterećenja koje treba odraditi. Svi čvorovi veruju najdužem lancu. Ukoliko neko želi da maliciozno izmeni vrednosti u blokčejnu, morao bi kontrolisati više od 50% svetske snage za heširanje kako bi obezbedio da uvek generiše poslednji blok i time upravlja najdužim lancem. Dobiti ovakvog pristupa su mnogo veće od utroška resursa koji se dešava prilikom kreiranja svakog bloka, i time PoW garantuje sigurnost blokčejna [89].

3.1.4.2. Proof-of-stake (PoS)

Proof-of-Stake je konsenzus protokol koji pruža moć odlučivanja, da li će dodati novi blok ili ne, onim učesnicima koji imaju uloge u sistemu bez obzira na dužinu lanca ili istoriju javnog ledger-a. Motivacija iza ovog konsenzus protokola je da se da moć odlučivanja zainteresovanim stranama. Takav postupak je urađen kako bi se osigurala bezbednost sistema ukoliko članovi budu pod pretnjom. Postupak je sličan PoW konsenzus algoritmu osim dela sa izračunavanjem. Šanse da akter uključi svoj blok u blokčejn proporcionalno zavisi od njegovog udela u čitavom sistemu. Ukoliko neki učesnik sistema ima 500 jedinica kriptovalute, on ima 5 puta veće šanse da bude odabran za kreiranje bloka od učesnika koji ima 100 jedinica kriptovalute. PoS konsenzus mehanizam zahteva da se uloge podele pre početka procesa, što nije slučaj u PoW prilazu. Inicijalni dizajn PoS je uključivao starost kriptovalute i totalnu sumu kako bi se odredila uloga svakog rudara u sistemu [90].

3.1.4.3. *Stellar* konsenzus protokol

Stellar konsenzus protokol je predložen kako bi se kreirala platforma otvorenog koda u kojem bi korisnici mogli da kreiraju aplikacije koristeći blokčejn arhitekturu. *Stellar* konsenzus protokol, koji je predložen od strane Davida Mazieresa, prati *Federated Byzantine Fault Tolerance* (FBFT). *Stellar* konsenzus uvodi koncept delimičnog kvoruma (eng. *quorum slice*). Kvorum je skup čvorova koji rade zajedno sa ciljem postizanja konsenzusa, dok je parče kvoruma njegov podskup, koji pomaže čvoru u njegovom procesu saglasnosti. *Stellar* je globalni konsenzus protokol, koji se sastoji iz protokola nominacije i protokola glasačkog listića. Inicijalno se pokreće protokol nominacije. U toku nominacije, nove vrednosti, koje se

nazivaju kandidatske vrednosti se predlažu za saglasnost. Svaki čvor koji dobije te vrednosti će glasati za jednu vrednost od njih. To konačno rezultira u jednoglasno odabranoj vrednosti tog slota. Nakon uspešno izvršenog protokola nominacije, čvorovi pokreću protokol glasačkoj listića. To uključuje ujedinjeno glasanje ili za odustanak od vrednosti dobijenih iz nominacionog protokola ili nastavljanjem i izvršavanjem. To rezultuje eksternalizaciji glasanja za trenutni slot. Prekinuti glasački listići će biti proglašeni za nevažeće. Postoje situacije u kojima čvorovi ne mogu doći do zaključka da li treba da se prihvati ili odustane od date vrednosti. Ta situacija se može izbeći prebacivanjem na glasački listić sa većom vrednosti. To pomaže u slučaju kada čvor misli da je takav zaglavljeni glasački listić izvršen. Time Stellar konsenzus protokol obezbeđuje izbegavanje i upravlja zaglavljenim stanjima, čime se obezbeđuje živost sistema. Stellar protokol tvrdi da je bez blokirajućih stanja, da pruža decentralizovanu kontrolu, asimptotsku sigurnost, fleksibilno poverenje i malo kašnjenje. Ono što Stellar protokol ne garantuje je bezbednost u svakom trenutku. Ukoliko čvor izabere neefikasno kvorum parče, bezbednost se ne može garantovati [91].

3.2. Klasifikacija blokčejn mreža

Ukoliko se vratimo na osnovnu CIA trijadu i pogledamo kroz prizmu poverljivosti, možemo sagledati važnost različitih tipova blokčejn mreža. Informacije koje mogu i treba da budu dostupne svima, mogu se skladištiti na blokčejn mrežama kojima svi imaju pristup, bez obzira da li učestvuju u kreiranju transakcija ili ne. Takav tip blokčejn mreža se naziva javni blokčejn i primeri su Bitkoin i Etereum. Javna blokčejn mreža je u potpunosti otvorena i svako može da joj se pridruži i učestvuje u transkacijama koje se kreiraju. Mreža tipično ima podsticajni mehanizam kojim se ohrabruju učesnici da se priključe mreži. Bitkoin je najveća javna blokčejn mreža i ona privlači korisnike time što za uspešno kreiran blok, daje nagradu u vidu bitkoina. Jedna od mana javnog blokčejna je količina računarske moći koja je potrebna da bi se održao distribuiran ledger na velikom nivou. Tačnije, da bi se postigao koncezus, svaki čvor u mreži mora da reši kompleksan i resursno zahtevan problem, pod nazivom Proof-of-Work, koji je ranije opisan. Pored toga, otvorenost javnih blokčejn mreža implicira da ne postoji privatnost transakcija, ili postoji u jako maloj meri. Ovo je bitno uzeti u razmatranje ukoliko se razmišlja o upotrebi blokčejn tehnologija u velikim sistemima.

Sa druge strane, ukoliko informacije ne smeju biti dostupne svima sa internet pristupom, odnosno aspekt poverljivosti mora biti uzet u obzir prilikom dizajniranja rešenja, kreirane su privatne blokčejn mreže, gde je pristup informacijama koje se skladište na takvoj mreži omogućen isključivo prethodno autentifikovanim i autorizovanim korisnicima. Kako bi se omogućila autentifikacija i autorizacija korisnika, privatne blokčejn mreže gube na osobini koju javne blokčejn mreže zadržavaju, a to je anonimnost. Budući da je potrebno utvrditi indentitet pre pristupa privatnoj mreži kako bi se odobrio ili onemogućio pristup, anonimnost ne može više biti zadržana za privatne blokejn mreže. Privatna blokčejn mreža zahteva poziv i takav učesnik mora biti odobren ili od strane kreatora mreže ili od strane niza pravila koje je

kreator napravio. Kompanije koje postavljaju privatni blokčejn će generalno postavljati mrežu sa dozvolama (eng. *permissioned network*), čime se postavljaju restrikcije ko sme da učestvuje u mreži i u transakcijama. Učesnici moraju imati ili poziv ili permisiju za priključivanje. Mehanizam kontrole pristupa se može razlikovati: postojeći učesnici bi mogli odlučivati o potencijalnim budućim učesnicima ili bi regulatorni organ mogao da izdaje licence za učešće ili bi konzorcijum mogao da donosi te odluke. Kada se učesnik priključi mreži, imaće ulogu u održavanju blokčejna na decentralizovani način. Primer mreže sa dozvolama je Hyperledger Fabric.

Pored podele na javne i privatne blokčejn mreže, autori [92] su predstavili podele blokčejn mreža prema dostupnosti podataka, potrebnom za autorizacijom i podrškom za pametne ugovore. U Tabelama 3 i 4 su prikazani tipovi mreža spram prethodne podele [92], [93], [94].

Tabela 3 Podela blokčejn mreža spram dostupnosti podataka

Dostupnost podataka

Javni	Privatni	Zajednica/Konzorcijum	Hibridni
Svako može da kreira i čita transakcije	Samo jedna organizacija ili njene podružnice u okviru grupe mogu da čitaju i kreiraju transakcije	Više organizacija kreira zajednicu odnosno konzorcijum čime dobijaju pravo čitanja i kreiranje transakcija	Bilo koja prethodno opisana kombinacija može predstavljati hibridni model blokčejn mreže

Tabela 4 Podela blokčejn mreža spram potrebe za autorizacijom

Potreba za autorizacijom

Bez dozvole	Sa dozvolom	Hibridni
Nije potrebna dozvola za učešće u čitanju i kreiranju transakcija. Svi imaju mogućnost pristupa mreži i učestvovanju u verifikacionom procesu, koristeći svoju računarsku moć.	Potrebno je prethodno odobrenje kako bi se pristupilo mreži sa dozvolom. Jedino prethodno autorizovani korisnici mogu učestvovati u verifikaciji transakcija.	Ukoliko se čvor koristi za rad sa blokčejn mrežama i sa i bez dozvole, kako bi omogućio lakšu komunikaciju takvih mreža, on se može smatrati hibridnim.

Tabela 5 Podela blokčejn mreža spram podrške za pametne ugovore

Podrška za pametne ugovore

Pamćenje stanja (eng. *Stateful*)	Bez pamćenja stanja (eng. *Stateless*)
Tip blokčejn mreže koji omogućava korišćenje pametnih ugovora, sa ciljem optimizacije i očuvanja logičkog stanja.	Fokusirano isključivo na optimizovanje transakcija i funkcionalnost lanca, odnosno na verifikaciju transakcija računanjem heš vrednosti. Nezavisno je nivoa sa pametnim ugovorima, čime je zaštićeno od greški i ranjivosti koje pametni ugovori mogu uneti u mrežu.

3.2.1. Bitkoin

Bitkoin predstavlja javni blokčejn korišćen za razmenu novca. Ideja iza najpoznatije kriptovalute, Bitkoina, leži u brzom slanju novca sa jedne strane sveta na drugu. Ono što razlikuje ovu kriptovalutu od običnih valuta koje su svakodnevno u upotrebi, jeste nepostojanje centralne organizacije koja svakom transakcijom dobija određeni procenat. Bitkoin je prvo bio upotrebljen 2009. godine. Bitktoin ima sličnosti sa digikešom (eng. *digicash*), koji je kreirao Dejvid Čaum (eng. *David Chaum*) 1989. godine. Digikeš, koji je zvanično ugašen 1998. godine, je koristio transakcije čiji učesnici su anonimni i koristio je javni i privatni ključ kako bi se očuvala anonimnost. Razlika između digikeša i Bitkoina je u decentralizovanom sistemu koji Bitkoin koristi, čime se izbegava potreba za trećim licem koji potvrđuje transakcije.

Svaka blokčejn mreža se sastoji od učesnika (eng. *node*) koji mogu imati različite uloge. Ukoliko učesnik u mreži želi da vrši transakcije, što u slučaju Bitkoin mreže predstavljati slanje/primanje bitkoin kriptovalute, tada je učesniku dovoljno da ima jedinstvenu adresu koja će pokazivati njegovo stanje Bitkoina. Jedinstvena adresa koju učesnik dobija prilikom prvog ulaska na mrežu se zove novčanik (eng. *wallet*) i tu se nalaze podaci o stanju računa. Novčanik nije vezan za neko lično obeležje učesnika, npr. ime, prezime ili jmbg, već predstavlja niz brojeva i slova (dužina zavisi od mreže koja se koristi, da li je učesnik deo Bitkoin ili Ethereum mreže), čime se korisniku omogućava anonimnost. Ukoliko učesnik primi određenu sumu kriptovaluta, npr. Bitkoina, njegovo stanje će se uvećati za dati broj, dok će se stanje učesnika koji je poslao bitkoine, umanjiti za dati iznos. Jedan od izazova predstavlja proveru da li učesnik koji šalje bitkoine ili neku drugu kriptovalutu, zaista poseduje sumu koju želi da pošalje. Budući da ne postoji centralna organizacija koja bi rukovala ovakvim transakcijama i vršila takve provere, sama provera i potvrda transakcije prebačena je na učesnike mreže koji žele da vrše takve potvrde. Takvi učesnici se zovu rudari (eng. *miners*). Da bi se potvrdila transakcija, potrebno je rešiti određene matematičke probleme, koji zahtevaju dosta jake računare. Iz tog razloga se ne odlučuje svaki učesnik mreže da bude i rudar. Kako rudari koriste

svoje resurse da bi potvrdili transakciju, čime troše i velike količine struje, rudari bivaju nagrađeni za svaku transakciju koju potvrde, u vidu kriptovalute, koja je korišćena u datoj transakciji. Ukoliko se radi o javnoj blockchain mreži, tada je spisak svih transakcija, svih učesnika na mreži, javno dostupan svima, bez obzira da li imaju kreiran novčanik ili ne [95].

Pojam koji se vezuje za Bitkoin jeste i slučajna vrednost (eng. *nonce*), koji u bloku Bitkoina predstavlja 32-bitno (4-byte) polje čija vrednost je tako postavljena da heš bloka uvek počinje sa određenim brojem nula. Sa druge strane, u okviru Ethereum blokčejna, razlikuju se 2 tipa slučajne vrednosti. Prvi je vezan za "račun", gde predstavlja broj transakcija koje su izvršene na tom računu. Drugi nonce je nasumični broj koji se koristi kako bi se zadovolji Proof-of-work, koji će biti objašnjen u nekom od narednih odeljaka.

Kako Bitkoin koristi Proof-of-Work (PoW) mehanizam za kreiranje blokova, potrebno je osvrnuti se i na veliku energetsku potrošnju koju taj mehanizam nosi sa sobom. Kako je prethodno pojašnjeno, PoW mehanizam omogućava kreatoru bloka nagradu za „iskopani" blok, čime se i plaća kreatorova uložena energija. Na samom početku Bitkoin mreže, korisnici su koristili obične računare za kreiranje blokova, ali su ubrzo došli do zaključka da su drugi dostupni hardveri profitabilniji. Tako su nakon upotrebe centralne procesorke jedinice (eng. *Central Processing Unit*, CPU), korisnici prešli na korišćenje grafičke procesne jedinice (eng. *Graphical Processing Unit*), ali i na korišćenje FPGA (eng. *Field-Programmable Gate Array*) i ASIC (eng. *Application-Specific Integrated Circuits*) [96].

3.2.2. Etereum

Poredeći mogućnosti koje je Etereum uveo u odnosu na Bitkoin, kreiranje pametnih ugovora koje izvršava Etereum Virtuelna Mašina (eng. *Ethereum Virtual Machine,* EVM) predstavlja najznačajniju razliku između dve blokčejn mreže. Uvidevši mogućnosti koje blokčejn kao koncept omogućava, Vitalik Buterin napisao je rad [97] u kojem opisuje Etereum mrežu, koju je godinu dana kasnije, 2015. godine i pustio na korišćenje. Novina koja je predložena u odnosu na Bitkoin su pametni ugovori koji omogućavaju pamćenje stanja, čime Etereum mreža postaje po klasifikaciji mreža sa pamćenjem stanja (eng. *stateful*). Time što su omogućeni pametni ugovori, dok Etereum blokčejn mreža postaje Tjuring kompletna, odnosno, omogućena su sva računanja, uključujući i petlje [98].

Pametni ugovor (eng. *smart contract*), pojam koji je uveden sa nastankom Etereum blokčejn mreže, se može posmatrati kao skup pravila i postupaka koji se automatski izvršavaju. Uslov za pokretanje koda koji je definisan pametnim ugovorom jeste da se dogodi odgovarajuća transakcija unutar blokčejn mreže. Ulazi, izlazi i stanja koja su deo pametnih ugovora se moraju izvršiti i potvrditi na svakom čvoru, čime se onemogućava manipulacija vrednostima u pametnim ugovorima i zapisanim vrednostima na blokčejn mreži. Ukoliko analiziramo primer upotrebe pametnih ugovora kod kriptovaluta, ugrađeni pametni ugovor prvo verifikuje transakciju tako što proverava njen potpis. Zatim verifikuje da li iznos na računu

izlazne adrese odgovara ulazu. Na kraju, primenjuje izmene na stanje. Sa druge strane, postoje pametni ugovori koje korisnici mogu sami da kreiraju.

Ukoliko se kreira pametni ugovor na Etereum platformi, postoji lista poznatih napada na koje treba obratiti pažnju prilikom pisanja ugovora, kako ne bi došlo do gubitka ili krađe sredstava. Jedan od identifikovanih napada je ponovni ulaz. U prvoj verziji ovog baga, primećeno je da se neke funkcije mogu zvati beskonačno mnogo puta, pre nego što se završi prvi poziv funkcije. To može dovesti do različitih poziva funkcije koji mogu rezultirati uništavanjem ugovora. Spisak identifikovanih napada se može naći na [99].

3.2.3. Hyperledger projekat

Još jedna od značajnijih blokčejn mreža koja se spominje u ovoj disertaciji je i Hyperledger Fabric, privatna blokčejn mreža okrenuta ka organizacijama koje se bave različitim aktivnostima u okviru određene industrije, koje žele da informacije na mreži ostanu zaštićene i dostupne samo određenoj grupi korisnika. Međutim, Hyperledger Fabric je samo jedno od rešenja koje je nastalo u okviru Hyperledger projekta, pokrenutog od strane Linuks fondacije, ali sa velikim doprinosom IBM. Rešenja su podeljena u četiri kategorije, distribuirane knjige, biblioteke, alati i domenski specifični projekti. Hyperledger Besu projekat je kreiran za preduzeća koja žele kako javne mreže, tako i privatne mreže sa dozvolom, sa implementiranom Etereum virtuelnom mašinom (eng. *Ethereum Virtual Machine*, EVM). Što se tiče Hyperledger Indy, taj projekat je kreiran sa ciljem pružavanja alata i biblioteka za kreiranje digitalnih identiteta koji su u osnovi vezani za blokčejn, na način da poseduju interoperabilnost u okviru administrativnih domena. Hyperledger Iroha je kreiran kako bi se olakšala integracija infrastrukturnih ili IoT (eng. *Internet of Things*) rešenja sa distribuiranim knjigama. Hyperledger Sawtooth je projekat koji omogućava odvajanje osnovnog sistema od aplikativnog domena, čime se mogu pisati pametni ugovori bez potrebe da se poznaju detalji osnovnog sistema. Kada je u pitanju Hyperledger Fabric projekat, on predstavlja osnovu za kreiranje aplikacija ili rešenja sa modularnom arhitekturom, gde će se detalji o samoj implementaciji pojasniti u narednom odeljku [100].

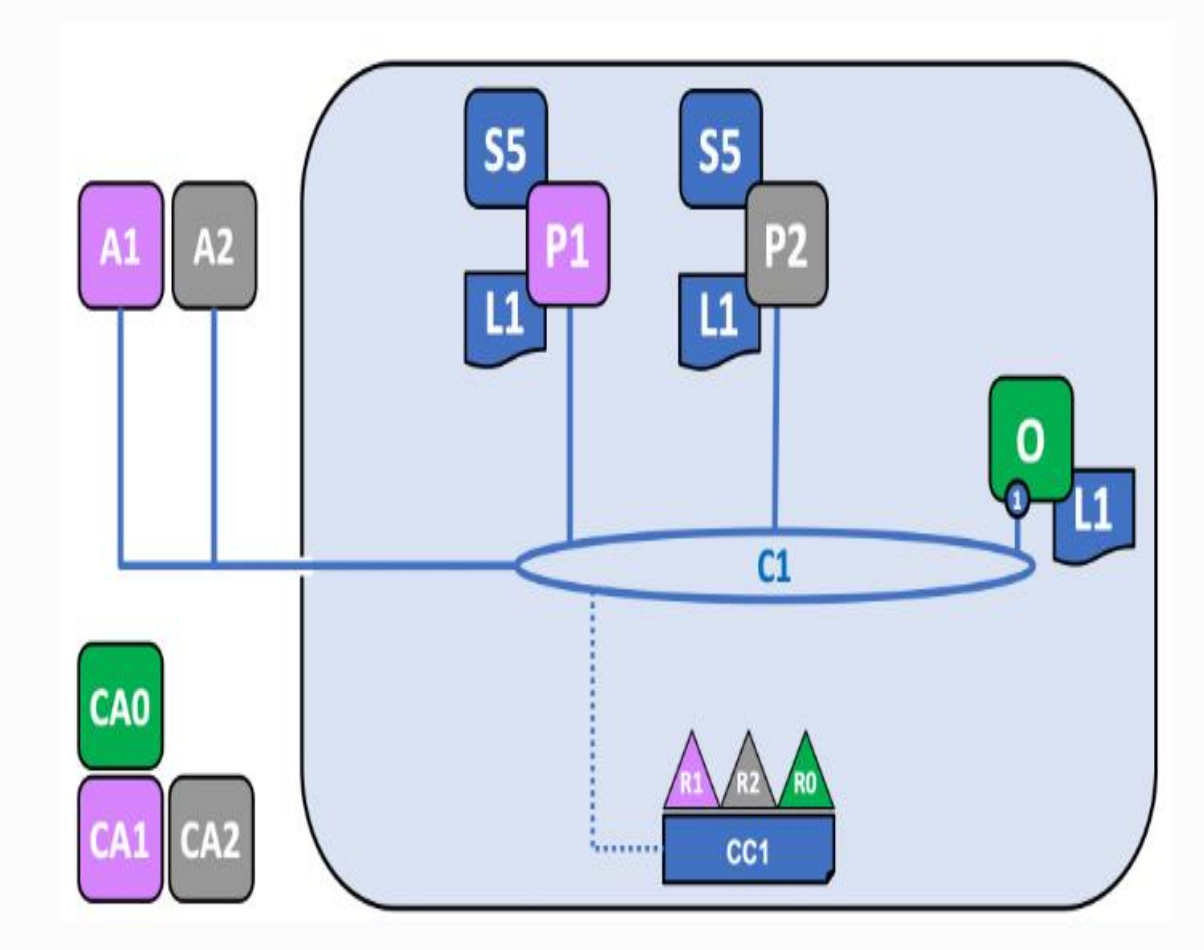
3.2.3.1. Hyperledger Fabric

Hyperledger Fabric je blokčejn implementacija koja podrazumeva mrežu koja je privatna i sa dozvolom, što znači da samo prethodno odobreni korisnici imaju mogućnost učestvovanja u kreiranju transakcija, čitanju i pisanju podataka u lanac podataka. Kao deo Hyperledger projekta, Hyperledger Fabric je omogućio korporacijama da koriste prednosti koje blokčejn mreža pruža, kao što su neporecivost, integritet, fleksibilnost, ali dodajući i osobinu poverljivosti, koju javne blokčejn mreže ne pružaju. Principi koji su ranije pomenuti, a tiču se kriptografije, pametnih ugovora, učesnika u mreži itd. se mogu uočiti i u okviru Hyperledger Fabric implementacije blokčejn mreže, ali je Hyperledger Fabric uneo i druge pojmove i

koncepte koje je potrebno obraditi kako bi se u potpunosti razumele prednosti Hyperledger Fabric rešenja koja su iskorišćena u okviru ove disertacije. Kratka pojašnjenja uvedenih pojmova i koncepata navedena su niže, dok će određeni pojmovi biti detaljnije pojašnjeni:

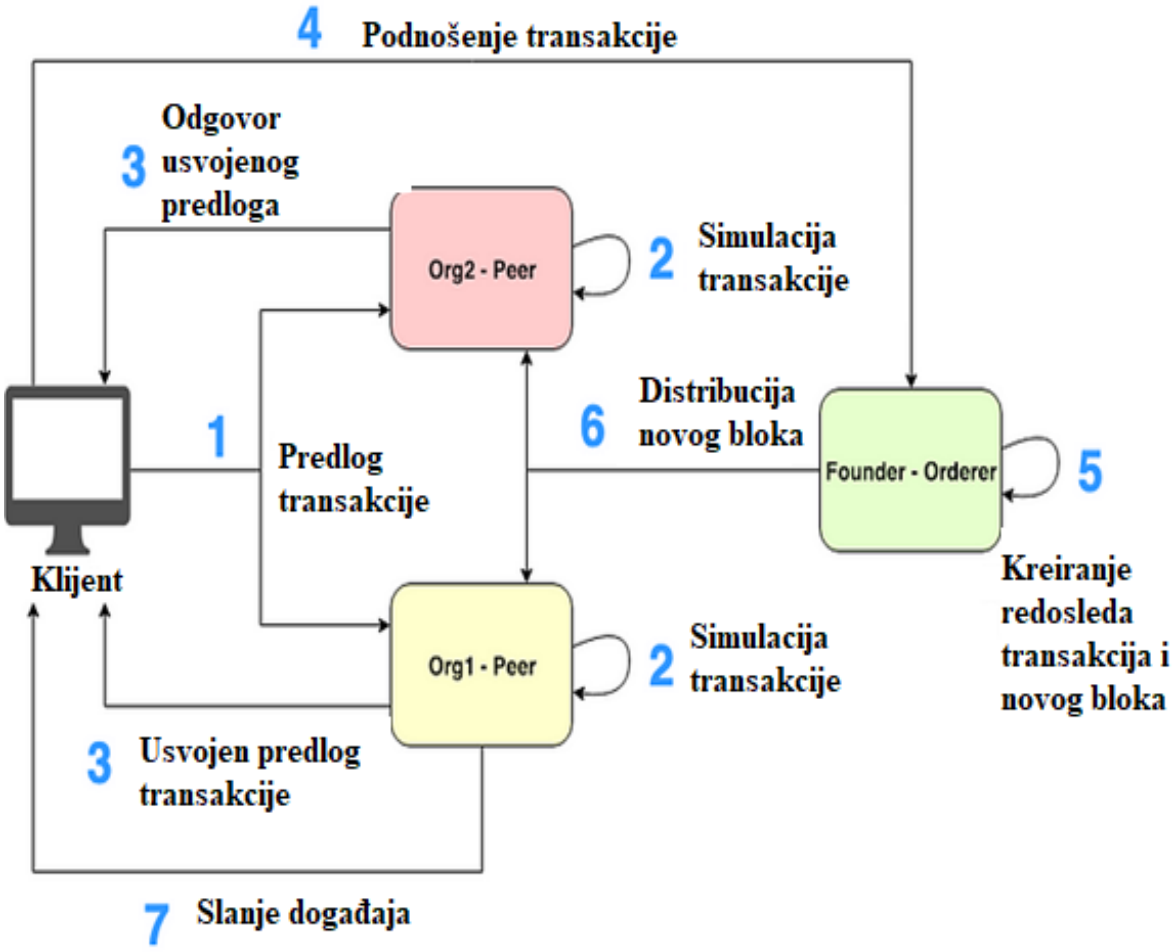
- Programski kod lanca (eng. *Chaincode*): predstavlja oblikovan pametni ugovor, distribuiran čvorovima u mreži i spreman za izvršavanje.
- Kanal (eng. *Channel*): uvidevši mogućnost za dodatno razdvajanje unutar jednog projekta, Hyperledger Fabric uvodi koncept kanala, koji predstavlja mehanizam za kreiranje nove privatne blokčejn mreže, unutar postojeće. Na taj način se omogućuje dodatna segregacija korisnika, pružajući izolaciju podataka i poverljivost za organizacije i aplikacije.
- Čvor (eng. *Peer*): entitet mreže koji je zadužen za održavanje mreže i na kojem se izvršavaju programski kodovi lanaca, kako bi se izvršile operacije čitanja i pisanja.
- Organizacija (eng. *Organization*): predstavljaju skup učesnika koji su pozvani da se priključe blokčejn mreži.
- Konzorcijum (eng. *Consortium*): skup organizacija koje mogu da kreiraju kanale i imaju svoje čvorove.
- Uređivač (eng. *Orderer* ili *Ordering service*): jedan ili više čvorova koji su zaduženi za pripremanje transakcija u blokove i njihovo distribuiranje kako bi se transakcije validirale i potvrdile (eng. *commit*). [101], [102], [103]

Nakon uvođenja osnovnih pojmova koji definišu jedno Hyperledger rešenje, možemo sagledati kako bi izgledala jednostavna mreža, a zatim i tok informacija. Na slici 9. prikazana je jednostavna blokčejn mreža, koja sadrži jedan kanal, na koji su povezani čvorovi **P1** i **P2**, zajedno sa uređivačem **O**. Kanal ima svoju konfiguraciju označenu sa **CC1**, koja, između ostalog, definiše prava za organizacije označene sa **R1**, **R2** i **R0**. Kako je blokčejn distribuirana i decentralizovana mreža, svaki čvor sadrži i poslednje stanje knjige (eng. *ledger*) koji je na slici označeno sa L1. Pristup kanalu i samoj knjizi omogućen je kroz aplikacije **A1** i **A2**, pri čemu su one odvojeno kreirane za čvorove i organizacije **P1** i **R1**, odnosno P2 i R2, respektivno. Hyperledger Fabric je privatna mreža sa dozvolom i zbog toga je potrebno kreirati sertifikacioni autoritet (eng. *Certificate Authority*) za svaku organizaciju i to je označeno sa **CA0**, **CA1** i **CA2** [104].



Slika 5 Jednostavna predstava arhitekture blokčejn mreže

Pored arhitekture prikazane na Slici 5, dodatni pogled kojim se posmatra put informacija od kada je korisnik putem aplikacije unese, do momenta kada se trajno skladišti u *ledger* su autori u [105] definisali kao *izvrši-poređaj-validiraj,* što principijelno odgovara trenutno dostupnoj dokumentaciji [106], koja te korake razlaže na dodatne radi lakšeg pojašnjenja. Čitav proces prikazan je na Slici 6 [107] i započinje slanjem zahteva od strane klijentske aplikacije. Svaki čvor nezavisno validira predlog transakcije, nakon čega se odgovor vraća klijentskoj aplikaciji. Odgovor se digitalno potpisuje i prosleđuje Uređivaču, kako bi se jednoznačno odredio redosled transakcije. Čvorovi mogu zatražiti informacije od Uređivača o novim blokovima ili se pretplatiti na izmene, koje se u tom slučaju šalju bez prethodnog traženja. Takođe, klijentska aplikacija može dobijati informacija od čvorova ukoliko je transakcija prihvaćena i skladištena na blokčejn.



Slika 6 Redosled transakcija [107]

3.3. Bezbednost blokčejn mreža

Informacije koje industrijski upravljački sistemi treba da čuvaju na blokčejn mreži mogu biti klasifikovane kao poverljive, budući da sadrže informacije o funkcionisanju sistema i načinu na koji se sam sistem štiti od napadača, potrebno je razmotriti bezbednost samih blokčejn mreža. Jedan od načina sagledavanja jeste i da se pogledaju prethodni napadi na blokčejn mreže i da se analiziraju greške koje su omogućile takve napade. Neki od najznačajnijih napada, posmatrajući kroz prizmu finansijskih gubitaka, navedeni su u okviru ovog odeljka.

3.3.1. Eclipse napad

Y.Marcus, E.Heilman i S.Goldberg u [108] prezentuju Eclipse napad na Ethereum čvorove koji eksploatišu peer-to-peer mrežu koja se koristi za otkrivanje komšija. Napad koji su opisali se može pokrenuti koristeći samo dva domaćina (host), gde svaki ima jednu IP adresu. Napadač monopolizuje sve dolazne i odlazne konekcije žrtava, čime se žrtva izoluje od ostatka čvorova u mreži. Napadač tada može da filtrira žrtvin pogled na blokčejn ili da preuzme žrtvine

računarske resurse, što bi predstavljalo sofisticiraniji napad. Ovaj rad je izlaz iz uspešnog unapređenja Ethereum-ovog alata za pristup mreži, koji je prihvaćen u februaru 2018.godine. Autori smatraju da je takav propust nastao usled Kademlia protokola koji je Ethereum bio implementirao. Takođe, u okviru rada predstavljaju kontramere za jačanje mreže protiv takvih napada [108].

3.3.2. The DAO napad

The DAO (Decentralized Autonomous Organization) je kompleksni pametni ugovor koji je imao mnoge mogućnosti i trebao je da omogući stvaranje organizacije putem pametnog ugovora. Taj pametni ugovor je takođe imao i grešku koja je iskorišćena za krađu Ether kriptovalute koju su posedovali korisnici mreže. Funkcija podele (eng. *split function*) je omogućila korisnicima da otkažu započetu transakciju i da povrate svoj novac. U junu 2016. godine, članovi Ethereum zajednice su primetili da sredstva sa njihovog *The DAO* pametnog ugovora nestaju i da se ukupna vrednost iznosa pametnog ugovora smanjuje. Ukupno 3,6 miliona Ether-a, što je u tom trenutku predstavljalo oko 70 miliona dolara, je skinuto sa računa u prvih nekoliko sati. Napad se desio usled greške koja je postojala u funkciji podele. Napadač(i) je(su) povlačili Ether sa *The DAO* pametnog ugovora više puta, koristeći iste DAO tokene, potražujući od pametnog ugovora Ether više puta, pre nego što pametni ugovor uspe da izmeni svoje stanje. Problem je bio što kreatori pametnog ugovora nisu uzeli u obzir rekurzivni poziv, kao i što su se unutar pametnog ugovora prvo slala sredstva, a tek nakon slanja se menjalo stanje [109].

3.3.2. Sybil napad

Kao jedan od mogućih napada na blokčejn mrežu identifikovan je *Sybil* napad koji može nastati kada se u velikim *peer-to-peer* sistemima, jedan maliciozni entitet predstavi sa više identiteta, što bi mu omogućilo da preuzme kontrolu nad značajnim delom sistema, potkopavajući redundantnost. Bitkoin prevazilazi ovaj problem tiče sto koristi PoW konsenzus model, u kojem je potrebno da se svaka transakcija verifikuje izračunavanjem određenog matematičkog problema, čime rudari pokazuju da nisu virtuelni entiteti [110].

3.3.3. Parity napad

U julu 2017.godine, dogodio se napad na Ethereum mrežu u kojem je ukradeno oko 150000 Ether-a, što je u tom trenutku vredelo oko 30 miliona dolara. U distribuiranim ugovorima, sistem se štiti od krađe tako što zahteva višestruke nezavisne strane da potpišu transakcije pre nego što se ona može smatrati validnom. To se postiže transakcijom sa više digitalnih potpisa (multi-signature) gde minimalno m od n ključeva, gde su m i n unapred definisani, mora potpisati transakciju, kako bi se tokeni mogli potrošiti [111]. U Parity napadu, napadač je poslao dve transakcije svakom od izabranih pametnih ugovora: prvi je bio da se dobije

ekskluzivno pravo na virtuelni novčanik, a druga je bila za prebacivanje svih sredstava. Funkcija koja je to omogućavala je verovatno kreirana kako bi se mogla izvući logika kreatora novčanika u posebnu biblioteku. To je dovelo do toga da sve funkcije budu javne, odnosno svako je mogao da ih pozove, uključujući i funkciju pod nazivom *initWallet*, koja je mogla da promeni vlasnika pametnog ugovora. Nažalost, unutar same funkcije nije postojala provera koja bi sprečila ponovno pozivanje funkcije, ukoliko je pametni ugovor već kreiran. Napadač je to uočio i jednostavno promenio stanje varijable pametnog ugovora, tako da odgovara adresama novčanika koje on poseduje, čime on postaje jedini odgovorni za prihvatanje transakcije [112].

4. Zahtevi za bezbedan razvoj softvera

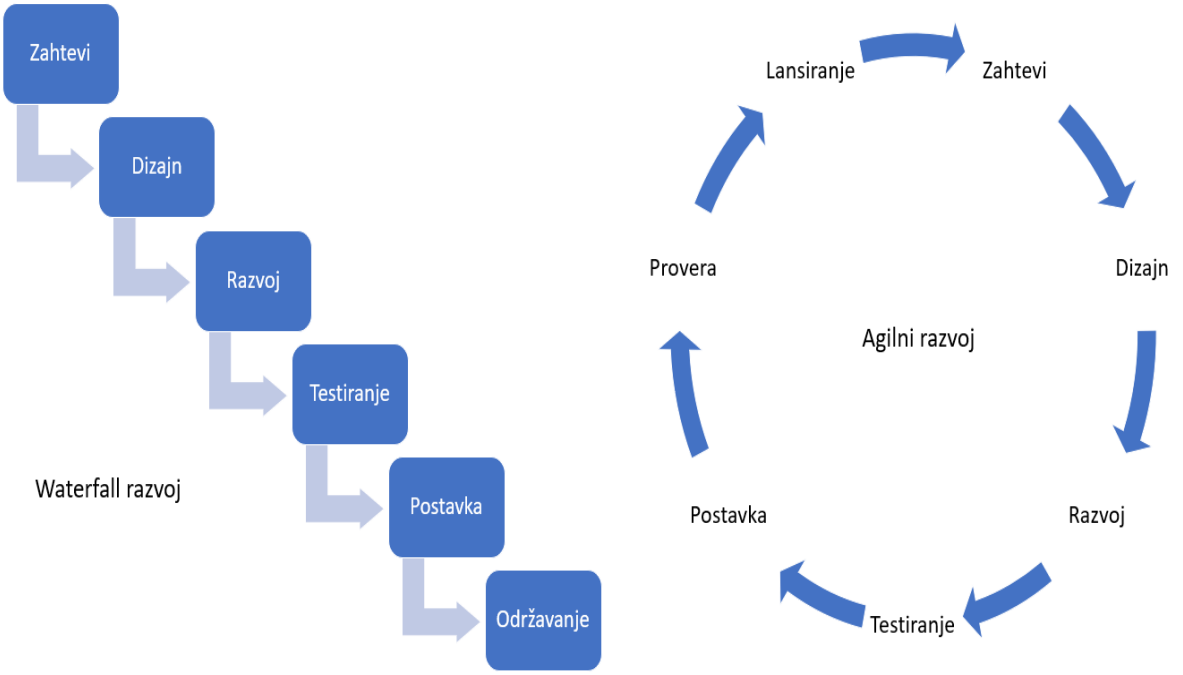
Sagledavši u prethodnom poglavlju osnovne delove blokčejn mreža, možemo uvideti benefite u različitim oblastima koje ta tehnologija donosi. Jedna od značajnijih karakteristika blokčejn mreže jeste da se jednom upisani podaci na blokčejn mrežu ne mogu obrisati. Treba imati na umu da podaci mogu promeniti svoje vrednosti, odnosno može doći do njihove izmene, ali istorija svih unetih informacija na jednu blokčejn mrežu se ne može obrisati. Svakako, ovakva osobina može predstavljati problem u različitim sferama bezbednosti, kao što je privatnost, budući da regulativa za zaštitu ličnih podataka (eng. *General Data Privacy Regulation*, GDPR) podrazumeva i mogućnost trajnog brisanja ličnih podatka podataka (eng. *right to be forgotten*). Sa druge strane, takva karakteristika pomaže u sistemima gde je potrebno sačuvati istoriju svih promena sa ciljem dokazivanja implementiranih kontrola.

U narednim odeljcima će biti analizirani standardi i smernice za bezbedan razvoj softvera. Standardi i smernice su napisani u obliku zahteva, opisujući kontrole koje je potrebno implementirati. Nakon implementacije samih kontrola, ukoliko je traženo zakonom, regulativom ili donosi prednost u datoj industriji, kontrole prolaze kroz verifikaciju od strane sertifikovanih revizora. Posmatrajući procese i procedure koje se primenjuju prilikom implementacije standarda, radi bolje organizacije samog posla, potrebno je jasno definisati role koje su uključene u procese i definisati njihova zaduženja.

4.1. Metodologije za razvoj softvera

Za potrebe razvoja softvera, za industrijske upravljačke sisteme ili bilo kog drugog sistema, neophodno je oformiti tim koji će raditi na definisanju zahteva, razvoju i održavanju kreiranog softvera. U zavisnosti od izabrane metodologije razvoja softvera, razlikuju se uloge, broj članova u timu, brzina realizovanja softvera i učestalost kreiranja novih mogućnosti softvera. Ukoliko uporedimo dve metodologije kao što su vodopad pristup (eng. *Waterfall*) i jednu od agilnih metodologija kao što je Skram (eng. *Scrum*), možemo uvideti te razlike. Vodopad pristup tipično podrazumeva četiri uloge, programer (eng. *developer*), tester (eng. *tester*), poslovni analitičar (eng. *business analyst*) i rukovodilac projekta (eng. *project manager*) [113]. Iako su definisane samo četiri uloge, timovi koji prate vodopad metodologiju mogu imati veći broj članova, gde više članova ima istu ulogu, kao što su programer ili tester. Sa druge strane, agilna metodologija kao što je Skram, omogućava veći broj uloga, dok su programer i tester dve zajedničke uloge sa vodopad pristupom. Dodatno, Skram omogućava definisanje uloge za vlasnika proizvoda (eng. *product owner*), skram master (eng. *scrum master*), vođa tima (eng. *team leader*), arhitekta (eng. *architect*), tehnički ili domenski ekspert (eng. *technical and domain expert*), DevOps (eng, *DevOps*), interfejs dizajner (eng. *UX designer*) [114]. Pored različitih uloga, dve metodologije se razlikuju i u fazama kroz koje softver prolazi, kao i da li postoji mogućnost povratka na prethodnu fazu. Prilikom razvoja softvera po vodopad principu,

faze se prolaze sekvencijalno, bez mogućnosti povratka na prethodnu fazu, kod agilne metodologije, kao što je Skram, omogućavaju povratak na prethodnu fazu, tačnije, sve faze se ponavljaju ciklično. Slikovit prikaz različitih faza razvoja prikazan je na Slici 7.



Slika 7 Razlike između Waterfall i Agilnog razvoja

Bez obzira na izabranu metodologiju razvoja proizvoda, timovi koji razvijaju softver u industrijskim upravljačkim sistemima i moraju da prate zahteve za bezbedan razvoj softvera, susreću se sa dodatnim zahtevima koje je neophodno pratiti i zadovoljiti. Kako usklađenost sa bezbednosnim zahtevima može predstavljati zakonsku ili regulativnu obavezu, očuvanje dokaza za usklađenost može predstavljati izazov za timove. U nastavku su diskutovane metodologije koje su usmerene na bezbedan razvoj softvera.

4.1.1. Standardi i prakse za bezbedan razvoj softvera

Podaci koji su dobijeni od Savezne Trgovinske Komisije (eng. *Federal Trade Commision*, F.T.C) i Federalnog Istražnog Biroa (eng. *Federal Biro of Investigation*, FBI), prikazuju gotovo eksponencijalni rast gubitaka koji su nastali kao posledica sajber napada u periodu od 2010. do 2021. godine [115]. Opisane vrste napada se mogu grupisati u prevare, krađu identiteta i ostale vrste napada, gde se broj povećao 290,21% u datom periodu. Sajber napada nisu usmerini isključivo na individue, banke ili specifične industrije, već se može posmatrati da sve što je "povezano na internet" je ranjivo [116]. Kako bi se unapredili sistemi, kompanije i ljudi u pogledu bezbednosti, Evropska Komisija je za period 2023-2024. godina pripremila budžet od 375 miliona evra, dok su Sjedinjene Američke Države pripremile budžet veći od 10 milijardi dolara [117].

Regulative, standardi i stručne smernice predstavljaju izvor zahteva koji su neophodni da se implementariju, budući da neusklađenosti sa istim mogu dovesti do velikih novčanih kazni. Primera radi, poslednje statistike vezane za naplatu kazni zbog kršenja Zaštite podataka o ličnosti (eng. *General Data Protection Regulation*, GDPR) pokazuju da je ukupan iznos kazni od 25. maja 2018.godine, odnosno datuma stupanja na snagu GDPR regulative, blizu 3 milijarde evra, a da je broj naplaćenih kazni 1574 [118]. Po broju plaćenih kazni, Španija zauzima prvo mesto sa plaćenih 618 kazni, dok je Republika Irska sumarno platila najveći iznos [118]. Dok je GDPR regulativa koja je usmerena na privatnost korisnika, postoje i standardi koji pokrivaju i druge oblasti bezbednosti.

Sa druge strane, standardi predstavljaju skup preporuka i smernica, kreiranih specifično za industriju u kojoj ih je potrebno primeniti. Kada je reč o standardima iz oblasti bezbednosti, cilj standarda je da se poboljšaju zaštite digitalnih sistema, čime će se sprečiti, ili makar ublažiti štete evenutalnih sajber napada. Standardi pružaju informacije o bezbednosnim kontrolama, načinu implementacije bezbednosnih kontrola, kao i metodologijama praćenja i unapređenja postojećih kontrola. Kontrole i smernice koje se prikazuju u standardima mogu biti definisane na različitim nivoima detalja. Takođe, postoje razlike u oblasti primene bezbednosnih standarda. Ukoliko uzmemo primer HIPPA (eng. *Health and Insurance Privacy Protection Agreement*), standard je usmeren isključivo na zdravsto i usluge osiguranja u zdravstvu, sa ciljem zaštite poverljivosti podataka vezanih za lične zdravstvene informacije. Slično HIPPA standardu, PCI DSS standard (eng. *Payment Card Industry Data Security Standard*) se vodi kao bezbednosni standard, ali je usmeren isključivo na bankarski sektor i načinu zaštite podataka vezanih za procesiranje, čuvanje i druge upotrebe platnih kartica. Kako bi se pokazala usklađenost sa određenim standardima, koji su u nekim slučajevima i obavezni zakonima, neophodno je proći kroz proces sertifikacije. Sam proces sertifikacije predstavlja način da se javno prikaže usklađenost sa standardima, koje mogu da izdaju sertifikacione kompanije.

U nastavku će biti prikazani standardi i stručne smernice koje se mogu koristiti prilikom razvoja softvera, budući da zahtevi unutar standarda i stručnih smernica, usmeravaju korisnike na implementaciju bezbednosti od samog početka razvoja softvera.

4.1.1.1. Microsoft SDL process

Gledajući hronološki, SDL (eng. S*ecurity Development Lifecycle*) proces predstavlja jedan od prvih sistematičnih izvora zahteva vezanih za bezbedan razvoj softvera, kreiran 2002. godine od strane Majkrosoft kompanije [119]. Tokom godina, Majkrosoftov SDL proces je evoluirao, prateći razvoj tehnologija, dostupnih alata, kao i sve veće potrebe da bezbednost postane sastavni deo razvojnog procesa. Od 2004. godine, SDL proces predstavlja integralni deo razvojnog procesa u Majkrosoft kompaniji [120]. U trenutnom obliku, Majkrosoftov SDL proces se sastoji iz 12 praksi, koje opisuju celine koje razvojni timovi treba da prate [121]. Ideja svake od praksi jeste da se ukaže razvojnim timovima na dobre bezbednosne prakse koje su neophodne u svakoj fazi životnog ciklusa razvoja softvera. Kroz prvu praksu [121], timovi

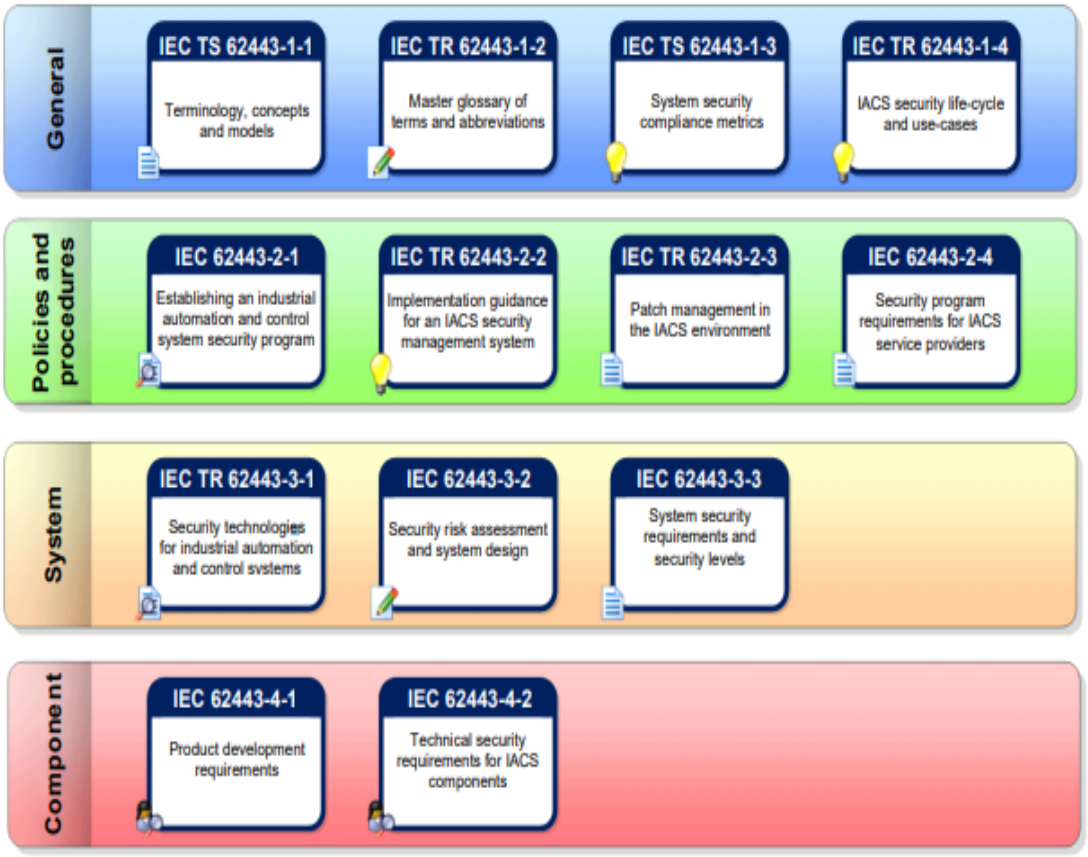
su u obavezi da prođu kroz predefinisane treninge, sa ciljem podizanja znanja i svesnosti kada je u pitanju bezbedan razvoj softvera. Nakon toga, u okviru prakse za definisanje bezbednosnih zahteva, neophodno je identifikovati zahteve koji sadrže bezbednosne aspekte, kao i aspekte privatnosti podataka. Budući da zahtevi zavise od industrije u kojoj se softver razvoja, jedan od izvora zahteva svakako predstavljaju standardi i regulative te industrije. Sledeća praksa govori o definisanju metrika i načinu izveštavanja usklađenosti, sa ciljem praćenja kompletnosti dogovorenih ciljeva. Nakon dogovorenih metrika, neophodno je napraviti model pretnji, sa ciljem identifikovanja pretnji pre samog početka razvoja, nakon čega se prati praksa za utvrđivanje zahteva za dizajn. Kako je preporuka industrija da se koriste isključivo kriptografski algoritmi koji su prethodno odobreni i testirani, u okviru naredne prakse neophodno je definisati i upotrebljavati odobrene kriptografske standarde. Kako softveri mogu biti sačinjeni i od komponenti koje su razvijane od strane drugih kompanija, kroz sledeću praksu neophodno je upravljati rizicima koje donose komponente koje su razvijane od strane drugih kompanija. Sličnu temu pokriva i naredna praksa, u okviru koje je neophodno koristiti prethodno odobrene alate za razvoj. Naredne dve prakse se odnose na statičko i dinamičko (eng. *Static Analysis Security Testing* (SAST) i *Dynamic Analysis Securiry Testing* (DAST)) testiranje softvera, čime se povećava verovatnoća detektovanja problema u izvornom kodu softvera. Sledeća praksa (pod brojem 11) takođe vezana za testiranje, ali je u ovom slučaju neophodno angažovanje specijalizovanog tima za penetraciono testiranje. Na samom kraju, kroz praksu za uspostavljanje procesa za rukovanje incidentima, kreiraju se planovi koji se mogu iskoristiti u slučaju napada, sa ciljem brzog oporavka sistema i povratka u normalan režim rada [121].

4.1.1.2. CLASP proces

Dok Majkrosoftov SDL proces predstavlja okosnicu i za veliki broj timova u današnje vreme (april 2023.godine), CLASP (eng. *Comprehensive, Lightweight Application Security Process*) proces predstavlja metodologiju koja je nastala u slično vreme kao Majkrosoftov SDL proces, ali nije zaživela. Cilj Majkrosoftovog SDL procesa i CLASP je veoma sličan, gde krajnji proizvod odnosno softver treba da minimizuje rizik od ekploatisanja ranjivosti, primenom dobrih praksi i smernica koje su definisane u okviru procesa. Pored aktivnosti koje CLASP definiše, koja se u nekoj meri poklapaju sa Majkrosoftovim SDL procesom, CLASP definiše i uloge u čitavom procesu. Uloge su podeljene na rukovodioca projekta, uloge zadužene za definisanje zahteva, arhitekte, dizajnera, implementatora, test analitičara i bezbednosnog auditora [122]. Poredeći CLASP i Majkrosoftov SDL proces 2009. godine, u preko 150 oblasti koje pokrivaju procesi, oko 10% je postojalo poklapanje u aktivnostima. U momentu poređenja dva procesa, autori [119] CLASP proces ocenili kao bogatiji po pitanju informacija vezanih za način implementacije predloženih aktivnosti. Kako CLASP proces nije nastavio da se razvija na način na koji je Majkrosoftov SDL proces evoluirao, CLASP se retko ili gotovo uopšte ne vidi u implementaciji bezbednog razvoja softvera.

4.1.1.3. IEC 62443-4-1 standard

Međunarodna elektrotehnička komisija (eng. *International Electrotechnical Commission*, IEC) je međunarodna organizacija osnovana 1906. godine sa ciljem publikovanja internacionalnih standarda iz oblasti električnih, elektronskih i sličnih tehnologija [123]. Kao organizacija, IEC je publikovao preko 11 hiljada standarda, zaključno sa krajem 2022. godine [124]. Kada su u pitanju bezbednost i razvoj softvera za industrijske upravljačke sisteme, skup standarda pod nazivom IEC 62443 se izdvaja kao sveprisutan. Na Slici 8 prikazan je spisak svih pojedinačnih standarda koji se nalaze pod okriljem IEC 62443 grupe [125]. U okviru IEC 62443 grupe standarda, definisane su podgrupe. Tako u podgrupi za opšte standarde, postoje 4 standarda, pod oznakama IEC 62443-1-1 do IEC 62443-1-4. U okviru sledeće podgrupe nalaze se četiri standarda koji su vezani za polise i procedure, označeni sa IEC 62443-2-1 do IEC 62443-2-4. U trećoj podgrupi, nalaze se standardi vezani za podizanje nivoa bezbednosti na nivou sistema i označeni su od IEC 62443-3-1 do IEC 62443-3-3. U poslednjoj podgrupi, nalaze se dva standarda koji su usmereni na bezbednost komponenti i označeni su oznakama IEC 62443-4-1 i IEC 62443-4-2.



Slika 8 Skup IEC 62443 standarda [125]

U okviru poslednje podgrupe, izdvaja se standard IEC 62443-4-1 pod nazivom Bezbednost sistema industrijske automatizacije i upravljanja – Deo 4-1: Zahtevi za bezbedan razvoj proizvoda (eng. *Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements*). Standard je publikovan u januaru 2018. godine i očekivana validnost standarda je postavljena na 2024. godinu [126], čime se obezbeđuje period u kojem zahtevi ostaju nepromenjeni. Cilj standarda je da pruži procesne zahteve za bezbedan razvoj softvera u okviru industrijskih upravljačkih sistema [125]. Standard IEC 62443-4-1 je podeljen u 8 celina – tzv. praksi, koje služe da se grupišu zahtevi po temama. Ukupan broj zahteva je 47, dok broj zahteva u okviru prakse varira od 2 do 13. Same prakse su definisane na taj način da se pokrije svaka faza životnog ciklusa razvoja softvera. Prakse su označene skraćenicom, nakon čega sledi broj zahteva u okviru prakse. Prakse u okviru standarda su:

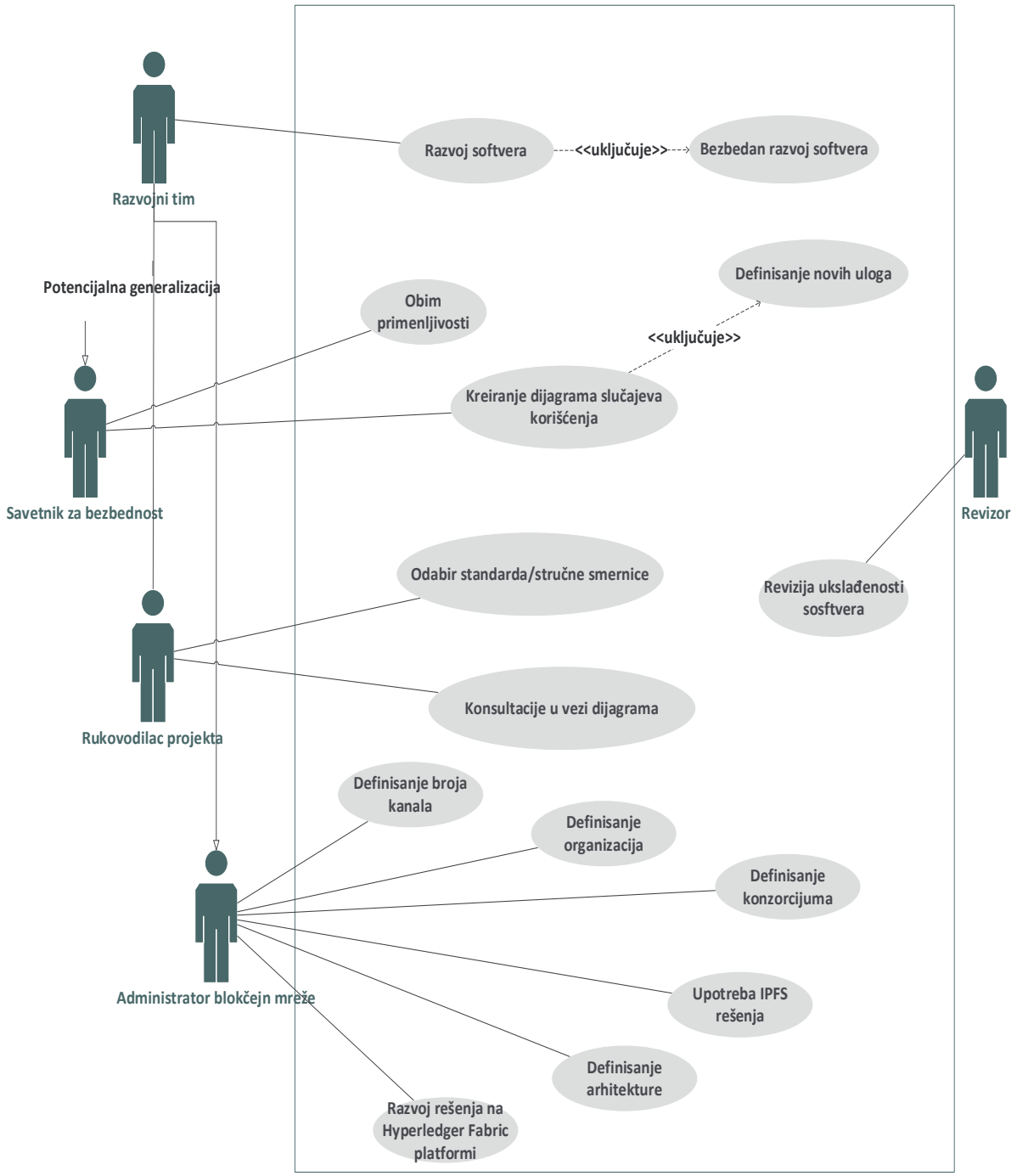
- Upravljanje bezbednošću (eng. *Security Management*)
- Specifikacija bezbednosnih zahteva (eng. *Specification of security requirements*)
- Bezbednost po dizajnu (eng. *Secure by design*)
- Implementacija bezbednosti (eng. *Secure implementation*)
- Bezbednosno verifikaciono i validaciono testiranje (eng. *Security verification and validation testing*)
- Upravljanje problemima vezanih za bezbednost (eng. *Management of security-related issues*)
- Upravljanje bezbednosnim zakrpama (eng. *Security update management*)
- Bezbednosne smernice (eng. *Security guidelines*).

Budući da su zahtevi procesne prirode (definišu procedure i prakse koje timovi moraju da prate tokom razvoja softvera, osiguravajući doslednost, kvalitet i efikasnost tokom procesa), samim zahtevima nije dodeljen bezbednosni nivo (eng. security level) kao što je slučaj sa zahtevima u okviru IEC 62443-3-3 standarda, već se za nivo usklađenosti koriste nivoi slični CMMI-DEV modelu [127]. Po CMMI-DEV (eng. *Capability Maturity Model Integration for Developemt*) modelu, postoje pet faza, odnosno pet nivoa zrelosti, u kojima se procesi mogu naći, od inicijalnog, koji predstavlja napore pojedinaca da se poštuju zahtevi, do poslednjeg, nazvanog optimizacija, u kojoj organizacija koje prate procese mogu da na osnovu jasno definisanih metrika, unapređuju svoje procese [127]. U okviru IEC 62443-4-1 standarda postoje 4 nivoa zrelosti, gde se prva 3 nivoa poklapaju sa CMMI-DEV modelom, dok je poslednji IEC 62443-4-1 nivou napravljen od poslednja dva CMMI-DEV nivoa. Takođe, opisi samih nivoa usklađenosti, definisani zasebno za IEC 62443-4-1 standard.

4.2. Model za praćenje zahteva za bezbedan razvoj softvera

Sa osvrtom na prethodno opisane metodologije razvoja softvera i standarde i prakse za bezbedan razvoj softvera, možemo uočiti nedostatke sa kojima se timovi koji razvijaju po vodopad ili agilnoj metodologiji mogu suočiti kada je neophodno uvesti i zahteve za bezbedan

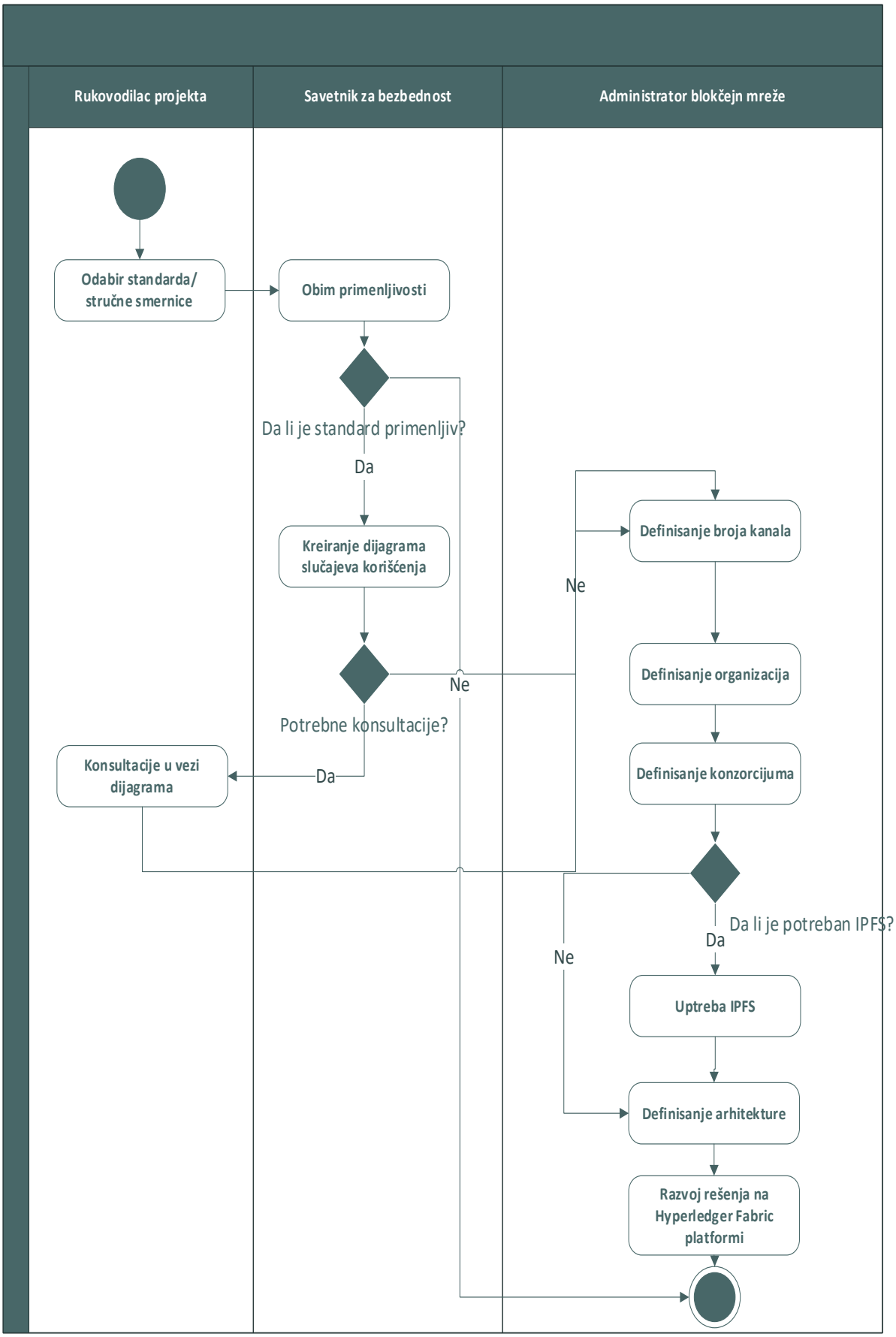
razvoj softvera. Ugledajući se na CLASP proces, timovi mogu angažovati osobu čija uloga će biti usmerena na bezbednost i time prevazići eventualni nedostatak u znanju potrebnom za bezbedan razvoj softvera. Pored uloge *savetnika za bezbednost*, čija zaduženja bi bila usmerena na tehničke i procesne aspekte bezbednosti, praćenje usklađenosti softvera sa zahtevima za bezbedan razvoj softvera se može pripisati ulozi *rukovodioca projekta*. *Rukovodilac projekta* bi obezbedio neophodne uslove za ispunjenje zahteva, pratio napredak usklađenosti zahteva, sve sa ciljem obezbeđivanja potvrde usklađenosti zahteva, koja može biti dobijena unutar razvojnog tima ili van tima. Ulogu koja bi radila potvrdu usklađenosti softvera sa zahtevima za bezbedan razvoj softvera preuzima *revizor*, koji može biti uloga izvan razvojnog tima, čime bi se obezbedila nepristrasna verifikacija. Ukoliko su *revizori* nisu deo razvojnog tima, neophodno je dostaviti pokazatelje usklađenosti koji ne narušavaju poverljivost podataka. Pored poverljivosti podataka, pokazatelji usklađenosti *revizorima* mogu dokazati sledljivost. Ukoliko želimo da obezbedimo poverljivost i sledljivost dokaza, možemo uvesti ulogu *administratora blokčejn mreže*, koji bi omogućio platformu za praćenja usklađenosti softvera sa zahtevima za bezbedan razvoj softvera. Na Slici 9, prikazan je dijagram slučajeva korišćenja, koji obuhvata prethodno pomenute uloge.



Slika 9 Dijagram slučajeva korišćenja za praćenje zahteva za bezbedan razvoj softvera

Na Slici 9 su prikazane različite uloge, kao što su *razvojni tim*, *revizor*, *savetnik za bezbednost*, *rukovodilac projekta* i *administrator blokčejn mreže*. Učesnik *razvojni tim*, je zadužen za razvoj softvera, što u slučaju razvoja softvera za industrijske upravljačke sisteme uključuje i razvoj softvera na bezbedan način. Time, *razvojni tim* uključuju bezbednosne zahteve koje dolaze iz različitih standarda ili smernica. U zavisnosti od izabrane metodologije razvoja softvera, razvojni tim može imati specijalizacije, koje su pojašnjene u odeljku 4.1. Metodologije za razvoj softvera, ali nisu prikazane na datoj slici radi jednostavnosti.

Potencijalne specijalizacije razvojnog tima predstavljaju učesnici *savetnik za bezbednost*, *rukovodilac projekta* i *administrator blokčejn mreže*. U zavisnosti od metodologije koju tim prati, veličine tima kao i broja timova koji rade na razvoju softvera za industrijske upravljačke sisteme, može biti doneta odluka da li su potencijalne specijalizacije novi članovi tima ili će se postojećim članovima tima dodeliti dodatna zaduženja. Analiza uvođenja novih članova tima, odnosno dodela novih zaduženja postojećim članovima tima napravljena je u narednom odeljku. Nezavisno od izabranog pristupa, *rukovodilac projekta* je zadužen za odabir standarda ili stručne smernice koju će razvojni tim da prati, sa ciljem praćenja zahteva za bezbedan razvoj softvera. Takođe, *rukovodilac projekta* je na raspolaganju *savetniku za bezbednost* za eventualne konsultacije u vezi dijagrama slučajeva korišćenja. Izabrani standard ili stručna smernica, ne moraju u potpunosti biti primenljivi na razvijani softver za industrijske upravljačke sisteme. Iz tog razloga, *savetnik za bezbednost* ima zaduženje da se odredi obim primenljivosti izabranog standarda ili stručne smernice. Definisanje obima primenljivosti, razvojni tim će imati uvid u zahteve koje je neophodno ispuniti, dok će *rukovodilac projekta* imati neophodne informacije za praćenje nivoa usklađenosti sa izabranim zahtevima. Pored toga, *savetnik za bezbednost* kreira dijagrame slučajeva korišćenja, čime će se definisati i uloge koje su neophodne za ispunjenje bezbednosnih zahteva. Budući da je jedan od ciljeva da se obezbedi sledljivost prilikom utvrđivanja usklađenosti sa bezbednosnim zahtevima, uvedena je uloga *administratora blokčejn mreže* koji je zadužen za razvoj platforme koja će omogućiti date osobine. Time je jedno od zaduženja *administratora za blokčejn mrežu* razvoj rešenja na Hyperledger Fabric platformi, budući da su osobine kao što su poverljivost, sledljivost, opisane u prethodnom poglavlju, dostupne na datoj platformi. Sam razvoj rešenja na Hyperledger Fabric platformi uključuje definisanje arhitekture rešenja. Kao neophodne informacije za definisanje arhitekture rešenja, neophodno je definisati broj kanala, definisati broj organizacija i definisati broj konzorcijuma. Na Slici 10, prikazani su opisani slučajevi korišćenja, u obliku dijagrama aktivnosti, čime se prikazuje redosled kojim aktivnosti treba da se odigravaju.



Slika 10 Model baziran na blokčejnu za praćenje bezbednosnih zahteva

Kako bi se omogućilo efikasno praćenje zahteva za bezbedan razvoj softvera, predložen je model, kojim su identifikovane su tri ključne uloge, kao i njihova zaduženja. Na Slici 10, prikazan je dijagram aktivnosti na kojem su definisane aktivnosti za *rukovodioca projekta*, *savetnika za bezbednost* i *administratora blokčejn mreže*. Dijagram aktivnosti predstavlja korake koje identifikovane uloge treba da načine kako bi se kreirala platforma koja će omogućiti i ostalim učesnicima projekta da doprinesu usklađivanju softvera sa predloženim zahtevima. Početni korak predstavlja odabir standarda ili stručnih smernica koji trebaju biti zadovoljeni, što je korak koji radi rukovodilac projekta. Nakon odabira standarda, *savetnik za bezbednost* određuje obim primenljivosti izabranog standarda, gde ukoliko standard nije primenljiv u dovoljnoj meri, dijagram aktivnosti se završava. U tom slučaju, *rukovodilac projekta* bi ponovno pokrenuo ceo postupak sa novo izabranim standardnom ili stručnom smernicom. U slučaju velikog obima primenljivosti standarda ili stručne smernice, *savetnik za bezbednost* kreira slučajeve korišćenja za zahteve koji se nalaza u odabranom standardu. *savetniku za bezbednost* je ostavljena mogućnost konsultacija sa *rukovodiocem projekta*, nakon čega *administrator blokčejn mreže* može da kreira arhitekturu same mreže. Koraci prilikom kreiranja same mreže obuhvataju definisanje broja kanala koje će mreža imati, broja organizacija i jednog ili više konzorcijuma. Kako blokčejn mreža nije namenjena za čuvanje različitih tipova dokumenata, *administrator blokčejn mreže* može koristiti IPFS (eng. *Interplanetary File System*) u svrhu čuvanja dokumenata. Nakon definisanja datih parametara, a*dministrator blokčejn mreže* može postaviti arhitekturu rešenja i razviti rešenje na Hyperledger Fabric rešenju.

4.2.1. Matrica odgovornosti

Radi boljeg razumevanja raspodele zaduženja između *Rukovodioca projekta*, *Savetnika za bezbednost* i *Administratora blokčejn mreže*, definisana je matrica odgovornosti, odnosno RACI matrica (eng. RACI *matrix*). Matrica odgovornosti ima za cilj da za definisane uloge, pokaže koja uloga je odgovorna operativno izvršavanje (eng. *responsible*, obeleženo slovom **R**), koja je odgovorna za krajnji rezultat (eng. *accountable*, obeleženo slovom **A**), a koja uloga se konsultuje (eng. *consulted*, obeleženo slovom **C**), odnosno informiše (eng. *informed*, obeleženo slovom **I**) [128]. Uloge koje su označene kao odgovorne za operativno izvršavanje, moraju izvršiti definisani posao i odgovorne su njihov završetak [129]. Posao koji je odrađen od strane uloge odgovorne za operativno izvršavanje mora biti odobrene od strane uloge koja je definisana kao odgovorna za krajnji rezultat čitavog projekta. Nakon odobravanja, uloga koja je definisana kao odgovorna za krajnji rezultat postaje odgovorna i za sve naknadne eventualne probleme. Bitno je napomenuti da može postojati više uloga koje su odgovorne za operativno izvršavanje, samo jedna uloga može biti odgovorna za krajnji rezultat [129]. Usled takvog prevoda reči *responsible* i *accountable* na srpski jezik, gde se obe reči prevode kao odgovorne, bitno je imati na umu da dve uloge imaju jasno definisane opise i spram toga je potrebno definisati odgovornosti. Uloge koje su definisane kao konsultovane, su uloge koje se

mogu konsultovati u slučaju da njihova ekspertiza može doprineti efikasnijem rešavanju predočenog posla [129]. Uloge koje su informisane, dobijaju redovne novosti o napretku aktivnosti i takvih uloga može biti po jednoj aktivnosti [129]. U Tabeli 6 prikazana je matrica odgovornosti između *rukovodioca projekta*, *savetnika za bezbednost* i *administratora blokčejn mreže*, gde su odgovornosti obeležene koristeći engleske skraćenice (R, A, C, I).

Tabela 6 Matrica odgovornosti

	Rukovodilac projekta	**Savetnik za bezbednost**	**Administrator blokčejn mreže**
Odabir standarda/stručne smernice	R/A	C	I
Obim primenljivosti	C, I	R/A	I
Konsultacije u vezi dijagrama	C, I	R/A	I
Kreiranje dijagrama slučajeva korišćenja	C, I	R/A	C, I
Definisanje broja kanala	I	C	R/A
Definisanje broja organizacija	I	C	R/A
Definisanje broja konzorcijuma	I	C	R/A
Upotreba IPFS	C, I	C	R/A
Definisanje arhitekture	I	C	R/A
Razvoj rešenja na Hyperledger Fabric platformi	I	C	R/A

Odgovornost za odabir standarda ili stručnih smernica koje je potrebno implementirati je data *rukovodiocu projekta*, koji je odgovoran kako za operativno izvršavanje tako i za krajnji rezultat. Budući da se odabir standarda i/ili stručnih smernica odnosi za oblast bezbednosti, *savetnik za bezbednost* je u ovoj aktivnost konsultovan od strane *rukovodioca projekta*. Prilikom aktivnosti odabira standarda i/ili stručne smernice, *administrator blokčejn mreže* je informisan o trenutnom napretku aktivnosti, čime se i prikazuje jednosmerna komunikacija koja se odvija od *rukovodioca projekta* kad *administratoru blokčejn mreže*. U okviru aktivnosti analize obima primenljivosti standarda i/ili stručne smernice, *savetnik za bezbednost* je odgovorna uloga jer se znanje potrebno za izvođenje ove aktivnost nalazi u toj ulozi.

Rukovodilac projekta može biti konsultovan za vreme aktivnost analize obima primenljivosti, gde se ta konsultacija ogleda u dvosmernoj komunikaciji između *savetnika za bezbednost* i *rukovodioca projekta*. Nakon završetka analize obima primenljivosti, savetnik za bezbednost obaveštava rukovodioca projekta i administratora blokčejn mreže o rezultatima analize, što je definisano informativnom ulogom rukovodioca i administratora u ovoj aktivnosti. Saveznik za bezbednost je takođe odgovoran za operativno izvršenje i krajnji rezultat u aktivnosti kreiranje dijagrama slučajeva korišćenja, dok su i rukovodilac projekta i administrator blokčejn mreže konsultovani po potrebi, a informisani na kraju same aktivnosti. Sledeće tri aktivnosti su poverene administratoru za blokčejn mrežu, budući da se znanje i ekspertiza nalaze u toj ulozi. Administrator blokčejn mreže je odgovor za definisanje broja kanala, organizacija i konzorcijuma, dok je savetnik za bezbednost konsultovan u ovoj aktivnosti, budući da je izlaz iz prethodne aktivnosti osnova za definisanje broja kanala, organizacija i konzorcijuma. Odgovornost za upotrebu IPFS rešenje je takođe stavljena na administratora blokčejn mreže, ali su i rukovodilac i savetnik konsultovani u vezi potrebe za IPFS rešenjem. Savetnik za bezbednost je konsultovan zbog postojeće ekspertize vezane za odabrane standarde, dok je rukovodilac projekta konsultovan zbog ekspertize vođenja projekata, te ima uvid najbolje prakse koje mogu doprineti jednostavnijem dokazu o usklađenosti sa izabranim standardom. Za aktivnosti definisanje arhitekture i razvoja rešenja na Hyperledger Fabric platformi odgovornost je dodeljena administratoru blokčejn mreže, dok je savetnik za bezbednost konsultovan po potrebi, a rukovodilac projekta informisan i napretku i rokovima završetka opisanih aktivnosti.

4.2.2. SWOT analiza

Jedan od izazova sa kojim se timovi mogu susresti jeste uvođenje novih uloga, sa ciljem jednostavnijeg usklađivanja sa zahtevima za bezbedan razvoj softvera. Prilikom odluke da li je postojećem timu neophodno dodati nove zaposlene koji će preuzeti novo-definisane uloge ili će se postojećim članovima tima proširiti zaduženja, neophodno je sagledati nekoliko različitih aspekata. Ti aspekti uključuju veličinu tima, buduće planove za proširenje tima, broj timova koji rade na istom razvoju, kao i budžeta. Kako bi se uporedile prednosti i mane jednog i drugog pristupa, kreirana je SWOT analiza za opciju proširenja zaduženja trenutnim članovima tima, koji je predloženo kao rešenje za male timove sa ograničenim budžetom. Budući da SWOT analiza predstavlja sagledavanje snaga (eng. *Strengths*) i slabosti (eng. *Weaknesses*), kao unutrašnjih faktora i prilika (eng. *Opportunities*) i pretnji (eng. *Threats*), kao spoljnih faktora, dobija se široka slika koja može pomoći prilikom donošenja odluke. Takva analiza je prikazana u Tabeli 7.

Tabela 7 SWOT analiza mali tim

Snaga	**Slabosti**
- Prilika za postojeći tim za učenje novih tehnologija, standarda, stručnih smernica - Zadržavanje postojećeg budžeta, budući da nema novog zapošljavanja	- Nedostatak znanja unutar tima - Neophodno je obezbediti adekvatne treninge kako bi se unapredilo postojeće znanje - Moguće kašnjenje usled proširenja zaduženja koja su dobili postojeći članovi tima - Nezadovoljstvo tima usled proširenja zaduženja
Prilike	**Pretnje**
- Prednost na tržištu - Mogućnost daljeg proširenja kroz servisne usluge	- Ograničenje resursa (vreme, novac)

Sa druge strane, ukoliko postoji velik broj timova ili velik broj članova tima, kao i odgovarajući budžet, moguće je uvesti nove članove tima, koji bi preuzeli nova zaduženja. Kako bi se napravilo što bolje poređenje sa drugom opcijom, napravljena je još jedna SWOT analiza u Tabeli 8, koja prikazuje snagu, slabosti, prilike i pretnje u slučaju uvođenja novih članova, koji bi imali definisane nove uloge u timu.

Tabela 8 SWOT analiza veliki tim ili puno timova

Snaga	**Slabosti**
- Povećana efikasnost, budući da postojeći članovi tima nastavljaju da rade ono što najbolje rade - Ravnopravnija raspodela posla - Raznovrsnost perspektivi koje donose novi članovi tima sa novim ulogama	- Razvojni tim sve nedoumice vezane za bezbednost prebacuje na nove članove - Zapošljavanje novih ljudi utiče na budžet - Vreme neophodno za uhodavanje novih članova tima - Spremnost tima na nove članove i drugačiji način rada
Prilike	**Pretnje**
- Nove uloge mogu omogućiti nove servise koji se mogu pružiti klijentima, čime se pozitivno utiče na zaradu - Zadovoljstvo klijenata na posvećenosti	- Izazov prilikom pronalaska odgovarajuće osobe sa neophodnim iskustvom za datu ulogu -

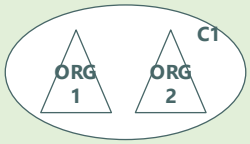
Poredeći prethodne SWOT analize za mali tim i jedan veliki tim ili puno manjih timova, mogu se sagledati prednosti i mane zapošljavanja novih ljudi, odnosno proširivanja zaduženja postojećim članovima tima.

4.2.3. Konfigurabilna blokčejn arhitektura

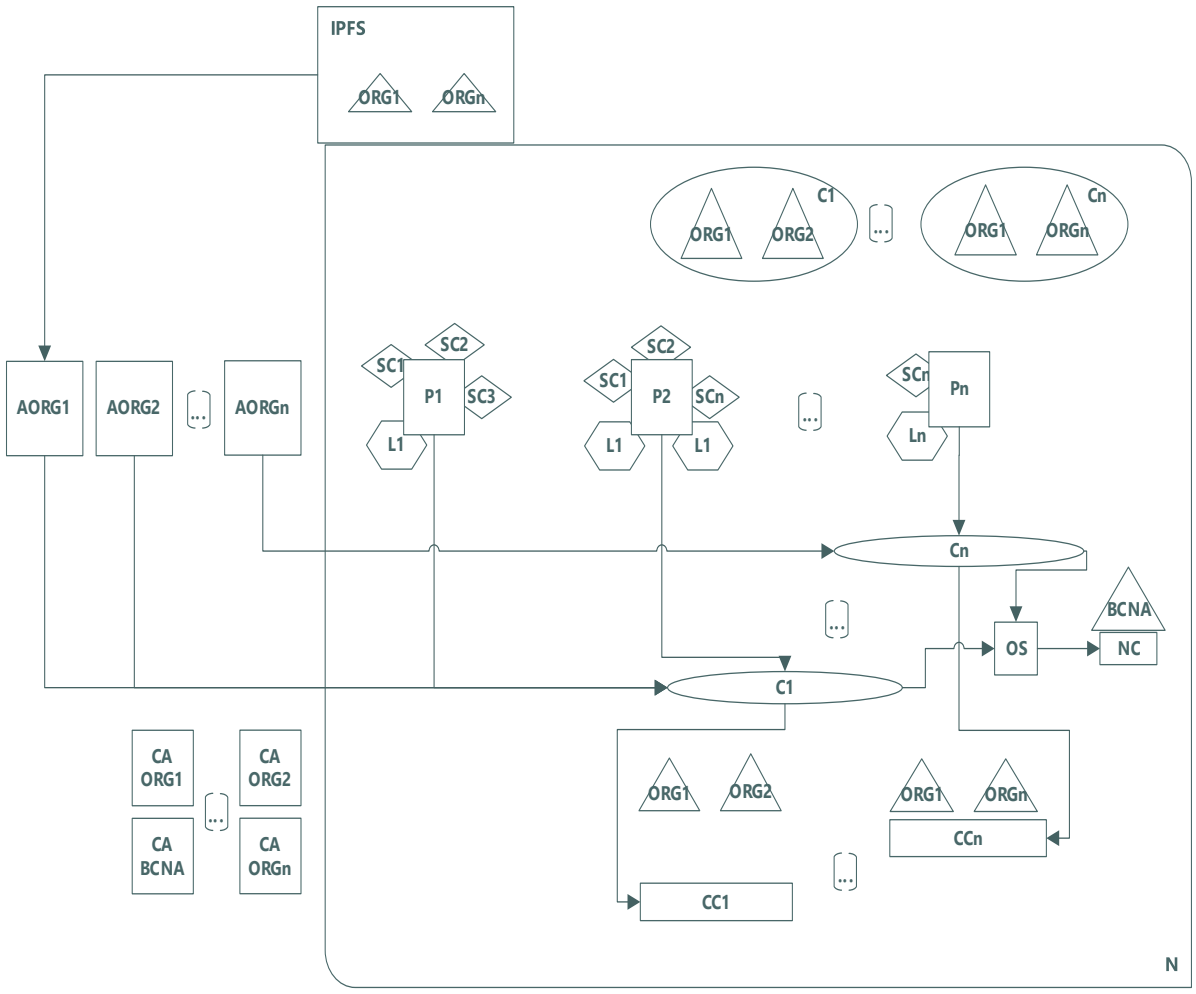
Kao jedna od aktivnosti koju Savetnik za bezbednost izvršava, a definisana je dijagramom na Slici 11, jeste aktivnosti kreiranja dijagrama slučajeva korišćenja, u okviru koje savetnik za bezbednost na osnovu zahteva koji se nalaze u odabranom standardu, definiše učesnike i njihove slučajeve korišćenja. Broj učesnika u svim dijagramima slučajeva korišćenje odgovara broju organizacija koje će biti kreirane kao rešenje na Hyperledger Fabric platformi. Takođe, na dijagramu slučajeva korišćenja se mogu uočiti i kanali i konzorcijumi koji će biti implementirani u rešenju. Kada se pogledaju parametri koji imaju ulogu u definisanju arhitekture rešenja, administrator blokčejn mreže može upotrebiti konfigurabilnu predloženu arhitekturu, koja uzima u obzir broj kanala, broj organizacija, broj konzorcijuma i potrebu za IPFS rešenjem. U okviru Tabele 7 predstavljene su oznake koje su korišćenje u konfigurabilnoj arhitekturi, kao i objašnjenja čime se definiše broj stavki [130].

Tabela 9 Oznake korišćene za potrebe definisanja konfigurabilne blokčejn arhitekture

Oznaka	Ime	Objašnjenje
	Organizacija	Organizacija predstavlja entitet koji grupiše učesnike po istoj ili sličnoj potrebi, dajući im pristup kanalu. Organizacija se u ovom modelu vezuje za učesnika iz dijagrama slučajeva korišćenja, tako da je broj organizacija definisan ukupnim brojem učesnika u svim dijagramima slučajeva korišćenja.
	Aplikacija organizacije	Kako organizacija grupiše korisnike po potrebi, svaka organizacija ima svoju dedikovanu aplikaciju za pristup blokčejn rešenju. Broj aplikacija je identičan broju organizacija. Treba imati na umu da broj aplikacija ne znači nužno i drugačiji način implementacije aplikacije za svaku organizacija, već predstavlja različit skup mogućnosti koji je dostupan različitoj organizaciji.
	Sertifikaciono telo	Kako je Hyperledger Fabric privatna blokčejn mreža sa dozvolom, svaki učesnik u okviru organizacije mora biti autentifikovan i autorizovan, što predstavlja aktivnost koju obavlja sertifikaciono telo. Broj kompanija za izdavanje sertifikata određena je brojem organizacija, budući da svaka organizacija ima

		svoje dedikovano sertifikaciono telo.
	Čvor	Entitet u blokčejn mreži koji održava knjigu i na kojem se izvršavaju programski kodovi lanaca.
	Konzorcijum	Skup organizacija koje kreiraju kanal, održavaju čvor i priključuju druge organizacije kanalu.
	Programski kod lanca	Program napisan tako da izvršava logiku koja je dogovorena među članovima mreže. Može biti napisan u Go, Node.js ili Java programskom jeziku.
	Kanal	Predstavlja dodatno ograničavanje u okviru blokčejn mreže, pružajući mogućnost izolacije određenih informacija, gde se definisane organizacije mogu priključiti određenom kanalu.
	Konfiguracija kanala	Konfiguracija kanala predstavlja spisak ograničenja i pravila koji prikazuju način pristupa organizacija datom kanalu.
	Uređivač	Skup čvorova koji uređuje redosled transakcija i prosleđuje ih ostalim čvorovima na validaciju i potvrđivanje.
	Ledger	Sadrži najažurnije stanje podatak, koji su predstavljeni kao spisak transakcija.

Konfigurabilna arhitektura je predstavljena na Slici 11, koja je kreirana koristeći principe jednostavne blokčejn mreže, opisane u ranijem poglavlju.



Slika 11 Konfigurabilna arhitektura blokčejn mreže

Konfigurabilna arhitektura prikazana na Slici 11 predstavlja blokčejn mrežu N, sa povezanim autoritetima za izdavanje sertifikata, aplikacijama za organizacije i IPFS. U okviru prikazne arhitekture broj organizacija, kanala, konzorcijuma i potreba za IPFS rešenjem je diktirano od strane rezultata dijagrama aktivnosti, što je prikazano brojevima od 1 do n. Blokčejn mreža N sadrži jedan servis za uređivanje (eng. *ordering service*, OS), jednu mrežnu konfiguraciju i organizaciju administratora blokčejna (BCNA). Takođe, broj konzorcijuma čvorova (Pn), pametnih ugovora (SCn), knjiga (Ln), sertifikaciono telo (CAn) i aplikacija (AORGn) definisan je kroz slučajeve korišćenja i njihove učesnike, koje kreira savetnik za bezbednost. Ako se IPFS identifikuje u slučajevima korišćenja, može se eksterno dodati u blokčejn mrežu i povezati sa odgovarajućim aplikacijama. Pristup IPFS rešenju se dodeljuje organizacijama koje za to imaju potrebu koja je identifikovana slučajevima korišćenja, a sam pristup je omogućen kroz aplikaciju svake organizacije.

4.2.4. Potvrda hipoteze H1

Definisanjem modela za praćenje usklađenosti sa zahtevima za bezbedan razvoj softvera koji se odnose na industrijske upravljačke sisteme, potvrđena je prethodno definisana hipoteza **H1**. Predloženi model predstavlja korak ka bezbednosti i pouzdanosti u industrijskim upravljačkim sistemima, čime se daje značaj očuvanju integriteta industrijskih procesa. Tokom razvoja softvera za industrijske upravljačke sisteme, neophodno je uvrstiti zahteve za bezbedan razvoj softvera, čime su smanjuju rizici koje softvera mogu predstavljati po okolinu i najvažnije, ljudske živote. Model obuhvata definisane uloge, zajedno sa svojim aktivnostima, koje praćenjem modela, pomažu podizanju bezbednosti softvera i omogućavaju sledljivo pokazivanje usklađenosti sa odabranim zahtevima. Pored uloga i aktivnosti, definisane uloge su posmatrane i kroz matricu odgovornosti, čime se dodatno pojašnjavaju razlike svih predloženih uloga. Takođe, model može se prilagoditi različitim veličinama i složenosti, što je pokazano u sekciji SWOT analiza.

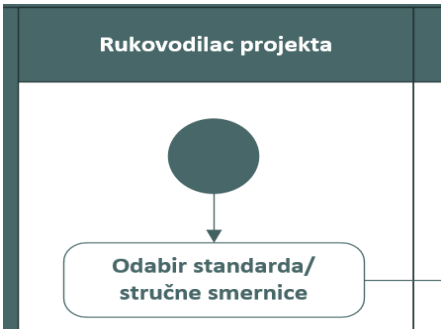
Kako bi se pokazala validnost modela, u nastavku će biti prikazan svaki korak predloženog modela, primenjen na koristeći zahteve za bezbedan razvoj softvera koji se nalaze u okviru standarda IEC 62443-4-1, pod nazivom „Zahtevi za bezbedan razvoj proizvoda“ (eng. *Secure product development lifecycle requirements*). Time će se steći uslovi za potvrđivanje preostalih hipoteza **H2** i **H3**.

5. Validacija modela za praćenje zahteva za bezbedan razvoj softvera

U prethodnim poglavljima su opisane podele blokčejn mreža, načini funkcionisanja kao i standardi za bezbedan razvoj softvera. Industrijski upravljački sistemi su identifikovani u literaturi kao sistemi čija bezbednost je od izuzetnog značaja, pogotovo u slučajevima njihove upotrebe u kritičnim infrastrukturama [131]. Softveri za industrijske upravljačke sisteme prolaze kroz posebne provere sa ciljem pružanja odgovarajućeg nivoa bezbednosti. Jedan od načina podizanja nivoa bezbednosti jeste praćenje standarda i stručnih smernica vezanih za bezbedan razvoj softvera. Prilikom implementacije standarda i stručnih smernica, potrebno je kreirati i sačuvati dokaz da je predložena kontrola implementirana, za potrebe budućih eventualnih provera. Za potrebe praćenja nivoa usklađenosti softvera za industrijske upravljačke sisteme sa zahtevima za bezbedan razvoj, u prethodnom poglavlju je prikazan model za praćenje zahteva za bezbedan razvoj softvera. Model je prikazan kroz dijagram aktivnosti i matricu odgovornosti, gde su opisane aktivnosti koje identifikovane uloge moraju da ispune. U okviru ovog poglavlja, model iz prethodnog poglavlja će biti validiran, koristeći zahteve za bezbedan razvoj softvera koji se nalaze u okviru standarda IEC 62443-4-1, pod nazivom „Zahtevi za bezbedan razvoj proizvoda“ (eng. *Secure product development lifecycle requirements*), budući da je standard usmeren na bezbednost sistema industrijske automatizacije i upravljanja. Sam standard je podeljen u osam celina, te će se validacijom modela pokazati koraci iz modela za zahteve u okviru prve celine, odnosno Upravljanje bezbednošću (eng. *Security management*). U narednim odeljcima biće prikazane aktivnosti opisane u modelu, tako da su prikazani kroz zahteve iz celine Upravljanja bezbednošću.

5.1. Odabir standarda/stručne smernice

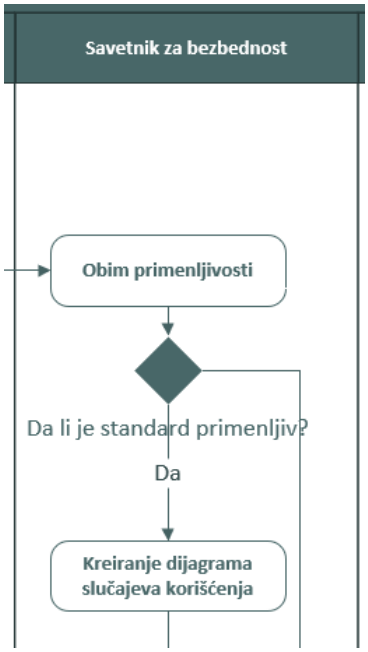
Prvi korak u prikazanom modelu predstavlja odabir standarda ili stručne smernice, čiji zahtevi će se implementirati i čije praćenje je potrebno. Rukovodilac projekta odabira IEC 62443-4-1 standard, koji smatra prikladnim za softvere koji se nalaze u okviru industrijskih upravljačkih sistema. Posmatrajući trenutno stanje u industriji [132], porast bezbednosnih problema u okviru industrijskih upravljačkih sistema [133] i smernice koje dolaze od vladinih agencija i službi [134], odabir IEC 62443-4-1 standarda rukovodioca projekta je utemeljen i kao takav se prosleđuje savetniku za bezbednost na utvrđivanje obima primenljivosti. U okviru dijagrama aktivnosti prikazanog na Slici 10, ovaj korak se nalazi na samom početku, kao što je prikazano na Slici 12.



Slika 12 Aktivnost – Odabir standarda/stručne smernice

5.2. Obim primenljivosti

Prateći model predstavljen u prethodnom poglavlju (isečak dostupan na Slici 13), *savetniku za bezbednost* je dostavljen predlog za praćenje zahteva koji se nalaze u okviru standarda IEC 62443-4-1. U okviru ovog koraka, savetnik za bezbednost analizira sam standard i definiše obim primenljivosti. Zarad čitljivosti same disertacije, u okviru ovog koraka smatraćemo da je savetnik za bezbednost identifikovao samo zahteve u okviru celine Upravljanje bezbednošću za dalje korake. Celina pod nazivom Upravljanje bezbednošću se sastoji od 13 zahteva i svi zahtevi će biti obrađeni u narednim koracima.



Slika 13 Aktivnosti – Obim primenljivosti i Kreiranje dijagrama slučajeva korišćenja

5.3. Kreiranje dijagrama slučajeva korišćenja

Kao jedna od aktivnosti koju Savetnik za bezbednost izvršava, a definisana je modelom na Slici 11 (isečak dostupan na Slici 13), predstavlja kreiranje dijagrama slučajeva korišćenja. Aktivnost koju savetnik za bezbednost treba da uradi na osnovu prethodnog koraka, obima

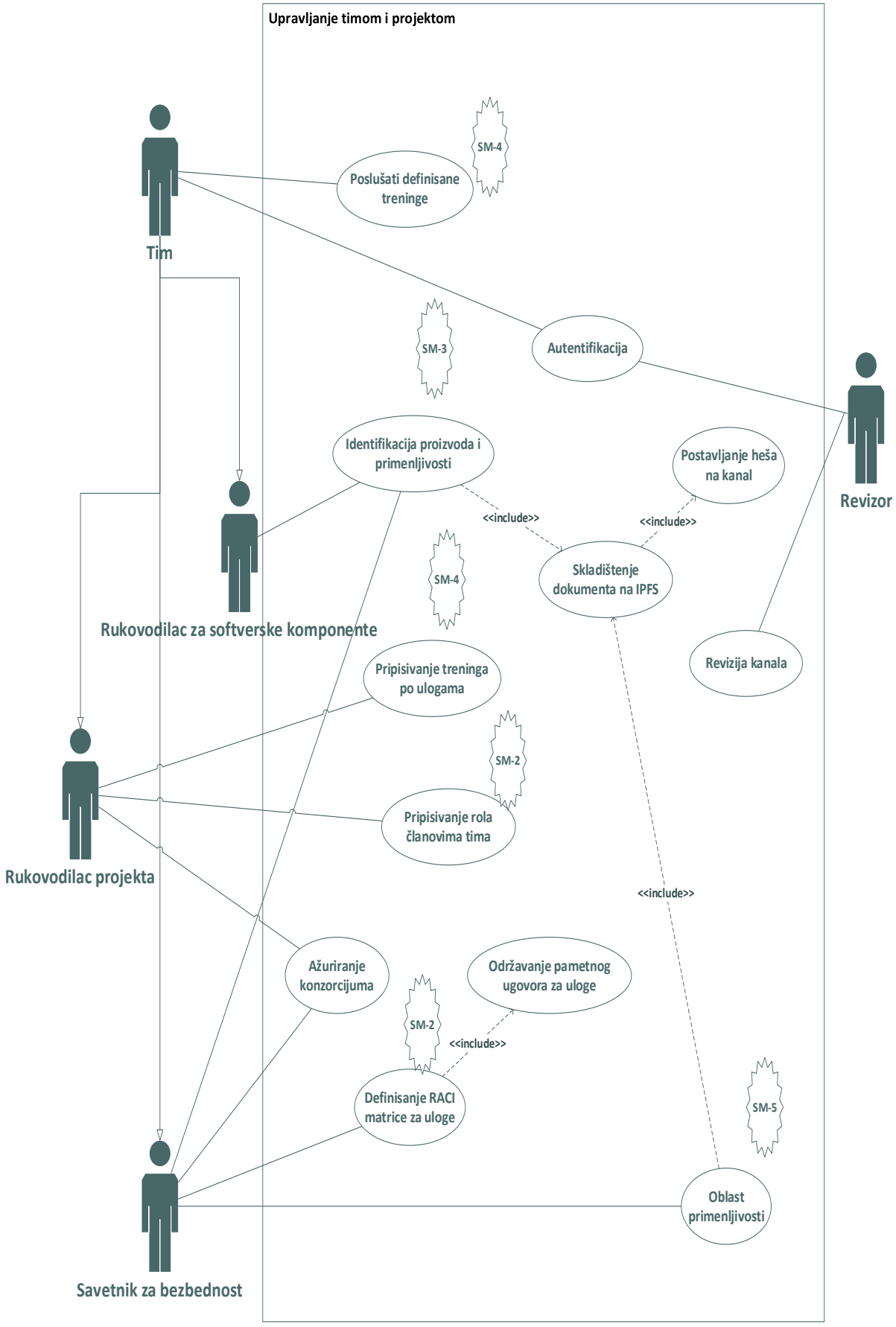
primenljivosti, predstavlja preduslov za sve naredne korake, koje administrator blokčejn mreže treba da nastavi. Savetniku za bezbednost su na raspolaganju 13 zahteva koji su deo celine Upravljanja bezbednošću, a deo su standarda IEC 62443-4-1. Analizom 13 zahteva iz celine Upravljanja bezbednošću, savetnik za bezbednost je kreirao dijagrame slučajeva korišćenja. Tom prilikom, identifikovani su učesnici i njihove aktivnosti. Usled velikog broja aktivnosti koje su uočene tokom analize zahteva, identifikovana su četiri dijagrama slučajeva korišćenja, koja su formirana na osnovu zajedničke tematike samih zahteva:

1. Upravljanje timom i projektom: ovim dijagramom su pokriveni zahtevi sa oznakama SM-2, SM-3, SM-4 i SM-5, koji se bave određivanjem uloga u timu (SM-2), njihovom edukacijom u oblasti bezbednosti softvera (SM-4) i analizom primenljivosti standarda na softver koji se razvija (SM-3, SM-5),
2. Okruženje za razvoj: u okviru ove celine, nalaze se zahtevi za pravljenje bezbednog okruženja za razvoj softvera (SM-7), načina na koji se softver razvija, odnosno zahtev za razvojni proces (SM-1), kao i zahtev za kontrole za privatne ključeve (SM-8) i zahtev za integritet fajlova (SM-6),
3. Upravljanje lancem snabdevanja: kreirani dijagram pokriva zahteve pod oznakama SM-9 i SM-10, koje su usmere na zahteve za lance snabdevanja (SM-9) i uključivanje drugih kompanija u razvoj komponenti (SM-10).
4. Kvalitet: poslednji dijagram pokriva zahteve od SM-11 do SM-13, koji govore o praćenju problema u softveru do njihovog rešavanja (SM-11), procesnoj verifikaciji koja je zadovoljna praćenjem predloženog modela u okviru ove disertacije (SM-12) i kontinualno unapređivanje procesa za bezbedan razvoj softvera (SM-13).

U svakom od narednih odeljaka, opisani su dijagrami slučajeva korišćenja, vodeći se podelom koja je prethodno opisana. Detalji samih dijagrama su opisani kroz prizmu svakog učesnika, dok je za svaku aktivnost prikazan i analiziran zahtev iz celine Upravljanja bezbednošću. Iz svakog pojedinačnog dijagrama slučajeva korišćenje će se izvući broj učesnika, koji će u narednim koracima predstavljati broj organizacija.

5.3.1. Upravljanje timom i projektom

U okviru dijagrama za upravljanje timom i projektom, identifikovani su učesnici Tim i Revizor. Učesnik Tim predstavlja generalizaciju nekoliko učesnika, budući da korisnici Rukovodioci projekata, Rukovodilac softverskih komponenti i savetnik za bezbednost imaju dva zajednička slučaja korišćenja. Jedan od tih slučaja korišćenja je deljen sa Revizorskom ulogom, što je predstavljeno kroz autentifikaciju koju sve uloge moraju da obave pre upotrebe sistema. Na Slici 14 su predstavljeni pomenuti korisnici, njihove aktivnosti, čime je kreiran dijagram slučajeva korišćenja za upravljanje timom i projektom.



Slika 14 Dijagram slučaja korišćenja - Upravljanje timom i projektom

Detalji svakog slučaja korišćenja su opisani kroz prizmu učesnika:

- Učesnik *Tim*: zajedničko za sve specijalizacije učenika Tim jeste kompletiranje obuke koja im je dodeljena. Kompletiranjem obuke se ispunjava zahtev u okviru celine Upravljanje bezbednošću, sa oznakom SM-4. Kako bi se pristupilo platformi za praćenje usklađenosti softvera, potrebno je da se svi korisnici autentifikuju, što je i predstavljeno slučajem korišćenja Autentifikacija.
    - Učesnik *Rukovodilac projekta*: u okviru ove uloge objedinjene su različite aktivnosti koji su vezane za vođenje projekata koji razvijaju softver u skladu sa zahtevima za bezbedan razvoj softvera. Pored uobičajenih aktivnosti koje Rukovodioci projekta imaju, kao što su vođenje tima, spram definisanih rokova, cilja i u skladu sa budžetom, kako bi se zadovoljili zahtevi u celini Upravljanje bezbednošću, identifikovane su dodatne aktivnosti:
        1. Dodeljivanje uloge u okviru tima: kako bi se zadovoljio zahtev u okviru celine Upravljanje bezbednošću, sa oznakom SM-2 čiji cilj je identifikovanje tima i dodeljivanje uloga spram postojećeg posla. Dodeljivanje uloga se može prilagoditi svakom projektu ponaosob, prateći postojeće smernice za razvoj softvera i druge smernice za razvoj softvera koji prepoznaju uloge u timu [135].
        2. Dodeljivanje obuka spram uloge: nakon dodeljivanja uloga u timu, svaki član tima prolazi kroz obuku spram svoje definisane uloge. Dati slučaj korišćenja predstavlja preduslov za kompletiranje obuke učesnika u razvoju softvera predstavlja zahtev koji je obeležen sa oznakom SM-4 u okviru celine Upravljanje bezbednošću.
        3. Ažuriranje konzorcijuma: identifikovan slučaj korišćenja nije proistekao kao zahtev u okviru celine Upravljanje bezbednošću, već je način kojim će se upravljati pravom pristupa i odobravati Revizorima pristup platformi. U okviru Hyperledger Fabric rešenja, konzorcijum je skup organizacija koje imaju mogućnost dodeljivanja prava za čitanje Revizorima kada je to potrebno, kao i ukljanjanje prava kada Revizorima pristup nije više potreban. Kako je konzorcijum sastavljen od više organizacije, na dijagramu slučajeva korišćenja su predstavljeni Rukovodilac projekta i Savetnik za bezbednost kao uloge koje imaju mogućnost ažuriranja konzorcijuma.
    - Učesnik *Rukovodilac za softverske komponente*: ovom ulogom je objedinjeno nekoliko različitih aktivnosti koje izvršava Rukovodilac za Softverske Komponente. Glavne aktivnosti su definisane slučajevima korišćenja u okviru ovog dijagrama, kao i narednim dijagramima, ali takođe može obavljati i neku od aktivnosti koje opisuju procesi za razvoj softvera, a definisani su ulogom DevOps inženjera [136]. U okviru ovog dijagrama, identifikovani su sledeći slučaji korišćenja:

1. Identifikacija proizvoda spram primenljivosti: u okviru zahteva sa oznakom SM-3, neophodno je pokazati da softver koji se razvija na bezbedan način u dovoljnoj meri može primeniti zahteve koji su definisani standardom IEC 62443-4-1. U okviru ovog zahteva, potrebno je označiti i delove softvera koji neće primeniti deo standarda ili čitav standard. Kako je jedan od načina da se pokaže primenjivost kreiranje dokumenta koji je opisuje, potrebno je skladištiti dokument na IPFS rešenje, budući da blokčejn mreža nije napravljena za skladištenje te vrste fajlova. Pored skladištenja dokumenta na IPFS rešenje, u okviru ovog slučaja korišćenja se dobijeni heš dokumenta skladišti na blokčejn mrežu, čime se pokazuje vreme kreiranja dokumenta, a heš omogućava integritet.

- Učesnik *Savetnik za bezbednost*: uloga savetnika za bezbednost je već diskutovana u samom modelu. Ideja uloge jeste da znanje neophodno za bezbedan razvoj softvera bude skoncentrisano u savetniku, koji može da pruža savete i dodatne smernice kako za sam razvoj softvera tako i za uopštene najbolje prakse u bezbednosti. Kada se posmatra uloga savetnika za bezbednost u zahtevima Upravljanja bezbednošću, konkretno na zahteve grupisane kao Upravljanje timom i projektom, savetnik za bezbednost ima sledeća zaduženja:
  1. Identifikacija proizvoda spram primenljivosti: u okviru zahteva sa oznakom SM-3, zajedno sa Rukovodiocem za Softverske Komponente, Savetnik za bezbednost označava delove softvera koji se razvija, na koji su primenljivi zahtevi iz IEC 62443-4-1 standarda.
  2. Definisanje obima primenljivosti: korak koji je savetnik za bezbednost ispunio u okviru modela, pokriven je zahtevom koji je označen sa SM-5. Kako je jedan od načina da se pokaže obim primenljivosti kreiranje dokumenta koji je opisuje, potrebno je skladištiti dokument na IPFS rešenje. Nakon skladištenja dokumenta na IPFS rešenje, u okviru ovog slučaja korišćenja se dobijeni heš dokumenta skladišti na blokčejn mrežu, čime se pokazuje vreme kreiranja dokumenta, dok heš omogućava integritet.
  3. Definisanje matrice odgovornosti: deo zahteva SM-2 je obrađen u sklopu uloge *rukovodioca projekta*, gde se definišu uloge članova tima. Međutim, znanje vezano za bezbednost se nalazi u *savetniku za bezbednost*, tako da se kompletiranje zahteva SM-2 dobija definisanjem matrice odgovornosti, koju kreira *savetnik za bezbednost*.
  4. Ažuriranje konzorcijuma: kao što je slučaj sa *rukovodiocem projekta*, ovaj slučaj nije zahtev iz celine Upravljanja bezbednošću, već posledica korišćenja Hyperledger Fabric rešenja. Savetnik za bezbednost i Rukovodilac projekta će predstavljati organizacije ko će se kreirati u

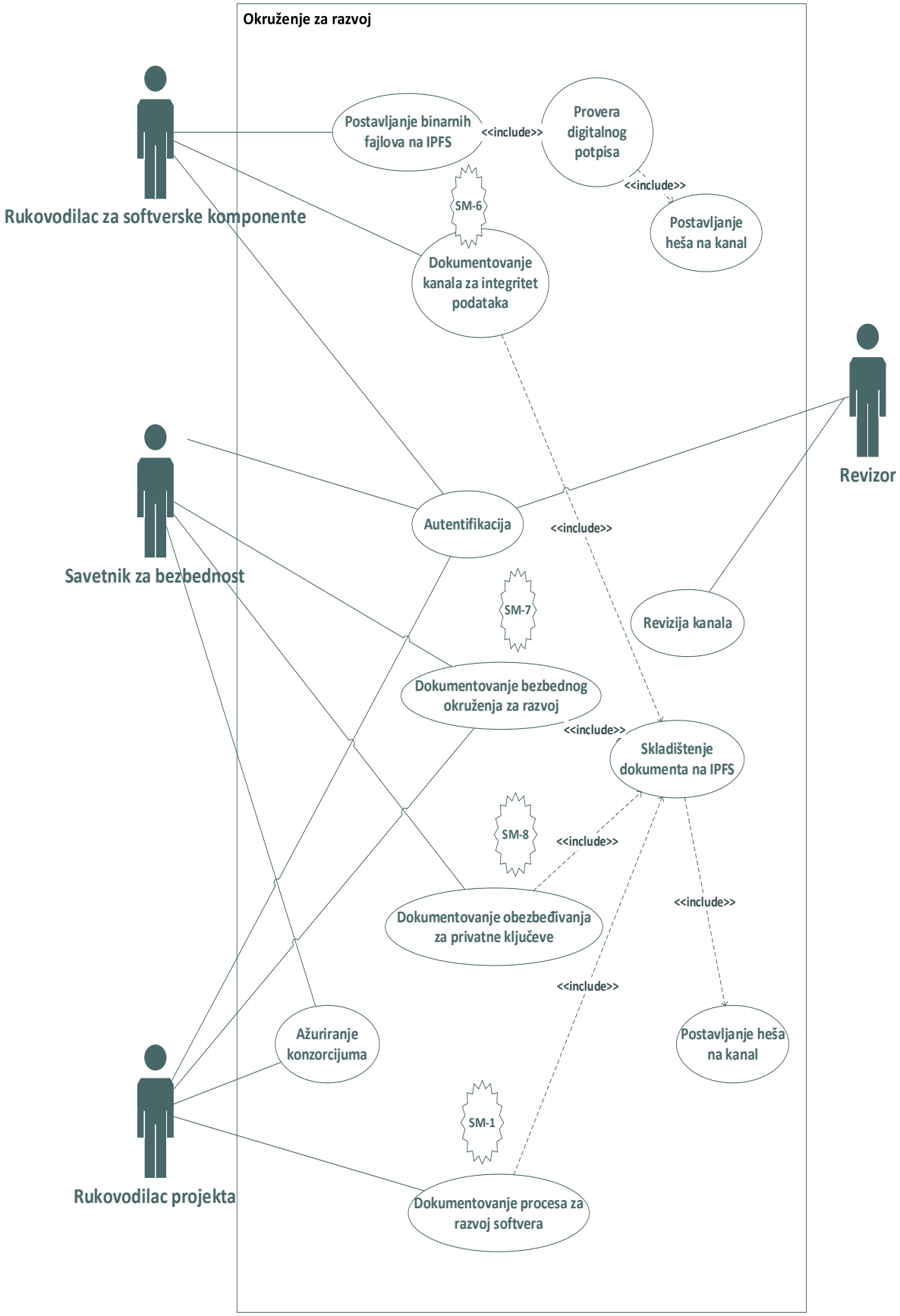
okviru blokčejn mreže i kroz konzorcijum će se omogućavati pristup informacijama koje su skladištene na blokčejn mrežu.

- Učesnik *Revizor*: implementiranje zahteva standarda ili stručne smernice je često revidirano od strane unutrašnjeg ili spoljašnjeg revizora čime se potvrđuje nivou usklađenosti sa izabranim standardom. Uloga *Revizora* jeste da se uvidom u informacije koje su skladištene na blokčejn mreži, nedvosmisleno utvrdi nivou usklađenosti sa standardom. Sama usklađenost može biti zahtevana od strane zakona ili regulativa. U slučaju spoljašnjih revizora, odnosno revizora koji ne pripadaju kompaniji koja i radi na samom razvoju softvera, često se angažuju kompanije koje su specijalizovane za reviziju. Kako bi se zadržao princip dovoljnog nivoa informisanosti (eng. *need to know*), *Revizorima* će biti omogućen pristup informacijama nakon autentifikacije, u skladu sa pravima koje su definisane u okviru kanala kojem pristupaju. Nakon obavljenog revizorskog posla, pristup se ukida, omogućavajući kontinuiranu bezbednost informacija skladištenih na blokčejn mreži. Iz tih razloga, učesnik *Revizor* ima sledeći slučaj korišćenja:
    1. Revizija kanala: ovo je jedini slučaj korišćenja koju Revizori mogu da rade. Uvidom u informacije koje su skladištene u okviru kanala, Revizor ima mogućnost da se uveri u nivo usklađenosti implementacije sa zahtevima iz standarda.

Naredni odeljci opisuju preostale dijagrame slučajeva korišćenja.

5.3.2. Okruženje za razvoj

Naredni dijagram slučajeva korišćenja grupiše zahteve koji su u okviru standarda IEC 62443-4-1 obeleženi oznakama SM-1, SM-6, SM-7 i SM-8. Zahtevi su usmereni ka obezbeđivanju okruženja za razvoj, upravljanju privatnim ključevima, integritetu fajlova i razvojnom procesu softvera. Učesnici dijagrama slučajeva korišćenja Okruženje za razvoj su već prikazani u prethodnom dijagramu, ali u okviru ovog dijagrama identifikovane su dodatne aktivnosti. Na Slici 15, prikazani su slučajevi korišćenja za *Rukovodioca za Softverske komponente*, *Savetnika za bezbednost*, *Rukovodioca projekta* i *Revizora*. Zajedničko za sve učesnike je autentifikovanje na platformu pre samog korišćenja.



Slika 15 Dijagram slučaja korišćenja - Okruženje za razvoj

Prateći način opisivanja prethodnog dijagrama slučajeva korišćenja, slučajevi korišćenja su opisani kroz prizmu učesnika:

- Učesnik *Rukovodilac za Softverske Komponente*: pored zaduženja koja su identifikovana u prethodnom dijagramu, Rukovodilac za Softverske Komponente proširuje svoje aktivnosti sledećim slučajevima korišćenja:
    1. Dokumentovanje procesa za integritet fajlova: kako bi se omogućila usklađenost softvera sa zahtevom SM-6, neophodno je dokumentovati proces koji će opisati način na koji se osigurava integritet fajlova. U literaturi [137] se mogu pronaći različita rešenja za održavanje integriteta fajlova, što je ostavljeno *Rukovodiocu za Softverske Komponente* da odabere kao najpogodnije rešenje za implementaciju. Budući da je neophodno zadržati dokumentovan proces, slučaj korišćenja uključuje i postavljanje dokumenta na IPFS platformu. Tim korakom se dokument skladišti na rešenje namenjeno distribuiranom čuvanju fajlova, dok se kao povratna informacija prilikom skladištenja dokumenta, heš vrednost, čuva na Hyperledger kanalu.
    2. Postavka izvršnih datoteka na IPFS: pored procesa za integritet fajlova, sami fajlovi se mogu skladištiti na IPFS platformi. Prilikom skladištenja, vršila bi se provera digitalnog potpisa fajla, potvrđujući da je fajl napravljen od strane kompanije koja to i tvrdi. Nakon postavljanja fajla na IPFS platformu, dobijena heš vrednost se čuva u okviru dodeljenog kanala na Hyperledger Fabric platformi.
- Učesnik *Savetnik za bezbednost*: slučajevi korišćenja u okviru ovog dijagrama pridodati Savetniku su:
    1. Zaštita privatnih ključeva: kako bi se ispunio zahtev SM-8, neophodno je napraviti mehanizam kojim se štite privatni ključevi. Neki od načina na koji se implementiraju mehanizmi za zaštitu privatnih ključeva su opisani u [138]. Sama implementacija se dokumentuje i skladišti na IPFS rešenje, budući da se fajlovi u tom formatu ne postavljaju na kanale u Hyperledger Fabric rešenju. Postavljanjem dokumenta na IPFS rešenje se dobija heš vrednost skladištenog dokumenta i ta heš vrednost se skladišti na kanalu Hyperledger Fabric platforme, čime se može verifikovati integritet skladištenog dokumenta.
    2. Zaštita okruženja za razvoj: standard IEC 62443-4-1 je usmeren bezbednom razvoju softvera za industrijske upravljačke sisteme i zahtevi objašnjavaju na koje načine se podiže bezbednost samog softvera, pokrivajući sve faze razvoja softvera, od prikupljanja zahteva do održavanja. Kako bi se osiguralo da softver koji se razvija ima i bezbedno okruženje, neophodno je ispuniti zahtev pod oznakom SM-7. Dokumentom kojim se opisuje okruženje za razvoj softvera koje je

zaštićeno implementiranim mehanizmima zaštite se skladišti na IPFS rešenje, dok se dobijena heš vrednost skladišti u okviru implementiranog pametnog ugovora, omogućavajući proveru integriteta dokumenta.

3. Ažuriranje konzorcijuma: sa istim razlozima kao u prethodnom slučaju korišćenja, Savetniku za bezbednost data je mogućnost da ažurira konfiguraciju kanala, čime će omogućiti pristup kanalu drugim učesnicima. Kako je konzorcijum sastavljen od *Savetnika za bezbednost* i *Rukovodioca projekta*, oba učesnika moraju biti saglasni sa izmenom.

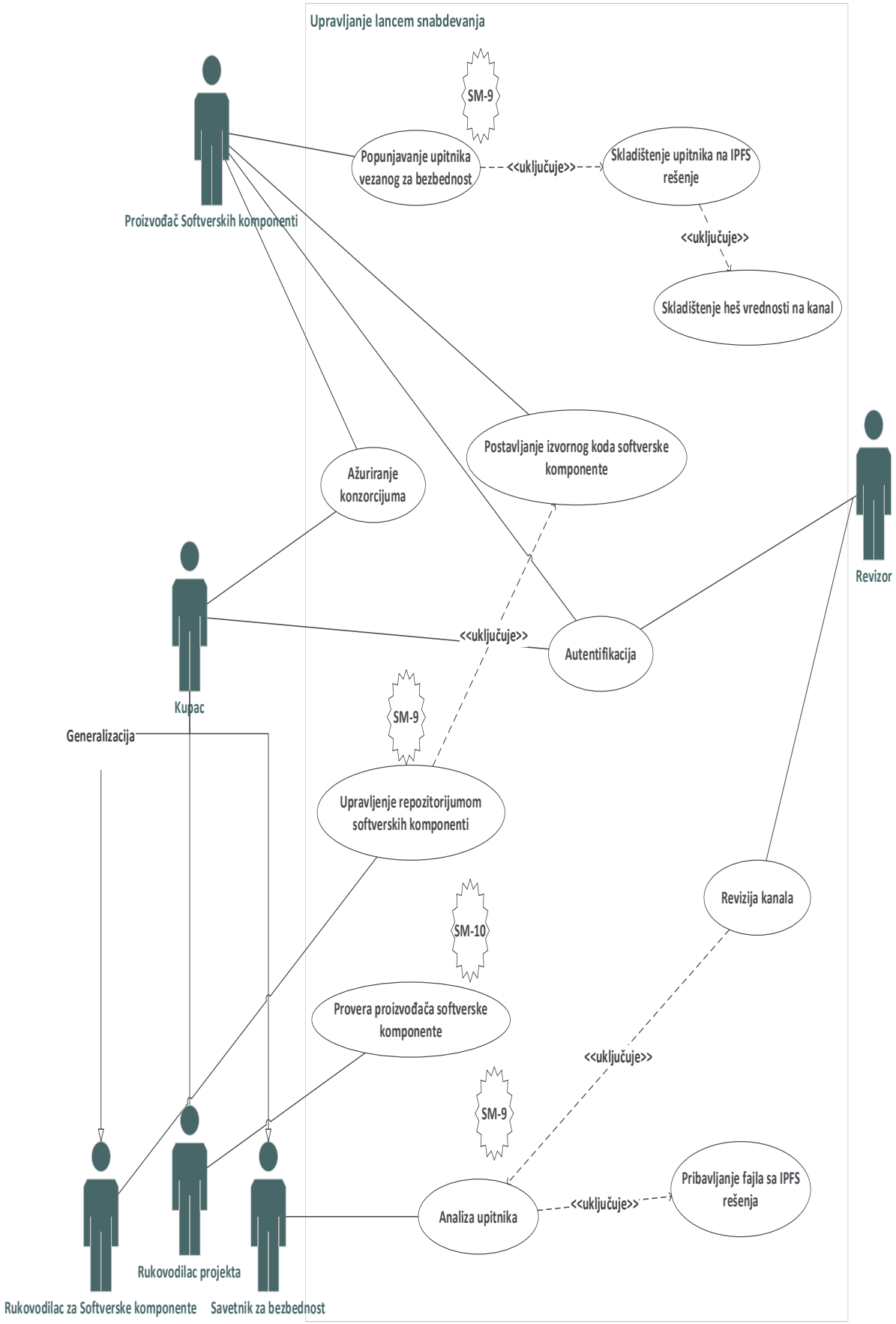
- Učesnik *Rukovodilac projekta*: prateći prethodne učesnike, *Rukovodiocu projekta* su dodati sledeći slučajevi korišćenja:
    1. Proces za razvoj: zahtev SM-1 je usmeren na kreiranje procesa za razvoj softvera. Kompanije se mogu okrenuti različitim metodama za razvoj softvera, kao što su agilne metode ili neke tradicionalne, kao što je vodopad [139]. Kako bi se pokazala usaglašenost sa zahtevom, propisani proces za razvoj softvera se dokumentuje i dokument se skladišti na IPFS rešenju, zajedno sa ostalim dokumentima.
    2. Ažuriranje konzorcijuma: kao što je opisano u prethodnom dijagramu slučajeva korišćenja i u okviru učesnika *Savetnik za bezbednost*, *Rukovodilac projekta* ima mogućnost izmene konfiguracije kako bi se omogućilo dodavanje ili sklanjanje prava pristupa organizacijama.
- Učesnik *Revizor*: uloga *Revizora* se nije proširila u okviru ovog slučaja korišćenja i jedini slučaj je:
    1. Revizija kanala: kako bi se potvrdila usklađenost softvera sa zahtevima, *Revizor* dobija pravo pristupa kanalu, nakon čega se vrši provera. Nakon obavljene revizije, *Revizor* gubi pravo pristupa, prateći princip dovoljan nivo informisanosti.

U naredna dva dijagrama slučajeva korišćenja, uvode se novi učesnici, koji su neophodni za učestvovanje u implementaciji zahteva.

5.3.3. Upravljanje lancem snabdevanja

Dijagramom slučajeva korišćenja pod nazivom Upravljanje lancem snabdevanja, grupisani su zahtevi iz celine Upravljanje bezbednošću standarda IEC 62443-4-1, pod oznakama SM-9 i SM-10. Praćenje lanca snabdevanja iz ugla bezbednosti predstavlja zahtev koji se pominje i u okviru drugih standarda, koji su vezani za industrijske upravljačke sisteme ili kritične infrastrukture [134]. Lanac snabdevanja u softverskom okruženju predstavlja skup svih računarskih biblioteka ili softvera koji su iskorišćeni za kreiranje drugog softvera. Da bi se krajnji softver smatrao bezbednim, potrebno je obezbediti da su sve komponente tog softvera, koji su deo lanca snabdevanja, bezbedne. Softver koji kasnije postaje takođe deo drugog lanca snabdevanja, zadržava nivo svoje bezbednosti i doprinosi bezbednosti softvera koji je dalje u

lancu snabdevanja. Učesnici u lancu snabdevanja predstavljaju treću stranu za kompaniju koja je dobila na korišćenje ili kupila komponentu. Jedan od načina da se kreira ili očuva bezbednost komponenti u lancu snabdevanja je bezbednosna provera svih komponenti u tom lancu. Identifikovani učesnici u dijagramu slučaja korišćenja za Upravljanje lancem snabdevanja su *Proizvođač Softverskih Komponenti*, *Kupac*, koji je generalizacija prethodno prikazanih učesnika *Savetnika za bezbednost*, *Rukovodioca projekta* i *Rukovodioca Softverskih Komponenti*, kao i *Revizor*. Generalizacija *Kupac* predstavlja istu generalizaciju kao u slučaju dijagrama Upravljanje timom i projektom, ali zbog mesta korišćenja je stavljeno ime *Kupac*. Na Slici 16, prikazan je dijagram slučajeva korišćenja za pomenute korisnike, gde je zajednički slučaj korišćenja sistema autentifikacija na sistem.



Slika 16 Dijagram slučajeva korišćenja - Upravljanje lancem snabdevanja

Slučajevi korišćenja razmatrani su kroz prizmu učesnika u dijagramu:

- Učesnik *Proizvođač Softverske komponente*: kako ovaj učesnik ne pripada ostatku organizacije koja razvija softver, pravljenjem novog dijagrama slučaja korišćenja, a posledično i drugih kanala na Hyperledger Fabric rešenju, *Proizvođač Softverske komponente* neće imati uvid u ostale kanale i podatke koji su skladišteni na platformi.
    1. Popunjavanje upitnika vezanih za bezbednost komponente: jedan od načina provere nivoa bezbednosti komponente je analiza upitnika koju popunjavaju proizvođači softverskih komponenti. Za komponente koje nisu otvorenog koda, proizvođači popunjavaju upitnik sa stanovišta bezbednosti, koji je *Kupac* kreirao. Tom aktivnošću se zadovoljava zahtev SM-9, iz celine Upravljanje bezbednošću. Nakon popunjenog upitnika, isti se postavlja na IPFS rešenje, za skladištenje, dok se heš vrednost koju je IPFS rešenje dalo nakon skladištenja dokumenta, smešta na blokčejn mrežu.
    2. Postavljanje izvornog koda softverske komponente: ukoliko je proizvođač softverske komponente napravio softversko rešenje koje je namenski napravljeno za određenog kupca, izvorni kod tog rešenja se postavlja na repozitorijum, kojem može da upravlja *Rukovodilac za Softverske Komponente*. Ovim slučajem korišćenja se pokazuje usklađenost sa zahtevom SM-10.
    3. Ažuriranje konzorcijuma: kako je prilikom revizije implementacije zahteva potrebno omogućiti učesniku *Revizor* da pristupi sistemu, neophodno je ažurirati konzorcijum tako da je Revizoru dato pravo pristupa. Konzorcijum u ovom dijagramu slučajeva korišćenja je napravljen između *Proizvođača Softverske Komponente* i generalizacije *Kupac*. Na taj način, biće neophodna saglasnost oba učesnika za izmenu konfiguracije i dodavanje novih učesnika, odnosno organizacija na Hyperledger Fabric platformi.
- Učesnik *Kupac*: posmatrajući dijagram slučajeva korišćenja, učesnik *Kupac* predstavlja generalizaciju tri učesnika, gde svaka specijalizacija ima određenu ulogu u nabavci softverskih komponenti od treće strane. Jedini zajednički slučaj korišćenja je:
    1. Ažuriranje konzorcijuma: pored učesnika *Proizvođač Softverskih Komponenti*, u ažuriranju konzorcijuma učestvuje i učesnik *Kupac*, budući da je potrebna saglasnost oba učesnika da se izmeni kanal i dozvoli pravo pristupa učesniku *Revizor*.
    - *Savetnik za bezbednost*: proširujući aktivnosti koje su dodeljene *Savetniku za bezbednost*, u okviru ovog dijagrama, identifikovan je sledeći slučaj korišćenja:
        1. Analiza upitnika vezanog za bezbednost komponente: nakon popunjenog upitnika vezanog za bezbednost komponente od strane *Proizvođača softverskih komponenti*, na *Savetniku za bezbednost* je da

proveri upitnik koji je postavljen na IPFS rešenje. Na taj način, upotpunjava se zahtev SM-9.

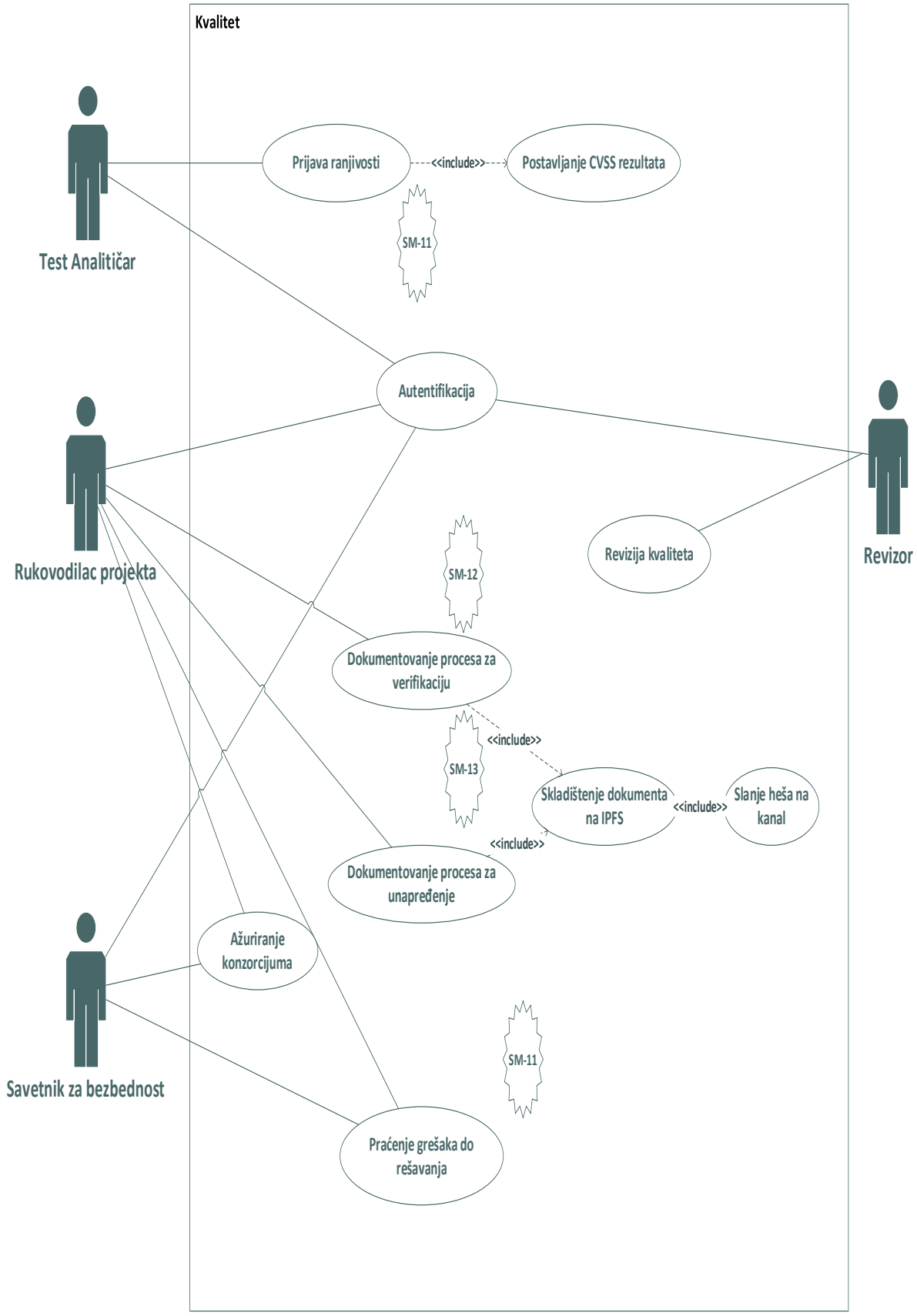
- *Rukovodilac projekta*: pored slučaja korišćenja koju ima kao specijalizacija učesnika *Kupac*, *Rukovodilac projekta* učestvuje u aktivnosti:
  1. Provera proizvođača softverske komponente: kako bi se zadovoljio zahtev SM-10, Rukovodilac projekta je potrebno da potvrdi ispravnost ugovora koji je sklopljen sa *Proizvođačem Softverskih Komponenti*.
- *Rukovodilac za Softverske komponente*: pored autentifikacije na sistem, *Rukovodilac za Softverske komponente* je identifikovan u slučaju korišćenja:
  1. Upravljanje repozitorijumom softverskih komponenti: kako bi se zadovoljio zahtev SM-10, gde *Proizvođač Softverskih Komponenti* razvija namenski softver za *Kupca*, neophodno je upravljati repozitorijumom na koji *Proizvođač* postavlja rešenje. Upravljanje repozitorijumom uključuje učestvovanje u aktivnosti postavljanja izvornog koda, za koji je zadužen *Proizvođač softverskih komponenti*.

- Učesnik *Revizor*: nakon što se promeni konfiguracija kanala i svi učesnici konzorcijuma budu saglasni da se *Revizor* doda, na *Revizoru* je:
  1. Revizija kanala: u okviru ovog kanala, *Revizor* ima mogućnost uvida u kreirane dokumente, kao i mogućnost uvida u upitnike vezane za bezbednost *Proizvođača*.

U okviru opisanog dijagrama slučajeva korišćenja, dodat je novi učesnik *Proizvođač Softverskih Komponenti*, sa ciljem ispunjenja zahteva SM-9 i SM-10 iz celine Upravljanje bezbednošću, u okviru standarda IEC 62444-4-1. U okviru poslednjeg dijagrama slučajeva korišćenja, razmatraju se poslednja tri zahteva celine Upravljanje bezbednošću.

5.3.4. Kvalitet

U okviru dijagrama slučajeva korišćenja pod nazivom Kvalitet, grupisani su zahtevi SM-11, SM-12 i SM-13 koji se bave problemima vezanim za ranjivosti, procesnom validacijom i kontinualnim napretkom, respektivno. Kako zahtevi utiču na kvalitet krajnjeg proizvoda, čineći ga se ranjivosti softvera mitiguju pre slanja proizvoda krajnjim korisnicima i da se čitav proces bezbednog razvoja softvera kontinualno unapređuje, identifikovan je dodatni učesnik, a ranijim učesnicima su dodate nove odgovornosti. Na Slici 17 prikazani su učesnici *Test analitičar*, *Rukovodilac projekta*, *Savetnik za bezbednost* i *Revizor*. Kao što je opisano i u prethodnim dijagramima, zajednički slučaj korišćenja za sve učesnike je autentifikacija na sistem pre korišćenja.



Slika 17 Dijagram slučaja korišćenja – Kvalitet

Kako bi se opisali načini na koje se ispunjavaju zahtevi SM-11, SM-12 i SM-13, dijagram slučajeva korišćenja je opisan kroz njegove učesnike:

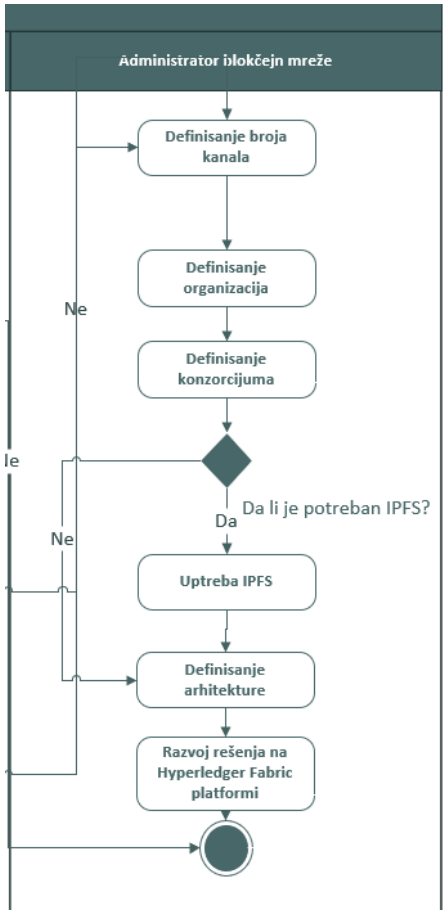
- Učesnik *Test Analitičar*: ranjivost softvera predstavlja slabost u softveru koju neka pretnja može da iskoristi. Kako su slabosti i greške u softveru povezane i najčešće ih pronalaze *Test analitičari*, uveden je učesnik koji je zadužen za sledeći slučaj korišćenja:
    1. Prijavljivanje ranjivosti: kako bi se zadovoljio zahtev SM-11 za praćenje ranjivosti i da krajnji proizvod nije predat klijentu sa poznatim ranjivostima, uloga Test analitičara je da zavodi sve ranjivosti na predefisan način. Kako bi se odredila kritičnost ranjivosti, a spram toga i rizik koja ta ranjivost nosi, prilikom zavođenja ranjivosti u sistem, potrebno je odrediti CVSS (eng. *Common Vulnerability Scoring System*) rezultat [140].
- Učesnik *Rukovodilac Projekta*: u okviru ovog dijagrama, učesniku su dodati sledeći slučaji korišćenja:
    1. Dokumentovanje procesa verifikacije: kako bi se obezbedilo zadovoljenje zahteva SM-12, neophodno je kreirati proces verifikacije, čime bi se osiguralo da novo kreirani proizvod ima dovoljan nivo implementirane bezbednosti pre nego što se ispostavi krajnjim korisnicima. Dokumentovani proces se, kao i ostali napravljeni dokumenti, skladišti na IPFS rešenju, a dobijeni povratni heš od IPFS rešenja se skladišti na blokčejnu, kako bi se mogao pokazati integritet samog dokumenta.
    2. Dokumentovanje proces kontinualnog unapređivanja: pored dokumentovanja procesa verifikacije, neophodno je dokumentovati proces za kontinualno unapređivanje. Kao i za prethodni slučaj korišćenja, nakon postavljanja dokumenta na IPFS rešenje, heš vrednost se skladišti na blokčejn mrežu. Ovim se može pokazati usklađenost sa zahtevom SM-12 koji govori o kontinualno unapređivanju.
    3. Praćenje grešaka: praćenjem grešaka i ranjivosti koje kreirani softver unosi, može se obaviti preciznija analiza rizika i dati uvid u bezbednosno stanje softvera. Obezbeđivanjem da su najrizičnije ranjivosti mitigovane pre isporuke softvera klijentu, *Rukovodilac projekta* doprinosi usklađenosti sa zahtevom SM-11.
    4. Ažuriranje konzorcijuma: kao što je prikazano i u ostalim dijagramima slučaja korišćenja, *Rukovodilac projekta* ima mogućnost ažuriranja konzorcijuma, čime se mogu vršiti izmene u konfiguraciji, omogućavajući novim učesnicima pravo pristupa.
- Učesnik *Savetnik za bezbednost*: u okviru predstavljenog dijagrama slučajeva korišćenja, *Savetniku za bezbednost* su dodati sledeći slučajevi korišćenja:
    1. Praćenje grešaka: kako greške mogu uneti ranjivosti u softver, a *Savetnik za bezbednost* je identifikovan kao učesnik koji poseduje znanje bezbednosti

softvera, u ovom slučaju korišćenja, neophodno je postarati se da su sve greške i ranjivosti mitigovane pre isporuke krajnjim korisnicima, što je u skladu sa zahtevom SM-11.

2. Ažuriranje konzorcijuma: prateći prethodno identifikovane dijagrame, *Savetniku za bezbednost* je data mogućnost ažuriranja konzorcijuma, čime se omogućava izmena konfiguracije kanala.

- Učesnik *Revizor*: nakon dobijanja pristupa kanalu, za *Revizora* je identifikovan slučaj korišćenja:
    1. Revizija kanala: u skladu sa prethodnim zaduženjima *Revizora*, neophodno je utvrditi usaglašenost sa zahtevima, što je omogućeno pristupu ovom kanalu, gde su objedinjeni zahtevi SM-11, SM-12 i SM-13.

Naredni korak u opisanom modelu je konsultacija *Savetnika za bezbednost* i *Rukovodioca projekta*. Kako su se konsultacije mogle obavljati u svakom od prethodnih koraka, neće se posvetiti poseban odeljak za taj korak u okviru modela.



Slika 18 Aktivnosti Administratora blokčejn mreže

Naredna poglavlja se bave definisanjem broja kanala, organizacija, konzorcijuma, upotrebnom IPFS rešenja i postavljanjem arhitekture, što predstavljaju naredne korake opisane u modelu u okviru poglavlja 4 (isečak dostupan na Slici 18).

5.4. Definisanje broja kanala

Sledeći korak koji je opisan u modelu u okviru poglavlja 4 predstavlja definisanje broja kanala, koji *Administrator blokčejn mreže* treba da uradi spram kreiranih slučajeva korišćenja. Kanali predstavljaju dodatnu segregaciju podataka koju nudi Hyperledger Fabric rešenje, čime su podaci skladišteni na jednom kanalu odvojeni od podataka na drugom kanalu. *Blokčejn administrator* analizira kreirane slučajeve korišćenja kako bi se donela odluka o broju kanala. U zavisnosti od načina na koji su kreirani dijagrami slučajeva korišćenja, odnosno nivoa detalja koji su predstavljeni na dijagramu slučajeva korišćenja, broj kanala se može razlikovati. Posmatrajući dijagrame slučajeva korišćenja opisane u prethodnom odeljku, koji su vezani za zahteve u celini Upravljanje bezbednošću u okviru standarda IEC 62443-4-1, *Administrator blokčejn mreže* može definisati tri kanala. Dijagrami slučajeva korišćenja Upravljanje timom i projektom i dijagram Okruženje za razvoj će biti implementirani u okviru jednog kanala, budući da su svi učesnici dijagrama isti, kao i konzorcijumi. Za dijagrame Upravljanje lancem snabdevanja i Kvalitet, biće kreiran po jedan kanal, budući da ti dijagrami imaju uloge koje se ne pojavljuju u ostalim dijagramima, te je potrebno implementirati drugačije pravo pristupa informacijama. Kako se neki učesnici u dijagramima slučajeva korišćenja ponavljaju u više dijagrama, mogao je biti kreiran i samo jedan dijagram slučajeva korišćenja, koji bi obuhvatao sve zahteve iz celine Upravljanje bezbednošću. U tom slučaju, bilo bi potrebno naznačiti kom kanalu pripadaju koji zahtevi iz celine Upravljanja bezbednošću, te se prikazani pogled, pokazao kao rešenje iz kojeg se jednostavnije mogu dobiti neophodne informacije.

5.5. Definisanje organizacija

Kako je ranije pojašnjeno, organizacija predstavlja element koji grupiše učesnike po istoj ili sličnoj potrebi i takvom elementu se daje pristup kanalu, u kontekstu Hyperledger Fabric platforme. Dijagrami slučajeva korišćenja su kreirani na takav način da će svaki učesnik dijagrama postati organizacija u kontekstu Hyperledger Fabric platforme. Uvidom u četiri dijagrama slučajeva korišćenja koji su kreirani za potrebe usklađenosti sa zahtevima celine Upravljanje bezbednošću, identifikovano je šest organizacija (po jedna za svakog učesnika: *Rukovodilac projekta, Savetnik za bezbednost, Rukovodilac za Softverske komponente, Revizor, Proizvođač softverskih komponenti* i *Test Analitičar*).

5.6. Definisanje konzorcijuma

Posmatrajući dokumentaciju, konzorcijum predstavlja skup organizacija koje formiraju kanal, priključuju se kanalu i održavaju čvorove [141]. Učesnik *Revizor*, odnosno organizacija u Hyperledger Fabric terminima, nije deo nijednog konzorcijuma budući da se dodaje samo u slučajevima kada je potrebno vršiti reviziju kanala. Kroz dijagrame slučajeva korišćenja, prikazani su slučajevi korišćenja u kojima je potrebno ažurirati konzorcijum. Budući da je

ažuriranje omogućeno samo članovima konzorcijuma, broj konzorcijuma *koje Administrator blokčejn mreže* treba da definiše odgovara i broju dijagrama slučajeva korišćenja, odnosno potrebna su tri konzorcijuma. Slično kao u slučaju definisanja broja kanala, broj konzorcijuma zavisi od načina kreiranja dijagrama slučajeva korišćenja, nivou detalja i načinu na koji su zahtevi grupisani.

5.7. Upotreba IPFS rešenja

Posmatrajući karakteristike IPFS rešenja, kao što su arhitektura ravnopravnih računara i odsustvo jedinstvene tačke pucanja, IPFS ima slične osobine koje se mogu videti u blokčejn implementacijama. Međutim, IPFS rešenje predstavlja distribuirani fajl sistem koji se oslanja na prethodno implementirana rešenja kao što su BitTorrent i Git [142]. Detaljan dizajn i način implementacije su opisani u okviru rada [142], dok su neki primeri upotrebe opisani kroz nekoliko radova [143], [144], [145]. Sa stanovišta upotrebe IPFS rešenje za potrebe prethodno opisanog modela, najbitniji aspekt jeste skladištenje podataka u okviru distribuiranog sistema, gde se kao potvrda skladištenja datoteke dobija heš vrednost, koja omogućava proveru integriteta skladištene datoteke.

U okviru modela koji je definisan u prethodnom poglavlju, IPFS rešenje bi se koristilo za skladištenje fajlova koji su potrebni za usklađivanje sa bezbednosnim zahtevima. Budući da je na dijagramima slučajeva korišćenja identifikovano više slučajeva u kojima je potrebno kreirati fajl i zatim ga postaviti na IPFS rešenje, IPFS rešenje je potrebno dodati na arhitekturu i povezati sa ostalim komponentama. Jedan od načina korišćenja IPFS rešenja je upotreba javno dostupne platforme [146] koja omogućava skladištenje datoteka i deljenje sa zainteresovanim stranama. Kako nije neophodno kreirati nalog, predloženi pristup doprinosi jednostavnosti upotrebe. Međutim, bezbednost i privatnost fajlova koje su skladištene na taj način je na niskom nivou. Kako je model opisan u prethodnom poglavlju usmeren ka skladištenju podataka koji mogu biti klasifikovani kao osetljivi, upotreba javnog IPFS rešenja bi narušila poverljivost. U tom slučaju potrebno je napraviti privatnu IPFS mrežu, što je obrađeno kroz nekoliko radova [147], [148], a dostupna uputstva omogućavaju lakšu postavku [149], [150]. U okviru narednih odeljaka će biti predstavljana arhitektura rešenja koja podržava identifikovane dijagrame slučajeve korišćenja i upotrebu privatne IPFS mreže, sa ciljem zadržavanja poverljivost podataka koje je potrebno skladištiti.

5.8. Potvrda hipoteze H2

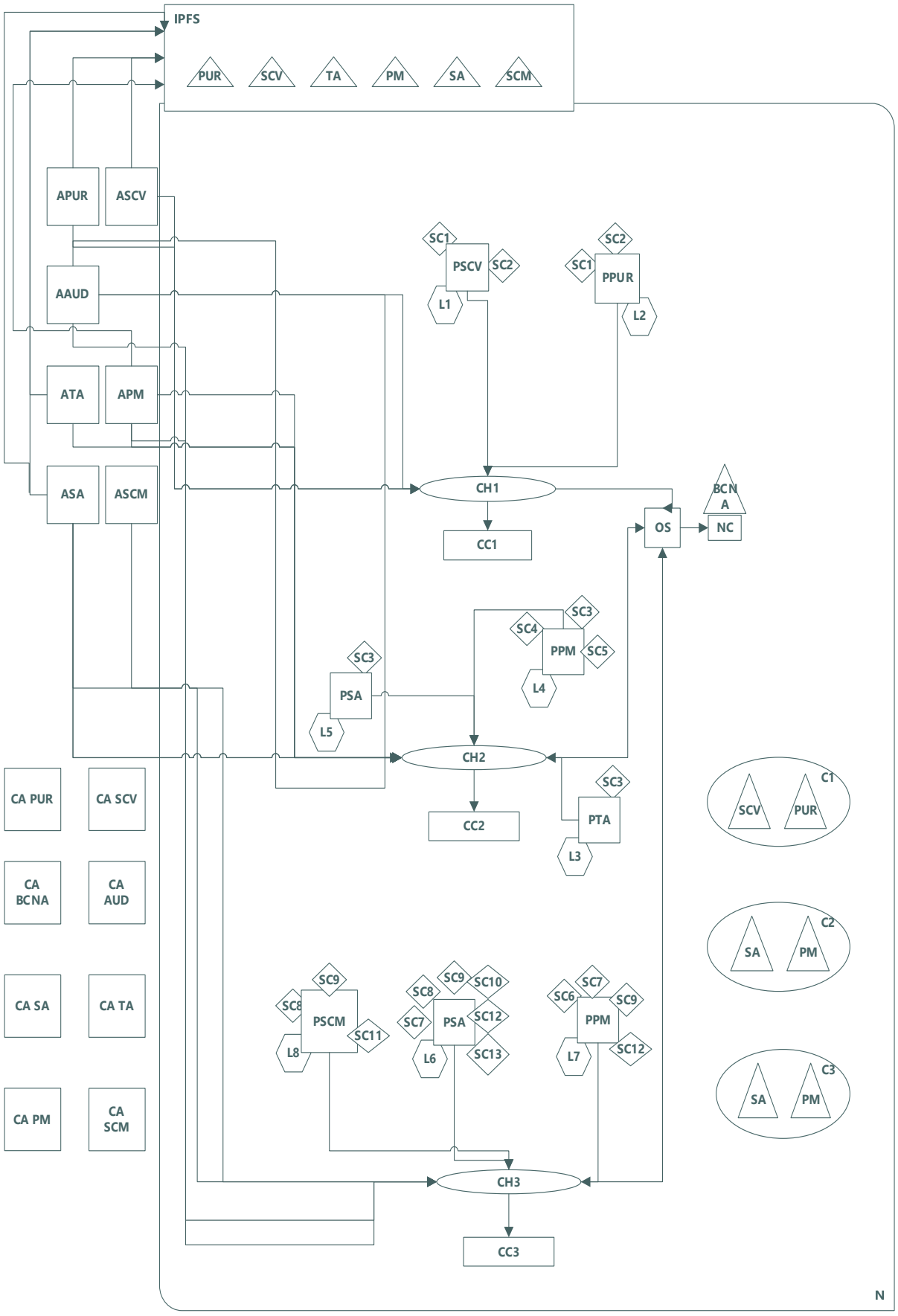
Osvrtom na definisanu hipotezu **H2** (*Moguće je definisati učesnike u procesu praćenja zahteva, njihove slučajeve korišćenja, uzimajući u obzir osetljivost podataka sa kojima učesnici rukuju i prateći princip da su podaci obezbeđeni samo ovlašćenim pojedincima, neophodnim za obavljanje svojih dužnosti*), zaključujemo da se koracima koji su opisani u prethodnim odeljcima potvrđuje hipoteza **H2**. Pored prethodno identifikovanih *Savetnika za bezbednost*,

Rukovodioca projekta i *Administratora blokčejn mreže*, identifikovani su i *Rukovodilac za Softverske Komponente*, *Revizor*, *Proizvođač Softverske komponente*, *Kupac* i *Test Analitičar*. Uloge su posmatrane kroz dijagrame slučajeva korišćenja, čime se jasno identifikuju odgovornosti svake od prepoznatih uloga. Tom prilikom, vođeno je računa o osetljivosti podataka, čime svaka uloga ima dostupan minimalan neophodan skup informacija. Takav princip obezbeđuje da se smanji rizik od neovlašćenog pristupa ili zloupotrebe podataka, što dodatno doprinosi ukupnoj pouzdanosti i sigurnosti procesa.

5.9. Postavka arhitekture rešenja

Nakon odrađenih prethodnih aktivnosti predloženih modelom, potrebno je kreirati arhitekturu rešenja, koju je moguće postaviti na Hyperledger Fabric platformu. Prilikom kreiranje arhitekture rešenja, potrebno je uzeti u obzir kreirane dijagrame slučajeva korišćenja, broj kanala, organizacija i potreba za IPFS rešenjem. Informacije analizirane u prethodnim odeljcima, iskorišćene su za kreiranje arhitekture koja je predstavljena na Slici 20. Osnovu za kreiranje arhitekture rešenja predstavljaju jednostavna arhitektura koja je predložena u okviru dokumentacije [106], kao i radovi [151], [152] u kojima je predstavljena arhitektura rešenja samo za dijagram slučajeva korišćenja Upravljanje lancem snabdevanja.

U okviru arhitekture predstavljene na Slici 19, prikazani su svi učesnici identifikovani u dijagramima slučajeva korišćenja. Učesnici su predstavljeni kao organizacije, koji mogu da pristupaju kanalima, na kojima su skladištene potrebne informacije za usklađivanje sa bezbednosnim zahtevima, kao i aplikacije preko kojih organizacije mogu da pristupe tim informacijama. Kako bi se proverilo da li svaka član organizacije ima odgovarajuće kredencijale, svaka organizacija ima svoju kompaniju za upravljanje sertifikatima. Takođe, na arhitekturi su prikazani konzorcijumi, kreirani po prikazanim dijagramima, kao i IPFS rešenje koje se koristi za skladištenje datoteka. Na predstavljenoj arhitekturi su 3 kanala, koja su identifikovana u prethodnim odeljcima.



Slika 19 Arhitektura rešenja za praćenje usklađenosti sa zahtevima u okviru celine Upravljanje bezbednošću

Prvi kanal koji je kreiran, predstavlja kanal koji je ranije opisan u radovima [151] i [152], a vezan je za dijagram slučajeva korišćenja Upravljanje lancem snabdevanja. Kada se posmatra kanal broj 1, označen sa CH1 i pripadajućom konfiguracijom, označenom sa CC1, uočavamo i čvorove PSCV i PPUR, na kojima se nalazi programski kodovi lanaca i knjige. Čvorovi su kreirani kako bi služili organizacijama SCV i PUR, koji su organizacije za učesnike *Rukovodilac za softverske komponente* i *Kupac*, koji su identifikovani na dijagramu slučajeva korišćenja. Na čvoru PSCV su postavljena dva programska koda lanaca, sa oznakama SC1 i SC2. Programski kodovi lanaca su napisani kako bi se mogle skladištiti informacije na knjizi L1, koja je dedikovana čvoru PSCV. Programski kod lanca SC1 je napisan za potrebe ispunjavanja zahteva SM-9 iz celine Upravljanja bezbednošću, dok je programski kod lanaca SC2 napisan za potrebe ispunjavanja zahteva SM-10 iz iste celine zahteva. Sa druge strane, na čvoru PPUR nalazi se isti programski kodovi lanaca, budući da obe organizacije doprinose usklađenosti za date zahteve. Jedina razlika je što čvor PPUR ima svoju instanciranu knjigu označenu sa L2. Kanal CH1 je povezan sa *Uređivačem*, označenim sa OS, koji će odrediti redosled transakcija koje će se upisati u knjigu. Kako bi se inicirao programski kod lanca, neophodno je imati aplikaciju koja pristupa čvoru i inicira programski kod lanca i knjigu. Svaka organizacija imaju svoju aplikaciju za pristup svom čvoru, te tako aplikacija ASCV pristupa čvoru PSCV kako bi upisala i pročitala informacije koje su skladištene u knjizi. Posledično, postoji i aplikacija APUR za pristup programskom kodu lanca i knjizi koji se nalaze na čvoru PPUR. Kako bi se proverili kredencijalni korisnika koji pristupaju kanalu, kreirane su kompanije za upravljanje sertifikatima, CASCV i CAPUR, za organizacije SCV i PUR, respektivno. Organizacije SCV i PUR čine konzorcijum koji je predstavljen na arhitekturi i označen sa C1. Na dijagramu slučajeva korišćenja predstavljen je slučaj u kojem je potrebno skladištiti datoteku na IPFS rešenje, te je na predstavljenoj arhitekturi dodato IPFS rešenje kojem mogu da pristupe aplikacije ASCV i APUR.

Kanal sa oznakom CH2, kreiran je za potrebe dijagrama slučajeva korišćenja Kvalitet, gde su identifikovana četiri učesnika, odnosno četiri organizacije, koje su označene SA, PM, AUD i TA. Kao i za prethodni kanal, kanal CH2 ima svoju konfiguraciju koja je označena sa CC2 i povezan je sa Uređivačem OS. Kada je u pitanju broj čvorova na kojima se nalaze programski kod lanca i knjiga, on je isti kao i broj organizacija, budući da će na taj način svaka organizacija pristupati kanalu. Na arhitekturi su predstavljeni PSA, PPM i PTA koji su čvorovi za organizacije *Savetnika za bezbednost*, *Rukovodioca projekta i Testnog analitičara*, respektivno. Na čvoru PTA nalazi se programski kod lanaca sa oznakom SC3 u okviru kojeg se nalazi kod koji omogućava skladištenje informacija na knjigu L3, kako bi se dostigla usklađenost sa zahtevom SM-11. Isti programski kod lanaca je postavljen i na čvorove PSA i PPM, budući da i te organizacije doprinose usklađivanju sa zahtevom SM-11 te im je neophodan mehanizam da isti način skladište informacije. Knjiga koja održava lanac na čvoru PSA je označena sa L4, dok je knjiga na čvoru PPM označena brojem L5. Dodatno, na čvoru PPM se nalaze još dva programska koda lanca jer je učesnik *Rukovodilac projekta*, odnosno organizacija, zadužena za usklađivanje sa zahtevima SM-12 i SM-13, zbog čega su kreirani programski kodovi lanaca i

označeni sa SC4 i SC5, respektivno. Takođe, na arhitekturi su predstavljene i aplikacije putem kojih organizacije pristupaju kanalu, označene sa ASA, APM i ATA, kao i odgovarajuće kompanije za upravljanje sertifikatima CASA, CAPM, CATA, za Savetnika za bezbednost, *Rukovodioca projekta* i *Testnog analitičara*, respektivno. Konzorcijum je kreiran između organizacija *Savetnika za bezbednost i Rukovodioca projekta* i označena je sa C2. Kao i za prethodni kanal, dodato je IPFS rešenje kako bi se mogle skladištiti datoteke neophodne za pokazivanje usklađenosti sa bezbednosnim zahtevima.

Poslednji kanal, koji je označen sa CH3, kreiran je kako bi se pratila usklađenost sa zahtevima koji su opisani na dijagramu slučajeva korišćenja Upravljanje timom i projektom i dijagramu Okruženje za razvoj. Budući da su konzorcijumi identifikovani za ta dva dijagrama slučajeva korišćenja, kao i ostali učesnici na dijagramu, identični, moguće je spojiti dva slučaja korišćenja na jedan kanal. Kanal CH3 ima svoju konfiguraciju CC3 i kao i prethodni kanali, povezan je sa Uređivačem, odnosno OS elementom na arhitekturi. Čvorovi na kojima su postavljeni programski kodovi lanaca i knjige su obeleženi sa PSA, PPM i PSCM, za organizacije *Savetnika za bezbednost*, *Rukovodioca projekta* i *Rukovodioca za softverske komponente*, respektivno. Čvorovi PSA, PPM i PSCM imaju svoje knjige koje održavaju i označene su sa L6, L7 i L8, respektivno. Broj programski kodova lanaca koji su postavljeni na čvorove je različit za svaki čvor i zavisi od slučajeva korišćenja i broj zahteva koje ta organizacije treba da ispuni. Jedini programski kod lanaca koji je postavljen na sva tri čvora je SC9, koji je potreban za ispunjenje zahteva SM-4, dok je pregled programskih kodova lanaca po čvorovima dat niže:

- Čvor PSCM: kako bi se ispunili zahtevi SM-3 i SM-6, opisani na dijagramima slučajeva korišćenja Upravljanje timom i projektom, odnosno dijagramom Okruženje za razvoj, na čvoru PSCM postavljeni su i programski kodovi lanaca sa oznakama SC8 i SC11.
- Čvor PPM: budući da je učesnik Rukovodilac projekat uključen u veći broj aktivnosti. koje su identifikovane na dijagramima slučajeva korišćenja Upravljanje timom i projektom, kao i na dijagramu Okruženje za razvoj, na čvoru PPM postoji veći broj programskih kodova lanaca. Tako su kreirani programski kodovi lanaca SC6, SC7 i SC12 kako bi se omogućilo skladištenje informacija koje su neophodne za usklađivanje sa zahtevima SM-1, SM-2 i SM-7, respektivno.
- Čvor PSA: kada je u pitanju dijagram slučajeva korišćenja Upravljanje timom i projektom, za potrebe usklađivanja sa zahtevima identifikovanim na tom dijagramu, definisani su programski kodovi lanaca SC7, SC8 i SC10, koji pomažu skladištenju informacija neophodnih za usklađivanje sa zahtevima SM-2, SM-3 i SM-5, respektivno. Za potrebe usklađivanja sa zahtevima koji su predstavljeni na dijagramu slučajeva korišćenja Okruženje za razvoj, na čvor PSA su dodati i programski kodovi lanaca SC12 i SC13, kako bi se omogućilo skladištenje informacija vezanih za zahteve SM-7 i SM-8, respektivno.

Organizacije *Savetnika za bezbednost* i *Rukovodioca projekta* koriste već kreirana sertifikaciona tela, dok je za organizaciju *Rukovodioca za softverske komponente* dodato sertifikaciono telo sa oznakom CASCM. Posledično, kreirana je i aplikacija putem koje će se pristupati čvoru PSCM i ima oznaku ASCM. Kao i za prethodna dva kanala, dodato je IPFS rešenje kako bi se mogle skladištiti datoteke neophodne za pokazivanje usklađenosti sa bezbednosnim zahtevima.

Radi lakšeg pregleda, u Tabeli 10, prikazani su prethodno opisani kanali, sa pripadajućim čvorovima, knjigama, programskim lancima i zahtevima u okviru IEC 62443-4-1 standarda koji su ispunjeni na tim kanalima.

Tabela 10 Pregled kanala, programskih lanaca i IEC 62443-4-1 zahteva

Ime kanala	Ime čvora	Ime knjige	Ime programskog lanca	IEC 62443-4-1 zahtev
CH1	PSCV	L1	SC1, SC2	SM-9, SM-10
	PPUR	L2	SC1, SC2	
CH2	PTA	L3	SC3	SM-11, SM-12, SM-13
	PPM	L4	SC3, SC4, SC5	
	PSA	L5	SC3	
CH3	PSA	L6	SC6, SC7, SC8, SC9, SC10, SC12, SC13	SM-1, SM-2, SM-3, SM4, SM-5, SM-6, SM-7, SM-8
	PPM	L7	SC6, SC7, SC9, SC12	
	PSCM	L8	SC8, SC9, SC11	

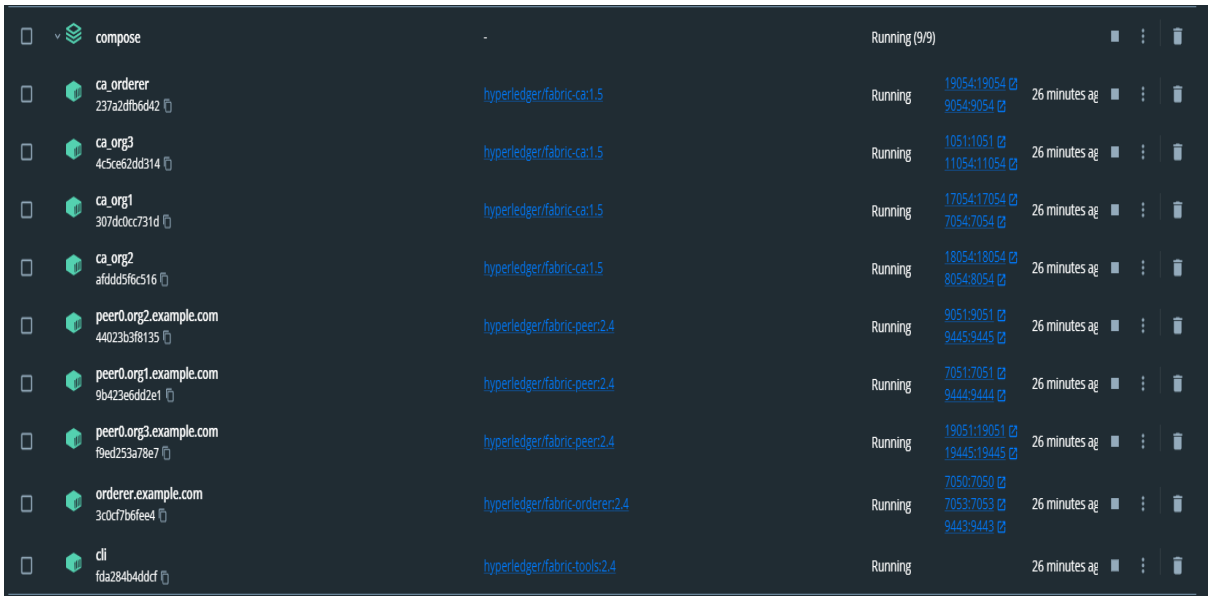
5.10. Postavljanje rešenja na Hyperledger Fabric platformu

Za potrebe verifikacije rešenja na Hyperledger Fabric platformu, iskorišćena je postojeća dokumentacija za postavljanje testne Hyperledger Fabric mreže [153]. Kako bi se kreirala testna mreža, potrebno je podesiti okruženje za rad i to se sastoji iz sledećih potrebnih alata [154]:

- Docker desktop: okruženje dostupno za različite operativne sisteme, kojim se omogućava kreiranje, pokretanje i razvoj aplikacija koristeći kontejnere.

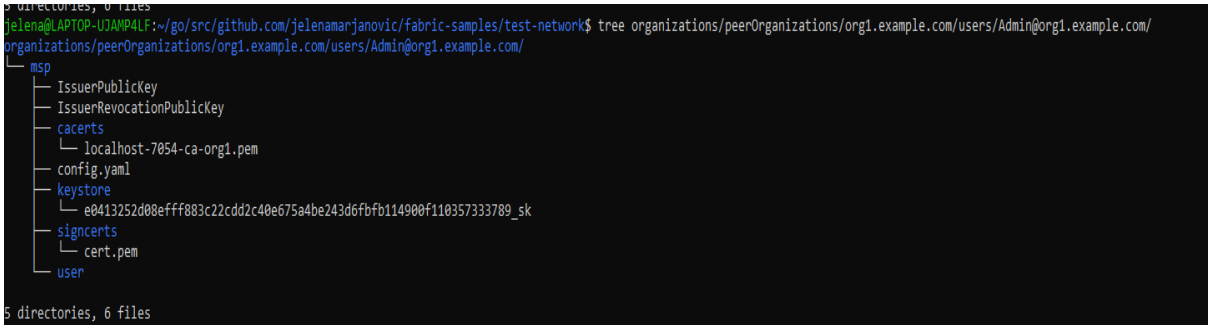
- Windows Subsystem for Linux verzija 2 (WSL2) [155]: sloj u okviru Windows operativnog sistema koji omogućava jednostavan pristup alatima i mogućnostima koje pružaju Linux operativni sistemi, bez potrebe da se kreira virtuelna mašina sa Linux instalacijom. Ukoliko se rešenje razvija na Linux platformi, ovaj korak nije potrebno realizovati.

Nakon podešavanja okruženja, proširenjem izvornog koda u [156] moguće je pokrenuti testnu mrežu koja se sastoji iz dve organizacije, dok svaka organizacije upravlja jednim čvorom, koji su imenovani *peer0.org1.example.com* i *peer0.org2.example.com*. Postojeći izvodni kod je modifikovan na način da se podrže tri organizacije, gde svaka organizacija ima svoj odgovarajući čvor. Takođe, svaka organizacija ima svoje sertifikaciono telo, te je za potrebe ovog rešenja dodata i kompanija za upravljanje sertifikatima koja služi novo dodatoj organizaciji. Nakon ponovnog podizanja testne mreže, na Slici 20 se vide Docker kontejneri za svaku organizaciju, pripadajući kontejneri za sertifikaciona tela, verzija *image*-a koja je iskorišćena za kreiranje kontejnera, status kontejnera (da li je operativan) i portovi koji su otvoreni za komunikaciju sa ostalim kontejnerima.

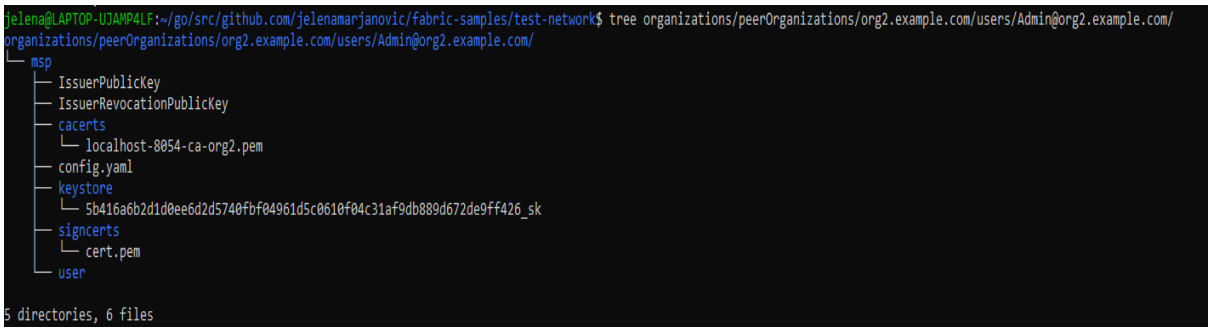


Slika 20 Kreirani Docker kontejneri za rad sa Hyperledger Fabric mrežom

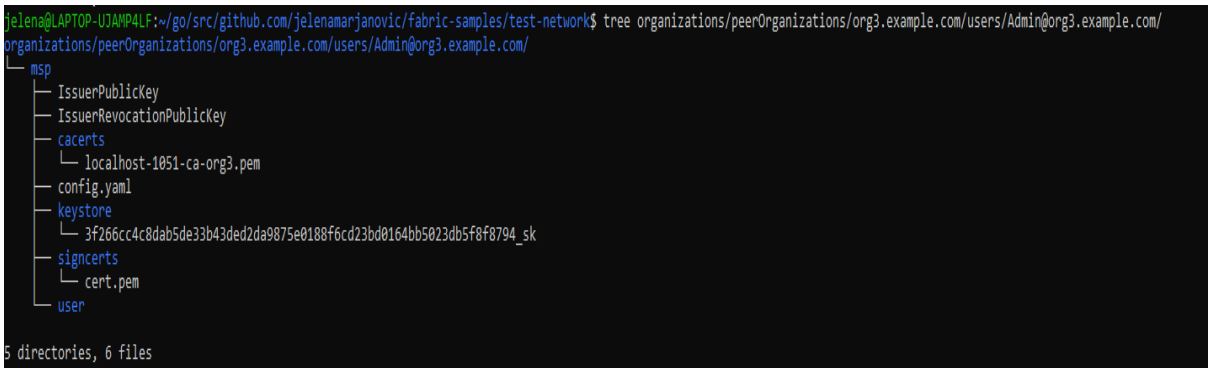
Podizanjem testne mreže kao što je prikazano na Slici 20, generišu se i sertifikati i privatni ključevi za svaki identitet. Na Slikama 21, 22, 23 prikazani su sertifikati administratora svake od organizacija org1, org2 i org3, respektivno, kao i njihove privatne ključeve u okviru foldera *Keystore*.



Slika 21 Sertifikat i privatni ključ za Org1



Slika 22 Sertifikat i privatni ključ za Org2



Slika 23 Sertifikat i privatni ključ za Org3

Da bi se omogućilo dodavanje nove organizacije, sa svojom kompanijom za upravljanje sertifikatima, potrebno je načiniti sledeće izmene na datim fajlovima javno dostupnog izvornog koda:

- *../fabric-samples/test-network/network.sh:* potrebno je dodati org3 kao novi CA
- *../fabric-samples/test-network/compose/compose-ca.yaml:* potrebno je dodati org3 kao novi CA
- *../fabric-samples/test-network/compose/compose-test-net.yaml:* potrebno je dodati org3

- *../fabric-samples/test-network/compose/docker/docker-compose-test-net.yaml:* potrebno je dodati org3
- *../fabric-samples/test-network/scripts/deployCC.sh:* fajl je potrebno izmeniti na takav način da se čejnkod postavlja na čvorove na kojima je to potrebno
- *../fabric-samples/test-network/configtx/configtx.yaml:* potrebno je dodati org3
- *../fabric-samples/test-network/scripts/createChannel.sh:* potrebno je dodati org3.

Ukoliko je testna mreža bila pokrenuta komandom *./network.sh,* koja se nalazi u okviru testnog foldera *../fabric-samples/test-network,* za dalji rad sa mrežom potrebno je kreirati kanale i postaviti čejnkod na odgovorajuće čvorove. Za potrebe kreiranja kanala, može se iskoristiti dostupna komanda *createChannel -c mychannel -ca,* dok se za postavljanje čejnkoda koristi komanda *deployCC -ccn basic -ccp ../chaincode-java/ -ccl java.* Nakon toga, mreža je spremna za skladištenje informacija, koji ostaju zapisani na kreiranim kanalima i knjigama. Deo čejnkoda za upravljanje dokumentima je prikazan niže u Listingu 1:

```
@DataType()
public final class Document {
	@Property()
	private final String documentHash;
	@Property()
	private final String documentTitle;
	@Property()
	private final String owner;
	public String getDocumentHash() {
		return documentHash;
	}
	public String getDocumentTitle () {
		return documentTitle;
	}
	public String getOwner() {
		return owner;
	}
	public Document(@JsonProperty("documentHash") final String
	documentHash, @JsonProperty("documentTitle") final String
	documentTitle, @JsonProperty("owner") final String owner) {
		this.documentHash = documentHash;
		this.documentTitle = documentTitle;
		this.owner = owner;
	}
}
```

Listing 1 Čejnkod za upravljanje dokumentima

U okviru Listinga 1, kreirana je klasa koja omogućava rad sa dokumentima koje je kasnije neophodno sačuvati na IPFS rešenju. Kao neophodne informacije koje se čuvaju u vezi dokumenata predložene su `documentHash` (heš vrednost dokumenta), `documentTitle` (naziv dokumenta koji je potrebno sačuvati) i `owner` (naziv vlasnika dokumenta). U slučaju dodatnih neophodnih informacija, neophodno je proširiti klasu `Document` sa potrebnim atributima. U okviru klase, definisane su i metode za pristup vrednostima atributa (`getDocumentHash, getDocumentTitle, getOwner`) , kao i konstruktor klase u okviru kojeg je moguće inicijalizovati vrednosti atributa klase. Za dalji rad sa dokumentima, prikazan je deo pametnog ugovora za upravljanje dokumentima, u okviru Listinga 2:

```
@Contract(
name = "testDocCont", info = @Info(title = "Docs to stub",
description = "Testing docs to stub"))
@Default
public final class DocumentContract implements ContractInterface {
    private final Genson genson = new Genson();
    @Transaction(intent = Transaction.TYPE.SUBMIT)
    public Document CreateDocument(final Context ctx, final String
documentHash, final String documentTitle, final String owner) {
        ChaincodeStub stub = ctx.getStub();
        Document doc = new Document(documentHash, documentTitle,
owner);
        String sortedJson = genson.serialize(doc);
        stub.putStringState(documentHash, sortedJson);
        return doc;
    }
}
```

Listing 2 Deo pametnog ugovora za upravljanje dokumentima

U okviru Listinga 2, prikazan je deo pametnog ugovor koji omogućava upravljanje dokumentima. U okviru klase `DocumentContract`, kreirana je metoda `CreateDocument`, u okviru koje se kreira objekat `doc`. Budući da takav objekat nije moguće postaviti na Hyperledger Fabric mrežu, iskorišćena metoda `serialize` klase `Genson` za serijalizaciju `doc`, čime se dobija JSON format objekta `doc`. Zajedno sa heš vrednosti dokumenta `documentHash`, `sortedJson` koji predstavlja JSON vrednost objekta `doc`, postavljano je na Hyperledger Fabric mrežu koristeći metodu `putStringState`. Za potrebe postavljanja fajla na IPFS rešenje, a zatim postavljanje dobijene heš vrednost na Hyperledger fabric, korišćen je deo koda prikazan niže u Listingu 3:

```
private final Contract contract;

private void createDocumentoOnIPFS(String documentLocation, String
documentTitle, String owner) {

      IPFS ipfs = new IPFS("/ip4/127.0.0.1/tcp/5001");

      NamedStreamable.FileWrapper file = new
NamedStreamable.FileWrapper(new File(documentLocation));

      MerkleNode uploadedDoc = ipfs.add(file).get(0);

      contract.submitTransaction("CreateDocument",
uploadedDoc.hash.toString(), documentTitle, owner);

}
```

Listing 3 Deo koda za postavljanje fajla na IPFS rešenje

U okviru Listinga 3, prikazan je deo koda koji je potreban za postavljanje fajla na IPFS rešenje i poziv pametnog ugovora za čuvanje dobijene heš vrednosti na Hyperledger Fabric rešenju. U okviru metode, IPFS rešenje koje je korišćeno je u lokalu, dok u slučaju drugog IPFS rešenja postavlja se drugačija putanja u odnosu na `/ip4/127.0.0.1/tcp/5001`. Nakon postavljanja fajla na IPFS rešenje, dobijeni heš `uploadedDoc` se zajedno sa ostalim informacijama o dokumentu (`documentTitle`, `owner`), šalje pametnom ugovoru `contract` kroz metodu `submitTransaction`, gde se parametrom `"CreateDocument"` ukazuje na metodu koja će biti pozvana u okviru pametnog ugovora.

5.11. Potvrda hipoteze H3

U prethodnim odeljcima prikazana je arhitektura rešenja, koja obuhvata tri kanala kroz koja definisane organizacije mogu da sarađuju, osam sertifikationih tela koja obezbeđuju da samo prethodno autentifikovane i autorizovane uloge mogu koristiti rešenje, kao i IPFS rešenje koje je neophodno za skladištenje dokumenata. Takođe, prikazani su kreirani kontejneri koji su neophodni za pokretanje koda. U okviru Listinga 1,2 i 3, predstavljeni su delovi koda koji omogućavaju rad sa dokumentima, njihovo skladištenje na IPFS rešenje, kao i čuvanje na Hyperledger Fabric rešenju. Ovim prethodnim odeljcima je potvrđena hipoteza **H3**, budući da je moguće validirati model koristeći principe rada blokčejn tehnologija, konkretno Hyperledger Fabric rešenja koje pruža pristup informacijama prethodno registrovanim i autorizovanim korisnicima.

6. Zaključak

U okviru ove disertacije, istražena je mogućnost upotrebe Hyperledger Fabric blokčejn rešenja, za praćenje usklađenosti softvera sa bezbednosnim zahtevima. Kroz hipoteze koje su definisane na početku istraživanja, pokazano je da je moguće definisati model za praćenje usklađenosti softvera sa zahtevima za bezbedan razvoj softvera, a koji se odnose na industrijske upravljačke sisteme. Takođe je pokazano da je moguće definisati učesnike u procesu praćenja zahteva, njihove slučaje korišćenja, uzimajući u obzir osetljivost podataka sa kojima učesnici rukuju i prateći princip da su podaci obezbeđeni samo ovlašćenim pojedincima, neophodnim za obavljanje svojih dužnosti. Na kraju, pokazano je da je moguće validirati model koristeći principe rada blokčejn tehnologija, konkretno Hyperledger Fabric rešenja koje pruža pristup informacijama prethodno registrovanim i autorizovanim korisnicima.

Pokazani pristup praćenja zahteva omogućava kako korisnicima sistema, tako i regulatornim telima da imaju uvid u usklađenost softvera sa zahtevima, bez potrebe za brigom o transparentnosti, neporecivosti, sledljivosti i dostupnosti, budući da su to osobine koje su dobijene upotrebom Hyperledger Fabric rešenja. Kako su softveri u industrijskim upravljačkim sistemima od izuzetne važnost u kritičnim infrastrukturama, jer informacije dobijene o takvim sistemima mogu dovesti do problema u slučaju zlonamernih aktivnosti hakera, poverljivost informacija obezbeđena je upotrebom privatne blokčejn mreže kao što je Hyperledger Fabric. Za potrebe validacije modela, analizirana je bezbednosna praksa Upravljanje bezbednošću, koja je deo standarda IEC 62443-4-1, koja se sastoji iz 13 zahteva. To predstavlja prvi korak modela za praćenje usklađenosti softvera sa bezbednosnim zahtevima. Nakon odabira standarda ili stručne smernice, izvršena je analiza obima primenljivosti, gde je konstatovano da su za dati primer, svi zahtevi iz prakse Upravljanje bezbednošću primenljivi. Sledeći korak je kreiranje dijagrama slučajeva korišćenja, gde su nastala četiri slučaja korišćenja, grupisana na takav način da su slični zahtevi postavljeni na zajednički dijagram slučajeva korišćenja. Nakon toga, definisani su kanali, organizacije i konzorcijumi koji su neophodni za funkcionisanje datog primera. Kako je ustanovljeno da je potrebno skladištiti dokumente kao deo rešenje, upotrebljeno je IPFS rešenje za skladištenje. Svi do sada opisani koraci bili su neophodni kako bi se definisala arhitektura rešenja. Kao poslednji korak, urađeno je postavljanje rešenja na Hyperledger Fabric blokčejn mrežu.

Sagledavši doprinose ove doktorske disertacije, kao naredni pravci istraživanja, ističu se sledećih nekoliko pravaca:

- Proširenje modela za podršku drugih sektora: bezbednost nije neophodno jedinu u kritičnim infrastrukturama, već i u drugim sektorima. I dok je IEC 62443-4-1 standard okrenut ka razvoju proizvoda na bezbedan način, prateći zahteve definisane u okviru tog standarda, postoje i drugi bezbednosni standardi, regulative i stručne smernice koje

mogu zahtevati izmenjeni model za praćenje usklađenosti. Stoga je neophodno sagledati i druge sektore, kako bi se mogao predložiti unificirani model za sve sektore.

- Upotreba drugih blokčejn mreža: ukoliko se u prethodnom pravcu istraživanja pokaže da je potrebno da je usklađenost sa zahtevima javno dostupna, potrebno je istražiti upotrebu javnih blokčejn mreža i prilagoditi model za praćenje usklađenosti, po potrebi.
- Optimizacija modela za praćenje usklađenosti softvera: ovaj pravac okrenut je velikom broju zahteva koji se mogu nametnuti industrijskim upravljačkim sistemima, kao i zahtevima koji su došli iz različitih standarda ili regulativa. Određeni zahtevi mogu biti prisutni u više različitih standarda, čija usklađenost može biti na različitom nivou, što može dovesti do postojanja duplih informacija o usklađenosti. Takva situacija dovodi do višestrukog održavanja usklađenosti, što može rezultirati nepotpunim informacijama.
- Integracija sa postojećim rešenjima: u određenim situacijama, usklađenost softvera sa zahtevima je već definisana i zato su korišćeni različiti trenutno dostupni alati. S obzirom na prednosti koje donosi upotreba privatnog blokčejna, kao što su otpornost, transparentnost, neporecivost i sledljivost, potrebno je istražiti slučaj kada je usklađenost već definisana, ali se kao deo unapređenja predlaže prelazak na blokčejn baziran model za praćenje usklađenosti.

Literatura

[1] I. a. K. P. a. P. M. a. A. C. a. L. J. Stellios, „A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Communications Surveys & Tutorials,* t. 20, br. 4, pp. 453--3495, 2018.

[2] S. K. a. P. J. a. S. R. Venkatachary, „Cybersecurity and cyber terrorism-in energy sector--a review," *Journal of Cyber Security Technology,* t. 2, br. 3-4, pp. 111--130, 2018.

[3] [Na mreži]. Available: https://energy.utexas.edu/research/ercot-blackout-2021.

[4] „https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience/critical-infrastructure-sectors," [Na mreži].

[5] D. Kushner, „The real story of stuxnet," *IEEE Spectrum,* pp. 48-53, 2013.

[6] M. Zhivich i R. K. Cunningham, „The real cost of software errors," *IEEE Security & Privacy,* pp. 87-90, 2009.

[7] J. Graham, J. Hieb i J. Naber, „Improving cybersecurity for industrial control systems," u *2016 ieee 25th international symposium on industrial electronics (isie)*, 2016, pp. 618--623.

[8] P. G. Neumann, „Risks to the public in computers and related systems," *ACM SIGSOFT Software Engineering Notes,* t. 25, br. 3, pp. 15-23, 2000.

[9] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos i R. Karri, „The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE,* t. 104, br. 5, pp. 1039--1057, 2016.

[10] D. C. Smith, „Cybersecurity in the energy sector: are we really prepared?," *Journal of Energy & Natural Resources Law,* t. 39, br. 3, pp. 265--270, 2021.

[11] T. H. Morris i W. Gao, „Industrial control system cyber attacks," u *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*, 2013, pp. 22--29.

[12] Z. Drias, A. Serhrouchni i O. Vogel, „Analysis of cyber security for industrial control systems," u *2015 international conference on cyber security of smart cities, industrial control system and communications (ssic)*, 2015, pp. 1--8.

[13] L. A. Maglaras, K.-H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras i T. J. Cruz, „Cyber security of critical infrastructures," *Ict Express,* t. 4, br. 1, pp. 42--45, 2018.

[14] M. A. Nasir, S. Sultan, S. Nefti-Meziani i U. Manzoor, „Potential cyber-attacks against global oil supply chain," u *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2015, pp. 1--7.

[15] Q. A. Al-Haija i S. Brahma, „Optimization of Cyber System Survivability Under Attacks Using Redundancy of Components," u *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, 2019, pp. 1-6.

[16] K. Bjerke-Gulstuen, E. W. Larsen, T. St{\aa}lhane i T. Dings{\o}yr, „High level test driven development--Shift left," u *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings 16*, Springer, 2015, pp. 239--247.

[17] I. E. C. a. others, „IEC 62443-4-1: 2018 Security for Industrial Automation and Control Systems--Part 4-1: Secure Product Development Lifecycle Requirements," *International Electrotechnical Commissio,* 2018.

[18] D. Pandey, U. Suman i A. K. Ramani, „An effective requirement engineering process model for software development and requirements management," u *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, 2010 , pp. 287--291.

[19] C. B. Haley, J. D. Moffett, R. Laney i B. Nuseibeh, „A framework for security requirements engineering," u *Proceedings of the 2006 international workshop on Software engineering for secure systems*, 2006, pp. 35--42.

[20] D. Mishra, A. Mishra i A. Yazici, „Successful requirement elicitation by combining requirement engineering techniques," u *2008 First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, 2008 , pp. 258--263.

[21] L. Fiorineschi, N. Becattini, Y. Borgianni i F. Rotini, „Testing a new structured tool for supporting requirements' formulation and decomposition," *Applied Sciences,* t. 10, br. 9, p. 3259, 2020.

[22] V. Gupta, J. M. Fernandez-Crehuet, T. Hanne i R. Telesko, „Requirements engineering in software startups: A systematic mapping study," *Applied Sciences,* t. 10, br. 17, p. 6125, 2020.

[23] A. Mengist, L. Buffoni i A. Pop, „An integrated framework for traceability and impact analysis in requirements verification of cyber--physical systems," *Electronics,* t. 10, br. 8, p. 983, 2021.

[24] S. U. Rehman i V. Gruhn, „An effective security requirements engineering framework for cyber-physical systems," *Technologies,* t. 6, br. 3, p. 65, 2018.

[25] J. Golosova i A. Romanovs, „The advantages and disadvantages of the blockchain technology," u *2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE)*, 2018 , pp. 1--6.

[26] S. JAIN, Programming Hyperledger Fabric: Creating enterprise blockchain applications, 2020.

[27] E. Venson, X. Guo, Z. Yan i B. Boehm, „Costing secure software development: A systematic mapping study," u *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1-11.

[28] M. Kainerstorfer, J. Sametinger i A. Wiesauer, „Software security for small development teams: a case study," u *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, 2011, pp. 305--310.

[29] P. Maier, Z. Ma i R. Bloem, „Towards a secure scrum process for agile web application development," u *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1-8.

[30] T. W. Thomas, M. Tabassum, B. Chu i H. Lipford, „Security during application development: An application security expert perspective," u *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1--12.

[31] K. Stouffer, J. Falco, K. Scarfone i others, „Guide to industrial control systems (ICS) security," *NIST special publication,* t. 800, br. 82, 2011.

[32] [Na mreži]. Available: https://www.techtarget.com/whatis/definition/industrial-control-system-ICS. [Poslednji pristup April 2023].

[33] J.-M. Flaus, Components of an industrial control system, Wiley Data and Cybersecurity, 2019.

[34] [Na mreži]. Available: https://app.dimensions.ai/discover/publication?search_mode=content&search_text=critical%20infrastructure%20cybersecurity&search_type=kws&search_field=full_search. [Poslednji pristup April 2023].

[35] [Na mreži]. Available: https://transformers-magazine.com/tm-news/395-power-transformer-shooting-could-be-terrorism/. [Poslednji pristup April 2023].

[36] [Na mreži]. Available: https://www.politico.com/newsletters/power-switch/2022/12/05/who-shot-the-north-carolina-power-grid-00072235. [Poslednji pristup April 2023].

[37] J. T. a. I. T. Force, „Security and privacy controls for federal information systems and organizations," *NIST Special Publication,* t. 800, br. 53, pp. 8--13, 2013.

[38] [Na mreži]. Available: https://iso.org.rs/iso-27001/. [Poslednji pristup April 2023].

[39] L. A. Alexei, „Ensuring information security in public organizations in the Republic of Moldova through the ISO 27001 standard," *Journal of Social Sciences,* t. 1, br. 4, pp. 84--94, 2021.

[40] V. Monev, „ISO 27001 Framework for Securing Election Infrastructure and Machine Voting," u *2022 International Conference on Information Technologies (InfoTech)*, IEEE, 2022, pp. 1--7.

[41] H. Prabowo, M. R. Shihab i R. F. Aji, „Practical Implementation Of Information Security Management In The Energy Sector Insights From An Oil And Gas Organization In Indonesia," u *2018 International Workshop on Big Data and Information Security (IWBIS)*, IEEE, 2018, pp. 159--163.

[42] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, G. Russell i J. Porcel-Bustamante, „Implementation and evaluation of physical, hybrid, and virtual

testbeds for cybersecurity analysis of industrial control systems," *Symmetry,* t. 13, br. 3, p. 519, 2021.

[43] F. Sicard, E. Hotellier i J. Francq, „An Industrial Control System Physical Testbed for Naval Defense Cybersecurity Research," u *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2022, pp. 413--422.

[44] M. Noorizadeh, M. Shakerpour, N. Meskin, D. Unal i K. Khorasani, „A cyber-security methodology for a cyber-physical industrial control system testbed," *IEEE Access,* t. 9, pp. 16239--16253, 2021.

[45] H. Gao, Y. Peng, Z. Dai, T. Wang, X. Han i H. Li, „An industrial control system testbed based on emulation, physical devices and simulation," u *Critical Infrastructure Protection VIII: 8th IFIP WG 11.10 International Conference, ICCIP 2014, Arlington, VA, USA, March 17-19, 2014, Revised Selected Papers 8*, Springer, 2014, pp. 79--91.

[46] K. Hemsley i R. Fisher, „A history of cyber incidents and threats involving industrial control systems," u *Critical Infrastructure Protection XII: 12th IFIP WG 11.10 International Conference, ICCIP 2018, Arlington, VA, USA, March 12-14, 2018, Revised Selected Papers 12*, Springer, 2018, pp. 215--242.

[47] T. Miller, A. Staves, S. Maesschalck, M. Sturdee i B. Green, „Looking back to look forward: Lessons learnt from cyber-attacks on industrial control systems," *International Journal of Critical Infrastructure Protection,* t. 35, p. 100464, 2021.

[48] R. Masood, Z. Anwar i others, „SWAM: Stuxnet worm analysis in metasploit," u *2011 Frontiers of Information Technology*, IEEE, 2011, pp. 142--147.

[49] M. Baezner i P. Robin, „Stuxnet," ETH Zurich, 2017.

[50] T. M. Chen i S. Abu-Nimeh, „Lessons from stuxnet," *Computer,* t. 44, br. 4, pp. 91--93, 2011.

[51] D. Kushner, „The real story of stuxnet," *ieee Spectrum,* t. 50, br. 3, pp. 48--53, 2013.

[52] J. R. Lindsay, „Stuxnet and the limits of cyber warfare," *Security studies,* t. 22, br. 3, pp. 365--404, 2013.

[53] R. Langner, „Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy,* t. 9, br. 3, pp. 49--51, 2011.

[54] [Na mreži]. Available: https://ics-cert.kaspersky.com/publications/reports/2022/09/08/h1-2022-a-brief-overview-of-the-main-incidents-in-industrial-cybersecurity/. [Poslednji pristup April 2022].

[55] [Na mreži]. Available: https://ics-cert.kaspersky.com/publications/reports/2023/03/15/h2-2022-brief-overview-of-main-incidents-in-industrial-cybersecurity/. [Poslednji pristup April 2022].

[56] D. Kavallieros, G. Germanos i N. Kolokotronis, „Profiles of cyber-attackers and attacks," u *Cyber-Security Threats, Actors, and Dynamic Mitigation*, CRC Press, 2021, pp. 1--26.

[57] I. Agrafiotis, J. R. Nurse, M. Goldsmith, S. Creese i D. Upton, „A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate," *Journal of Cybersecurity,* t. 4, br. 1, 2018.

[58] [Na mreži]. Available: https://dataprot.net/statistics/cyber-security-statistics/. [Poslednji pristup April 2023].

[59] [Na mreži]. Available: https://techjury.net/blog/how-many-cyber-attacks-per-day/. [Poslednji pristup April 2023].

[60] M. I. Tariq i V. Santarcangelo, „Analysis of ISO 27001: 2013 Controls Effectiveness for Cloud Computing," u *ICISSP*, 2016, pp. 201-208.

[61] D. Steuperaert, „COBIT 2019: A significant update," *EDPACS,* t. 59, br. 1, pp. 14-18, 2019.

[62] C. Gikas, „A general comparison of fisma, hipaa, iso 27000 and pci-dss standards," *Information Security Journal: A Global Perspective,* t. 19, br. 3, pp. 132--141, 2010.

[63] [Na mreži]. Available: https://www.pcisecuritystandards.org/standards/. [Poslednji pristup April 2023].

[64] [Na mreži]. Available: https://www.hhs.gov/hipaa/index.html. [Poslednji pristup April 2023].

[65] J.-L. Boulanger, „2 - Requirements Management," u *Certifiable Software Applications 3*, Elsevier, 2018, pp. 7-27.

[66] S. Sankhwar, V. Singh i D. Pandey, „Requirement engineering paradigm," *Global Journal of Multidisciplinary Studies,* t. 3, br. 3, pp. 1--8, 2014.

[67] B. Cheng i J. Atlee, „Research directions in requirements engineering. In 2007 Future of Software Engineering," *IEEE Computer Society,* pp. 285-303, 2007.

[68] M. Hoffmann, N. Kuhn, M. Weber i M. Bittner, „Requirements for requirements management tools," u *Proceedings. 12th IEEE International Requirements Engineering Conference*, 2004.

[69] Z. Jin, „Requirements engineering methodologies," u *Environment Modeling-Based Requirements Engineering for Software Intensive Systems*, Elsevier, 2018, pp. 13-27.

[70] C. C. Agbo, Q. H. Mahmoud i J. M. Eklund, „Blockchain technology in healthcare: a systematic review," u *Healthcare*, MDPI, 2019, p. 56.

[71] P. zalachowski, „Blockchain-based tls notary service," *arXiv preprint arXiv:1804.00875,* 2018.

[72] M. Nofer, P. Gomber, O. Hinz i D. Schiereck, „Blockchain," *Business & Information Systems Engineering,* t. 59, pp. 83--187, 2017.

[73] G. Sladic, B. Milosavljevic, S. Nikolic, D. Sladic i A. Radulovic, „A blockchain solution for securing real property transactions: a case study for Serbia," *ISPRS international journal of geo-information,* t. 10, br. 1, p. 35, 2021.

[74] G. M. Hastig i M. S. Sodhi, „Blockchain for supply chain traceability: Business requirements and critical success factors," *Production and Operations Management,* t. 29, br. 4, pp. 935--954, 2020.

[75] S. Demi, M. Sanchez-Gordon i R. Colomo-Palacios, „What have we learnt from the challenges of (semi-) automated requirements traceability? A discussion on blockchain applicability," *IET Software,* t. 15, br. 6, pp. 391--411, 2021.

[76] S. :. S.-G. M. Demi i M. Kristiansen, „Blockchain for requirements traceability: A qualitative approach," *Journal of Software: Evolution and Process,* 2022.

[77] S. Demi, M. Sanchez-Gordon i R. Colomo-Palacios, „A blockchain-enabled framework for requirements traceability," u *Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI 2021*, Krems, Austria, September 1--3, 2021.

[78] S. Demi, „Blockchain-oriented requirements engineering: A framework," u *2020 IEEE 28th International Requirements Engineering Conference (RE)*, IEEE, 2020, pp. 428--433.

[79] A. Bahga i V. K. Madisetti, „Blockchain platform for industrial internet of things," *Journal of Software Engineering and Applications,* t. 9, br. 10, pp. 533--546, 2016.

[80] R. Schollmeier, „A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," u *Proceedings First International Conference on Peer-to-Peer Computing*, IEEE, 2001, pp. 101--102.

[81] [Na mreži]. Available: https://towardsdatascience.com/mechanisms-securing-blockchain-data-9e762513ae28.

[82] A. a. W. G. Antonopoulos, Mastering Ethereum: implementing digital contracts, O'Reilly Media: Sebastopol, CA, USA, 2018.

[83] A. M. Antonopoulos, Mastering Bitcoin: unlocking digital cryptocurrencies, O'Reilly Media, Inc., 2014.

[84] D. R. Morrison, „PATRICIA—practical algorithm to retrieve information coded in alphanumeric," *Journal of the ACM (JACM),* t. 15, br. 4, pp. 514--534, 1968.

[85] „https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/," [Na mreži]. [Poslednji pristup August 2023].

[86] L. Lamport, S. Robert i P. Marshall, „The Byzantine generals problem," u *Concurrency: the Works of Leslie Lamport*, 2019, pp. 203-226.

[87] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei i C. Qijun, „A review on consensus algorithm of blockchain," u *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, 2017, pp. 2567--2572.

[88] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi i J. Wang, „Untangling blockchain: A data processing view of blockchain systems," *IEEE transactions on knowledge and data engineering,* t. 30, br. 7, pp. 1366--1385, 2018.

[89] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei i C. Qijun, „A review on consensus algorithm of blockchain," u *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, 2017, pp. 2567--2572.

[90] D. K. Tosh, S. Shetty, X. Liang, C. Kamhoua i L. Njilla, „Consensus protocols for blockchain-based data provenance: Challenges and opportunities," u *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 2017, pp. 69--474.

[91] L. S. Sankar, M. Sindhu i M. Sethumadhavan, „Survey of consensus protocols on blockchain applications," u *2017 4th international conference on advanced computing and communication systems (ICACCS)*, 2017, pp. 1-5.

[92] M. K. Shrivas i T. Yeboah, „The disruptive blockchain: types, platforms and applications," *Texila International Journal of Academic Research,* pp. 17--39, 2019.

[93] M. Gupta i others, „A survey of blockchain security issues and challenges," *Int. J. Netw. Secur,* t. 1919, br. 55, pp. 653--659, 2017.

[94] G. Hileman i M. Rauchs, „2017 global blockchain benchmarking study," *Available at SSRN 3040224,* 2017.

[95] D. D. J. Antoni, C. Perez-Sola i J. Herrera-Joancomarti, „The bitcoin P2P network," u *Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18*, 2014, pp. 87--102.

[96] H. Vranken, „Sustainability of bitcoin and blockchains," *Current opinion in environmental sustainability,* t. 28, pp. 1--9, 2017.

[97] [Na mreži]. Available: https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf. [Poslednji pristup April 2023].

[98] D. Vujicic, D. Jagodic i S. Randic, „Blockchain technology, bitcoin, and Ethereum: A brief overview," u *2018 17th International Symposium on INFOTEH-JAHORINA, INFOTEH 2018-Proceedings*, 2018.

[99] [Na mreži]. Available: https://consensys.github.io/smart-contract-best-practices/attacks/. [Poslednji pristup April 2023].

[100] [Na mreži]. Available: https://landscape.hyperledger.org/. [Poslednji pristup April 2023].

[101] [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/glossary.html. [Poslednji pristup April 2023].

[102] [Na mreži]. Available: https://developer.ibm.com/articles/blockchain-basics-hyperledger-fabric. [Poslednji pristup April 2023].

[103] L. Lukic i I. Jovicic, „OSNOVE FUNKCIONISANJA HYPERLEDGER FABRIC BLOCKCHAIN MREŽE," *Info M.,* t. 17, br. 68, pp. 4-10, 2018.

[104] [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/network/network.html#the-sample-network. [Poslednji pristup April 2023].

[105] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich i others, „Hyperledger fabric: a distributed operating system for permissioned blockchains," u *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1-15.

[106] [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/txflow.html. [Poslednji pristup April 2023].

[107] P. Thummavet, „Demystifying Hyperledger Fabric (1/3): Fabric Architecture," [Na mreži]. Available: https://medium.com/coinmonks/demystifying-hyperledger-fabric-1-3-fabric-architecture-a2fdb587f6cb. [Poslednji pristup Jul 2023].

[108] Y. Marcus, E. Heilman i S. Goldberg, „Low-resource eclipse attacks on ethereum's peer-to-peer network," *Cryptology ePrint Archive,* 2018.

[109] [Na mreži]. Available: https://www.utwente.nl/en/ces/sal/exams/digital-exams/Blockchain-and-Distributed-Ledger-Technology-test/6-Consensus/the-dao-the-hack-the-soft-fork-and-the-hard-fork.pdf. [Poslednji pristup April 2023].

[110] M. Conti, E. S. Kumar, C. Lal i S. Ruj, „A survey on security and privacy issues of bitcoin," *IEEE Communications Surveys & Tutorials,* t. 20, br. 4, pp. 3416--3452, 2018.

[111] N. Z. Aitzhan i D. Svetinovic, „Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing,* t. 15, br. 5, pp. 840--852, 2016.

[112] [Na mreži]. Available: https://blog.zeppelin.solutions/on-the-parity-wallet-multisig-hack-405a8c12e8f7. [Poslednji pristup April 2023].

[113] „https://hygger.io/blog/team-roles-in-waterfall-methodology/," [Na mreži]. [Poslednji pristup Avgust 2023].

[114] „https://teamhood.com/agile-resources/agile-team-roles/," [Na mreži]. [Poslednji pristup Avgust 2023].

[115] M. H. U. Sharif i M. A. Mohammed, „A literature review of financial losses statistics for cyber security and future trend," *World Journal of Advanced Research and Reviews,* t. 15, br. 1, pp. 138--156, 2022.

[116] M. Hypponen, If It's Smart, It's Vulnerable, Wiley, 2022.

[117] [Na mreži]. Available: https://www.statista.com/statistics/675399/us-government-spending-cyber-security/]. .

[118] [Na mreži]. Available: https://www.enforcementtracker.com/?insights.

[119] B. De Win, R. Scandariato, K. Buyens, J. Gr{\'e}goire i W. Joosen, „On the secure software development process: CLASP, SDL and Touchpoints compared," *Information and software technology,* t. 51, br. 7, pp. 1152--1171, 2009.

[120] [Na mreži]. Available: https://www.microsoft.com/en-us/securityengineering/sdl/about.

[121] [Na mreži]. Available: https://www.microsoft.com/en-us/securityengineering/sdl/practices.

[122] [Na mreži]. Available: https://cwe.mitre.org/documents/sources/TheCLASPApplicationSecurityProcess.pdf.

[123] [Na mreži]. Available: https://www.iec.ch/who-we-are.

[124] [Na mreži]. Available: https://www.iec.ch/understanding-standards.

[125] [Na mreži]. Available: https://webstore.iec.ch/preview/info_iec62443-4-1%7Bed1.0%7Db.pdf.

[126] [Na mreži]. Available: https://webstore.iec.ch/publication/33615.

[127] M. R. Ayyagari i I. Atoum, „CMMI-DEV Implementation Simplified," *International Journal of Advanced Computer Science and Applications,* t. 10, br. 4, 2019.

[128] [Na mreži]. Available: http://men.fon.bg.ac.rs/wp-content/uploads/2015/10/RACI-matrica.pdf. [Poslednji pristup April 2023].

[129] C. Cabanillas, M. Resinas i A. Ruiz-Cortes, „Automated resource assignment in BPMN models using RACI matrices," u *On the Move to Meaningful Internet Systems: OTM 2012: Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2012, Rome, Italy, September 10-14, 2012. Proceedings, Part I*, Springer, 2012, pp. 56-73.

[130] [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.3/peers/peers.html. [Poslednji pristup April 2023].

[131] M. a. M. A. L. a. V. S. S. Malatji, „Cybersecurity capabilities for critical infrastructure resilience," *Information & Computer Security,* t. 30, br. 2, pp. 255--279, 2022.

[132] M. K. a. H. A. A. a. S. Z. a. I. F. a. I. S. a. R. M. A. Hasan, „Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations," *Journal of Network and Computer Applications,* t. 209, p. 103540, 2023.

[133] „ICS/OT Cybersecurity 2022 TXOne Annual Report Insights," [Na mreži]. Available: https://www.trendmicro.com/en_us/research/23/c/ics-ot-cybersecurity-2022-txone-annual-report-insights.html. [Poslednji pristup July 2023].

[134] „Critical Infrastructure Security and Resilience," [Na mreži]. Available: https://www.cisa.gov/topics/critical-infrastructure-security-and-resilience. [Poslednji pristup July 2023].

[135] S. a. R. K. Ashmore, Introduction to agile methods, Addison-Wesley Professional, 2014.

[136] A. a. L. A. a. R. I. a. H. S. Qumer Gill, „DevOps for information management systems,“ *VINE Journal of Information and Knowledge Management Systems,* t. 48, br. 1, pp. 122--139, 2018.

[137] J. a. W. M. Kaczmarek, „Modern approaches to file system integrity checking,“ u *2008 1st International Conference on Information Technology*, IEEE, 2008, pp. 1-4.

[138] H. a. M. M. a. Y. A. Galal, „Blindfold: Keeping private keys in PKIs and CDNs out of sight,“ *Computers & Security,* t. 118, p. 102731, 2022.

[139] M. a. S. M. Mahalakshmi, „Traditional SDLC vs scrum methodology--a comparative study,“ *International Journal of Emerging Technology and Advanced Engineering,* t. 3, br. 6, pp. 192--196, 2013.

[140] „Common Vulnerability Scoring System SIG,“ [Na mreži]. Available: https://www.first.org/cvss/. [Poslednji pristup Jul 2023].

[141] „Channel Configuration (configtx),“ [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/configtx.html?highlight=consortium. [Poslednji pristup Jul 2023].

[142] J. Benet, „Ipfs-content addressed, versioned, p2p file system,“ *arXiv preprint arXiv:1407.3561,* 2014.

[143] C. Bieri, „An Overview into the InterPlanetary File System (IPFS): Use Cases, Advantages, and Drawbacks,“ *Communication Systems XIV; University of Zurich: Zurich, Switzerland,* p. 78, 2021.

[144] E. Nyaletey, R. M. Parizi, Q. Zhang i K.-K. R. Choo, „BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability,“ *2019 IEEE International Conference on Blockchain (Blockchain),* pp. 18--25, 2019.

[145] H. Huang, J. Lin, B. Zheng, Z. Zheng i J. Bian, „When blockchain meets distributed file systems: An overview, challenges, and open issues,“ *IEEE Access,* t. 8, pp. 50574--50586, 2020.

[146] [Na mreži]. Available: https://share.ipfs.io/#/. [Poslednji pristup April 2023].

[147] P. Kang, W. Yang i J. Zheng, „Blockchain Private File Storage-Sharing Method Based on IPFS,“ *Sensors,* t. 22, br. 14, p. 5100, 2022.

[148] F. H. Halim, N. A. M. Rashid, N. F. M. Johari i M. A. H. A. Rahman, „Decentralized Children's Immunization Record Management System for Private Healthcare in Malaysia Using IPFS and Blockchain,“ *JOIV: International Journal on Informatics Visualization,* t. 6, br. 4, pp. 890--896, 2022.

[149] [Na mreži]. Available: https://medium.com/@s_van_laar/deploy-a-private-ipfs-network-on-ubuntu-in-5-steps-5aad95f7261b. [Poslednji pristup April 2023].

[150] [Na mreži]. Available: https://labs.eleks.com/2019/03/ipfs-network-data-replication.html. [Poslednji pristup April 2023].

[151] J. a. D. N. a. S. G. Marjanovic, „Improving critical infrastructure protection by enhancing software acquisition process through blockchain,“ u *7th Conference on the Engineering of Computer Based Systems*, 2022, pp. 1-7.

[152] J. a. D. N. a. S. G. Marjanovic, „Blockchain-based model for tracking compliance with security requirements," *Computer Science and Information Systems,* t. 20, br. 1, pp. 359--380, 2023.

[153] „Using the Fabric test network," [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/test_network.html. [Poslednji pristup Jul 2023].

[154] [Na mreži]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/prereqs.html. [Poslednji pristup April 2023].

[155] P. Singh i P. Singh, „Exploring WSL2," *Learn Windows Subsystem for Linux: A Practical Guide for Developers and IT Professionals,* pp. 75--98, 2020.

[156] „Hyperledger Fabric samples," [Na mreži]. Available: https://github.com/hyperledger/fabric-samples. [Poslednji pristup Jul 2023].

[157] [Na mreži]. Available: https://www.microsoft.com/en-us/securityengineering/sdl/practices.

[158] D. Boneh i V. Shoup, Applied Cryptography, 2021.

[159] [Na mreži]. Available: https://keccak.team/files/Keccak-reference-3.0.pdf.

[160] [Na mreži]. Available: https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions?pub_id=919061.

[161] R. C. Merkle, „One way hash functions and DES," u *Advances in Cryptology—CRYPTO'89 Proceedings*, 2001, pp. 428--446.

[162] R. MO, „Digitalized signatures," *Foundations of secure computation,* pp. 155--168, 1978.

[163] I. B. Damgard, „Collision free hash functions and public key signature schemes," u *Advances in Cryptology—EUROCRYPT'87: Workshop on the Theory and Application of Cryptographic Techniques Amsterdam, The Netherlands, April 13--15, 1987 Proceeding*, 2000, pp. 203--216.

[164] I. Damgard, „The Application of Claw-Free Functions in Cryptography," *Doctor Thesis, Aarhus University,* 1988.

[165] B. Preneel, „Analysis and design of cryptographic hash functions," Katholieke Universiteit te Leuven Leuven, 1993.

[166] L. M. F. d. :. B. D. F. :. J.-M. R. Moura, „Blockchain and a technological perspective for public administration: a systematic review," *Revista de Administração Contemporânea ,* t. 24, pp. 259--274, 2020.

[167] F. K. a. T. J. Y. Chan, „Acceptance of agile methodologies: A critical review and conceptual framework," *Decision support systems,* t. 46, br. 4, pp. 803--814, 2009.

Biografija

Jelena Marjanović, rođena Stankovski, završila je osnovnu školu Jovan Popović u Novom Sadu, nakon čega upisuje gimnaziju Jovan Jovanović Zmaj u istom gradu. Nakon završene srednje škole 2011. godine, Jelena upisuje osnovne studije na Fakultetu tehničkih nauka, smer Računarstvo i automatika. Diplomu diplomiranog inženjera Elektrotehnike i računarstva stiče 9.7.2015. godine, završišvi osnove akademske studije sa prosekom 9,89. Iste godine upisu je master akademske studije na Fakultetu tehničkih nauka, smer Primenjeno softversko inženjerstvo na departmanu Energetike, elektronika i telekomunikacije. Master akademske studije završava godinu dana kasnije, 6.7.2016. godine, sa prosekom 10. Nakon završenih master akademski studija, odlučila je da nastavi svoje akademsko usavršavanje kroz rad na fakultetu i doktorske studije. Trenutno radi kao asistent na Fakultetu tehničkih nauka, smer Primenjeno softversko inženjerstvo na departmanu Energetika, elektronika i telekomunikacije. Pored rada na fakultetu, stekala je vredno iskustvo i u industriji. Od 2016. godine zaposlena je u kompaniji Schneider Electric (prethodno Schneider Electric DMS), gde je radila na pozicijama junior softver inženjera, medior softver inženjera, specijaliste za bezbednost softvera, a od oktobra 2021. godine zaposlena je na poziciji senior inženjera bezbednosti proizvoda. Ova iskustva su mi omogućila da se upoznam s realnim izazovima industrijskog sektora i proširim svoju perspektivu.

Bibliografija

1. Marjanović, J., Dalčeković, N., & Sladić, G., Blockchain-based model for tracking compliance with security requirements, Computer Science and Information Systems 20 (1), 359-380

2. Marjanović, J., Dalčeković, N., & Sladić, G. (2021, May). Improving critical infrastructure protection by enhancing software acquisition process through blockchain. In 7th Conference on the Engineering of Computer Based Systems (pp. 1-7).

3. Dejanovic, S., Marjanovic, J., Lendak, I., & Erdeljan, A. (2019). Using blockchain to decentralize and protect user privacy in compliance with GDPR.

4. Kovačević, I., Erdeljan, A., Vukmirović, S., Dalčeković, N., & Stankovski, J. (2017, May). Combining real-time processing streams to enable demand response in smart grids. In 2017 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

5. Dalceković, N., Vukmirović, S., Kovacević, I., & Stankovski, J. (2017, May). Providing flexible software as a service for smart grid by relying on big data platforms. In 2017 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

6. Bjeljac, P., Zečević, I., Stankovski, J., Tegeltija, S., & Nićin, M. (2015). Arhitektura povezivanja bankarskih kioska na sistem platnog prometa Republike Srbije. Infoteh Jahorina 2015.

7. Tegeltija, S., Dejanović, S., Feng, H., Stankovski, S., Ostojić, G., Kučević, D., & Marjanović, J. (2022). Blockchain Framework for Certification of Organic Agriculture Production. Sustainability, 14(19), 11823.

8. Dalčeković, I., Erdeljan, A., Dalčeković, N., & Marjanović, J. (2021). Graph Modeling for Efficient Retrieval of Power Network Model Change History. Energies, 14(24), 8351.

9. Marjanović, J., Dejanović, S., Smirnov, V., Miljković, P., Lendak, I. Applying Blockchain Technology to Oil and Gas Industry. In: Zdravković, M., Konjović, Z., Trajanović, M. (Eds.) ICIST 2020 Proceedings, pp.56-59, 2020

10. Stankovski, J., Dejanović, S., Erdeljan, A., Lendak, I. Benefit analysis of blockchain technology on Hyperledger and Ethereum platform. In: Konjović, Z., Zdravković, M., Trajanović, M. (Eds.) ICIST 2018 Proceedings, pp.305-308, 2018

11. Dejanović, S., Stankovski, J., Stanojević, M., Lendak, I. Cost-benefit analysis of migrating the ADMS to the computing cloud. In: Zdravković, M., Konjović, Z., Trajanović, M. (Eds.) ICIST 2017 Proceedings, pp.90-92, 2017

12. Stankovski, J., Erdeljan, A., Kovačević, I., & Dalčeković, N. Computing Data Encrypted by Paillier Encryption Scheme Using Cassandra Database.

Овај Образац чини саставни део докторске дисертације, односно докторског уметничког пројекта који се брани на Универзитету у Новом Саду. Попуњен Образац укоричити иза текста докторске дисертације, односно докторског уметничког пројекта.

План третмана података

Назив пројекта/истраживања
Блокчејн базиран модел за праћење усклађености софтвера у индустријским управљачким системима са захтевима за безбедан развој софтвера
Назив институције/институција у оквиру којих се спроводи истраживање
а) Универзитет у Новом Саду, Факултет техничких наука б) в)
Назив програма у оквиру ког се реализује истраживање
1. Опис података
1.1 Врста студије *Укратко описати тип студије у оквиру које се подаци прикупљају* Докторска дисертација _____ _____ _____ 1.2 Врсте података а) квантитативни **б) квалитативни**

1.3. Начин прикупљања података

а) анкете, упитници, тестови

б) клиничке процене, медицински записи, електронски здравствени записи

в) генотипови: навести врсту _____

г) административни подаци: навести врсту _____

д) узорци ткива: навести врсту_____

ђ) снимци, фотографије: навести врсту_____

е) текст, навести врсту ____ **Актуелна литература у области истраживања** _____

ж) мапа, навести врсту _____

з) остало: описати _____

1.3 Формат података, употребљене скале, количина података

1.3.1 Употребљени софтвер и формат датотеке:

a) Excel фајл, датотека _____

b) SPSS фајл, датотека _____

c) PDF фајл, датотека _____

d) Текст фајл, датотека _____

e) JPG фајл, датотека _____

f) Остало, датотека _____

1.3.2. Број записа (код квантитативних података)

а) број варијабли _____

б) број мерења (испитаника, процена, снимака и сл.) _____

1.3.3. Поновљена мерења

а) да

б) не

Уколико је одговор да, одговорити на следећа питања:

а) временски размак између поновљених мера је _____

б) варијабле које се више пута мере односе се на _____

в) нове верзије фајлова који садрже поновљена мерења су именоване као _____

Напомене: _____

Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?

а) Да

б) Не

Ако је одговор не, образложити _____

_____

2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

а) експеримент, навести тип _____

б) корелационо истраживање, навести тип _____

ц) анализа текста, навести тип _____ **Анализа доступне литературе**

_____

д) остало, навести шта _____

2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).

_____

_____

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

а) Да ли матрица садржи недостајуће податке? Да **Не**

Ако је одговор да, одговорити на следећа питања:

а) Колики је број недостајућих података? _____

б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не

в) Ако је одговор да, навести сугестије за третман замене недостајућих података

_____

2.2.2. На који начин је контролисан квалитет података? Описати

_____

_____

2.2.3. На који начин је извршена контрола уноса података у матрицу?

_____

_____

3. Третман података и пратећа документација

3.1. Третман и чување података

3.1.1. Подаци ће бити депоновани у _____ репозиторијум.

3.1.2. URL адреса _____

3.1.3. DOI _____

3.1.4. Да ли ће подаци бити у отвореном приступу?

а) Да

б) Да, али после ембарга који ће трајати до _____

в) Не

Ако је одговор не, навести разлог _____

3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.

Образложење

_____

_____

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен? _____

3.2.1. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

_____

_____

Ако је потребно, навести методе које се користе за преузимање података, аналитичке и процедуралне информације, њихово кодирање, детаљне описе варијабли, записа итд.

_____

_____

_____

_____

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму? _____

3.3.2. Да ли ће подаци бити депоновани под шифром? Да Не

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да Не

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена?

Да Не

Образложити

_____

_____

4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размотрити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности *(https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html)* и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да **Не**

Ако је одговор Да, навести датум и назив етичке комисије која је одобрила истраживање

_____

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? Да **Не**

Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

а) Подаци нису у отвореном приступу

б) Подаци су анонимизирани

ц) Остало, навести шта

_____

_____

5. Доступност података

5.1. Подаци ће бити

а) јавно доступни

б) доступни само уском кругу истраживача у одређеној научној области

ц) затворени

Ако су подаци доступни само уском кругу истраживача, навести под којим условима могу да их користе:

_____

_____

Ако су подаци доступни само уском кругу истраживача, навести на који начин могу приступити подацима:

_____

_____

5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.

_____

6. Улоге и одговорност

6.1. Навести име и презиме и мејл адресу власника (аутора) података

Јелена Марјановић jelena.stankovski@uns.ac.rs

_____

6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима

_____

6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима

_____