Pametna pogodba z metodo za zagotavljanje ravni storitev z uporabo pametnih prerokov

Sandi Gec1,2, Dejan Lavbič1, Vlado Stankovski2
[1]Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija
[2]Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, Ljubljana, Slovenija
sandi.gec@fri.uni-lj.si, dejan.lavbic@fri.uni-lj.si, vlado.stankovski@fgg.uni-lj.si

Izvleček

Od razvoja tehnologije veriženja blokov ter pristopov, kot so pametne pogodbe, je zanimanje za vključitev tovrstnih pristopov v obstoječe arhitekture računalništva v oblaku v vzponu. Pametne pogodbe so v splošnem porazdeljene aplikacije, ki omogočajo prenos digitalnih sredstev med pogodbenimi strankami po vnaprej dogovorjenih pogojih, so se izkazale kot obetavne zlasti zaradi transparentnega izvajanja med različnimi deležniki. S povečanjem algoritmičnih kompleksnosti funkcionalnosti, kot v našem primeru metoda za izvajanje sporazuma na ravni storitve v pametnih pogodbah, lahko postane vprašljiva smortnost izvedljivosti predvsem s stroškovnega stališča, zato je mogoče uporabiti sodoben pristop komunikacije pametnih pogodb z zunanjimi viri, ki ga imenujemo pametni preroki. Poenostavitev pametnih pogodb je mogoča, če del jedra funkcionalnosti ali algoritma izvajamo zunaj verige. V tem prispevku predlagamo novo metodo za izvajanje sporazuma na ravni storitve na pametni pogodbi s prenosom operacij z verige na namenske storitve zunaj verige z uporabo pametnih prerokov. Predlagana metoda dokazuje, da je mogoče zmanjšati skupne stroške izvajanja pametne pogodbe in hkrati izboljšuje raven distribucije rešitve.

Ključne besede: tehnologija veriženja blokov, pametne pogodba, pametni prerok

Abstract

Since the development of blockchain technology such as Smart Contracts, the attempts for the integration into cloud computing architectures have been on the rise. Smart Contracts are distributed applications that facilitate the transfer of digital assets between parties under the agreed-upon terms and have proven to be promising mechanisms that are able to interact with different stakeholders. However, with the increase of the algorithmic complexity of Smart Contract functionalities, such as the Service Level Agreement, the cost feasibility may become questionable. As a result, we have used the approach known as Smart Oracles, which enables the interaction of Smart Contracts with off-chain data. It is possible to migrate the computational core of the algorithms to off-chain execution. In this paper, we propose a novel method for the Service Level Agreement by migrating on-chain operations to off-chain dedicated services via Smart Oracles. The proposed method shows that it is possible to reduce the overall Smart Contract execution cost and at the same time increase the distribution level of the solution.

Keywords: Blockchain, Smart Contract, Smart Oracle

1 UVOD

S pojavom tehnologije veriženja blokov ter omrežij, kot je Ethereum [Buterin, 2015] zagnan leta 2015, so se odprle nove možnosti za izvajanje operacij zaupanja. Te vključujejo uporabo pametnih Pogodb (PP), ki poleg prvotne interakcije s podatki v verigi omogočajo neposredno interakcijo s podatki izven verige s pomočjo pristopa pametnih prerokov (PPR) ali tradicionalno, vendar neučinkovito z uporabo načrtovalskega vzorca preroka, predlaganega od [Wöhrer in Zdun, 2018]. PPR lahko v osnovi opišemo kot decentralizirano storitev, ki omogoča interakcijo PP izven verige.

V primerih, kjer so scenariji dinamični in posledično bolj zapleteni, se ta kompleksnost odraža v funkcijah PP, ki so stroškovno dražje in obenem so njihovi klici časovno manj učinkoviti (npr. se izvedejo v 2 ali več blokovnih ciklih). Takšne funkcije PP

zato nimajo realnih temeljev uporabe v produkcijskih okoljih oblakovnega računalništva, v našem primeru pri izvajanju sporazuma o ravni storitev (ang. service-level agreement - SLA).

Cilj tega dela je predlagati novo metodo v PP za SLA, ki temelji na uporabi pametnih prerokov ter jo primerjati s konvencionalno metodo SLA, podprto v celoti v PP. Smotrnost nove SLA metode je podkrepljena z empirično analizo, kjer dokažemo, da se v primerjavi z obstoječo metodo zmanjša skupni strošek izvajanja PP. Obenem se poveča stopnja distribucije sistema ter posledično tudi razpoložljivosti na več ravneh, kot so PPR vozlišča. Nova PPR vozlišča lahko po potrebi namestimo v efektivnem času na poljubnih geolokacijah. Torej, našo rešitev lahko predstavimo z analogijo tri-nivojskih sistemov, kjer želimo doseči tankega oz. lahkega odjemalca v PP, ki je sestavljen iz manj kompleksnih in posledično cenejših funkcij.

2 SORODNO DELO

V tem poglavju so predstavljena sorodna raziskovalna dela, ki temeljijo na tehnologiji veriženja blokov. Pri tem se osredotočamo na dela PP in PPR, kot temeljna pristopa za doseganje naše raziskave.

Za pripravo robustnih PP smo uporabili orodja GasReducer avtorjev [Chen idr., 2018] ter Oyente avtorjev [Luu idr., 2016], ki nam nudijo za identifikacijo najdražjih funkcij v PP. Po identifikaciji funkcij smo pripravili prototipno rešitev brez uporabe pametnih prerokov na podlagi oblikovalskega vzorca PPR, ki sta ga predlagala [Wöhrer in Zdun, 2018]. Ker predlagan oblikovalski vzorec vsebuje preveč varnostnih pomanjkljivosti (npr. kako zagotoviti varen pretok podatkov, integriteta zunanje storitve in drugi), smo uporabili zaupanja vreden pristop z uporabo Chain-Link1 PPR. Sorodno našemu predhodnemu delu, kjer je bila osredotoče- nost na zaupanju PPR z uporabo ogrodja Oraclize2 in našega pristopa [Kochovski idr., 2019], kjer v tem delu predstavimo smotrnost migracije segmentov najdražjih funkcij izven verige. Avtorji [Zhou idr., 2019] predlagajo SLA metodo v celoti na PP brez uporabe PPR, vendar je stroškovni vidik pri takšnem pristopu vprašljiv za produkcijske namene.

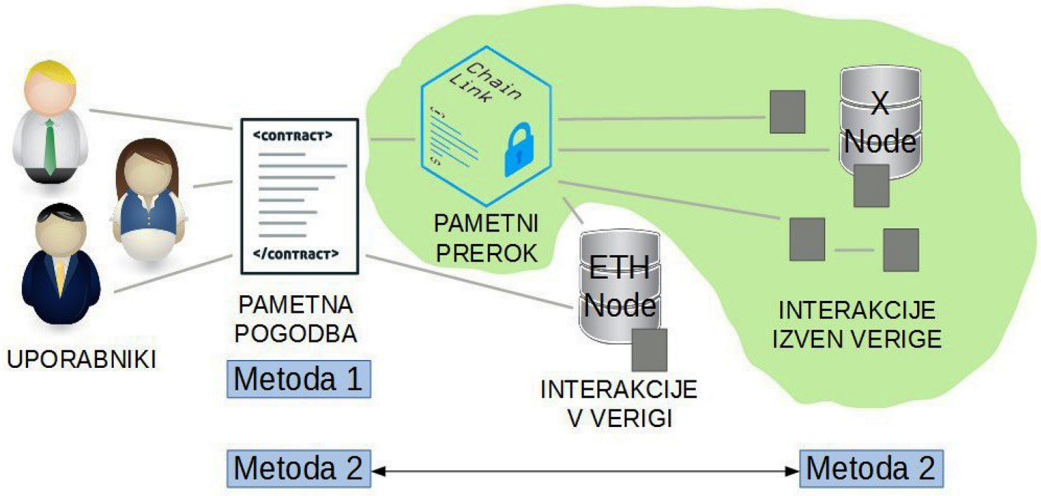
V našem delu predlagamo novo SLA metodo, ki je definirana na PP ter zunanji storitvi dostopni s pomočjo PPR. Poleg visoke stopnje distribucije takšnega pristopa, metoda omogoča večjo skalabilnost, saj je metodo mogoče nadgraditi v realnem času obenem pa hraniti obstoječe rezultate v sorodni verigi in s tem ohraniti integriteto delovanja.

3 INTERAKCIJA PAMETNIH POGODB NA VERIGI Z ZUNANJIMI SISTEMI

PP v osnovi omogočajo posredno interakcijo s podatki izven verige. Na primer, uporabnik lahko podatke izven verige posreduje preko vhodnih atributov funkcije PP. Raziskovalno so se [Carminati idr., 2018] ukvarjali z naslovitvijo konceptov varnega pretoka podatkov v poslovnih procesih z uporabo tehnologije veriženja blokov. Od pojavitve javnih PPR rešitev so PP sposobne komunicirati s storitvami tretjih oseb (npr. podatki, ki so del drugih sistemov na tehnologiji veriženja blokov) preko (de)centraliziranih storitev PPR, ki omogočajo sprožitev zunanjih storitev in neposredno pridobivanje rezultatov PP v jedru same funkcije. Interakcija s podatki zunaj verige je omogočena s sledečimi entitetami: vozlišči tehnologij veriženja blokov vključno z drugimi sorodnimi sistemi, javnimi storitvami ali kombinacijo sistemov ter storitev. Proces, ki opisuje predstavljeno interakcijo med PP in PPR, je prikazan na sliki 1, kjer se interakcija poleg omrežja Ethereum (predstavljena z ETH Node) razširja tudi na druge blokovne rešitve (predstavljena kot X Node). Naša ideja je, da izkoristimo prednosti uporabe PPR z redefinicijo obstoječih funkcij PP, ki niso primerne za produkcijsko rabo – pomanjkanje preverljivosti, razširljivosti in predvsem stroškovne smotrnosti.

V našem delu je primer uporabe obravnavan na PP, ki ga lahko v grobem povzamemo kot SLA, kjer je namen PP digitalno preveriti pogoje uporabe specifične oblakovne storitve in posledično olajšati konsenz med deležniki, uporabnikom ter sistemom, ki ponuja oblakovno storitev. Na primer, končni uporabnik želi uporabljati storitve v oblaku pod določenimi SLA pogoji (npr. dostopnost storitve 99.9 %, latenca, manjša od specificirane vrednosti itd.). Najprej se izvede dogovor o ceni in SLA pogojih – končni uporabnik plača določeno ceno, ki je zaklenjena v PP in začne uporabljati storitev v oblaku. Ko končni uporabnik preneha uporabljati to storitev, se izvede funkcija preverjanja SLA. Potek funkcije je opisan v algoritmu 1. V takšni obliki je funkcija PP za preverjanje SLA draga zaradi zahtevnejših vhodnih podatkov (npr. tabele) in jedra funkcije z vrsto zank, ki so potrebne za samo izvedbo funkcije.

[1]https://chain.link/



Slika 1: **Prikaz procesa interakcije PP med uporabniki, instanco PP in vozliščem Ethereum. Pri pristopu uporabe PPR, ki je prikazano z zelenim ozadjem, je razvidna povezljivost z zunanjimi sistemi na tehnologiji veriženja blokov in storitvami tretjih oseb. Osnovni algoritem (Metoda 1) na PP in novi algoritem (Metoda 2), kjer del algoritma predstavlja zunanja storitev.**

Vhodni podatki: tabela SLAOsnovneMeritve, tabela ločila, decimalnoŠtevilo CiljSLA
Izhodni podatki: operacijaFormalneLogike jeDosežen tabela (razdeli SLAOsnovneMeritve po ločniku);
decimalnoŠtevilo RezultatSLA = 0; tabela MeritveSLA = razdeli(SLAOsnovneMeritve, ločila);
dokler *MeritveSLA obstajajo* **naredi**
decimalnoŠtevilo vsotaMeritev = 0; število števec = 0;
// za vsak seznam SLA meritev;
ZA število i = 0, *i*++, dokler *i* < *MeritveSLA.dolina*; vsotaMeritev += meritev; števec++;

RezultatSLA += *pragPomembnosti* (*vsotaMeritev*/*števec*);
konec
če *RezultatSLA CiljSLA* **potem**
// SLA je bil dosežen; vrni pravilen;
drugo
// SLA ni bil dosežen; vrni napačen;
konec

Algoritem 1: **Primer funkcije za izračun SLA, ki temelji na interakciji s podatki na Ethereum verigi brez uporabe PPR.**

V SLA primeru uporabe nadgrajenim s PPR interakcijo, je algoritem preoblikovan na takšen način, da je računanje oz. jedro funkcije v celoti preneseno v zunanjo storitev. Pri tem je potrebno pametno pogodbo razširiti s *ChainlinkClient* pametno pogodbo, ki definira ogrodje za podporo ChainLink PPR. Ta neposredno komunicira s SLA podatki, shranjenimi v namenski sorodni verigi (npr. IOTA[3]) pri čemer so takšni podatki obenem tudi vedno preverljivi. Obenem se vhodni podatki poenostavijo iz tabel v nize ali so dodatno nizi ločeni z ločili kot reference na SLA podatke ter s časovnimi okvirji relevantnimi za izračun SLA. Povzetek PP funkcije podprte s PPR predstavlja algoritem 2.

4 EKSPERIMENTALNA ŠTUDIJA

V tem poglavju je predstavljena eksperimentalna metodologija in prvotno doseženimi empiričnimi rezultati

[3]https://www.iota.org/

Vhodni podatki: tabela SLAOsnovneMeritve, tabela ločila, decimalnoŠtevilo CiljSLA
Izhodni podatki: operacijaFormalneLogike jeDosežen podatki = (SLAOsnovneMeritve, ločila, CiljSLA);
// gradnja nove poizvedbe;
Chainlink.Request memory poizvedba = buildChainlinkRequest(jobId, this, this.izpolnjevalnaFunkcija);
// dodamo parametre poizvedbe in sicer URL ter tip REST klica; poizvedba.add(„get“, „API-URL“);
// dodamo pot (ang. path) z vsemi vhodnimi podatki v JSON obliki; poizvedba.add(„path“, podatki);
// pošiljanje poizvedbe z 1 LINK žetonom v PPR; RezultatSLA = sendChainlinkRequest(poizvedba, 1 * LINK); **če** $RezultatSLA \geq CiljSLA$ **potem**
// SLA je bil dosežen; vrni pravilen;
drugo
// SLA ni bil dosežen; vrni napačen;
konec

Algoritem 2: **Primer funkcije za izračun SLA, ki temelji na interakciji s podatki na Ethereum verigi nadgrajen z uporabo PPR.**

4.1 Eksperimentalna metodologija

Eksperimentalno okolje temelji na testnem javnem Ethereum vozlišču Rinkeby, ki omogoča največjo stopnjo funkcionalnosti (npr. naprednejše poslušalce vozlišča). Druga možnost, kjer se izognemo vzpostavitvi lastnih vozlišč, je spletno orodje Infura4, ki povezljivost vozlišč zagotavlja preko storitev in s tem omogoča interakcijo z javnimi Ethereum testnimi Rinkeby omrežji. V tem primeru smo nekoliko funkcionalno omejeni, saj ne moremo zagnati poslušalcev na nivoju bloka ali transakcijskih funkcij za bloke, imenovanih opazovalci (ang. observators). V našem primeru poganjamo svoja testna vozlišča in tako imamo možnost zagnati poslušalca na nivoju bloka, kar pomeni da smo o potrjeni transakciji (izvedeni funkciji PP) nemudoma obveščeni, ko je blok na novo ustvarjen. Skozi empirično analizo želimo odgovoriti na ključni vprašanji:

- Ali je v našem primeru bolj primeren pristop z uporabo zgolj PP ter s tem interakcijo podatkov na verigi ali je bolj smotrno del podatkov obravnavati zunaj verige z uporabo PPR?
- V primeru pomanjkanja PPR vozlišč ali jih je mogoče namestiti v obstoječem ciklu (trajanje cikla cca 15 sekund) na poljubno geolokacijo?

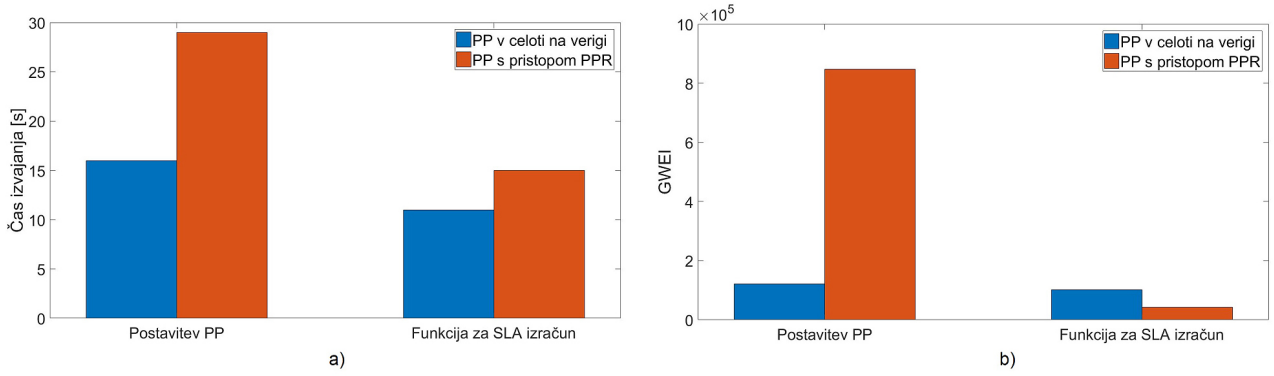
4.2 Rezultati in razprava

V okviru eksperimentalne metodologije in implementacije našega primera uporabe smo izvedli evalvacijo, kjer smo naš primer uporabe izvedli 10-krat na testnem Ethereum omrežju Rinkeby. Rezultati so prikazani na sliki 2, kjer se v primeru metrike za ceno uporablja denomenacijo GWEI5.

Rezultati delovanja kažejo, da se postavitev PP v primeru klasičnega pristopa izvede v enem bloku, medtem ko v primeru pristopa z uporabo PPR le ta potrebuje čas dveh blokov. Naš pristop torej ni primeren takrat, ko so performančne zahteve ključnega pomena.

V fazi integracije ChainLink PPR vozlišč, ki služijo kot osnovne komponente za interakcijo s PPR, smo le te uporabljali v obliki kontejnerskih slik (ang. Container images). Pri tem smo instance kontejnerskih slik namestili s storitvijo Kubernetes v Sloveniji na različne geografske lokacije. Storitev Kubernetes je nameščena na strojni opremi s 4 virtualnimi procesorskimi jedri s strojnim pospeševanjem procesorja (ang. CPU) Intel Xeon E5649 jedri s posamičnimi frekvencami 2.53GHz, 2GB delovnega pomnilnika in trdega diska kapacitete 20GB. V poskusu namestitve ChainLink vozlišč smo namestitev kontejnerskih slik izvedli 10-krat na različnih geolokacijah kot prikazuje tabela 1. Visoka stopnja distribucije nudi hitrejšo interakcijo PP s PPR in je ključna v primerih, ko se interakcija PPR izvaja na globalnem nivoju.

[4]https://infura.io
[5]1 Ethereum = 1, 000, 000, 000 GWEI (10^9)



Slika 2: **Performančni časi (a) in rezultati stroškov z uporabo denominacije GWEI (b) na testnem Ethereum omrežju Rinkeby.**

Tabela 1: **Časi namestitve kontejnerskih slik ChainLink vozliščna različnih geolokacijah.**

Ime oblaka	Kontinent	Časi namestitve [ms]
Arnes	Evropa	3133
flexiOps	Združeno kraljestvo	80332
GKE-EU-WEST	Evropa	1866
GKE-ASIA-EAST	Azija	5133
GKE-US-WEST	Združene države Amerike	2266

Z vidika ocene stroškov je pri uporabi našega pristopa s PPR (podatki PP deloma izven verige), kjer so v PP globalne spremenljivke bistveno kompleksnejše in sicer večinoma tipa *uint256* in *bajtov*. To kot posledica terja večje stroške postavitve PP. Poleg tega je s pristopom PPR obvezna uporaba dodatnih knjižnic (npr. matematične, vmesniki, odločevalske itd.), ki jih je treba vključiti v PP, kar tudi terja svoj strošek pri postavitvi PP. Po drugi strani pa s prenosom logike na storitve, ki niso povezane z verigo, postane naša funkcija SLA bistveno cenejša v primerjavi z običajnim pristopom. Obenem je za nadaljnjo analizo potrebno podrobneje definirati zahteve uporabe in arhitekturo v oblaku, opraviti pregled smiselnih funkcij za prenos ter izvedbo v PP takšnih funkcij, ki se izvajajo večkrat, da bi zmanjšali skupne stroške PP, ki se običajno ocenjujejo, če se ujema s statističnim pristopom zlatega križa (ang. golden cross pattern).

5 ZAKLJUČEK

V okviru tega dela smo ugotovili, da je nova predlagana metoda SLA v PP z interakcijo PPR smotrna v tistih primerih uporabe, kjer je uporaba funkcij pogosta in posledično na dolgi rok privede do nižjih celovitih stroškov izvedbe funkcije oz. transakcij na verigi. Učinkovitost sistema pri uporabi nove metode na sam sistem ne vpliva negativno. Kljub temu študija ni obravnavala podrobnejših statističnih analiz, da bi našla ustrezne mejne vrednosti za ocenitev smotrnosti izbranega primera uporabe. Poleg tega lahko v času trajanja generiranja bloka namestimo poljubno novo vozlišče ChainLink PPR.

To delo temelji na prvotnih prizadevanjih, ki so namenjena boljšemu razumevanju PPR, novega pristopa PP v interakciji s podatki izven verige. Predstavljena metoda predstavlja segment primera uporabe v projektu ABC. Predstavljeni primer uporabe prikazuje potencialnost PPR pristopa za uporabo v drugih scenarijih računalništva v oblaku.

ZAHVALA

Raziskava je bila finančno podprta s sredstvi projekta Evropske unije Horizon 2020 program za raziskave in inovacije na podlagi sporazuma št. 815141 (DECENTER projekt: Decentralised technologies for orchestrated cloud-to-edge intelligence).

LITERATURA

[1] [Buterin, 2015] Buterin, V. (2015). Ethereum white paper, posodobljeno september 30, 2015. https://github.com/ethereum/wiki/wiki/White-Paper.

[2] [Carminati idr., 2018] Carminati, B., Ferrari, E., in Rondanini, C. (2018). Blockchain as a platform for secure inter-organizational business processes. *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 122–129.

[3] [Chen idr., 2018] Chen, T., Li, Z., Zhou, H., Chen, J., Luo, X., Li, X., in Zhang, X. (2018). Towards saving money in using smart contracts. In *Proceedings of the 40th International*

Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER '18, pages 81–84, New York, NY, USA. ACM.

[4] [Kochovski idr., 2019] Kochovski, P., Gec, S., Stankovski, V., Bajec, M., in Drobintsev, P. D. (2019). Trust management in a blockchain based fog computing platform with trustless smart oracles. *Future Generation Computer Systems*, 101:747 – 759.

[5] [Luu idr., 2016] Luu, L., Chu, D.-H., Olickel, H., Saxena, P., in Hobor, A. (2016). Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 254–269, New York, NY, USA. ACM.

[6] [Wöhrer in Zdun, 2018] Wöhrer, M. in Zdun, U. (2018). Design patterns for smart contracts in the ethereum ecosystem.

[7] [Zhou idr., 2019] Zhou, H., Ouyang, X., Ren, Z., Su, J., de Laat, C., in Zhao, Z. (2019). A blockchain based witness model for trustworthy cloud service level agreement enforcement. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 1567–1575.

▪

Sandi Gec je zaposlen kot asistent na Fakutelti za računalništvo in informatiko, Univerze v Ljubljani. V raziskovalno-razvojnih projektih se je ukvarjal z uporabo semantičnih tehnologij pri razovoju oblačnih sistemov, bazami znanja ter integracijo podpornih rešitev v oblaku. Svoje znanje je apliciral na Horizon 2020 projektih SWITCH in ENTICE. Trenutno se v okviru Horizon 2020 projekta DECENTER ukvarja z novimi pristopi tehnologije veriženja blokov, predvsem s pametnimi pogodbami ter komunikacijo med verigami in zunaj verige.

▪

Vlado Stankovski je izredni profesor računalništva na Univerzi v Ljubljani. Ima več kot 15 let izkušenj s področja oblakovnega računalništva, porazdeljenih sistemih, semantike, programskega inženirstva, strojnega učenja in podatkovnega rudarjenja. Dr. Stankovski ima izkušnje s področja integracije programske opreme, kjer je svoje sodobne smernice področja gradil na več projektih EU, vključno z DataMiningGrid (2004–2006), InteliGrid (2004–2007), mOSAIC (2011–2013), ENTICE (2015–2018), SWITCH (2015–2018) in trenutno DECENTER (2018–2021) projektu. Sodeluje v nedavno oblikovanem konzorcijskem superračunalniškem centru Slovenije in pri slovenskih projektih pametne specializacije, kot je IQ DOM. Vlado Stankovski trenutno aktivno sodeluje v gruči projektov programskega inženiringa Horizon 2020, kot predstavnik projektov ENTICE, SWITCH in DECENTER.

▪

Dejan Lavbič je leta 2010 doktoriral na področju računalništva in informatike na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je zaposlen kot docent. Na raziskovalnem področju se ukvarja z inteligentnimi agenti, večagentnimi sistemi, ontologijami, poslovnimi pravili, semantičnim spletom, odkrivanjem plagiatorstva s pomočjo socialnih omrežij, kakovostjo informacij in naprednimi pristopi za porazdelitev podatkov. Sodeloval je pri številnih gospodarskih in raziskovalnih projektih s področja strateškega planiranja, metodologij razvoja informacijskih sistemov, uporabe inteligentnih sistemov, avtomatizacije poslovnih procesov in obvladovanja ter porazdelitve velike količine podatkov.