

## PREDLOG ARHITEKTURE SISTEMA ZA PODRŠKU BEŽIČNIM SENZORSKIM MREŽAMA ZASNOVANOG NA MIKROSERVISIMA

Milan Malić<sup>1</sup>, Dalibor Dobrilović<sup>2</sup>, Dušan Malić<sup>3</sup>

<sup>1</sup>Panonit, Novi Sad, milanmalic@outlook.com

<sup>2</sup>Univerzitet u Novom Sadu – Tehnički Fakultet “Mihajlo Pupin”, Zrenjanin,  
dalibor.dobrilovic@uns.ac.rs

<sup>3</sup>Visoka tehnička škola strukovnih studija u Zrenjaninu, Zrenjanin, dmalic@sbb.rs

**Sadržaj:** *Veoma brz razvoj novih tehnologija koje su doprinele sve većoj primeni bežičnih komunikacionih tehnologija u senzorskim mrežama, kao i pad cena hardvera potrebnog za njihovu realizaciju, primorale su softverske inženjere da razmatraju i razvijaju nova rešenja za arhitekture tih sistema. Dosadašnji pristup u razvoju aplikacija pokazao se nedovoljno dobrim u ispunjavanju novih zahteva koje takve aplikacije treba da ispune. Upravo zbog navedenog, model arhitekture sistema baziranog na mikroservisima ukazao se kao potencijalno rešenje koje bi moglo da prevaziđe nametnute probleme. U ovom radu je prikazan predlog rešenja arhitekture takvog sistema, kao laganog (eng. lightweight) modela mikroservisa koji se može koristiti kao osnova u razvoju kompleksnih sistema, npr. za kućnu automatizaciju, monitoring u pametnim gradovima i sl. Zbog kompleksnosti u radu su predstavljeni samo najznačajniji segmenti arhitekture kao i potencijalni problemi i prednosti koje ista nosi sa sobom.*

**Ključne reči:** *bežične senzorske mreže, softverska aritektura, mikroservisi, sistemi za monitoring*

### 1. Uvod

Prema istraživanju *International Data Corporation (IDC)* [1] u 2018. godini obrt kapitala, kada je reč o Internetu stvari ili IoT (eng. *Internet of Things*) uređajima i sistemima premašiće 772 milijardi dolara što predstavlja povećanje od približno 15% u odnosu na prethodnu godinu. Ova prognoza rasta očekuje se do 2021. godine kada se očekuje da obrt kapitala pređe 1.1 trilion dolara. Porast primene IoT sistema i sistema baziranim na senzorskim mrežama otvorilo je i novo poglavlje u razvoju aplikacija. Prikupljanje velike količine informacija u realnom vremenu, u senzorskim mrežama uz pomoć bežičnih komunikacionih tehnologija, postavilo je niz problema sa kojima se suočavaju softverski inženjeri pri razvoju aplikacija. Klasične monolitne softverske arhitekture, pa čak i moderne kao što je servisno orijentisana arhitektura ili SOA (*Service Oriented Architecture*), nisu u mogućnosti da ispune stroge zahteve nametnute od strane velikog broja sistema koji su zasnovani na bežičnim senzorskim mrežama. Heterogenost

hardvera, mreža, operativnih sistema i velikog broja proizvođača postavile su traženu kompleksnost ovakvih sistema na veoma visok nivo [2].

Kao jedan od odgovora na navedene izazove u ovom radu je prikazan predlog laganog (eng. *lightweight*) modela mikroservisa koji se koristi u sistemu za monitoring *data* centra, a koji može poslužiti kao osnova u razvoju kompleksnijih sistema kao što su monitoring industrijskog centra ili, pak, sistema pametnih gradova, kućne automatizacije i dr. Postoji veliki broj definicija, od strane velikog broja autora, koje opisuju kako aplikacija zasnovana na ovoj arhitekturi treba da funkcioniše. Daya [3] definiše mikroservise kao arhitekturni stil, u kome su veliki kompleksni softveri sačinjeni od jednog ili više servisa (mikroservisa), pri čemu svaki mikroservis treba da se fokusira na izvršavanje jednog zadatka, pri čemu taj zadatak treba da obavi efikasno.

U ovom radu izvršiće se najpre poređenje između klasičnih softverskih arhitektura i mikroservisa, nakon čega su prezentovana dosadašnja istraživanja u svetu koji koriste ovaj pristup (sekcija 2). U nastavku (sekcija 3) opisan je sistem za monitoring *data* centra za koji se razvija aplikacija, a detaljniji model arhitekture sistema dat je u sekciji 4. Na kraju rada su *data* završna razmatranja.

## 2. Dosadašnja istraživanja

Razvoj novijih komunikacionih tehnologija doprineo je ogromnom porastu količine prikupljenih podataka, što je uticalo na pristup razvoju aplikacija. Tako se u okruženjima koja su zasnovana na bežičnim senzorskim mrežama (WSN) sve više razmatra upotreba *big data* tehnika [4]. U takvim sistemima moguća je upotreba raznovrsnih *big data* tehnika kao što su fajl sistemi za *big data* (npr. *Hadoop distributed file system* - HDFS), noSQL baze (npr. HBase), modeli obrada podataka (npr. MapReduce), *streaming* tehnike (npr. Storm), *query engines* (npr. Impala), *big data* arhitekture (npr. *lambda architecture*) i sl. U istom izvoru navode se i dva primera monitoring sistema. Jedan je sistem za bezbednost od požara, a drugi je sistem za monitoring parametara okruženja, tj. kvaliteta vazduha. U drugom primeru, sistem baziran na bežičnim senzorskim mrežama nadzire nivo štetnih gasova kao što su: SO<sub>2</sub>, CO, NO<sub>2</sub>, i dr. Sistem se sastoji iz tri dela, a jedan njegov deo *Message Oriented Middleware* (MOM) je zadužen za asinhronu komunikaciju između druga dva modula: *Data Acquisition Module* (DAM) i *Data Processing Module* (DPM). Kao predlog *message queue* sistema se predlaže upotreba RabbitMQ ili Kafka tehnologija [4].

Trenutno u svetu postoji veliki broj istraživanja o primeni arhitekture bazirane na mikroservisima u sistemima koji su generalno bazirani na senzorskim mrežama. Ti sistemi imaju primenu u različitim okruženjima, od sistema pametnih gradova, preko industrije, do kućne automatizacije. U [5] razmatra se pristup za upotrebu postojećih standarda *publish/subscribe* servisa kao što su DDS, XMPP, MQTT [6], CoAP [7] i AMQP za komunikaciju između senzora u IoT sistemima. Primena arhitekture zasnovane na mikroservisima u razvoju *Smart City* IoT platforme za raznovrsne aplikacije sa ciljem povećanja energetske efikasnosti grada prikazan je sa opisom prednosti i izazova arhitekture bazirane na mikroservisima u poređenju sa tradicionalnim SOA pristupom u razvoju servisne platforme za međudomske aplikacije [8].

Sistem za smanjenje rizika kod transporta hazardnih materija razvijen je na arhitekturi baziranoj na mikroservisima u *cloud* okruženju. Takav tip arhitekture se

zasniva na skupu slabo povezanih nezavisno implementiranih servisa za skalabilne aplikacije. S obzirom na distribuiranu prirodu mikroservisa sistem je razvijen kao skup mikroservisa koji su povezani sa procesom upravljanja rizikom [9]. Sistem je zasnovan na MQTT (*Message Queue Telemetry Transport*) protokolu. MQTT protokol prenosi podatke sa senzora i uređaja na vozilima preko GPRS mreže. On spada u *publish/subscribe messaging transport* protokole i podesan je za ovakvu primenu. U [10] je prikazana platforma InterSCity. Radi se o platformi koja koristi arhitekturu baziranu na miroservisima koja ima za cilj da podrži kolaborativni rad na istraživanjima, razvoju i implementaciji inicijativa za pametne gradove. U radu je prikazano kako pristup zasnovan na mikroservisima omogućava fleksibilno, proširivo i slabo povezano okruženje.

Još jedan pristup u upotrebi arhitekture bazirane na mikroservisima dat je na primeru platforme *Lightweight Edge Gateway for the Internet of Things* (LEGIoT). Platforma se oslanja na modularne karakteristike mikroservisa i lagane tehnologije za virtualizaciju sa ciljem garantovanja proširivih i fleksibilnih rešenja [11]. U radu je predstavljena arhitektura zasnovana na mikroservisima za platformu (LEGIoT) koja se oslanja na slojevitu arhitekturu i svestranost koje pružaju virtualizacione tehnologije u razvoju. Drugačiji pristup prikazan je na primeru radnog okvira (eng. *framework*) okruženja *IoT as a Service* (iTaaS) koje transformiše korisnički uređaj (npr. pametni telefon) u IoT *gateway* koji omogućava brz i efikasan prenos podataka u *cloud* okruženje. Radi se o dvoslojnom rešenju baziranom na mikroservisima za IoT okruženje kao i *back-end* servisima na *cloud* strani [12]. Arhitektura bazirana na mikroservisima koja je razvijena i implementirana u *cloud* okruženju, a koja omogućava fleksibilno prikupljanje podataka data je u [13]. Arhitektura obuhvata *mashup* aplikacije koje su dostupne preko različitih platformi (eng. *cross-device*) i imaju podršku za različite forme interakcije u *cloud* okruženju. Uopšteno rečeno, *mashup* aplikacije upravljaju ogromnim količinama heterogenih podataka koji se dobijaju iz različitih izvora i opslužuju različite tipove korisnika.

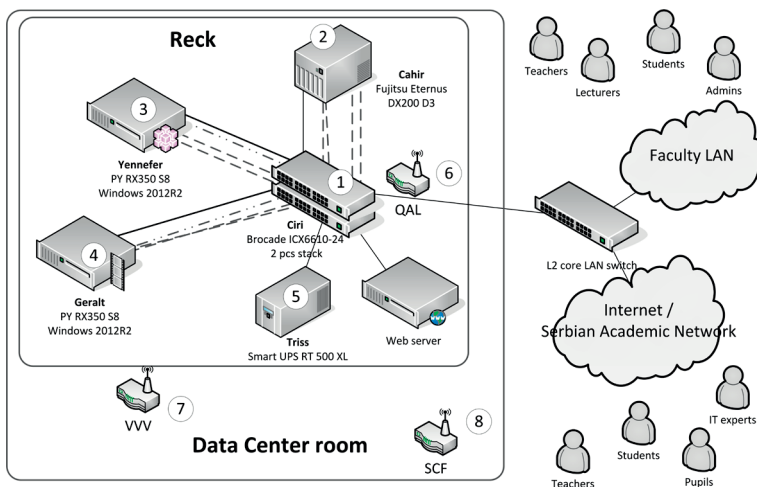
Primer kako najnoviji trendovi u softverskim tehnologijama, kao što su mikroservisi, mogu doprineti upotrebi manjih softverskih funkcija je prikazan u [14]. Te novine se mogu primeniti i u okruženjima kao što su *Cloud-based serverless computing*, poznati i kao *Function-as-a-Service* (FaaS). U takvom sistemu je izvršen pokušaj da se umanjí kompleksnost pomoću deljenja servisa na manje individualne funkcije koje se mogu implementirati i izvršavati nezavisno [14]. FaaS omogućava detaljnu i veoma dinamičnu konfiguraciju i deli mikroservise na male softverske delove koji se mogu izvršavati veoma brzo, pri tome treba imati na umu da cene hardvera imaju veliki uticaj na potrebu za optimizacijom izvršavanja funkcija.

Sličan primer mikroservis zasnovane arhitekture dat je na sistemu za praćenje saobraćaja koji je implementiran u gradu Mohamedija (*Mohammedia*) u Maroku. Ta arhitektura obuhvata tri sloja: IoV (*Internet of Vehicles*), *Fog Computing* i *Cloud Computing Layer*. Fokus je usmeren na *Fog Computing Layer* na kojem je razvijen radni okvir (*framework*) za prikupljanje i obradu događaja u realnom vremenu. Razvijeni radni okvir je zasnovan na Kafka, Spark, MongoDB tehnologijama i mikroservisima, a mikroservisi su razvijeni sa *Spring Boot* i implementirani pomoću *Docker container-a* [15].

### 3. Sistem za monitoring data centra

Prikaz sistema za koji se razvija aplikacija za monitoring *data* centra, a koji se zasniva na mikroservisima, dat je na slici 3. Razvoj aplikacije i nadogradnja sistema su u toku (u početnoj fazi) i oslanjaju se na dosadašnja istraživanja istraživačkog tima u okviru bežičnih senzorskih mreža u zatvorenim i otvorenim okruženjima [16, 17, 18, 19]. *Data* centar je implementiran u okviru projekta MIS ETC 1379 “*Cross-border access infrastructure to high-level education through web-casts (EduWebCast)*” koji je finansiran sredstvima iz programa prekogranične saradnje Rumunija – Srbija koji je finansiran od strane Evropske unije u okviru instrumenta za predpristupnu pomoć (IPA) i sufinansiran od strane država učesnica programa. Funkcija i detaljan opis sistema dati su u [20, 21]. Centralni deo sistema predstavljaju dva sviča Brocade ICX 6610 su prikazani sa na slici 3. kao (1). Ova dva uređaja daju sledeća očitavanja senzora temperature i to za: ventilator, MAC, CPU, senzor A, senzor B, senzor C, senzor D i *stack* karticu. Temperatura MAC se koristi kao referentna temperatura uređaja, jer se ona prikazuje i na veb stranici za administraciju istog uređaja kao glavna temperatura uređaja. Važno je reći da su opisana dva sviča postavljena u ormanu jedan ispod drugog i da tu zauzimaju centralnu poziciju. Donji *Brocade* svič ima višu izmerenu temperaturu, što je i očekivano jer je donji svič bliži centru ormara i okružen je drugim uređajima. Iz tog razloga se temperatura ovog sviča uzima kao referentna temperatura celog ormara i kao jedna od najvažnijih za monitoring. Uzimajući u obzir da postoje dva *Brocade ICX 6610* sviča, postoji ukupno 16 očitavanja senzora temperature koji se prate.

Sledeći element u sistemu je (2) mrežni uređaj za skladištenje podataka (*Fujitsu Eternus DX200 D3*) koji ima 12 SAS diskova kapaciteta 600GB, što daje ukupno 7,2 TB kapaciteta za skladištenje. Deset diskova je u režimu rada, a dva imaju ulogu rezervnih diskova. Broj diskova nije ograničen na 12 i sistem za skladištenje se lako može proširiti u slučaju potrebe. Disk RAID je postavljen na *High Performance (RAID1+0)* sa ukupnim operativnim kapacitetom od 2,67 TB. Jedan deo diska ima kapacitet od 900 GB i ovaj deo koristi server (3). Drugi deo diska ima kapacitet od 1,79 TB i ovaj disk koristi server (4).



Slika 1. Šema sistema za monitoring Data centra

Server (4) je Fujitsu PY RX350 S8 sa 64GB RAM-a, dva Xeon E5-2697 v2 12C/24T 2.7GHz CPU i dva GPU Tesla K20X. Server (2) Fujitsu RX300 S8 je slabije konfiguracije i ima dva Xeon E5-2697 v2 sa 10 jezgara. Sa oba servera se mogu očitati podaci o temperaturi komponenata ta dva servera. Prvi server ima dva Intel Xeon E5-2697v2 procesora sa 12 jezgara. Svaki procesor može da izvrši očitavanje temperature za 12 jezgara, što daje ukupno 24 senzorska očitavanja. Drugi server ima dva Xeon E5-2697 v2 sa 10 jezgara, dajući ukupno 20 dodatnih senzora temperature. Pored temperature postoji mogućnost da se prati potrošnja CPU, potrošnja DRAM memorije, kao i upotreba hard diska i opterećenje memorije. Svi ovi podaci mogu biti dodati u sistem monitoringa, ali se u postojećem sistemu ne koriste u ovoj fazi. U sistemu se koristi i očitavanje temperature sa dva grafička akceleratora (GPU) Tesla K20X. Ostala očitavanja sa grafičkog procesora se ne primenjuju u trenutnom prototipu sistema koji se razvija u ovoj fazi istraživanja. Tako, ukupno dva grafička procesora GPU Tesla K20X i dva servera sa njihovim CPU obezbeđuju očitavanje temperature sa ukupno 46 CPU / GPU senzora za nadgledanje. Dodatno se vrši očitavanje temperaturnih senzora sa matičnih ploča oba servera, što daje ukupno 15 dodatnih vrednosti.

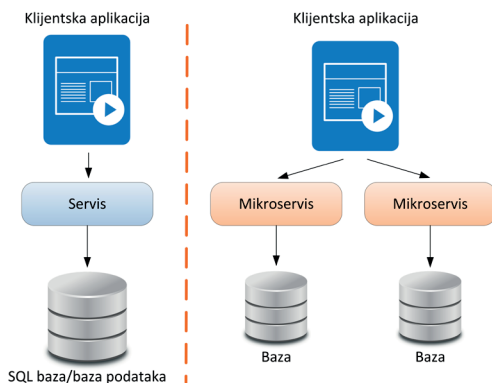
Sistem je osiguran sa 5000kVA UPS sistemom za kontinuirano napajanje (5). Na ovom uređaju prate se očitavanja temperature, izlaznog napona, frekvencije, snage i struje. Sistem dodatno nadgledaju senzorski uređaji koji se nalaze na senzorskim stanicama koji koriste razvojnu ploču Arduino/Genuino UNO i hardver otvorenog koda. Ukratko stanica SCF koristi DHT11 i BH1750FVI senzor za praćenje temperature, vlažnosti vazduha i osvetljenja. Stanica QAL koristi Grove temperaturni senzor i Bosch BMP85 senzor za praćenje temperature, odnosno temperature i barometarskog pritiska. Stanica VVV koristi takođe BMP85 senzor i DS18B20 za praćenje temperature i barometarskog pritiska, odnosno samo temperature. Komunikacija između Arduino/Genuino uređaja zasniva se na bežičnoj ZigBee tehnologiji, a centralni ZigBee uređaj (*Coordinator*) nalazi se u ormanu, direktno povezan sa serverom preko USB kabla na kome se nalazi prototip aplikacije. Da bi se mogli prikupljati podaci o temperaturi izvan opreme, tri senzorske stanice su postavljene na različite lokacije u *data* centru. Stanica QAL (6) se nalazi unutar ormara, na vrhu više postavljenog Brocade sviča, stanica VVV (7) nalazi se na vrhu ormara, a treća stanica SCF (8) nalazi se na udaljenosti od 2m od ormara blizu, prozora prostorije u kojoj se nalazi data centar. Konfiguracija senzorskih stanica u ovom slučaju je fokusirana na senzore temperature, tako da se na svakoj stanici nalazi više različitih senzora temperature. To ne predstavlja neko opterećenje za sistem s obzirom na nisku cenu senzora, a time se postiže i kontrola tačnosti izmerenih vrednosti.

#### **4. Pregled laganog modela arhitekture aplikacije**

U ovom poglavlju će biti prikazan lagani (eng. *lightweight*) model aplikacije koja je bazirana na mikroservisima, a koja se razvija sa ciljem da bude dodata u sistem za monitoring data centra. Važno je napomenuti da kada je reč o mikroservis arhitekturi, svaki od mikroservisa ima svoj logički domen kao i domen podataka. Kada je reč o životnom ciklusu svaki od domena mora biti autonoman. To zahteva da se svaki od domena međusobno razlikuje između mikroservisa. Svaki od njih će imati unikatan set

atributa i podataka koji će definisati kontekst operacija. Navedeno je veoma slično domenski vođenom dizajnu, gde je kontekst povezan sa tačno jednim mikroservisom.

Razlika u arhitekturi između tradicionalno napisanih aplikacija i aplikacija dizajniranih na osnovu mikroservis arhitekture se može videti na slici 2. Tradicionalne aplikacije imaju centralizovan sistem skladištenja podataka (jednu ili više baza podataka), dok je kod mikroservisa slučaj da svaki mikroservis ima svoju bazu podataka. Na slici 2. dat je uporedni prikaz između skladištenja podataka kod aplikacije napisane na tradicionalni način i aplikacije napisane u mikroservis arhitekturi.



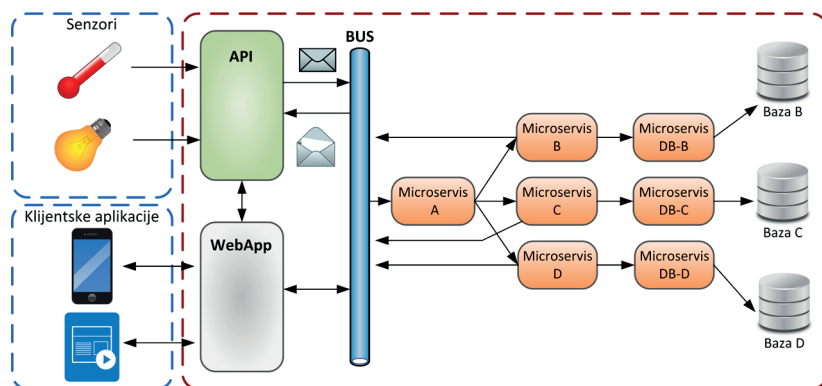
Slika 2. Uporedni prikaz skladištenja podataka između aplikacije napisane na tradicionalni način i aplikacije napisane u mikroservis arhitekturi

Kao što se može uočiti sa slike 2. tradicionalni pristup razvoja aplikacije je dosta jednostavniji, ili bar tako izgleda. Dosadašnja iskustva su pokazala da se sistem za skladištenje podataka svodi na ogromne tabele i baze kojima različiti podsistemi istovremeno pristupaju dovodeći do niza problema u radu.

Sa druge strane aplikacije zasnovane na mikroservis arhitekturi često koriste različite tipove sistema za skladištenje podataka. Pored relacionih baza podataka, moderne aplikacije danas često koriste i ne relacione baze podataka kao što su MongoDB, različiti tipovi digitalnih dokumenata, itd. Ovo se čini kako bi se postigle bolje performanse i skalabilnost. Ipak nisu retki slučajevi kada postoji kombinacija oba tipa sistema poznatija još kao poliglotski perzistentni pristup. Ovakav pristup upravo omogućava slabo povezan softver između servisa, bolje performanse, smanjenje troškova i lakšu upravljivost u procesu održavanja.

Na slici 3. dat je šematski prikaz arhitekture aplikacije zasnovane na laganom (eng. *lightweight*) modelu mikroservisa. Prednosti ovakvog razvoja aplikacije su višestruke, ali kao i druge arhitekture i ova ima svoje nedostatke. Najveća prednost u ovakvom pristupu je lako održavanje mikroservisa koji su relativno mali i međusobno nezavisni, i tako se postiže veća agilnost u razvoju i primeni novih verzija softvera. Razvoj može biti podeljen između više timova pri čemu je zavisnost između njih smanjena na minimum, a održavanje i uklanjanje potencijalnih problema je izolovano na nivo mikroservisa. Sa druge strane, nedostaci ovakve arhitekture se ogledaju u kompleksnoj komunikaciji između mikroservisa i primeni različitih komunikacionih protokola. Isto tako, porastom broja mikroservisa raste i kompleksnost primene istih u produkcijskim okruženjima. Na kraju, treba istaći da mikroservisi zahtevaju više

računarskih resursa kao što su RAM kapacitet, uređaji za skladištenje podataka, mrežni resursi, itd.

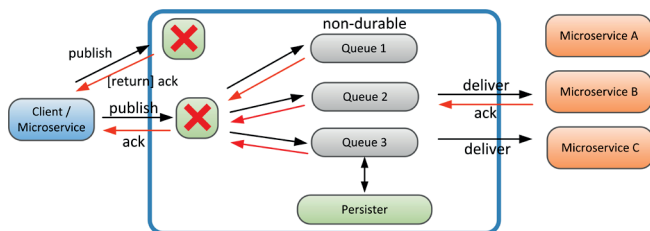


Slika 3. Lagani model mikroservis arhitekture za podršku sistemu za monitoring data centra

Na slici 3. se vidi i šematski prikaz toka poruka u aplikaciji zasnovanoj na mikroservis arhitekturi. Kada je reč o tipovima komunikacije u takvim sistemima se najčešće koriste dva tipa prenosa poruka, a to su asinhrono zasnovana komunikacija i komunikacija zasnovana na HTTP protokolu. Asinhrono-zasnovana komunikacija izvršava se uz pomoć magistrale događaja (*event bus*) koja propagira informacije, o novim događajima - mikroservisima. Propagirani događaj može biti interesantan jednom ili više mikroservisima u istom trenutku. Na ovaj način se može kontrolisati koji će mikroservis i u kom trenutku biti pokrenut. Kada je reč o implementaciji za magistralu događaja, u ovom sistemu se predviđa upotreba RabbitMQ [22] zbog mogućnosti postizanja visokih performansi i lakog skaliranja aplikacije. S druge strane, kada je reč o HTTP zasnovanoj komunikaciji ona se koristi u spoljašnjim pozivima ka WebApp segmentu arhitekture. Reč je o tradicionalnim HTTP zahtevima koji koriste upite za prosleđivanje podataka. Važno je istaći da komunikacija između mikroservisima i magistrale događaja nije jednosmerna, u pravcu od magistrale ka mikroservisu, već se može kretati i u drugom smeru. Na ovaj način ostvaruje se brza komunikacija između mikroservisima i postiže slabija povezanost sistema, kao i mogućnost da i drugi mikroservisi mogu da koriste obrađene podatke. Takođe, kao što se može uočiti svaki mikroservis ima svoju bazu podataka kroz koju perzistira podatke. Model podataka koji koristi mikroservis nalazi se u bazi podataka, a time se postiže veća dostupnost podataka kao i mogućnost lakšeg skaliranja sistema.

Kada je reč o brokerima poruka i servisnim magistralama (eng. *bus*) postoje različite implementacije kada je upitanju mikroservis arhitektura. RabbitMQ kao broker poruka često je rešenje u produkcijskoj implementaciji, ali se i druga rešenja često mogu koristiti, sve u zavisnosti koji se nivo skalabilnosti želi postići. Na kraju, uvek je moguće razviti i sopstveni protokol na osnovu već postojećeg. Kao što se može uočiti sa slike 3., predloženi lagani model arhitekture zasnovane na mikroservisu koristi kao backbone magistralu (RabbitMQ) kao komunikacioni protokol zasnovan na događajima. Svaki od mikroservisima se pretplaćuje na komunikacioni protokol i čeka da se određeni događaj desi, odnosno na redu (*queue*) RabbitMQ-a pojavi određena poruka (zahtev) koji je stigla

iz samog sistema ili spolja. Na slici 4. dat je shematski prikaz razmene poruka između mikroservisa i RabbitMQ-a.



Slika 4. Shematski prikaz komunikacije između mikroservisa i RabbitMQ [23]

## 5. Zaključak

Ovo istraživanje je usmereno u pravcu odgovora na zahteve u razvoju aplikacija koje se postavljaju pred softverskim inženjerima današnjice, a koji su prvenstveno prouzrokovani naglim i veoma širokim razvojem novih tehnologija, kao i njihovom raznovrsnošću. To je proces koji traje već skoro deceniju, a njegov dalji tok se ne može predvideti u potpunosti. U radu je dat fokus na novije pristupe u razvoju sistema koje se prilagođavaju okruženjima bežičnih senzorskih mreža (WSN), Internetu stvari (IoT) i okruženjima pametnih gradova.

Centralni deo istraživanja je pristup u razvoju softverskog sistema zasnovanog na mikroservisima za podršku senzorskim mrežama. Osnovna namena sistema je monitoring data centra i obuhvata kombinovano i žične i bežične senzorske stanice. Model sistema se može primeniti i u industrijskim uslovima, sistemima pametnih gradova, kućnoj automatizaciji i sl. Razvoj sistema je u početnoj fazi i u radu je prvenstveno prikazan pristup u dizajnu njegove arhitekture, kao i prikaz tehnologija koje su za to potrebne. Pored prikaza samog sistema navedene su prednosti i nedostaci takvog pristupa, ali i primeri sličnih sistema u svetu.

## Zahvalnica

Ministarstvo prosvete, nauke i tehnološkog razvoja Republike Srbije podržava ovo istraživanje u okviru projekta TR32044 "Razvoj softverskih alata za analizu i poboljšanje poslovnih procesa", 2011-2018.

## Reference

- [1] International Data Corporation (IDC). IDC Corporate USA, dostupno na: <https://www.idc.com/getdoc.jsp?containerId=prUS43295217>, pristupljeno oktobar 2018.
- [2] L. Sun, Y. Li and R. A. Memon, "An open IoT framework based on microservices architecture," *China Communications*, Vol. 14, no. 2, pp. 154-162, February 2017. doi: 10.1109/CC.2017.7868163
- [3] S. Daya, N. Van Duy, K. Eati, C.M. Ferreira, D. Glozic, V. Gucer, M. Gupta, S. Joshi, V. Lampkin, M. Martins, *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*. IBM Redbooks, 2016.



- [4] C. Hung, C. Hsieh, Chapter 5 - Big Data Management on Wireless Sensor Networks, *In Intelligent Data-Centric Systems*, „Big Data Analytics for Sensor-Network Collected Intelligence“, Academic Press, 2017, pp 99-116, <https://doi.org/10.1016/B978-0-12-809393-1.00005-2>.
- [5] C. Esposito, A. Castiglione, F. Palmieri, M. Ficco, C. Dobre, G. V. Iordache, F. Pop, „Event-based sensor data exchange and fusion in the Internet of Things environments“, *Journal of Parallel and Distributed Computing*, Vol. 118, Part 2, 2018, pp 328-343, <https://doi.org/10.1016/j.jpdc.2017.12.010>.
- [6] The Constrained Application Protocol (CoAP), IEEE Std. rfc7252, 2014.
- [7] Message Queue Telemetry Transport (MQTT), OASIS Std., Rev. 3.1.1, 2014.
- [8] A. Krylovskiy, M. Jahn and E. Patti, "Designing a Smart City Internet of Things Platform with Microservice Architecture," *Proceedings of 3rd International Conference on Future Internet of Things and Cloud*, Rome, Italy. 2015, pp. 25-30. doi: 10.1109/FiCloud.2015.55
- [9] G. Cherradi, A. EL Bouziri, A. Boulmakoul, K. Zeitouni, „Real-Time Microservices Based Environmental Sensors System for Hazmat Transportation Networks Monitoring“, *Transportation Research Procedia*, Vol. 27, 2017, pp 873-880, <https://doi.org/10.1016/j.trpro.2017.12.087>
- [10] A. de M. Del Esposte, F. Kon, F. M. Costa and N. Lago. “InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities”. *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems*, 2017.
- [11] R. Morabito, R. Petrolo, V. Loscri, N. Mitton, “LEGIoT: A Lightweight Edge Gateway for the Internet of Things”, *Future Generation Computer Systems*, Vol. 81, 2018, pp 1-15, <https://doi.org/10.1016/j.future.2017.10.011>.
- [12] E. G. M. Petrakis, S. Sotiriadis, T. Soultanopoulos, P. T. Renta, R. Buyya, Nik Bessis, “Internet of Things as a Service (iTaaS): Challenges and solutions for management of sensor data on the cloud and the fog”, *Internet of Things*, Vol. 3–4, 2018, pp 156-174, <https://doi.org/10.1016/j.iot.2018.09.009>.
- [13] A. J. Fernández-García, L. Iribarne, A. Corral, J. Criado, J. Z. Wang, “A flexible data acquisition system for storing the interactions on mashup user interfaces”, *Computer Standards & Interfaces*, Vol. 59, 2018, pp 10-34, <https://doi.org/10.1016/j.csi.2018.02.002>.
- [14] L. M. Vaquero, F. Cuadrado, Y. Elkhatib, J. Bernal-Bernabe, S. N. Srirama, M. F. Zhani, “Research challenges in nextgen service orchestration”, *Future Generation Computer Systems*, Vol. 90, 2019, pp 20-38, <https://doi.org/10.1016/j.future.2018.07.039>.
- [15] M. Nahri, A. Boulmakoul, L. Karim, A. Lbath, “IoV distributed architecture for real-time traffic data analytics”, *Procedia Computer Science*, Vol. 130, 2018, pp 480-487, <https://doi.org/10.1016/j.procs.2018.04.055>.
- [16] D. Dobrilović, Ž. Stojanov, Z. Čović, J. Simon and N. Petrov, "Model of data center temperature monitoring system with the use of open source hardware," *Proceedings of IEEE 14th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, 2016, pp. 221-226. doi: 10.1109/SISY.2016.7601501
- [17] D. Dobrilovic, B. Odadzic, Z. Stojanov and V. Sinik, "Testing Zigbee RF module applicability for usage in temperature monitoring systems," *Proceedings of 22nd Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia, 2014, pp. 415-418. doi: 10.1109/TELFOR.2014.7034436

- [18] D. Dobrilovic, Z. Stojanov, V. Brtko, Z. Čović and N. Bilinac, "Software application for analyzing ZigBee network performance in university courses," *Proceedings of IEEE 12th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, 2014, pp. 73-77. doi: 10.1109/SISY.2014.6923560
- [19] Milan Malić, Dalibor Dobrilović, Dušan Malić, Željko Stojanov, Srdan Sladojević, „Prototip uređaja za merenje performansi LoRa tehnologije baziran na hardveru otvorenog koda,, Tehničko rešenje, 2018. Dostupno na: [http://www.tfzr.rs/Content/files/0/Teh\\_Resenje\\_LoRa.pdf](http://www.tfzr.rs/Content/files/0/Teh_Resenje_LoRa.pdf)
- [20] M. Marcu, S. Ficiu, D. Dobrilovic, M. Popa, B. Odadzic, „Cross-border infrastructure for educational webcasting“, *Proceedings of International Conference on Applied Internet and Information Technologies AIIT 2015*, pp 225-230, October 23, Zrenjanin, Serbia, 2015.
- [21] S. Fuicu, M. Popa, D. Dobrilovic, M. Marcu, R. Bogdan, “Developing Distance Learning Environments in the Context of Cross-Border Cooperation”, *BRAIN Broad Research in Artificial Intelligence and Neuroscience*, pp 52-58, Vol. 8, Issue 1, April, 2017.
- [22] G. M. Roy, *RabbitMQ in Depth*, Manning Publications, 2017.
- [23] Pivotal Software, Inc., dostupno na: <http://www.rabbitmq.com/blog/2011/02/10/introducing-publisher-confirms/> pristupljeno oktobar 2018.

**Abstract:** *The rapid development of new technologies that have contributed to the increasing use of wireless communication technologies in sensor networks, as well as the decrease in hardware prices required for their implementation, have forced software engineers to consider and develop new solutions for the architecture of these systems. The current approach to application development has proven to be insufficient in meeting the new requirements that such applications should fulfill. Because of this, the architecture of the system based on microservices has emerged as a potential solution that could overcome imposed problems. This paper presents the proposal for the solution of the architecture of such a system, as a lightweight model of microservices that can be used as the basis for the development of complex systems, e.g. for home automation, monitoring in smart cities, etc. Because of the complexity, in this paper, only the most important segments of architecture are presented, as well as the potential problems and advantages that this architecture carries with them.*

**Keywords:** *wireless sensor networks, software architecture, microservices, monitoring systems*

## **PROPOSAL OF THE SYSTEM ARCHITECTURE BASED ON MICROSERVICES FOR WIRELESS SENSOR NETWORKS SUPPORT**

Milan Malić, Dalibor Dobrilović, Dušan Malić