



PREDLOG ARHITEKTURE DISTRIBUIRANE MOBILNE APLIKACIJE ZA PROCENU RIZIKA

Srđan Popov¹, Miroslav Bender¹, Jelena Tucakov²

¹Fakultet tehničkih nauka Novi Sad

²Visoka tehnička škola strukovnih studija u Zrenjaninu

Abstract:

U savremenom pristupu obradi podataka postoji težnja da se događaji obrađuju na mestu njihovog nastanka. Jedna od situacija koja zahteva takav pristup je procena rizika prilikom ostvarenja nekog hazarda. Ona podrazumeva prikupljanje i obradu podataka na samom mestu ostvarenja. Napredak tehnologije i trenutno stanje u računarstvu i komunikacijama su omogućili razvijanje takve aplikacije. U radu su analizirane moguće arhitekture zasnovane na Android mobilnoj platformi, Json formatu za razmenu podataka i NoSql rešenjima za skladištenje prikupljenih podataka.

Key words:

Android,
JSON,
NoSQL.

UVOD

Svakodnevne operativne aktivnosti u bavljenju rizikima od hazardnih događaja sa katastrofalnim posledicama, suočavaju se sa problemom podrške u odlučivanju na terenu. Odlučivanje se zasniva na nizu parametara (hazardi, ranjivost, izloženost, izdržljivost, ...) koji se izračunavaju upotrebom indikatora. Parametri su u slučaju odlučivanja ili poznati ili zavise od indikatora koje treba lokalno prikupiti.

U prethodnom periodu proces prikupljanja podataka i podrška odlučivanju o proceni rizika od hazardnih događaja sa katastrofalnim posledicama zahtevao je odlazak na lokaciju događaja. Na licu mesta bi se evidentirali podaci i opis njihovog prikupljanja, a zatim bi se podaci unosili u bazu podataka za procenu rizika. Takvim načinom obrade otvarale su se mogućnost za nepreciznosti i greške zbog različitih razloga: nepreciznih podataka, neodgovarajuće interpretacije ili nestandardizovanog pretstavljanja prikupljenih podataka. Osim toga postojalo je i vremensko kašnjenje od trenutka obrade na mestu događaja do trenutka kopmpletne obrade u sistemu procene rizika.

U tom smislu može se reći da se od samog početka primene automatizovane obrade podataka težilo da se događaji obrađuju na mestu njihovog nastajanja. Međutim, problemi obezbeđivanja komunikacija, gabariti i procesna moć računarske opreme su uglavnom nalagale da se vrše tzv. pozadinske obrade podataka koji su prikupljeni s terena. Danas su, međutim, u uslovima brzog napredovanja u oblasti komunikacija, interneta i mo-

bilnih uređaja svih vrsta, stvorene mogućnosti razvoja mobilnih aplikacija za najrazličitije primene. Jednostavno se kao standard nameće primena softverskih rešenja koja koriste mobilne uređaje s internet vezama. Postavlja se pitanje kako koncipirati arhitekturu aplikacije za mobilne uređaje. U radu je analizirana moguća arhitektura distribuirane mobilne aplikacije za prikupljanje podataka za procenu rizika.

ARHITEKTURA

Osnovni Aspekti

Generalno, kad su u pitanju aplikacije koje koriste mobilne uređaje za prikupljanje podataka, moguća su dva pristupa. Razvoj samostalne aplikacije ili korišćenje standardnog web pretraživača na strani mobilnog uređaja. U nekim slučajevima potrebno je obezbediti rad ne samo u *on-line* režimu već, vodeći računa o tome da mobilni uređaj nije u mogućnosti da neprestano obezbedi mrežnu konekciju, potrebno je obezbediti i mogućnost rada u *off-line* režimu. Kada postoje takvi funkcionalni zahtevi, jedino rešenje je samostalna mobilna aplikacija jer se u slučaju korišćenja web pretraživača teško može obezbediti potpuna *off-line* funkcionalnost. Pri tome treba voditi računa da je razvoj samostalne mobilne aplikacije skuplje rešenje, u smislu potrebnih napora, veština i vremena za njen razvoj. Zbog toga se razvoj samostalne aplikacije predlaže samo u slučaju eksplicitnih zahteva za obezbeđivanjem, uglavnom ograničene, funkcionalnosti i uslovima kada mobilni uređaj ostaje bez veze sa ostatkom sistema.



U pogledu arhitekture, mobilne aplikacije imaju mnoge zajedničke karakteristike kao i klijent-server ili web aplikacije sa višeslojnom arhitekturom. Međutim, postoje i segmenti u kojima se arhitektura i projektovanje mobilnih aplikacija razlikuju od klasičnih klijent-server ili višeslojnih web aplikacija. Zbog toga je tim aspektima arhitekture mobilnih aplikacija potrebno posvetiti pažnju na samom početku procesa razvoja aplikacije, da bi se izbegle posledice loše definisane arhitekture.

Uopšteno govoreći, aplikacije na mobilnim platformama potrebno je integrisati u informacionu strukturu celog sistema, ali zbog ograničenih resursa mobilnih uređaja interakcija s bazom podataka postaje jedan od važnih problema. Zbog toga (ograničenih resursa: procesor, memorija, baterija, veličina ekrana) je potrebno posvetiti pažnju formatima za efikasnu razmenu, parsiranje i interpretaciju podataka, i odabrati formate koji su efikasni u pogledu potrošnje resursa mobilnog uređaja i komunikacije s web servisima i bazom podataka.

Važan segment pri projektovanju mobilnih aplikacija je obezbeđivanje njihove *off-line* funkcionalnosti. Iako se *off-line* funkcionalnost smatra redukovanom, ipak se otvaraju mogućnost pristupa svim komponentama mobilnog uređaja, kao što su kamera ili GPS, a za smeštanje veće količine podataka se može koristiti fajl sistem samog uređaja, takođe se javljaju i problemi o kojima treba voditi računa. Mora se obezbediti sinhronizacija uređaja s bazom podataka kada se ponovo uspostavi veza s ostatkom sistema. Pri tome se moraju obraditi moguće konfliktne situacije da nebi došlo do gubljenja podataka ili situacije kada bi podaci postali nepouzdana.

Moguća rešenja

Trenutno aktualne arhitekture koje mogu zadovoljiti spomenute zahteve u adekvatnoj meri su:

- ◆ Microsoft, Windows Distributed Network Architecture (Windows DNA)/.NET,
- ◆ Object Management Group, Common Object Request Broker Architecture (CORBA),
- ◆ Sun, Java 2 Enterprise Edition (JavaEE) Architecture.

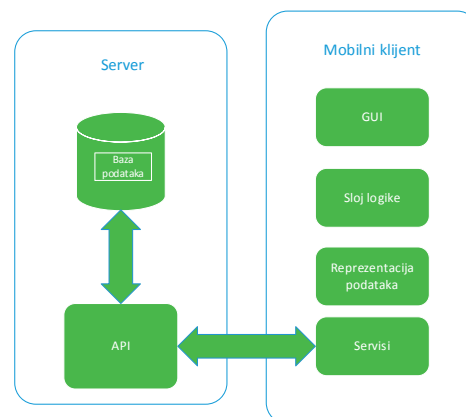
Tradicionalni nedostatak Microsoft zasnovanog rešenja je nemogućnost izvršavanja na drugim platformama, pre svega danas veoma aktuelnoj Android platformi. Takođe evidentan i teško premostiv problem se mala ponuda prenosnih uređaja koji imaju podršku za ovu platformu.

CORBA bazirano rešenje je prekomplikovano za potrebnu dinamiku adaptivnog razvoja aplikacije za podršku u odlučivanu oblasti koja je zavisna od primene savremenih, aktuelnih i često prototipskih komponenti i senzorskih sistema.

Kao jedina prihvatljivo rešenje se nameće JAVAEE. Sun-ova specifikacija i standard donosi dinamičnu i široko podržano rešenje koje obuhvata tehnologije kao što su: EnterpriseJavaBeans (EJB), Java Servlets API and Java Server Pagers (JSP), i mnoge druge. JAVAEE je izvršiva na različitim platformama i dominantna je na danas ak-

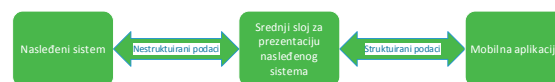
tuelnoj i ekonomski dostupnoj Android zasnovanoj platformi.

Ako JavaEE takođe posmatramo iz aspekta razvoja mobilne aplikacije čija arhitektura predstavlja sistem elemenata visokog nivoa apstrakcije i potrebne interakcije između elemenata, i potrebnih ograničenja, sa potrebom da se zadovolji tipična mobilna višeslojna arhitektura prikazana na Slici 1., jedno veoma kvalitetno rešenje sa potrebnim tehnologijama za svaku pojedinačnu komponentu.



Sl. 1. Tipična arhitektura mobilne višeslojne aplikacije

Još jedan, bitan aspekt u korist JavaEE je potreba za interakcijom mobilne aplikacije sa nasleđenim sistemima poput onih koje imaju RHMZ-a, RGZ, VGI, osiguravajuća društva i dr. [1] [2] [3] sl. 2.



Sl. 2. Interakcija mobilne aplikacije sa nasleđenim sistemima

Poslednji aspekt koji pominjemo je taj da je transfer podataka značajan aspekt arhitekture mobilne aplikacije, pogotovo u svetlu koraka koje podatak treba da obavi. Pri tome mora biti zadovoljena potreba za bezbednošću i dostupnosti podataka i jednostavnosti implementacije rešenja.

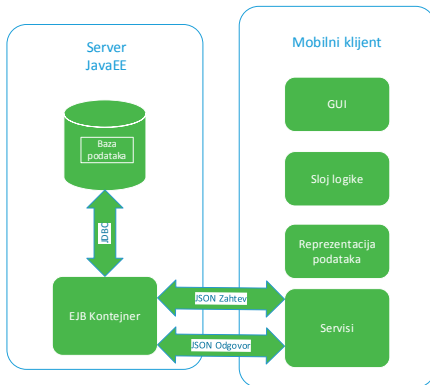
FORMAT PORUKA

Kada smo izabrali JAVAEE kao ciljanu platformu, sledeći korak nam je izbor formata za razmenu podataka. Kada u Java aplikaciji imamo potrebu za razmenom podataka imamo dve aktuelne mogućnosti, XML i JSON format. Oba formata imaju svoju prostorno zasnovanu verziju GML i GeoJSON, stoga se može reći da su funkcionalno u potpunosti sposobne da zadovolje potrebe želeje mobilne aplikacije.

Format dokumenta JSON, pogodan je za jednostavno rukaovanje podacima, takođe je pogodan i kao format za razmenu podataka u uslovima ograničenih resursa mobilnih uređaja. Prilikom rukovanja JSON-om nema konverzije podataka, i jednostavniji je u odnosu na XML format. Takođe JSON pogodniji i zbog manje potrošnje resursa pri obradi na strani mobilnog uređaja [4] [8].



Stoga tipičnu arhitektura mobilne višeslojne aplikacije sa sl. 1. specijalizujemo upotrebom JSON formata dokumenata. Na sl. 3. Prikazana je arhitektura mobilne višeslojne aplikacije u kojoj se koristi JSON.



Sl. 3. Arhitektura JAVAEE mobilne višeslojne aplikacije sa JSON razmenom podataka

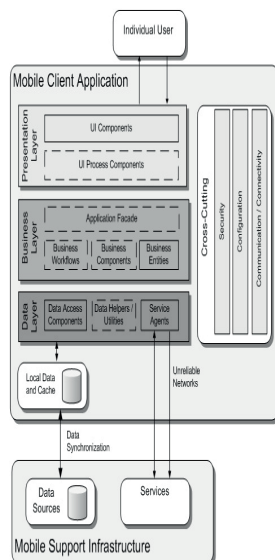
MOBILNA PLATFORMA

Kao što je ranije pomenuto JavaEE je dominantna na danas aktuelnoj i ekonomski dostupnoj Android zasnovanoj platformi [5]. Savremeni mobilni operativni sistem zasnovan na Linux kernelu dozvoljava, a rekli bi i motiviše razvoj mobilnih aplikacija kodiranih u Java programskom jeziku. Ovako kodirane aplikacije koriste Google-enabled Java biblioteke za pristup mobilnom uređaju. Za razvoj aplikacija Google je takođe obezbedio Android SDK, dok je za sam proces kodiranja na raspolaganju više integrisanih okruženja, između kojih se ističe Eclipse IDE.

Android platforma, kako javno ističe Google, ima četiri osnovne osobine:

- ♦ otvorenog je koda,
- ♦ aplikacije su jednake,
- ♦ razvoj aplikacija je jednostavan
- ♦ nema ograničenja primene,

i kako se može videti na sl. 4. softverski stek koji uključuje tri osnovna sloja operativni sistem, srednji sloj i aplikacioni sloj.



Sl. 4. Arhitektura Androida [6]

Primarna funkcija Android aplikacije u sistemu za procenu rizika je prikupljanje podataka. Zbog toga je potrebno obazbediti korisnicima da podatke uskladište bez obzira na status mrežne konekcije.

Android aplikacija preko Java interfesa komunicira sa serverom na kome se nalazi CouchDB baza podataka. Za prenos podataka (JSON objekata) od klijenta ka serveru koristi se HTTP protokol. Svaki zahtev koji se šalje serveru treba da sadrži, pored podataka koji će biti sačuvani, parametre odgovarajućeg CouchDB servera (IP i port servera centralne baze podataka).

UKLOVANJE PODACIMA

Prilikom izbora arhitekture servera baze podataka i sistema za upravljanje bazom, do skoro nije bilo nikakvih dilema, relacioni sistemi su decenijama bili jedini mogući izbor. Relacione baze podataka su najbolje rešenje za rukovanje strukturiranim podacima, kada je šema podataka fiksna i unapred definisana.

Međutim, kada su u pitanju geo prostorni podaci susrećemo sve veći priliv nestruktuiranih i slabo strukturiranih podataka za koje relacione baze podataka pokazuju određene slabosti. U takvim situacijama neke od NoSQL baza podataka mogu obezbediti veću raspoloživost, skalabilnost, jednostavnije rukovanje podacima i bolje performanse.

NoSQL baze podataka su došle kao odgovor na potrebu čuvanja i rukovanja velikim, uglavnom nestruktuiranim, količinama podataka kao i potrebe za horizontalnim skaliranjem u takvim slučajevima. S tim konceptom prvi su, zbog sopstvenih potreba, krenuli Google s konceptom *Google's BigTable* [9] i Amazon sa svojim konceptom *Amazon's Dynamo* [10]. Od tog perioda do danas razvijena su mnoga NoSQL rešenja čiji je zadatak da zadovolje zahteve i potrebe različitih organizacija u cilji skladištenja podataka. Skoro svaki sistem baza podataka koji ne podleže principima relacionih baza podataka može se svrstati u NoSQL sisteme. Danas se NoSQL baze podataka dele u sledeće grupe [11]:

- ♦ Ključ-Vrednost baze podataka,
- ♦ Kolonski orijentisane baze podataka,
- ♦ Dokument orijentisane baze podataka
- ♦ Graf orijentisane baze podataka

Zajedničke karakteristike NoSQL baza podataka su: visoka skalabilnost i pouzdanost, jednostavan model podataka, nedostatak transakcionog mehanizma za rukovanje podacima i mehanizma za rukovanje ograničenjima i integritetom.

Prednosti NoSQL baza podataka u odnosu na relacione baze su: rukovanje različitim tipovima podataka kao što su grafovi, objekti, slabo strukturirani i nestruktuirani podaci. Osim toga, NoSQL baze podataka su pogodnije pri obradi većih količina podataka i podržavaju jednostavniju skalabilnost što ih posebno kvalifikuje za upotrebu u distribuiranim okruženju.

Prednosti koje donose i sve veća primena NoSQL baza podataka u različitim oblastima [12] dovela je do ideje njihove primene i u geo sistemima. U tom smislu je i ovaj rad



pokušaj da se neke od prednosti koje donose NoSQL baza podataka integrišu u informacionu strukturu integralnog sistema za procenu rizika.

Postavlja se pitanje koju vrstu NoSQL baza podataka izabrati za podršku serverskoj strani i kako rukovati podacima na strani mobilnog klijenta sistema za procenu rizika.

Server baze podataka

Obzirom na činjenicu da NoSQL sistemi baza podataka podacima rukuju bez šeme baze podataka to im omogućava slobodnu izmenu strukture slogova baze podataka i jednostavno dodavanje ili uklanjanje obeležja bez potrebe izmene šeme baze podataka.

Kada su u pitanju mobilne aplikacije NoSQL baze podataka su interesantne zbog jednostavne izmene strukture poruka kojima mobilni uređaj komunicira s ostatkom sistema i samih zapisa baze podataka.

Osim pomenutih razloga, kod geoinformacionih sistema postoje oblasti primene u kojima se mogu iskoristiti i neke druge prednosti NoSQL baza podataka koje dolaze do izražaja kada se javi potreba za rukovanjem velikom količinom podataka. Jedna od tih situacija kada u nekom geoinformacionom sistemu dolazi do intezivnog priliva velikog obima podataka je komunikacija geo servera s klijentima koje predstavljaju senzori raspoređeni na terenu i koji u definisanim intervalima vremena dostavljaju izmerene podatke. U tim slučajevima, ukoliko je izabrana neka od arhitektura servera baze podataka koju podržavaju NoSQL baze, mogu se relativno jednostavno iskoristiti sve pogodnosti koje nude NoSQL sistemi baza podataka kao što su: horizontalna skalabilnost i distribucija.

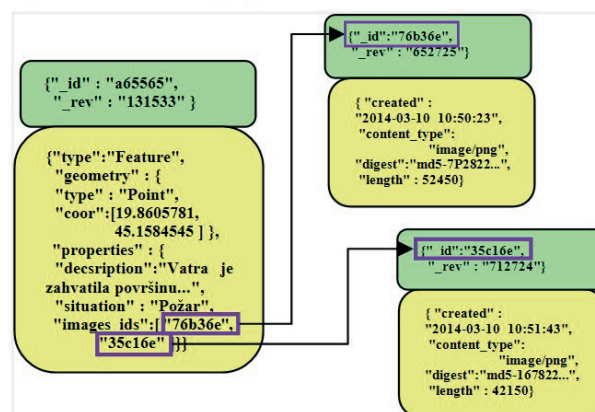
Analizirajući svaku od navedenih grupa NoSQL baza podataka, kao predlog servera baze podataka sistema za procenu rizika nameće se izbor neke od dokument orijentisanih baza podataka. U većini dokument orijentisanih baza podataka dokument koji se smešta u bazu je u JSON formatu. Struktura dokumenta je proizvoljna i može sadržati druge dokumente. U opštem slučaju dokumenti ne moraju da imaju istu strukturu. Ovakav pristup olakšava razvoj aplikacija gde unapred nije poznato sa kakvim će se podacima raditi i gde je potrebno čuvati više različitih tipova podataka unutar jednog dokumenta. U tom smislu je ova grupa NoSQL baza pogodna ne samo za jednostavno rukovanje podacima već i za pojednostavljenu razmenu podataka u komunikaciji između modula i komponenti sistema.

Danas su najčešće korišćene CouchDB i MongoDB dokument orijentisane NoSQL baze podataka.

CouchDB je *open source* dokument orijentisana NoSQL baza podataka. Za smeštanje dokumenata u bazu koristi JSON format. Pristup dokumentima vrši se putem HTTP protokola. Za realizaciju upita i transformacije dokumenata koristi se *JavaScript*. CouchDB pokazuje solidne performanse pri radu sa modernim web i mobilnim aplikacijama. Podaci se efikasno distribuiraju koristeći inkrementalnu replikaciju.

Činjenica da se podaci skladište u JSON formatu, kvalifikuje CouchDB kao mogući izbor za server baze podataka sistema za procenu rizika sa Android mobilnim klijentom. U tom slučaju bi se obezbedio identičan format razmene poruka u sistemu i format dokumenata u bazi podataka, čime bi se izbegla potreba konverzije i mapiranja podataka iz jednog formata u drugi. MongoDB je takođe dokument orijentisana NoSQL baza podataka, ali se kod nje dokumenti čuvaju u binarnom JSON formatu (*Binary JSON - BSON*). BSON format je efikasniji za razmenu jer je kompaktniji i poruke su kraće ali u slučaju izbora MongoDB-a za server baze podataka morala bi se vršiti konverzija komunikacionih poruka.

Dokumenti koji će se razmenjivati u sistemu će biti složeni JSON dokumenti za skladištenje podataka vezanih za lokacije, tzv. GeoJSON. Na sl. 5 prikazana je planirana struktura dokumenata. Svaki od dokumenata u bazi sadrži meta podatke (na slici 5 su označeni zelenim pravougaonicima) i podatke unete od strane korisnika (na slici 5 su označeni žutim pravougaonicima). Ukoliko korisnik želi da pošalje više fotografija vezanih za datu situaciju, svaka fotografija će biti snimljena u bazu kao zaseban dokument. Dokumenti vezani za lokaciju i dokumenti sa fotografijama poveziće se preko meta podataka kao što je prikazano na slici 5.



Sl. 5. Struktura JSON dokumenata koje Android klijent šalje serveru

Rukovanje podacima na strani mobilnog klijenta

Razvoj samostalne aplikacije na strani mobilnog klijenta na Android platformi obezbeđuje prednosti korišćenja svih komponenti samog uređaja i podrške za *off-line* funkcionalnost.

Kod mobilnog klijenta na Android platformi potrebno je rukovati relativno malom količinom podataka, zbog čega je moguće koristiti i memorijski prostor samog uređaja za privremeno čuvanje prikupljenih podataka do trenutka slanja ka serveru i njegove potvrde prijema.

Kada se na strani mobilnog klijenta rukuje većom količinom podataka, zbog pouzdanosti i efikasnosti rukovanja podacima potrebno je koristiti NoSQL bazu za mobilne klijente. Na taj način je moguće izbeći konverziju formata podataka jer je obezbeđeno da format podataka na serverskoj strani, mobilnom klijentu i sama komunikacija budu u istom formatu.



Kada je u pitanju mobilni klijent sistema za procenu rizika potrebno je već sada predvideti mogućnost da se u narednoj fazi razvoja, bez obzira na količinu podataka, i na strani mobilnog klijenta koristi NoSQL baza podataka u cilju sticanja iskustava u radu s njima jer se radi o potpuno novim softverskim rešenjima.

ZAKLJUČAK

NoSQL baze u velikoj većini skladište JSON u izvornom formatu, što je pogodno da se implementira koncept mobilne aplikacije koja kao format podataka koristi JSON. Zato se dokument orijentisane NoSQL baze podataka predlažu za korišćenje u sistemu procene rizika. CouchDB, kao skladište podataka orijentisano ka dokumentima predstavlja jedno od idealnih rešenja za čuvanje velikih količina podataka pa je upravo to rešenje izabrano u cilju implementacije centralne baze podataka koja skladišti podatke koji pristižu sa Android aplikacije za akviziciju podataka.

U poslednje vreme se pojavljuju i NoSQL sistemi za mobilne uređaje tako da se u narednom preiodu funkcionalnost mobilnog klijenta može proširiti korišćenjem takvog sistema i kada je u pitanju manja količina podataka, kao što je slučaj kod mobilnog klijenta sistema za procenu rizika.

LITERATURA

- [1] M. Bender, E. Šećerov, V. Šenk, S. Popov, Application Gateway between Open and Legacy Systems, EUROCON 2005 Serbia & Montenegro, Belgrade, November 22-24, 2005
- [2] S. Popov, A. Pavlović, Đ. Ćosić M. Hlebjan, Interfacing Data Structures of Legacy Systems, Intelligent Systems and Informatics (SISY), 8th International Symposium, Subotica, September 10-11, 2010
- [3] J. McCormick, Mainframe-web middleware, Government Computer News, Jun 2000
- [4] N. Nurseitov, M. Paulson, R. Reynolds, C. Izurieta, Comparison of JSON and XML Data Interchange Formats: A Case Study, Proceedings of the ISCA 22nd International Conference on Computer Applications in Industry and Engineering, CAINE 2009, November 4-6, USA, 2009
- [5] S. Holla, M. M. Katti, Android based mobile application development and its security, International Journal of Computer Trends and Technology, volume3 Issue3, 2012
- [6] What is android? <http://developer.android.com/guide/basics/what-is-android.html>
- [7] M. Butler, "Android: Changing the mobile landscape", IEEE Journal of Pervasive Computing, vol. 10(1), pp. 4-7, 2011.
- [8] Maeda K. Performance evaluation of object serialization libraries in XML, JSON and binary formats. Digital Information and Communication Technology and its Applications (DICTAP), 2012 Second International Conference on. IEEE, 2012
- [9] Fay Chang, Jeffery Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A Distributed Storage Sistem for Scructured Data. OSDI'06: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, 26(2): 205-218, November, 2006
- [10] G Decandia, D Hastorun, M Jamani, G Kakulapati, A Lakshman, A Pilchin, S Sivasubramanian, P Vosshall, and W Vogels. Dynamo: Amazon's Highly Available Key-value Store. ACM SIGOPS OperatingSystems Review, 41(6):205-220, 2007
- [11] Moniruzzaman, A. B. M., and Syed Akhter Hossain. "NoSQL Database: New Era of Databases for Big data Analytics-Classification, Characteristics and Comparison." *International Journal of Database Theory & Application* 6.4, 2013
- [12] Matti, M and Kvernik, T; "Applying Big-Data Tehnologies to Network Architecture"; Ericsson Rewiew, October 2012

ARCHITECTURE PROPOSAL OF THE DISTRIBUTED MOBILE APPLICATION FOR RISK ASSESSMENT

Abstract:

In today approach to data processing, there is a tendency that the events handle in the place of their occurrence. Situation that demands this approach is the risk assessment in situation of realization a hazardous situation. It involves aquisition and processing of data on the site. Progress in technology and the current state of computing and communications have made it possible developing of such application. This paper presents an analysis of the architecture based on the Android mobile platform, Json format for data exchange and NoSQL solutions for the storage of collected data.

Key words:

Android,
JSON,
NoSQL.