



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ



Сунчица Миливојша

# Интеграција система паметне куће са Амазон Алекса сервисом

ДИПЛОМСКИ РАД  
- Основне академске студије -

Нови Сад, 2017



## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Сунчица Миливојша		
Ментор, МН:	Проф. др Иштван Пап		
Наслов рада, НР:	Интеграција система паметне куће са Амазон Алекса сервисом		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2017		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	7/30/0/11/15/0/0		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Интернет ствари, гласовна контрола, систем за кућну аутоматизацију		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	У овом раду је приказан концепт једног решења за имплементацију модула за гласовну контролу у оквиру ОБЛО система за паметне куће. Представљена је имплементација гласовне функционалности користећи Амазон Алекса сервис за препознавање говора. Дати модул за гласовну контролу је интегрисан у ОБЛО облак систем, и обезбеђује гласовну контролу паметним уређајима као и надгледање.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Доц. др Богдан Павковић	
	Члан:	Доц. др Јелена Ковачевић	Потпис ментора
	Члан, ментор:	Проф. др Иштван Пап	



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO:</b>			
Identification number, <b>INO:</b>			
Document type, <b>DT:</b>	Monographic publication		
Type of record, <b>TR:</b>	Textual printed material		
Contents code, <b>CC:</b>	Bachelor Thesis		
Author, <b>AU:</b>	Sunčica Milivojša		
Mentor, <b>MN:</b>	Ištvan Papp, PhD		
Title, <b>TI:</b>	Integration of Amazon Alexa service with smart home automation system		
Language of text, <b>LT:</b>	Serbian		
Language of abstract, <b>LA:</b>	Serbian		
Country of publication, <b>CP:</b>	Republic of Serbia		
Locality of publication, <b>LP:</b>	Vojvodina		
Publication year, <b>PY:</b>	2017		
Publisher, <b>PB:</b>	Author's reprint		
Publication place, <b>PP:</b>	Novi Sad, Dositeja Obradovica sq. 6		
Physical description, <b>PD:</b> (chapters/pages/ref./tables/pictures/graphs/appendices)	7/30/0/11/15/0/0		
Scientific field, <b>SF:</b>	Electrical Engineering		
Scientific discipline, <b>SD:</b>	Computer Engineering, Engineering of Computer Based Systems		
Subject/Key words, <b>S/KW:</b>	Internet of things, voice control, smart home automation		
<b>UC</b>			
Holding data, <b>HD:</b>	The Library of Faculty of Technical Sciences, Novi Sad, Serbia		
Note, <b>N:</b>			
Abstract, <b>AB:</b>	<p>This paper presents the concept of solution for implementation of voice control module within OBLO system for smart homes.</p> <p>For the implementation of voice functionality is used Amazon Alexa Voice Service. The given voice control module is integrated into OBLO cloud system, providing smart device control and monitoring.</p>		
Accepted by the Scientific Board on, <b>ASB:</b>			
Defended on, <b>DE:</b>			
Defended Board, <b>DB:</b>	President:	Bogdan Pavković, PhD	
	Member:	Jelena Kovačević, PhD	Mentor's sign
	Member, Mentor:	Ištvan Papp, PhD	

## **Zahvalnost**

Zahvaljujem se pre svega mentoru Pap Ištvanu i profesorki Mariji Antić na konstantnoj podršci, edukaciji i savetima.

Roditeljima i prijateljima koji su mi uvek pružali podršku u toku školovanja i kolegama iz tima na odličnom timskom radu i saradnji.



## SADRŽAJ

1.	Uvod .....	1
2.	Teorijske osnove.....	2
2.1	Amazon Aleksa (Amazon Alexa) .....	2
2.1.1	Amazon glasovni servis (Amazon Voice Service) i Amazon uređaji .....	2
2.1.2	Aleksa funkcionalnost (Alexa Skill) .....	4
2.1.3	Grafička sprega za podešavanje Aleksa funkcionalnosti (Alexa Skills Kit)..	5
2.1.3.1	Informacije za Aleksa funkcionalnosti (Skill Information) .....	5
2.1.3.2	Komunikacioni model (Interaction Model) .....	6
2.1.3.3	Konfiguracija i SSL sertifikat (Configuration & SSL Certificate) .....	8
2.1.3.4	Testiranje.....	10
2.1.3.5	Informacije za objavljivanje i politika privatnosti (Publishing Information & Privacy and Compliance) .....	10
2.2	OBLO sistem .....	11
2.3	MQTT protokol.....	12
3.	Koncept rešenja .....	14
4.	Programsko rešenje.....	17
4.1	Implementacija servisa za glasovnu kontrolu .....	17
4.1.1	Format ulazne JSON poruke .....	17
4.1.2	Format izlazne JSON poruke.....	20
4.1.3	Verifikacija pošiljaoca.....	22
4.1.4	Autentikacija.....	22
4.1.5	Mapiranje ulaznog teksta na komandu .....	23
4.1.5.1	Komande za kontrolu pojedinačnih uređaja.....	23
4.1.5.2	Komande za kontrolu zona.....	25

4.2 Implementacija komunikacionog modela .....	25
4.2.1 Obrasci za pojedinačne uređaje .....	25
4.2.2 Obrasci za zone.....	26
5. Rezultati.....	27
6. Zaključak .....	29
7. Literatura .....	30

## SPISAK SLIKA

Slika 2.1 Aleksa uređaji .....	3
Slika 2.2 Funkcionalnost Aleksa servisa .....	3
Slika 2.3 Grafička sprega za podešavanje - informacija.....	5
Slika 2.4 Primer intent šeme .....	6
Slika 2.5 Grafička sprega za podešavanje - konfiguracija.....	8
Slika 2.6 Konfiguracija SSL sertifikata .....	9
Slika 2.7 Simulator za testiranje .....	10
Slika 2.8 Šema OBLO sistema .....	11
Slika 3.1 Predlog integracije glasovne funkcionalnosti.....	14
Slika 4.1 Ulagana Aleksa poruka.....	18
Slika 4.2 Primer intent objekta .....	20
Slika 4.3 Izlazna Aleksa poruka .....	21
Slika 4.4 Deo programskog koda za autentikaciju.....	22
Slika 4.5 Primer filtriranih podataka i odabrani uređaj .....	23
Slika 4.6 MQTT Poruka za kontrolu uređaja.....	24

## SPISAK TABELA

Tabela 2.1 Primeri enumeracija .....	7
Tabela 2.2 Podešavanje fraza za paljenje i gašenje uređaja.....	7
Tabela 2.3 Podešavanje fraza za promenu boje pametne sijalice .....	8
Tabela 4.1 Detalji „session“ objekta.....	19
Tabela 4.2 Detalji „request“ objekta.....	19
Tabela 4.3 Detalji „intent“ objekta .....	20
Tabela 4.4 Detalji izlazne poruke .....	21
Tabela 4.5 Mapiranje komandi na servise .....	24
Tabela 4.6 Primer fraza za kontrolu pojedinačnim uređajima.....	25
Tabela 4.7 Primer fraza za kontrolu zona .....	26
Tabela 5.1 Rezultati testiranja .....	27

## **SKRAĆENICE**

**API** - Application Programming Interface

**HTTP** - Hypertext Transfer Protocol

**HTTPS** - Hypertext Transfer Protocol Secure

**JSON** - JavaScript Object Notation

**MQTT** - Message Queue Telemetry Transport

**TCP/IP** - Transmission Control Protocol/Internet Protocol

**SSL** - Secure Sockets Layer

**URL** - Uniform Resource Locator

**QoS** - Quality Of Service

## 1. Uvod

Cilj ovog rada je predstavljanje jednog rešenja za implementaciju glasovne kontrole i nadgledanja u pametnim kućama. Upravljanje glasom već uveliko pronalazi primenu u mobilnim uređajima, a sve više se osmišljavaju uređaji isključivo namenski za glasovnu kontrolu.

Dato rešenje je integrisano sa OBLO [1] sistemom za kućnu automatizaciju. S obzirom da se postojećim OBLO sistemom upravlja korišćenjem klijentske mobilne ili internet aplikacije orijentisane na pretraživač, za pristup kontroli je potreban konstantan pristup mobilnom uređaju ili računaru. Uvođenje glasovne kontrole u sistem kao što je OBLO, pruža alternativu za postojeću mobilnu aplikaciju u kućnom okruženju, i čini upravljanje kućnim sistemom efikasnijim.

Korišćeni servis za glasovno prepoznavanje je Amazon Aleksa servis [2]. Prednosti ovog servisa u odnosu na druge, sa strane programera, jesu jednostavnija implementacija i veća sloboda u kreiranju glasovnih funkcionalnosti. Korisniku je omogućena intuitivna interakcija kao i mnoštvo već postojećih funkcionalnosti, informativnog, zabavnog karaktera kao i pojedine funkcionalnosti za pametne kuće, koje su besplatne za korišćenje. Ideja je da se Aleksa funkcionalnost integriše u OBLO oblak sistemu kao zaseban servis koji bi korisnici proizvoljno aktivirali. Amazon je proizveo posebne uređaje, koji su integrirani sa Aleksa servisom, a to su su Echo i Echo Dot uređaji, koji se lako instaliraju u kućno okruženje i omogućavaju jednostavno upravljanje kućnim sistemom.

## 2. Teorijske osnove

Teorijske osnove opisuju detalje potrebne za razumevanje arhitekture i funkcionalnosti OBLO sistema za kućnu automatizaciju, Amazonovog servisa za prepoznavanje govora i potrebnih protokola za komunikaciju i upravljanje.

### 2.1 Amazon Aleksa (Amazon Alexa)

Amazon je kompanija, osnovana 1994. godine koja prvo bitno pruža usluge internet trgovine, a danas je proširena na pružanje mnogobrojnih usluga od kojih su najpopularnije usluge platforme oblaka, pre svega za masovno skladištenje i obradu podataka. 2014. godine, Amazon prvi put izbacuje na tržište Aleksa uređaj za glasovnu kontrolu. U ovom poglavljju su opisane funkcionalnosti Amazon Aleksa uređaja, a potom i Amazonovog servisa za prepoznavanje govora koji je integriran sa ovim uređajem.

#### 2.1.1 Amazon glasovni servis (Amazon Voice Service) i Amazon uređaji

Alexa je uređaj cilindričnog oblika, povezan na internet koji sadrži ugrađeni *bluetooth* zvučnik i mikrofonski niz. Ovaj uređaj je povezan sa Amazonovim servisom za prepoznavanje govora koji je inteligentni personalni asistent koji omogućava interaktivnu komunikaciju u realnom vremenu. Interakcija započinje aktiviranjem uređaja na određene ključne reči, od kojih je najpoznatija „Alexa“, a postoje i druge, kao što su „Amazon“, „Echo“ i „Computer“. Po invokaciji, Alexa asistent je spremna za izdavanje komande i korisnik može da započne sa glasovnom sesijom.

Alexa omogućava saznavanje širokog spektra svakodnevnih informacija, počevši od vesti, vremenske prognoze, internet pretraživanja, a i zabavnog je karaktera pa omogućava i

slušanje muzike, audio knjiga, igranje igrica. Broj njenih mogućnosti se svakodnevno povećava.

Danas postoji više verzija ovog uređaja koji sa svakog generacijom uvode nove karakteristike i unapređenja. Na slici 2.1 su prikazani Amazon uređaji koji se integrišu sa Amazon Aleksa servisom, svaki sa boljim i poboljšanim karakteristikama.



Slika 2.1 Aleksa uređaji



Slika 2.2 Funkcionalnost Aleksa servisa

Na slici 2.2 su prikazani koraci u toku jedne glasovne sesije.

- Prvi korak je izdavanje komande Aleksa uređaju. Uloga Aleksa uređaja je da prihvati glasovnu komandu i prosledi je u vidu audio datoteke na obradu Amazon Aleksa servisu.
- Amazon Aleksa servis vrši konverziju iz glasovnog oblika u tekstualni uz pomoć predefinisanog komunikacionog modela koji se pominje u poglavljju 2.1.3.2 , a potom mapira tekstualnu komandu na određeni zahtev.

- Po izvršavanju zahteva, npr. preuzimanje određenih podataka sa interneta, Aleksa servis konvertuje tekstualni odgovor u audio formu i prosleđuju ga Amazon uređaju.
- Krajnji rezultat je glasovni odgovor korisniku.

Svaki korisnik koji poseduje Aleksa uređaj, potrebno je da ima i Amazon korisnički nalog. Amazon nalog omogućava pristup Aleksa internet ili mobilnoj aplikaciji preko kojih se konfiguriše uređaj, što podrazumeva povezivanje na bežičnu mrežu, registraciju uređaja na nalog kao i moguć uvid u istorijat zadatih komandi. Pored toga aplikacija pruža mogućnost da se instaliraju različite Aleksa funkcionalnosti. Te funkcionalnosti se nazivaju “*Alexa Skills*” i njihov detaljniji prikaz sledi u poglavljju 2.1.2.

### **2.1.2 Aleksa funkcionalnost (Alexa Skill)**

Broj mogućnosti Aleksa servisa raste svakodnevno. Zahvaljujući Amazonovojoj opciji da pojedinac ili organizacija, poštujući odredbe i standarde, implementira Aleksa glasovnu funkcionalnost koja može postati globalno dostupna. Sve Aleksa glasovne aplikacije su besplatne za instalaciju i korišćenje.

Postoje više tipova glasovnih funkcionalnosti:

#### **1. Custom Skills**

Ovaj tip aplikacije ima najveću kontrolu nad modelom za komunikaciju. Zbog toga je i najkompleksniji za implementaciju, sa obzirom da je potrebno detaljno definisati komunikacioni model. Detalji komunikacionog modela su objašnjeni u poglavljju 2.1.3.2.

#### **2. Smart Home Skills**

Aplikacija za pametne kuće omogućava definisanje modela za kontrolu uređajima poput sijalica i termostata. Ovaj tip aplikacije nije fleksibilan kao prethodni zato što se ne definiše model za komunikaciju, već se koriste predviđene fraze za upravljanje ograničenim skupom uređaja. Takođe se ne koristi ime za invokaciju date funkcionalnosti, kao u pretodnom slučaju. Programer definiše *adapter*, koji pretstavlja deo koda koji se izvršava po okidanju određene komande.

#### **3. Flash Briefing Skills**

Ovaj tip aplikacije pruža kratak pregled različitog sadržaja, kao što su na primer vesti. Moguć je prikaz sadržaja koristeći tekstualni ili u audio format.

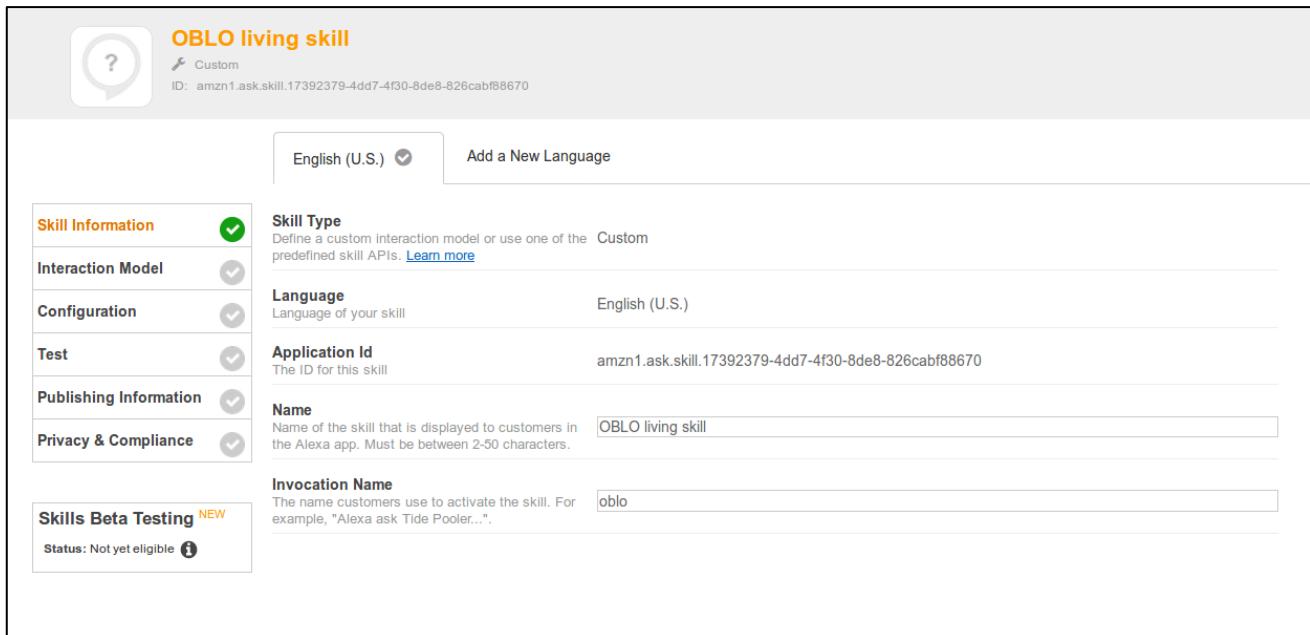
#### **4. Video Skills**

Omogućava upravljanje video uređajima i pregled video sadržaja.

## 2.1.3 Grafička sprega za podešavanje Aleksa funkcionalnosti (Alexa Skills Kit)

Grafička sprega za podešavanje [3] definiše pravila za glasovnu komunikaciju korisnika sa Aleksa funkcionalnošću. Sledi opisi koraka za konfiguraciju jedne Aleksa funkcionalnosti.

### 2.1.3.1 Informacije za Aleksa funkcionalnosti (Skill Information)



Slika 2.3 Grafička sprega za podešavanje - informacija

Ove informacije predstavljaju prvi korak u konfiguraciji Aleksa funkcionalnosti. Prikazan je na slici 2.3.

Informacije se sastoje iz sledećih konfiguracionih polja:

- Tip aplikacije (*Skill Type*)
- Jezik (*Language*)

Polje koje definiše jezik u kojem će biti definisana Aleksa funkcionalnost.

Trenutno su podržani engleski i nemački.

- Identifikacioni kod aplikacije (*Application Id*)
- Ime (*Name*)

Predstavlja ime aplikacije koje će biti prikazano unutar Aleksa korisničkog interfejsa i koje služi za lako da filtriranje i pronalaženje aplikacije.

- Ime za invokaciju (*Invocation Name*)

Služi za aktiviranje funkcionalnosti prilikom inicijalizacije glasovne sesije. Ono ukazuje sa kojom aplikacijom se započinje konverzacija.

Primer - Alexa, ask *Oblo* to turn on lamp. Gde je *Oblo* invokaciona reč.

### 2.1.3.2 Komunikacioni model (Interaction Model)

Kroz jedan primer je objašnjeno konfiguriranje glasovnog komunikacionog modela.

- Šema za opis namere korisnika (Intent Schema)

Intent predstavlja objekat u JSON formatu koji definiše polja koja korisnički zahtev može da sadrži, kao i njihove moguće vrednosti. Sledi primer jednog intenta.

```
{
  "intents": [
    { "slots": [
      {
        "name": "Devices",
        "type": "LIST_OF_DEVICES"
      },
      {
        "name": "LightRequests",
        "type": "LIST_OF_LIGHT_REQUESTS"
      },
      {
        "name": "Colors",
        "type": "LIST_OF_COLORS"
      }
    ],
      "intent": "LightIntent"
    }
  ]
}
```

Slika 2.4 Primer intent šeme

Naziv *intent-a* iz primera je „LightIntent“.

Sadrži tri polja :

*Device* - ime uređaja

*LightRequest* - ime zahteva

*Color* – vrednost RGB boje svetla

Svako od ovih polja ima svoj tip (LIST\_OF\_DEVICES, LIST\_OF\_COLORS, itd). Tip polja definiše se listom mogućih vrednosti polja, te polja praktično predstavljaju enumeracije. Enumeracije mogu biti proizvoljene, a moguće je koristiti enumeracije predefinisane od strane Amazon Alekse (kao što su: lista brojeva, datuma, država, objekata itd.).

- Proizvoljne enumeracije (Custom Slot Types)

Radi lakšeg formiranja modela definišu se tipovi koji predstavljaju enumeracije tj. liste reči grupisane u različite kategorije. U ovom primeru definišu se enumeracije LIST\_OF\_DEVICES, LIST\_OF\_LIGHT\_REQUESTS i

**LIST\_OF\_COLORS**. Potrebno je definisati što veći broj primera kako bi glasovno prepoznavanje bilo što tačnije.

<b>LIST_OF_DEVICES</b>	<b>LIST_OF_LIGHT_REQUESTS</b>	<b>LIST_OF_COLORS</b>
Lamp	Turn on	Blue
Light	Turn off	Green
Bulb	Change	Red
Dimmer	Set	Yellow

Tabela 2.1 Primeri enumeracija

- **Model glasovnih komandi (Sample Utterances)**

Predstavlja moguće fraze koje korisnik koristi u toku glasovne interakcije. Za svaki *intent* je potrebno omogućiti određeni broj fraza, tako da svaka od fraza trigeruje određeni intent. Pošto je navođenje velikog broja primera ovakvih fraza veoma kompleksno, enumeracije olakšavaju formiranje velikog broja kombinacija. Navođenjem samo jednog primera koji sadrži različite enumeracije, olakšano je pokrivanje velikog broja slučajeva. Enumeracije se uvek navode u vitičastim zagradama, a običan tekst bez.

LightIntent {LightRequests} {Devices}	Turn on lamp.
	Turn on light.
	Turn on bulb.
	Turn off lamp.

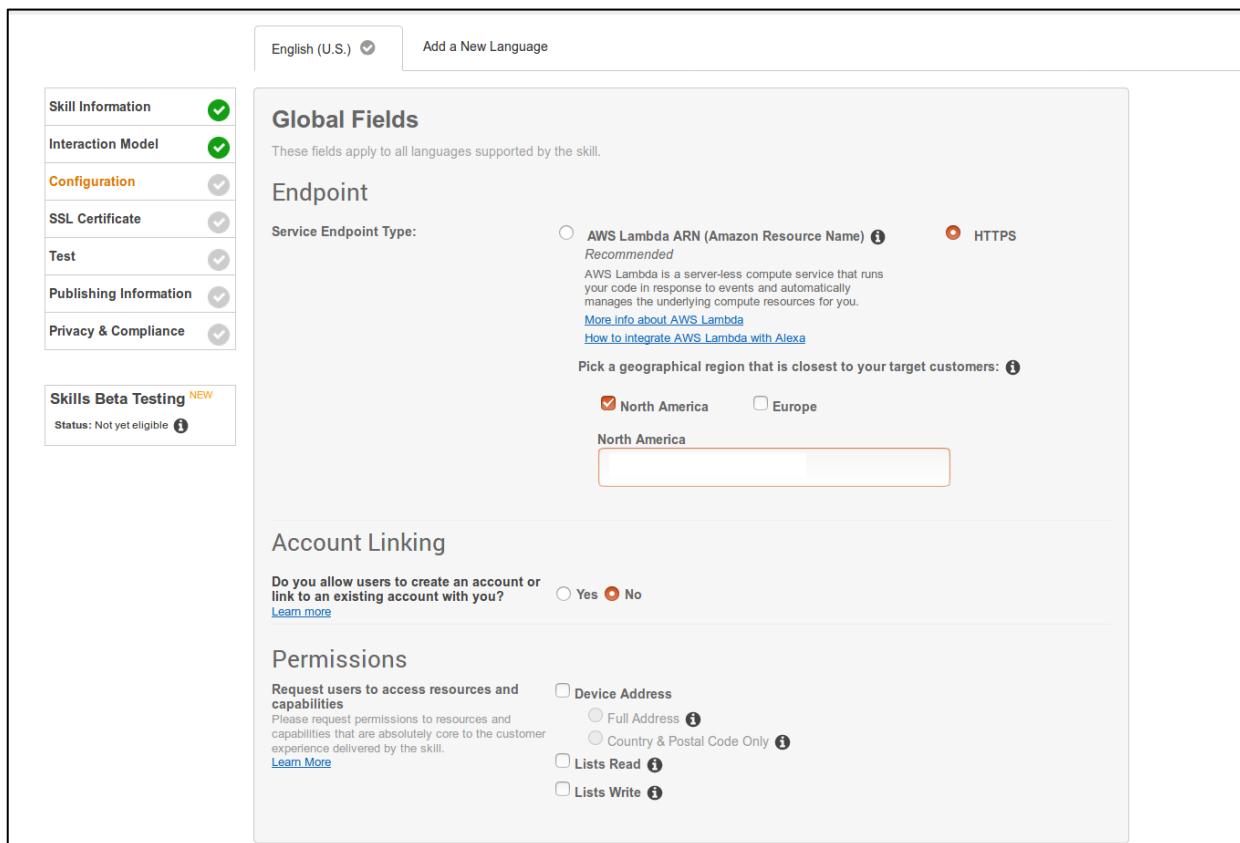
Tabela 2.2 Podešavanje fraza za paljenje i gašenje uređaja

	Change light color to blue.
LightIntent {LightRequests} {Things} color to {Colors}	Change lamp color to green.
	Change lamp color to red.
	Change dimmer color to red.

Tabela 2.3 Podešavanje fraza za promenu boje pametne sijalice

### 2.1.3.3 Konfiguracija i SSL sertifikat (Configuration & SSL Certificate)

U ovom poglavlju je prikazana sledeća tačka u konfiguraciji koja podrazumejava podešavanje polja u odeljku Configuration sa slike 2.5.



Slika 2.5 Grafička sprega za podešavanje - konfiguracija

- Servis (Service Endpoint Type)

Servis je krajnja tačka koja definiše programsku logiku Aleksa funkcionalnosti. Dato polje za konfiguraciju predstavlja lokaciju servisa sa kojom Aleksa funkcionalnost komunicira. Servis može da postoji na udaljenoj tački i tada se

definiše komunikacija preko HTTPS protokola. Potrebno je da data lokacija servisa bude javna na internetu.

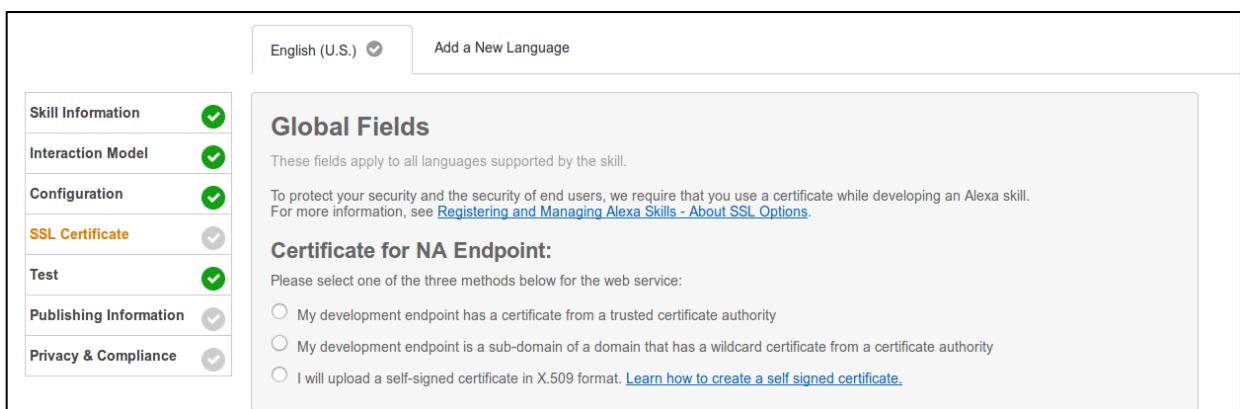
Drugi izbor za skladištenje servisa je AWS Lambda. To je Amazon servis koji omogućava izvršavanje proizvoljnog programskog koda. U slučaju AWS Lambda servisa moguće je koristiti samo određene programske jezike, dok za servis na udaljenoj tački ne postoji ovakvo ograničenje. AWS Lambda podržava Node.js (JavaScript), Python, Java (Java 8) i C# (.NET Core) programske jezike.

- Linkovanje korisničkog naloga (Account Linking)

Neke aplikacije zahtevaju mogućnost da povežu identitet korisnika sa njegovim identitetom u nekom drugom sistemu. Ovo je poznato kao povezivanje korisničkog naloga i krajnji cilj je da se kreira veza između Aleksa korisnika i njegovog naloga u drugom sistemu. Ukoliko se Aleksa funkcionalnost implementira koristeći *Smart Home Skill API*, povezivanje korisničkih naloga je obavezno. U slučaju *Custom Skill API*, povezivanje naloga može ali ne mora da se koristi. Ipak, u slučaju integracije sa OBLO sistemom neophodno je povezivanje korisničkog naloga čiji detalji slede u poglavlju 3.

- Dozvole (Permissions)

Omogućava zahtev za pristup informacijama vezanim za korisnički Amazon nalog tako što aplikacija zatražuje korisnikovo odobrenje. Korisnik odobrava dozvolu tek kada aplikacija to zahteva od njega. Trenutno postoje tri tipa dozvola: potraživanje adrese vezane za određeni Aleksa uređaj, pisanje i čitanje iz lista (čitanje ili pisanje u korisnikovu *ToDo* i *Shopping* listu)

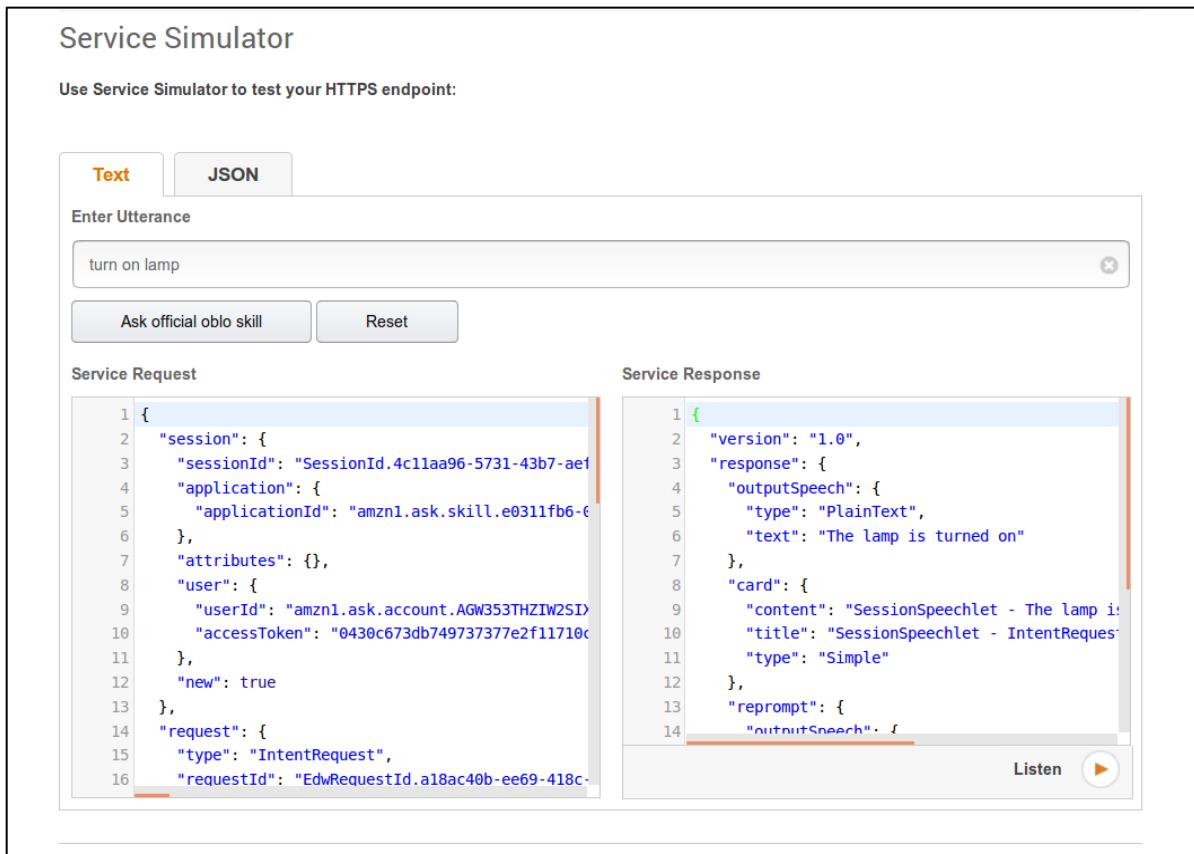


Slika 2.6 Konfiguracija SSL sertifikata

Podešavanje odgovarajućeg SSL sertifikata je prikazano na slici 2.6.

SSL sertifikat osigurava da će se komunikacija preko interneta biti zaštićena i poverljiva. Zbog toga Aleksa servis zahteva korišćenje HTTP konekcije preko SSL/TSL.

#### 2.1.3.4 Testiranje



Slika 2.7 Simulator za testiranje

Aleksa korisnička sprega za dizajn funkcionalnosti sadrži simulator za jednostavno testiranje zadavanjem komandi u tekstualnom obliku. Na slici 2.5 se može videti generisani izlaz iz servisa u JSON formatu, kao i povrati odgovor.

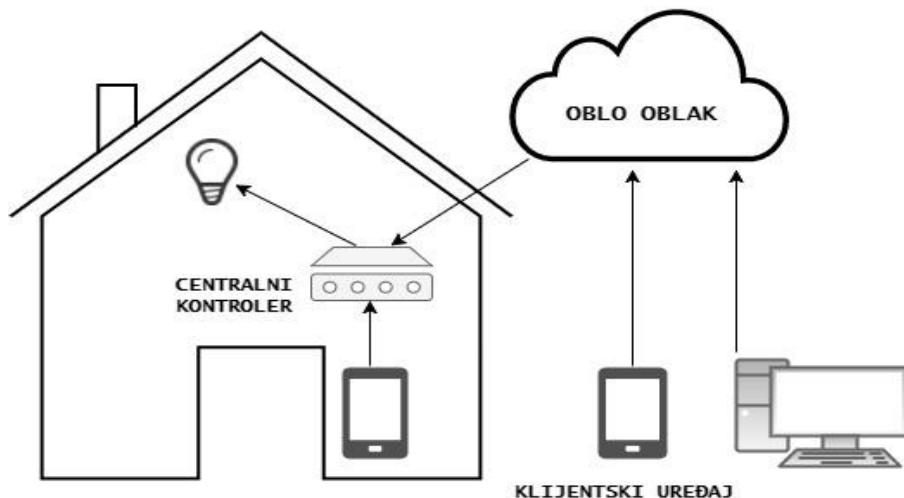
#### 2.1.3.5 Informacije za objavljivanje i politika privatnosti (Publishing Information & Privacy and Compliance)

Kada je Aleksa funkcionalnost u potpunosti implementirana i sve funkcije su zadovoljene, sledeći korak je objavljivanje date aplikacije. Tek posle objavljivanja, funkcionalnost će biti dostupna i drugim korisnicima.

Kako bi se aplikacija objavila potrebno je da prođe kroz par faza verifikacije. U delu za objavljivanje potrebno je omogućiti osnovne informacije o Aleksa funkcionalnosti kako bi se

krajnjem korisniku olakšalo korišćenje. To podrazumeva uputstvo za interakciju i sve mogućnosti date funkcionalnosti. U sledećem delu podešavanja, potrebno je omogućiti URL politike privatnosti i uslove korišćenja.

## 2.2 OBLO sistem



Slika 2.8 Šema OBLO sistema

OBLO je sistem za kućnu automatizaciju koji omogućava kontrolu i nadgledanje stambenih objekata koristeći pametne uređaje. Neki od pametnih uređaja koji su podržani u OBLO sistemu su pametna svetla, dimeri, pametne utičnice, senzori i kamere. Upravljački deo celog sistema je centralni kontroler (*gateway*), koji objedinjuje pametne uređaju u jedan inteligentan sistem.

Centralni kontroler prosleđuje pristigne komande ka uređajima koristeći ZigBee [5] i Z-Wave [4] komunikacione protokole pogodne za malu propusnost i malu potrošnju energije. OBLO sistem omogućava kontrolu sa bilo koje lokacije uz pomoć klijentske mobilne aplikacije podržane na iOS i Android operativnim sistemima. Klijentska aplikacija omogućava jednostavnu kontrolu i uvid u trenutna stanja svih uređaja, uz uslov da korisnik ima pristup internetu.

OBLO oblak rešenje predstavlja servis dostupan na internetu koji koordinira upravljanje pametnim sistemom preko centralnih uređaja krajnjih korisnika. Ovo rešenje omogućava korisnicima kućnu kontrolu izvan lokalne mreže. OBLO oblak omogućava korisnički interfejs

koja sadrži sve funkcionalnosti kao i mobilna aplikacija sa tim da ima i mogućnost za prikupljanje i analizu podataka, uvid u statistiku potrošnje i istorijat promene stanja uređaja.

Upravljanje OBLO uređajima je omogućeno preko servisa koji kontrolišu njihove funkcionalnosti. Svaki uređaj sadrži više različitih servisa koje je moguće koristiti ili čitajući ili pišući vrednost u njihova polja. Neki servisi služe samo za čitanje vrednosti, kao npr. "TemperatureService", koji omogućava čitanje trenutne temperature. Sa druge strane tu su i servisi koji omogućavaju pisanje vrednosti u njihova polja, kao što je "LightService" koji omogućava menjanje stanja sijalice (upaljena ili ugašena). Stanje servisa se može promeniti ili pročitati sa njega, slanjem posebnog formata poruke, korišćenjem MQTT [6] protokola čije kratko objašnjenje sledi u poglavlju 2.3.

## 2.3 MQTT protokol

MQTT [9] protokol se zasniva na principu pretplati - objavi, korišćenjem mrežnog protokola TCP/IP [8]. To je lagan i jednostavan protokol za implementaciju koji se koristi za komunikaciju između uređaja u sistemima sa malim kapacitetom memorije i ograničenim mogućnostima procesiranja.

Centralna deo ovakvog sistema je MQTT broker. On koordinira vezu između pošaljioца i primaoca poruka. Pošaljioč objavljuje poruku MQTT brokeru, koja sadrži telo poruke koja se naziva "*payload*", kao i naziv teme [7] na koju objavljuje. Na osnovu teme, broker zna kako i kom klijentu da rutira informaciju.

Svaki klijent koji želi da dobije određene informacije se pretplati na željenu temu. Kada npr. neki drugi klijent objavi poruku na tu temu, broker dostavlja datu poruku svim klijentima pretplaćenim na tu temu. U ovakovom sistemu klijent nikada ne mora da šalje upit da preuzme nove podatke, već broker prosleđuje informacije do njega ukoliko se desio novi događaj. Ukoliko MQTT klijent sluša na događaj on ima stalno otvorenu TCP vezu ka brokeru.

MQTT protokol definiše metode koje indikuju koju akciju klijent želi da započne. To sumetode:

- Connect – otvaranje konekcije ka brokeru
- Disconnect – zatvaranje konekcije ka brokeru
- Subscribe – pretplaćivanje na određenu temu
- Unsubscribe – prestanak pretplaćivanja na temu
- Publish – objavljuvanje na određenu temu

Teme se formiraju hijararhijski, krišćenjem “/” separatora i mogu da sadrže više nivoa. One služe brokeru za filtriranje poruka za svakog klijenta prisutnog u sistemu.

Primer teme:

/home/livingroom/bulb/status

Postoje i znakovi koji omogućavaju slanje poruka i preplatu na više različitih tema (*wildcards*).

- Znak „+“ predviđa zamenu za jedan nivo u hijararhiji.  
home/groundfloor/+/temperature  
validna zamena: home/groundfloor/kitchen/temperature
- Znak „#“ predstavlja zamenu za proizvoljan broj nivoa u hijararhiji. Uvek mora da se nalazi na poslednjoj poziciji u hijararhiji teme.  
home/groundfloor/#  
validna zamena: home/groundfloor/livingroom/temperature

Pošto MQTT koristi TCP protokol, oslanja se na garantovanu isporuku. Pored toga, MQTT uvodi kvalitet usluge (QoS) kojim broker ili klijent osigurava da je poruka uspešno primljena. Što je viši kvalitet usluge, razmena poruka je pouzdanija, ali uvodi veće kašnjenje i zahtevaju veću propustnost.

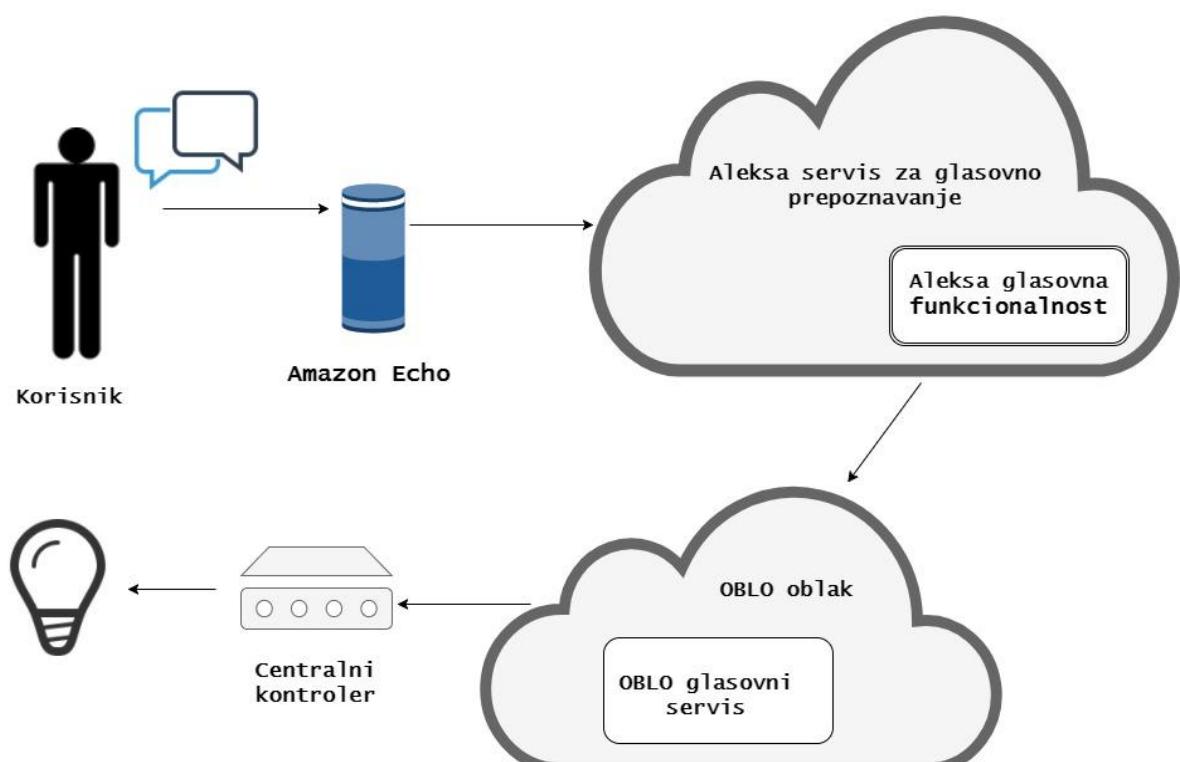
Tri nivoa kvaliteta usluge su:

- 0: broker ili klijent šalju poruku samo jednom, bez potvrde
- 1: broker ili klijent šalju poruku bar jednom, sa neophodnom potvrdom
- 2: broker ili klijent šalju poruku samo jednom, sa neophodnom potvrdom

### 3. Koncept rešenja

Najveći izazov u implementaciji sistema za glasovnu kontrolu je osobina ovakvih sistema da rade u realnom vremenu. Potrebno je da se na što efikasniji i što tačniji način obezbedi korisniku da izvrši željene komande. U ovom poglavlju je prestavljena problematika date teme rada, a potom i predlog rešenja.

Predlog integracije servisa za glasovnu kontrolu je kao jedan nezavisani modul unutar već postojećeg OBLO oblak sistema. Na slici 3.1 je predstavljen predlog integracije.



Slika 3.1 Predlog integracije glasovne funkcionalnosti

Koraci u komunikaciji sa slike 3.1 :

1. Korisnik izdaje naredbu uređaju.
2. Aleksa glasovni servis mapira komandu na Aleksa funkcionalnost.
3. Aleksa funkcionalnost trigeruje OBLO glasovni servis implementiran kao modul u OBLO sistemu za pametne kuće.
4. U OBLO glasovnom servisu se izvršava logika parsiranja ulaznog zahteva i potom mapiranje na odgovarajuću komandu.
5. Komanda se prosleđuje do OBLO centralnog kontrolera koristeći MQTT protokol.
6. Od OBLO centralnog kontrolera se komanda prosleđuje ka krajnjim nodovima (pametnim uređajima) koristeći ZigBee ili Z-Wave protokol.
7. Uređaj menja stanje ili se čita stanje sa uređaja.

Prvi korak u implementaciji je kreiranje Aleksa funkcionalnosti, koja definiše osnovne informacije za datu aplikaciju i model komunikacije sa korisnicima. U poglavlju 2.1.2 su izloženi detalji tipova Aleksa funkcionalnosti. U ovom radu se predlaže implementacija “*Custom Skill*” interfejsa sa obzirom da ima najveću fleksibilnost u implementaciji i mogućnost definisanja proizvoljnog i kompleksnijeg komunikacionog modela.

„*Home Skill API*“ ima ograničenje u izboru fraza za komunikaciju, koje se svodi na jednu frazu po komandi uređaja. Jedina prednost je neobavezno korišćenje invokacione reči, sa obzirom da je omogućeno direktno zadavanje komandi.

Glasovna funkcionalnost ne izvršava direktno kontrolu preko centralnog čvora, već se prvo bitno uspostavlja konekcija ka OBLO oblaku servisu koji je prelazna tačka ka centralnom kontroleru.

Kako bi glasovna aplikacija mogla da upravlja kućnim sistemom preko centralnog čvora, potrebno je da se predstavi OBLO oblaku kao dati korisnik. To je moguće korišćenjem korisnikovih podataka: jedinstvenog identifikatora korisnika i jedinstvenog identifikatora korisničkog centralnog kontrolera. Da bi glasovnoj aplikaciji bio omogućen pristup korisničkim podacima, zahteva se od korisnika da se poveže sa OBLO korisničkim nalogom prilikom aktiviranje glasovne funkcionalnosti u Aleksa aplikaciji. Tada OBLO oblak sistem dodeljuje Amazon klijentu određeni pristupni kod, koji će biti dostupan glasovnom servisu u svakom sledećem zahtevu poslatom od Amazon servisa.

Glasovni servis komunicira sa modulom za autentikaciju pre početka svake sesije, šaljući u zahtevu pristupni kod (*token*), a za uzvrat dobija potrebne korisničke podate za uspostavljanje MQTT konekcije za kontrolu uređajima.

Svaki put kada Alekса funkcionalnost pogađa rutu na kojoj se nalazi servis za glasovno upravljanje, u zahtevu se šalje JSON objekat koji pored informacija o zahtevu sadrži i detalje odabranog intenta, i parsirani glasovni ulaz. OBLO glasovni servis parsira taj objekat izdvajajući sve potrebne informacije za izvršavanje krajnje komande. Ukoliko bilo koja od neophodnih informacija fali, servis obustavlja obradu i vraća deskriptivan odgovor ukazujući korisniku na grešku koju je napravio. Ukoliko je zahtev validan prelazi se na sledeći zadatak a to je modul za kontrolu. Modul ispituje koji zahtev je u pitanju a potom započinje MQTT sesiju sa OBLO oblak sistemom. Kada je konekcija uspešno uspostavljena, klijent šalje odgovarajuću poruku ka OBLO oblak sistemu koji je potom prosleđuje ka centralnom kontroleru, koji konačno izvršava kontrolu krajnjih nodova.

Rešenje za glasovnu kontrolu se implementira koristeći JavaScript programski jezik iz razloga što je i OBLO oblak sistem implementiran na isti način. Na taj način je omogućena lakša integracija i jednostavnija komunikacija između postojećih modula.

Mogući slučajevi za korišćenje ovakvog glasovnog sistema:

- Kontrola pojedinačnih uređaja
  - paljenje i gašenje uređaja
  - podešavanje nivoa osvetljenja
  - čitanje temperature, vlažnosti vazduha, i različitih stanja sa senzora
- Kontrola grupnih uređaja
  - grupna kontrola po sobama
  - grupna kontrola po tipovima uređaja
- Kontrola scena i statusa
  - kontrola proizvoljnog broja uređaja različitog tipa po određenom scenariju

## 4. Programsко rešenje

### 4.1 Implementacija servisa za glasovnu kontrolu

Modul za glasovnu kontrolu je implementiran kao servis koji komunicira sa Amazon servisom koristeći HTTP protokol preko SSL/TSL. Poruke koja se prenose od i do Amazon glasovnog servisa su u JSON formatu. Kada glasovna sesija započne, Aleksa servis izvršava HTTP POST metodu na lokaciju servisa. Zahtev koji se šalje sadrži neophodne parametre za servis, kako bi na odgovarajući način bilo moguće izvršiti logiku sistema. Sledi prikaz i detaljno objašnjene polja ulazne i izlazne HTTP poruke, te detalji funkcionisanja sistema.

#### 4.1.1 Format ulazne JSON poruke

Format ulazne poruke se može videti na slici 4.1. U okviru ulazne poruke se razlikuju dva zasebna objekta „session“ i „request“. „Session“ objekat sadrži bitne informacije o korisniku, kao i detalje o Aleksa funkcionalnosti od koje je stigao zahtev. „Request“ objekat sadrži detalje odabranog intenta i vrednosti njegovih polja, koja su od značaja za funkcionalnost OBLO glasovnog servisa.

Sledi detaljan opis polja ulazne poruke.

```
{
  "session": {
    "sessionId": "SessionId.xxx-xxx-xxx",
    "application": {
      "applicationId": "amzn1.ask.skill.xxx-xxx-xxx"
    },
    "attributes": {},
    "user": {
      "userId": "amzn1.ask.account.xxx",
      "accessToken": "xxx"
    },
    "new": true
  },
  "request": {
    "type": "IntentRequest",
    "requestId": "EdwRequestId.xxx-xxx-xxx",
    "locale": "en-US",
    "timestamp": "YYY-MM-ddTHH:mm:ssZ",
    "intent": {
      "name": "LightIntent",
      "slots": {
        "LightRequests": {
          "name": "LightRequests",
          "value": "turn on"
        },
        "Device": {
          "name": "Device",
          "value": "lamp"
        }
      }
    },
    "version": "1.0"
  }
}
```

Slika 4.1 Ulagana Aleksa poruka

U tabeli 4.1 koja sledi su opisana polja „Session“ objekta.

Naziv polja	Objašnjenje
sessionId	Jedinstveni identifikator sesije, koji svaki put menja vrednost po završetku sesije. Može se koristiti ukoliko želimo da omogućimo interaktivnu konverzaciju između Alekse i korisnika.
applicationId	Jedinstveni identifikator aplikacije koji se dodeljuje prvi put pri kreiranju glasovne aplikacije.

attributes	Omogućava skladištenje tekstualnih podataka koji će istrajati tokom jedne sesije. Ti podaci se prosleđuju pri svakom sledećem zahtevu.
userId	Jedinstveni identifikator za korisnika koji se dodeljuje prilikom instalacije aplikacije.
accessToken	Pristupni kod koji identificuje korisnika u drugom sistemu. Šalje se u zahtevu jedino ako ako je korisnik uspešno linkovan na svoj korisnički nalog.
new	Vrednost koja ukazuje na početak sesije. Može imati vrednost „true“ ili „false“.

Tabela 4.1 Detalji „session“ objekta

Informacije sadržane u okviru „request“ objekta prikazane su u Tabeli 4.2.

Naziv polja	Objašnjenje
type	Postoje tri tipa : „LaunchRequest“, „IntentRequest“ i „SessionEndedRequest“
requestId	Jedinstveni identifikator zahteva.
locale	Ukazuje na jezik na kome je podešen Aleksa uređaj.
timestamp	Datum i vreme slanja zahteva.
intent	Šalje se samo ukoliko je u pitanju „IntentRequest“. Sadrži korisnikov zahtev u tekstualnoj formi.

Tabela 4.2 Detalji „request“ objekta

U tabeli 4.2 je prikazano da polje „type“ može da ima tri vrednosti. Vrednost tog polja zavisi od korisnikovog zahteva.

LaunchRequest označava da je korisnik prozvao Alexa funkcionalnost bez naznačavanja određenog zahteva. Korisnik može da startuje funkcionalnost na ovaj način koristeći ključnu reč „open“ ili neke od drugih invokacionih reči kao „ask“, „launch“, „start“, zajedno sa invokacionim imenom.

**Primer:** „Alexa, open Oblo skill.“ Odgovor koji korisnik dobija treba da bude kratak opis funkcionalnosti i pravila za korišćenje.

IntentRequest indikuje da je korisnik pored pozivanja imena funkcionalnosti naveo i određenu komandu ili upit. Tada se u zahtevu nalazi i „intent“ polje koje odgovara korisnikovom zahtevu.

**Primer:** „Alexa, ask Oblo to turn on lamp.“

SessionEndedRequest označava kraj sesije, i moguć je korišćenjem ključne reči „exit“.

```

"intent": {
    "name": "LightIntent",
    "slots": {
        "LightRequests": {
            "name": "LightRequests",
            "value": "turn on"
        },
        "Device": {
            "name": "Device",
            "value": "lamp"
        }
    }
}

```

Slika 4.2 Primer intent objekta

Naziv polja	Objašnjenje
intent.name	Ime odabranog intenta definisano u intent šemi pri definisanju komunikacionog modela.
name	Naziv polja definisan u intentu. Uvek se nalazi u intent objektu čak i ako nema vrednost.
value	Vrednost polja. Ukoliko nije definisano u zahtevu, ovo polje ne postoji.

Tabela 4.3 Detalji „intent“ objekta

#### 4.1.2 Format izlazne JSON poruke

Format izlazne poruke se može videti na slici 4.3. Izlazna poruka sadrži objekat „response“ koji definiše informacije o glasovnoj poruci koja se prikazuje korisniku u slučaju da je zahtev uspešno obrađen, ili ukoliko se desila greška. Moguće je konfigurisati parametar

koji definiše da li se sesija zavšava posle svakog primljenog zahteva ili nastavlja do eksplicitnog zatvaranja sesije od strane korisnika. Sledi detaljni opis polja izlazne poruke.

```
{
  "version": "1.0",
  "sessionAttributes": {},
  "response": {
    "outputSpeech": {
      "type": "PlainText",
      "text": "Your device command is executed."
    },
    "card": {
      "type": "Simple",
      "title": "Device Command",
      "content": "Welcome to smart home skill."
    },
    "reprompt": {
      "outputSpeech": {
        "type": "PlainText",
        "text": "I didn't quite understand your command. Can you repeat again?"
      }
    },
    "shouldEndSession": false
  }
}
```

Slika 4.3 Izlazna Aleksa poruka

Aleksa servis očekuje specifičan format odgovora. Odgovor predstavlja glasovno obaveštenje korisniku da se određena radnja izvršila ili da je došlo do greške. Takođe pruža i dodatne informacije za kontrolu sesije kao i izgled kartice koja se prikazuje u istoriji komandi Amazon aplikacije.

Naziv polja	Objašnjenje
sessionAttributes	Omogućava skladištenje tekstualnih podataka koji će istrajati tokom jedne sesije. Podaci se prosleđuju pri svakom sledećem zahtevu dok traje jedna sesija.
response.outputSpeech	Tekst koji će biti izgovoren krajnjem korisniku.
card	Detalji o kartici koja će biti prikazana u istoriji izgovorenih komandi u web ili mobilnoj Aleksa aplikaciji.
reprompt	Tekst koji je izgovoren korisniku ukoliko je sesija otvorena a servis ne uspe da mapira ni jednu korisnikovu komandu na odgovarajuću.
shouldEndSession	Označava da li sesija treba da se zatvori posle izgovaranja datog odgovora. Ima vrednosti „true“ ili „false“.

Tabela 4.4 Detalji izlazne poruke

### 4.1.3 Verifikacija pošiljaoca

Kako bi se osiguralo da ni jedan drugi servis ne šalje zahteve ka servisu za glasovnu kontrolu sem Aleksa servisa, potrebna je da se izvrši verifikacija i validnost ulaznih podataka. Validacija ima par bitnih aspekata koji je potrebno ispuniti.

Provera da li je URL koji je specificiran u zaglavljtu zahteva ima isti format kao što je i specificirano od strane Amazona. Potrebno je proveriti vremenski pečat i odbaciti zahteve koji prelaze starost od 150 sekundi.

U svakom zahtevu se šalje identifikacioni kod Aleksa funkcionalnosti koji je potrebno proveriti da bi se osiguralo da zahtev zaista dolazi od željene funkcionalnosti.

### 4.1.4 Autentikacija

Kako bi bilo omogućeno preuzimanje korisničkih podataka od strane OBLO oblak sistema, potrebno je poslati zahtev koji sadrži pristupni kod. Modul za autentikaciju proverava da li je pritsupni kod korisnika ispravan i potom modulu za glasovnu kontrolu prosleđuje potrebne podatke (identifikator korisnika i identifikator centralnog kontrolera). Modul koristi date kredencijale za formiranje MQTT tema koje su potrebne za kontrolu uređaja.

Kod na slici 4.4 predstavlja pribavljanje korisnikovih kredencijala i formiranje MQTT tema.

```
ClientAuth.prototype.getUserInformations = function(callback) {
    var credentials = {};

    credentials.aToken = this._aToken;
    credentials.cliId = 'c-amazon-' + this._aToken;

    request.get(options.SERVER, {
        'auth': {
            'bearer': this._aToken
        }
    },
    function(error, response, body) {
        if (!error && response.statusCode === 200) {
            var temp = JSON.parse(body);
            console.log(temp);
            if (temp.gtwSerial && temp.userId) {
                credentials.subTopic = 'oblo/' + temp.userId + '/cli/123/rsp';
                credentials.pubTopic = 'oblo/' + temp.userId + '/gtw/' + temp.gtwSerial + '/ohm/req';
                callback(false, credentials);
            } else {
                callback(true);
            }
        } else {
            callback(true);
        }
    });
};

};
```

Slika 4.4 Deo programskog koda za autentikaciju

#### 4.1.5 Mapiranje ulaznog teksta na komandu

Prvi korak u implementaciji servisa je parsiranje i filtriranje ulaznih podataka. Ulazni podaci sadrže dovoljno informacija da se doneše odluka koja komanda se izvršava nad uređajima. Potrebno je definisati različitu logiku za svaki intent sa obzirom da svaki sadrži različitu strukturu. Drugi deo u fazi filtriranja je validacija ulaznih podataka. Potrebno je proveriti da li je šablon poruke koja je izdata validna, tj. da li je korisnikov zahtev formiran na korektan način.

##### 4.1.5.1 Komande za kontrolu pojedinačnih uređaja

Posle filtriranja ulaznih podataka, formira se objekat koji sadrži bitne informacije za trenutni intent. Ideja je da se intenti mapiraju u na funkcionalne servise uređaja. Na primer, ukoliko je Amazon servis mapira korisnički glasovni zahtev na „LightIntent“, tada će se koristiti „LightService“ servis itd.

Prvi korak u kontroli je preuzimanje liste aktivnih uređaja sa OBLO centralnog kontrolera datog korisnika. Iz liste dobijenih uređaja potrebno je preuzeti sledeće neophodne podatke:

- Naziv uređaja
- Jedinstveni identifikator uređaja
- Sve servise uređaja

Naziv uređaja koji korisnik koristi prilikom izdavanja glasovne komande treba da bude isti kao i naziv uređaja u OBLO sistemu.

```
----Chosen Light request-----
{"request":"turn on","color":null,"device":"lamp"}

-----Chosen device(s)-----
[ { dev_index: 5,
  dev_id: 46,
  dev_name: 'lamp',
  dev_class: 'LeedarsonRGB',
  state: 0,
  dev_services:
    [ 'BasicService',
      'DeviceInfoService',
      'DimService',
      'HueService',
      'LightColorTemperatureService',
      'LightService',
      'ObloMaintenanceService',
      'ZigBeeService' ] } ]
```

Slika 4.5 Primer filtriranih podataka i odabrani uređaj

Naziv uređaja koji se nalazi u parsiranim podacima se koristi za filtriranje liste preuzetih uređaja. Primer filtriranih podataka se nalazi na slici 4.5. Ukoliko je pronađen bar jedan uređaj sa istim imenom, sledeći korak je provera da li taj uređaj ima servis na koji se mapira komanda.

U tabeli 4.5 sledi primer mapiranja komanda na servise

Naziv komande	Servis uređaja
Turn on / Turn off	LightService / OutletService
Dim / Set	DimService
Get temperature	TemperatureService
Roll shutter	RollerShutterService

Tabela 4.5 Mapiranje komandi na servise

Ukoliko uređaj ima mapirani servis, moguće je izvršavanje korisničke komande. Prosleđuje se kontrolna MQTT poruka sa vrednošću polja servisa koji treba da se promeni ili čita.

```
formCommandPayload: function(id, uid, serviceName, propertyName, PropertyValue) {
    return {
        sender: "cli/123",
        uid: uid,
        name: "set_device_property_value",
        params: {
            property_value: PropertyValue,
            property_name: propertyName,
            device_id: id,
            service_name: serviceName
        },
        type: "command"
    };
}
```

Slika 4.6 MQTT Poruka za kontrolu uređaja

Primer funkcije koja formira MQTT poruku za kontrolu na slici 4.5.

- property\_value – vrednost koja se menja npr. „true“ ili „false“ za kontrolu paljenja i gašenja
- property\_name – ime polja koji se menja npr. „state“
- service\_name – naziv servisa
- id – identifikacioni broj uređaja
- uid – identifikacioni broj poruke

#### 4.1.5.2 Komande za kontrolu zona

Komanda za zone se razlikuje po potrebi da prvo bitno izvrši filtriranje uređaja po zonama. Zone mogu biti sobe, spratovi ili delovi unutar istih. U OBLO sistemu zone su definisane kao grupe u koje mogu da se razvrstavaju uređaji različitih tipova.

Kako bi postojao parametar filtriranja, korisnik je potrebno da definiše naziv zone u glasovnoj komandi.

Primer. „Turn on all lights in *kitchen*“.

Kako bi se dobavili uređaji koji se nalaze u određenoj zoni potrebno je poslati zahtev kao jednu predefinsanu MQTT poruku koja zatražuje listu svih zona i uređaja u njima.

Filtriraju se svi uređaji u odabranoj zoni u zavisnosti od mapiranog servisa. Kontrola se izvršava kao i kontrola pojedinačnih uređaja, samo sve uređaje u grupi, redom.

## 4.2 Implementacija komunikacionog modela

U poglavlju 2.1.3.2 su definisane osnove konfiguracije komunikacionog modela. U ovom poglavlju je dat prikaz moguće forme komunikacionog modela koji je implementiran za OBLO sistem.

#### 4.2.1 Obrasci za pojedinačne uređaje

Dat prikaz se odnosi na moguće obrasce za glasovnu kontrolu pojedinačnim uređajima. Primeri iz tabele 4.6 su formirani na osnovu postojećih funkcionalnosti uređaja u OBLO sistemu.

Obrazac za komande	Primer	Servis
[ZAHTEV] + [UREĐAJ]	Turn on/Turn off lamp	LightService, OutletService
[ZAHTEV] + [UREĐAJ] + “to” + [VREDNOST] + “percentage”	Set dimmer to 80 percentage.	DimService
[ZAHTEV] + [UREĐAJ] + “color to” + [BOJA]	Change light color to blue.	HueService
[ZAHTEV] + [SERVIS SENZORA] + “from” + [UREĐAJ]	Get temperature from window sensor.	TemperatureService, FloodService, HumidityService, itd.

Tabela 4.6 Primer fraza za kontrolu pojedinačnim uređajima

#### 4.2.2 Obrasci za zone

U tabeli 4.7 su predstavljeni neki od mogućih obrazaca za glasovnu kontrolu zonama.

Obrazac za komande	Primer
[ZAHTEV] + "all" + [UREĐAJI] + "in" + [ZONA]	Turn on all lights in ground floor.
[ZAHTEV] + "all" + [UREĐAJI] + "in" + [ZONA] + "to" + [VREDNOST] + "percentage"	Set all dimmers in living room to fifty percentage.
[ZAHTEV] + "all" + [UREĐAJ] + "in" + [ZONA] + "color to" + [BOJA]	Change all lamps in bedroom color to red.
[ZAHTEV] + [SERVIS SENZORA] + "in" + [ZONA]	Get temperature in kitchen.

Tabela 4.7 Primer fraza za kontrolu zona

## 5. Rezultati

Testiranje servisa za glasovnu kontrolu se vršilo nad odabranim testnim skupom koji će biti prikazan u ovom poglavlju. Testovi su podeljeni u dve grupe: test glasovnih komandi koje su testirane direktno u interakciji sa Amazon Echo Dot uređajem i testiranje procesiranja komande od strane glasovnog servisa.

U tabeli 5.1 su predstavljeni dati rezultati.

<b>Br.</b>	<b>Glasovna komanda</b>	<b>Tk</b>	<b>To</b>	<b>Uspešnost</b>
1.	Turn on lamp	1.782 s	1.559 s	95%
2.	Turn on all lamps in ground floor	2.709 s	1.973 s	87%
3.	Run intrusion in master bedroom scene	0.98 s	0.774 s	98%
4.	Get temperature from sensor	2.5 s	1.995 s	76%
5.	Set dimmer to fifty	1.82 s	1.552 s	82%

Tabela 5.1 Rezultati testiranja

**Tk** je vremenski odziv čije merenje počinje u trenutku zadavanja glasovne komande do trenutka dobijanja povratne poruke od Aleksa servisa. Kod rezultata prvog testiranja potrebno je uzeti u obzir grešku sa obzirom da su ručno generisani. Glasovno testiranje je sprovedeno na uzorku od troje ljudi koji su izdavali svaku komandu po deset ponavljanja. Za računanje srednjeg vremena su uzeti u obzir po deset validnih rezultata, dok su nevalidni rezultati uzeti za izračunavanje uspešnosti izvršavanja.

**To** je vreme obrade, dobijeno korišćenjem programske podrške ApacheBench [10] koja meri performanse glasovnog servisa. Vreme izvršavanja komandi takođe zavisi i od broja MQTT poruka koje su potrebne da se razmene sa OBLO sistemom. Kod kontrole paljenja ili gašenja jednog uređaja razmeni se samo jedna MQTT poruka za upravljanje, dok za grupnu kontrolu se razmenjuje broj poruka u zavisnosti od broja uređaja u grupi.

Uspešnost rezultata se odnosi na broj tačnih rezultata u odnosu na ukupan broj zadatih komandi. Rezultati mogu biti nevalidni zbog lošeg prepoznavanja Amazon glasovnog servisa, okolnih šumova, kao i internet konekcije.

## 6. Zaključak

Predstavljeno je jedno, od više mogućih rešenja za integraciju glasovne funkcionalnosti u postojeći OBLO sistem. Dati glasovni servis je integriran u OBLO oblak sistemu. Planirane funkcionalnosti su implementirane i testirane. Rešenje je modularno, i stoga omogućava proširenje sistema, kao i eventualno uključivanje ili isključivanje bez uticaja na druge module. Postoji mogućnost unapređenja aplikacije u vidu dodavanja novih funkcionalnosti kao i podrške za nove uređaje. Sistem je implementiran na dva jezika – engleskom i nemačkom. Cela logika za jezik je implementirana u samoj sprezi za podešavanje funkcionalnosti za Aleksu i stoga dodavanje novih jezika ne podrazumeva velike izmene u konkretnoj implementaciji glasovnog sevisa. Prednosti ovakvog sistema je prilično laka konfiguracija i integracija u kućno okruženje.

Jedna mana ovakvog sistema je zavisnost rada OBLO glasovnog servisa od Amazonovog servisa za prepoznavanje govora. Što znači da glasovno prepoznavanje zavisi isključivo od datog servisa, i ne postoji nikakav uticaj na poboljšavanje glasovnog prepoznavanja. Spoljašnji uticaji kao što su zvukovi, udaljenost korisnika od uređaja, kao i sam način izgovaranja i akcenat su neki od problema koji utiču na stohastičko prepoznavanje, s tim da se svakodnevno teži ka poboljšanju takvog sistema.

## 7. Literatura

- [1] Oblo Living <http://www.obloliving.com/tehnologija>
- [2] Amazon Developer <https://developer.amazon.com/alexa-voice-service>
- [3] Amazon <https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/docs>
- [4] Dr. Christian Paetz, “Z-Wave Basics: Remote Contreol in Smart Homes”, Jun, 2013
- [5] ZigBee alliance <http://www.zigbee.org>
- [6] HiveMQ <http://www.hivemq.com/blog/how-to-get-started-with-mqtt>
- [7] Mosquitto <https://mosquitto.org/man/mqtt-7.html>
- [8] I.Bašičević, M. Popović, V. Kovačević, Osnove računarskih mreža 1, FTN, 2013.
- [9] Milan Tucić, “Protokol za razmenu poruka u sistemima pametnih kuća”, Master rad,  
2016
- [10] Apache <https://httpd.apache.org/docs/trunk/programs/ab.html>