Зборник радова Факултета техничких наука, НовиСад

УДК: 4.4
ДОИ: https://doi.org/10.24867/34BE15Dragicevic

Имплементација система за Over-the-Air ажурирање IoT уређаја

Implementation of an Over-the-Air update system for IoT devices

Марко Драгићевић, *Факултет техничких наука, Нови Сад*

Студијски програм – СОФТВЕРСКО ИНЖЕЊЕРСТВО И ИНФОРМАЦИОНЕ ТЕХНОЛОГИЈЕ

Кратак садржај – *Рад се бави дизајном, архитектуром и имплементацијом система за сигурно и скалабилно Over-the-Air (OTA) ажурирање фирмвера, са фокусом на IoT уређаје базиране на ESP32 микроконтролеру. За реализацију серверског дела система користи се serverless приступ на Amazon Web Services (AWS) платформи, уз употребу сервиса као што су Lambda, S3, DynamoDB и IoT Core. Клијентска апликација на уређају имплементира робустан механизам за верификацију дигиталног потписа и сигуран процес инсталације. Администраторски кориснички интерфејс, који омогућава управљање верзијама и покретање процеса ажурирања, развијен је коришћењем React библиотеке.*
Кључне речи: *OTA, ESP32, AWS, Serverless, бежично ажурирање микроконтролера*

Abstract – *This thesis presents the design, architecture, and implementation of a comprehensive framework for secure and scalable Over-the-Air (OTA) firmware updates, focusing on IoT devices based on the ESP32 microcontroller. The server-side portion of the system leverages a serverless paradigm on the Amazon Web Services (AWS) platform, utilizing services such as Lambda, S3, DynamoDB, and IoT Core. The client-side application implements a robust mechanism for digital signature verification and a secure installation process. The administrative user interface, for managing versions and initiating the update process, is developed using the React framework.*

Keywords: *OTA, ESP32, AWS, Serverless, wireless updating microcontrollers*

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је био др Милан Видаковић, ред. проф.

1. УВОД

Експанзија Internet of Thing (IoT) донела је фундаментални изазов управљања софтвером на милионима географски дистрибуираних уређаја [1]. Једном распоређени, ови уређаји захтевају континуирано ажурирање ради додавања нових функционалности, побољшања перформанси, а пре свега, ради затварања критичних безбедносних пропуста. Без ефикасног механизма за даљинско ажурирање, IoT екосистеми постају рањиви, тешки за одржавање и брзо застаревају.

Рад се бави овим проблемом кроз дизајн и имплементацију комплетног система за сигурно и скалабилно Over-the-Air (OTA) [2,3] ажурирање фирмвера. Приступ је заснован на cloud-native принципима, где је целокупна серверска (backend) инфраструктура реализована на Amazon Web Services (AWS) платформи, пратећи serverless парадигму. Овакав приступ омогућава високу доступност и аутоматско скалирање, док истовремено минимизира оперативне трошкове.

Кључни фокус система је на вишеслојној безбедности. Решење имплементира сигурносни модел који укључује дигитално потписивање фирмвера на серверу и криптографску верификацију на самом уређају, чиме се гарантује аутентичност и интегритет софтвера. Рад такође истражује и демонстрира еволуцију комуникационог модела, од иницијалног приступа са периодичном провером (енгл. *polling*) до финалног, ефикаснијег решења које користи MQTT pub/sub протокол за нотификације у реалном времену. Циљ је да се прикаже целовит и практичан модел за развој робусног OTA система, применљив на реалне сценарије употребе, са ESP32 микроконтролером као циљном хардверском платформом.

2. АРХИТЕКТУРА СИСТЕМА

Развијени систем је пројектован као дистрибуирана, cloud-native апликација са јасно дефинисаним компонентама и одговорностима. Архитектура је осмишљена тако да задовољи кључне нефункционалне захтеве: скалабилност, сигурност, поузданост и ефикасност. У овом поглављу биће представљен дизајн сваке од кључних целина: backend инфраструктуре, модела података, комуникационог модела и софтверске архитектуре на самом уређају.

2.1. Cloud архитектура

За серверску инфраструктуру усвојена је serverless парадигма на Amazon Web Services (AWS) платформи. Овај избор елиминише потребу за традиционалним управљањем серверима, омогућавајући систему да

аутоматски скалира ресурсе у складу са тренутним оптерећењем, док се трошкови генеришу искључиво по потрошњи [4]. Архитектура обједињује неколико кључних AWS сервиса који раде у синергији како би пружили комплетну функционалност. Amazon API Gateway служи као сигурна улазна тачка за све HTTP захтеве; AWS Lambda функције садрже сву пословну логику; Amazon S3 се користи као издржљиво складиште за бинарне фајлове фирмвера; Amazon DynamoDB је NoSQL база података за метаподатке; док AWS IoT Core функционише као MQTT брокер за комуникацију у реалном времену.

2.2. Архитектура података

Моделирање података је кључан аспект система, директно утичући на његове перформансе и скалабилност. Уместо традиционалног релационог модела, за DynamoDB је примењен Single-Table Design. Овај напредни NoSQL приступ подразумева складиштење свих типова података у једној табели, користећи композитни примарни кључ (партициони кључ PK и сортирајући кључ SK) за ефикасно организовање и дохватање. Табела је дизајнирана да ефикасно одговори на кључне обрасце приступа (енгл. *access patterns*) и садржи два типа ставки: VERSION#... ставке, које су историјски записи о свакој верзији фирмвера, и GROUP#... ставке, које служе као „показивачи“ на тренутно активну верзију за одређену deployment групу. Ради оптимизације, GROUP ставке садрже денормализоване (дуплиране) податке, што омогућава да уређај добије све потребне информације у једном упиту, елиминишући потребу за операцијама налик на JOIN.

2.3. Архитектура комуникације

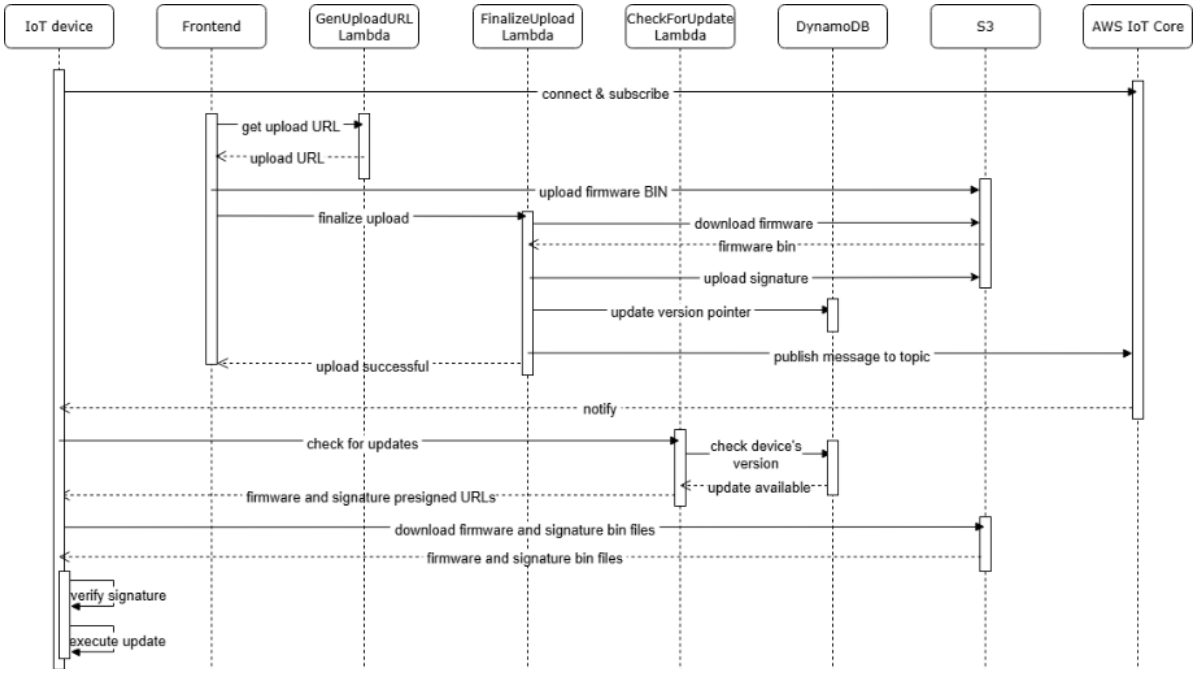
Архитектура комуникације је еволуирала током развоја. Иницијални polling модел [5], где уређај периодично шаље HTTP захтев серверу, показао је значајне недостатке: уносио је кашњење у детекцији ажурирања и генерисао велики број сувишних захтева. Због тога је финална архитектура усвојила MQTT pub/sub модел. Овај приступ, реализован преко AWS IoT Core сервиса, фундаментално мења динамику. Уређај успоставља једну дуготрајну, сигурну конекцију и претплаћује се на специфичну тему (енгл. *topic*). Backend систем објављује кратку нотификациону поруку само онда када је ново ажурирање заиста доступно. Ова архитектура вођена догађајима (event-driven architecture) је значајно ефикаснија и омогућава ажурирања у реалном времену.

2.4. Архитектура на уређају (Embedded)

Софтвер на ESP32 уређају је дизајниран са циљем да буде модуларан и робустан. Комплексност OTA процеса је енкапсулирана унутар самосталне Arduino библиотеке (SecureOtaUpdater), чиме се главна апликација поједностављује. Примењен је принцип инјекције зависности, где се кључни објекти (као WiFiClientSecure) креирају ван библиотеке и прослеђују јој, што повећава флексибилност. Архитектура се ослања на уграђени A/B партициони систем ESP32 меморије, који гарантује да уређај може да се опорави и врати на претходну верзију у случају неуспешног ажурирања.

3. ИМПЛЕМЕНТАЦИЈА СИСТЕМА

Ово поглавље детаљно описује техничку реализацију најважнијих делова система, са фокусом на имплементацију на серверском и клијентском делу.



Слика 1 - *Дијаграм секвенце*

На слици 1 је приказан обједињен дијаграм секвенце који илуструје комплетан животни циклус Over-the-Air (OTA) процеса, од иницијализације уређаја до финалне инсталације новог фирмвера.

3.1. Имплементација backend-а

Backend је реализован кроз комбинацију AWS CDK-а за дефинисање инфраструктуре и Python кода за пословну логику у Lambda функцијама. Коришћењем AWS CDK, целокупна инфраструктура је дефинисана у Python програмском језику. Ово омогућава аутоматизовано, поновљиво и верзионисано постављање свих cloud ресурса, укључујући S3 bucket, DynamoDB табелу, Lambda функције са прецизним IAM дозволама, и API Gateway ендпоинте. Овај приступ је у складу са модерним DevOps праксама и значајно смањује могућност људске грешке.

Процес објаве нове верзије фирмвера је оркестриран унутар FinalizeUpload Lambda функције. Кључни корак је дигитално потписивање. Да би се максимално заштитио приватни кључ, он се чува у AWS Secrets Manager сервису и дохвата се у тренутку извршавања. Lambda функција затим користи cryptography библиотеку за израчунавање ECDSA потписа [6], који уписује назад на S3. Након тога, ажурира DynamoDB табелу и објављује MQTT поруку на AWS IoT Core, чиме се процес завршава.

CheckForUpdate Lambda функција је оптимизована за брзину. Њен најважнији задатак је генерисање сигурних S3 Pre-signed URL-ова. Након што једним упитом ка DynamoDB-у утврди да је ажурирање потребно, она користи boto3 SDK да генерише временски ограничене URL-ове. Овај механизам омогућава уређају привремен приступ иначе потпуно приватним фајловима на S3, што је кључни сигурносни елемент.

3.2. Имплементација на ESP32 уређају

Имплементација на уређају је реализована у C++ језику кроз развијену Arduino библиотеку.

За сигурну конекцију са AWS IoT Core, уређај користи TLS протокол са аутентификацијом заснованом на X.509 сертификатима. WiFiClientSecure објекат се у тренутку повезивања конфигурише са Root CA сертификатом, сертификатом уређаја и приватним кључем уређаја. Овај процес гарантује да само ауторизовани уређаји могу да се повежу на систем и да је комуникација заштићена од прислушкивања. Имплементиран је и робустан механизам за аутоматско поновно повезивање у случају губитка везе.

Након пријема MQTT нотификације, уређај покреће процес ажурирања. Да би се уштедела RAM меморија, фирмвер се не преузима цео у меморију, већ се користи „streaming“ приступ. Преузима се у малим деловима (енгл. *chunks*), док се истовремено рачуна његов SHA-256 хеш и уписује у неактивну OTA партицију. Најважнији корак је верификација дигиталног потписа, која се врши помоћу mbedtls библиотеке. Уколико је потпис исправан, позива се Update.end(true) да би се ажурирање потврдило и променио показивач у otadata партицији; у супротном, позива се Update.end(false) и процес се сигурно прекида, остављајући уређај у претходном, функционалном стању.

3.3. Имплементација администраторског веб интерфејса

Да би се омогућило ефикасно и интуитивно управљање целокупним OTA системом, развијен је и администраторски веб интерфејс. Ова апликација представља централну контролну таблу и служи као „људски интерфејс“ за комплексну позадинску инфраструктуру, омогућавајући администратору да извршава све кључне операције без потребе за директним познавањем AWS сервиса или API-ја.

Frontend апликација је развијена као модерна Single-Page Application (SPA) коришћењем React библиотеке. Овај избор је направљен због React-ове архитектуре засноване на компонентама, која омогућава креирање комплексног корисничког интерфејса од малих, изолованих и поново употребљивих делова кода. Као SPA, апликација пружа брзо и флуидно корисничко искуство, где се све интеракције одвијају динамички, без поновног учитавања странице.

За хостовање и испоруку апликације примењена је стандардна, високо скалабилна пракса за статичке веб сајтове на AWS-у. Након процеса изградње (енгл. *build*), оптимизовани статички фајлови (HTML, CSS, JavaScript) се постављају у Amazon S3 bucket који је конфигурисан за хостовање статичких веб сајтова. Испред овог bucket-а постављен је Amazon CloudFront, глобална мрежа за испоруку садржаја (CDN). Оваква архитектура доноси две кључне предности: перформансе – кешира се садржај на локацијама широм света, што смањује време учитавања апликације за крајње кориснике, и безбедност – интеграцијом са AWS Certificate Manager (ACM), CloudFront омогућава да апликација буде доступна преко сигурне HTTPS везе, што је стандард за све модерне веб апликације.

Администраторски интерфејс омогућава извршавање свих неопходних радњи за управљање OTA процесом: преглед и управљање типовима уређаја, преглед верзија фирмвера, објављивање новог фирмвера.

4. ЗАКЉУЧАК

У раду је успешно пројектован и имплементиран комплетан систем за сигурно и скалабилно Over-the-Air (OTA) ажурирање фирмвера за IoT уређаје. Коришћењем serverless архитектуре на Amazon Web Services платформи, креирано је робусно решење које је ефикасно, скалабилно и лако за одржавање. Кључни допринос рада лежи у имплементацији вишеслојног сигурносног модела, са дигиталним потписивањем фирмвера (ECDSA) на серверу и криптографском верификацијом на ESP32 уређају, чиме је осигуран интегритет и аутентичност софтвера. Демонстрацијом преласка са традиционалног polling механизма на реактивни MQTT pub/sub модел, показана је супериорност архитектуре вођене догађајима у погледу ефикасности и брзине одзива. Развојем модуларне Arduino библиотеке и администраторског интерфејса, систем је заокружен као целовит производ

спреман за практичну примену. Могућности за даљи развој укључују увођење система за управљање корисницима (енгл. *multi-tenancy*) помоћу AWS Cognito сервиса, оптимизацију перформанси на уређају коришћењем двојезгарне обраде, као и имплементацију сигурносног „provisioning“ процеса за аутоматско додељивање јединствених идентитета уређајима.

5. ЛИТЕРАТУРА

[1] O. Mazhelis, E. Luoma и H. Warma, „Defining an internet-of-things ecosystem“, Conference on internet of things and smart spaces, 2012.

[2] S. Halder, A. Ghosal, M. Conti, „Secure over-the-air software updates in connected vehicles: A survey“, Computer Networks, 2020.

[3] M. Kubaščík, „OTA firmware updates on ESP32 based microcontrolers“, 2024 IEEE 17th International Scientific Conference on Informatics (Informatics), 2024

[4] I. Baldini, „Serverless computing: Current trends and open problems“, Research advances in cloud computing, 2017

[5] H. Takagi, „Queuing analysis of polling models“, ACM Computing Surveys (CSUR), 1988

[6] D. Toradmalle, „Prominence of ECDSA over RSA digital signature algorithm“, 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018

Кратка биографија:

Марко Драгићевић, рођен 10.02.2001. године у Лозници. Средње образовање стекао 2019. године, и исте године уписао Факултет техничких наука, смер Софтверско инжењерство и информационе технологије. Након завршених основних студија уписао је мастер студије на истом факултету, смер Софтверско инжењерство и информационе технологије. Положио је све испите предвиђене планом и програмом.
Контакт: mdragicevic58@gmail.com