Testiranje uporabniškega vmesnika naprav IoT

Tijana Krutil[1], **Andrej Kos**[1]

[1]*Laboratorij za telekomunikacije, Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška 25, Ljubljana*
E-pošta: tijana.krutil@gmail.com

IoT User Interface Testing

Abstract.

In this paper we first present the five-layer architecture of Internet of Things which consists of perception, network, processing, application, and business layer. Then we briefly describe the most important testing areas in IoT and explain why its testing is becoming significantly more and more important. We mainly focus on user interface testing and also describe it on a real-world example from the field of energy IoT and smart grid.

1 Uvod

V zadnjih dveh desetletjih je internet stvari (angl. Internet of Things - IoT) revolucionaliziral večino področij našega življenja. Gre za fuzijo resničnega in digitalnega sveta, ki nam omogoča komunikacijo ne le med ljudmi, pač pa tudi med samimi napravami in procesi. IoT je olajšal vsakdan posameznika s pametnimi napravami, ki prevzemajo opravila, ki jih je prej moral opraviti človek sam, na primer sesanje in košnja, omogoča tudi upravljanje doma in gospodinjskih aparatov prek aplikacij tudi na daljavo. Poleg vsakodnevne uporabe pa tehnologija IoT pomaga tudi bolj generalno in ponuja rešitve na družbeni ravni – pomaga pri optimizaciji procesov v bolnišnicah, proizvodnih linijah, kmetijstvu, turizmu in na drugih področjih. Na praktično vseh teh področjih prispeva k večji učinkovitosti procesov, zmanjša delovne obremenitve in hitro ter zanesljivo zbira, obdeluje in analizira podatke. IoT pa je tudi gonilo razvoja pametnih mest, katerih glavni cilji so med drugim povečanje učinkovitosti rabe električne energije in s tem minimizacija energijskih izgub, optimizacija prometa, zmanjšanje onesnaževanja, tudi varnost v mestu. [1][2]

S takšno rastjo IoT pa smo postavljeni pred velik izziv, in sicer testiranje naprav IoT. Zaradi razpršenosti in večplastne arhitekture je sistem IoT potencialno podvržen veliko napakam, zato je testiranje teh naprav zelo pomembno. Ker se IoT razširja na čedalje več področij, poleg tega pa upravlja z občutljivimi podatki, na pomembnosti pridobiva varnostni vidik testiranja naprav IoT. Ta pravzaprav postaja nujen in zahteva strožjo standardizacijo na tem področju. [4]

2 Arhitektura IoT

Internet stvari je orodje, ki napravam iz vsakodnevne rabe omogoča brezžično povezovanje v omrežje, zbiranje in medsebojno izmenjavo ter analizo podatkov, na podlagi teh pa sprejema odločitve. S tem zagotavlja kakovostnejše delovanje naprav in boljšo uporabniško izkušnjo.

Za naprave interneta stvari so najbolj značilni senzorji, ki iz okolja pridobivajo podatke in aktuatorji za interakcijo z okoljem. Osnova sistema (arhitekture) interneta stvari so podatki, ki se izmenjujejo med različnimi napravami.[3][5] Ker pa arhitektura IoT ni enotna, obstaja veliko različnih oblik arhitekture. V tem članku bomo opisali petslojno arhitekturo, ker je tudi v literaturi najpogosteje uporabljena.

2.1 Zaznavna plast

Zaznavna plast oziroma tudi plast senzorskih naprav je sestavljena iz fizičnih naprav in predstavlja najnižji sloj v petslojni arhitekturi IoT. Senzorji so lahko RFID (angl. Radio Frequency Identity), infrardeči senzorji, 2D črtne kode, QR (angl. Quick Response) kode, senzorji za zaznavanje lokacije, temperature, vlage v zraku, gibanja, vibracij, pospeška, kemičnih sprememb v vodi ali zraku. Zaznavna plast je namenjena identifikaciji predmetov ali zaznavi fizikalnih pojavov, kar počne s pomočjo senzorjev. Zbrani podatki se nato pretvorijo v digitalne signale in se posredujejo v omrežno plast. [6][8]

2.2 Omrežna plast in API-ji

Omrežna ali prenosna plast je zadolžena za varno prenašanje podatkov iz senzorskih naprav do aplikacij prek vmesnikov, prehodov (angl. gateway) med različnimi omrežji ter z uporabo različnih komunikacijskih tehnologij in protokolov. Pri tem sodelujejo tehnologije, kot so 3G, 4G, 5G, Wi-Fi, Bluetooth, ZigBee ali druge. [6][9]

Mehanizmi, ki omogočajo komunikacijo med dvema komponentama programske opreme z uporabo niza različnih protokolov, se imenujejo aplikacijski programski vmesniki (angl. Application Programming Interface - API).

2.2.1 HTTP protokol

HTTP (angl. Hypertext Transfer Protocol) je komunikacijski protokol aplikacijskega sloja za prenos hipermedijskih dokumentov, na primer HTML (angl. HyperText Markup Language). Zasnovan je za komunikacijo med

spletnimi brskalniki in spletnimi strežniki, lahko pa se uporablja tudi v druge namene, na primer za IoT. HTTP sledi klasičnemu modelu odjemalec-strežnik. Povezava deluje le v to eno smer, prav tako se lahko obdeluje le ena zahteva naenkrat. HTTP je protokol brez stanja, kar pomeni, da strežnik ne hrani nobenih podatkov (stanja) med dvema zahtevama. [11]

2.2.2 MQTT

MQTT (angl. Message Queue Telemetry Transport) je komunikacijski protokol, ki je bil zasnovan posebej za IoT. Ko je povezava MQTT vzpostavljena, je mogoče preko nje poslati poljubno število sporočil v obe smeri - torej od senzorja v zaledje (angl. back-end) in obratno. Za razliko od HTTP-ja se pri MQTT-ju lahko shrani zadnji dober podatek. Omogočeno je tudi preprosto dodajanje več porabnikov in proizvajalcev podatkov.

2.3 Procesna plast

Procesna plast je odgovorna za shranjevanje, analiziranje in obdelavo podatkov, ki jih je prejela iz omrežne plasti. Primarna in najpomembnejša tehnologija te plasti je računalništvo v oblaku. Računamo na to, da se bodo v prihodnosti pojavile nove računalniške tehnologije, ki bodo primernejše za IoT in zato so raziskave in razvoj na procesni plasti zelo pomembni za razvoj interneta stvari. [6][7] V to plast spadata tudi umetna inteligenca in strojno učenje.

2.4 Aplikacijska plast

Aplikacijska plast od procesne plasti prejme podatke, ki jih potem uporabi za zagotavljanje zahtevanih storitev. Aplikacijska plast predstavlja tudi most med IoT in uporabnikom, odgovorna je za to, da uporabniku na prijazen in učinkovit način zagotavlja storitve, specifične za to aplikacijo. Določa različne uporabe IoT, na primer pametne domove, pametna mesta in pametno zdravstvo.[6][13] V tej plasti je zelo pomembno zagotavljanje človeku prijazne interakcije.

2.5 Poslovna plast

Poslovni sloj se nahaja nad aplikacijskim slojem in je najvišji sloj v petslojni arhitekturi. Ta sloj je odgovoren za upravljanje celotnega sistema IoT, vključno z aplikacijami in storitvami. Na podlagi podatkov, ki jih prejme iz aplikacijskega sloja, gradi poslovne modele, grafe in diagrame poteka. Na podlagi analize rezultatov ta plast pomaga določiti prihodnje poslovne strategije. [6]

3 Testiranje IoT naprav

S strmim naraščanjem števila naprav IoT je tudi potreba po varnosti in učinkovitosti čedalje pomembnejša. Zato testiranje naprav IoT postaja zelo velik in pomemben del pri sami razvoju naprav rešitev IoT. Testiranje naprav IoT pa se razlikuje od tradicionalnega testiranja programske opreme, saj predstavlja kombinacijo tako programske kot strojne opreme. To se je pred IoT testiralo vsako posebej, zdaj pa je treba testirati cel ekosistem IoT, kar je velik izziv. [14][16]

IoT je sistem, ki velikokrat komunicira v realnem času, kar pomeni, da lahko težave z zmogljivostjo ali z varnostjo v katerem koli njegovem delu povzroči težave z delovanjem drugih delov omrežja. Testiranje IoT je nabor testov, s katerimi se preverjajo funkcionalnost, varnost in zmogljivost. Pri tem obstajajo izzivi zaradi razdrobljenosti sistema IoT, kar pa je mogoče odpraviti z uvedbo več testnih skupin, ki preverjajo zanesljivost vseh komponent na več platformah in napravah. [14][16]

3.1 Testiranje združljivosti

Pri testiranju združljivosti je vključenih več naprav, brskalnikov, operacijskih sistemov in načinov komunikacije. Bistvo tega testiranja je izpostavljanje sistema različnim okoljem in preverjanje, ali se v vseh obnaša enako. Preden izide nova različica, mora ta biti popolnoma združljiva z vsemi različicami drugih naprav in ostalih delov sistema. Enako velja tudi za posodobitve programske opreme, Pomembno je torej zagotoviti, da po posodobitvi vse komponente še delujejo tako, kot morajo. [4]

3.2 Testiranje zmogljivosti

V tej fazi testiranja se preverja zmogljivost pri največji obremenitvi, testiranje sistema za več naprav hkrati, spremlja se tudi poraba baterije. Naprave se tudi testirajo pri različnih omrežnih pogojih, pri aplikacijah pa je treba testirati hitrost pri velikih količinah podatkov. [4]

3.3 Testiranje povezljivosti

Pri testiranju povezljivosti se preverja, kako zanesljivo so komponente sistema IoT povezane med sabo. Zagotoviti je treba, da je sistem ves čas odziven, poleg tega pa je treba poskrbeti za ustrezna opozorila v primeru izgube povezave. Ob kratkotrajnih izgubah povezave mora sistem sam poskusiti nazaj vzpostaviti povezavo. [15]

3.4 Testiranje uporabniškega vmesnika

Testiranje uporabniškega vmesnika zagotavlja, da ima končni uporabnik intuitivno programsko opremo, čim preprostejšo za uporabo. Uporabniški vmesnik mora biti tudi čim bolj pregleden in estetsko zadovoljiv. Sem spada preverjanje prikaza zaslona, menijev, gumbov, ikon in ostalih stvari, s katerimi rokuje uporabnik. [4]

3.5 Testiranje varnosti

Testiranje varnosti je zagotovo ena najpomembnejših faz testiranja. Gre za iskanje in odpravljanje pomanjkljivosti, ki bi jih hekerji lahko izkoristili za dostop do spreminjanja ali krajo podatkov. Z razširitvijo naprav IoT in s čedalje večjo priljubljenostjo in aplikativnostjo se veča tudi število vstopnih točk in možnih vektorjev napada kibernetskih napadalcev . Naprave IoT uporabljamo za namen nadzora nad svojimi domovi, avtomobili, bančništvom, pa tudi za avtomatizacijo kompleksnih naprav in takšne stvari so zelo ranljive za kibernetske napade, v primeru teh pa je škoda neodpravljiva. Ne gre namreč le za finančno škodo, pač pa tudi za krajo identitete, krajo podatkov itd. [4][17] Ker so od IoT odvisni tudi mnogi drugi sistemi, to lahko pomeni tudi zaustavitev drugih sistemov, na primer finančnih, elektroenergetskih in mnogih

drugih sistemov, od katerih jih veliko spada v kritično infrastrukturo z vidika varnosti.

3.6 Testiranje interoperabilnosti

Interoperabilnost je sposobnost sistemov in naprav, da med sabo komunicirajo kljub različnim implementacijam. Več kot je povezanih naprav, več je prostora za napake tako s strani strojne kot tudi programske opreme. Testiranje interoperabilnosti je preizkušanje programske opreme in tehnologij, če so združljive z ostalimi. To je treba testirati, da se prepričamo, da se bodo različne tehnologije sposobne vključiti v arhitekturo z veliko različnimi elementi, pri čemer je ključna brezhibna integracija in usklajeno delovanje vseh naprav, ki so del sistema. [16][18][19]

3.7 Testiranje skladnosti

Testiranje skladnosti določa, v kolikšni meri je implementacija določenega standarda v skladu s posameznimi zahtevami tega standarda. V zadnjih letih je bilo predstavljenih veliko standardov na področju povezljivosti v internet, različnih protokolov komunikacije in na vseh ostalih področjih, ki so prav tako del arhitekture IoT. Takšni standardi so na primer IPv6 (angl. Internet Protocol version 6), Bluetooth Low Energy, ZigBee. Vse tehnologije, ki so del nekega sistema IoT, morajo biti v skladu s temi že obstoječimi standardi. [5][19]

4 Testiranje uporabniškega vmesnika na konkretnem primeru

4.1 eIoT

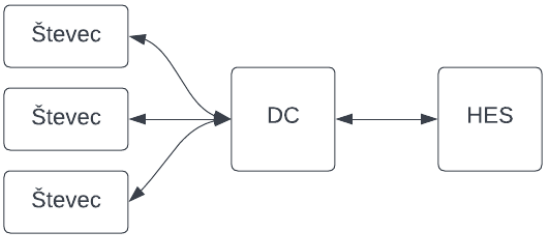
IoT zajema ogromno področij in s tem veliko različnih aplikacij. Eno izmed najpomembnejših področij je elektroenergetika, uporaba IoT na tem področju pa se imenuje energetski IoT (angl. energy IoT - eIoT). Cilj eIoT je avtomatizacija in nadgradnja elektroenergetske infrastrukture, s čimer bi se izboljšala učinkovitost. Prav tako naj bi pomagal zmanjšati ogljični odtis. [22][23][24]

V eIoT spadajo pametna omrežja (angl. Smart Grid - SG), ta tehnologija je pomembno orodje za zagotavljanje trajnostne in varne energetske prihodnosti. Smart Grid je nadgradnja tradicionalnega elektroenergetskega omrežja in omogoča prilagodljivo upravljanje energetskih virov, optimizacijo energetskih virov, izboljšanje vpogleda v delovanje, boljše napovedovanje vzdrževanja in dodatno varnost. [20]

Elektroenergetskim podjetjem eIoT omogoča, da ocenijo stanje sistema bistveno bolje, kot je bilo to mogoče prej. Na primer, s pametnimi števci lahko elektroenergetska družba odkrije potrebe po energiji v realnem času z veliko natančnostjo.[21]

4.2 Testiranje DC

Testirali smo uporabniški vmesnik, ki je del podatkovnega koncentratorja (angl. Data Concentrator - DC) sistema eIoT na sliki 1. Podatkovni koncentrator je programska in strojna rešitev, ki povezuje več podatkovnih kanalov. [25] Podatkovni kanali so v našem primeru števci, ki preko protokola COSEM (angl. Companion Specification for Energy Metering) pošiljajo podatke v podatkovni



Slika 1: Arhitektura obravnavanega sistema

koncentrator. Z druge strani je povezan na glavni sistem (ang. Head End System - HES), v našem primeru preko protokola ethernet. HES je zadolžen za sprejemanje podatkov iz števcev preko podatkovnega koncentratorja brez posredovanja človeka.

Testirali smo uporabniško izkušnjo (angl. User Experience - UX) za uporabnika DC in upravljalca elektroenergetskega omrežja. Zaradi velike količine podatkov, ki se prenašajo po sistemu, je bilo treba uporabniški vmesnik temeljito testirati, saj mora biti za učinkovito uporabo hiter in intuitiven.

4.2.1 Testiranje gumbov

Pri testiranju moramo biti pozorni na vsako malenkost. Začeli smo s testiranjem gumbov, preverili smo, kaj se zgodi ob različnih klikih in iskali kakšne napake, ki bi lahko vplivale na funkcionalnost ali na uporabniško izkušnjo.

4.2.2 Testiranje vnašanja podatkov

Naslednja stvar, ki smo jo testirali, je bilo vnašanje podatkov. Preverjali smo, kako aplikacija uporabniku sporoči oziroma pokaže, da je podatke vnesel pravilno in kako se obnaša, kadar jih vnese narobe, torej ali ga opozori, ali pa preprosto ne uboga ukaza uporabnika. Na primer v koledar, kamor se mora vnesti interval, smo vnašali nesmisle. Vnašali smo datume v prihodnosti, preverili smo, kaj se zgodi, če je začetni datum kasneje od končnega ali pa če sta slučajno enaka. Preverjali smo tudi zelo dolge intervale in iskali mejo, pri kateri funkcija ni več delovala, kot bi morala. Ta del testiranja uporabniškega vmesnika je zelo pomemben, namreč če aplikacija dovoli vnesti nesmisle, se lahko sistem posledično obnaša nepredvidljivo, takšne napake pa je težko odkriti.

Predlagali smo rešitev, kjer sistem preveri vnešene podatke in tudi avtomatsko dopolne vnos podatkov, ki so že bili uporabljeni. Predlagali smo tudi izboljšanje načina vnosa podatkov, da bi bila uporabniška izkušnja čim lažja in hitrejša.

4.2.3 Testiranje dostopa do uporabniškega vmesnika

V nadaljevanju smo testirali tudi vstop v aplikacijo z različnimi funkcijami in preverjali možne scenarije pri prijavi uporabnikov. To smo potem ponovili še v drugih brskalnikih in v anonimnih zavihkih, da bi videli, če pride do kakšnih sprememb v delovanju.

Aplikacijo smo testirali še na raznih drugih formatih – na manjšem računalniku, na veliko večjem zaslonu in na tabličnem zaslonu. Pri tem smo ugotovili, da se delovanje aplikacije spremeni odvisno od velikosti zaslona, kar se sicer ne bi smelo zgoditi, zato bo to napako tudi treba odpraviti.

4.3 Testiranje uporabniške izkušnje

Pri vsem tem smo bili pozorni na kakšne zoprne malenkosti, ki bi jih lahko izboljšali in bi uporabniku s tem olajšali uporabo te aplikacije. Predlagali smo natančnejša opozorila pri napačnem vnosu, da uporabnik točno ve, kaj se na tistem mestu od njega pričakuje.

5 Zaključek

Natančno in vseobsegajoče testiranje delovanja naprav IoT je zelo pomembno. Ko naprave IoT niso dovolj dobro testirane, preden gredo na trg, to povzročili veliko težav - nepredvidljivo obnašanje v situacijah, ki jih testiranje ni predvidelo, nedelovanje, ker je naprava nekompatibilna z nekaterimi komunikacijskimi protokoli in če bi bila aplikacija slabo testiranega IoT sistema avtonomna na kakšnem bolj občutljivem področju, so lahko posledice zelo hude.

Testirali smo UX za uporabnika naprave IoT. Pri tem smo ugotovili, da je največ manjših napak pri postavitvi gumbov. Bolj problematično pa je bilo nepredvidljivo obnašanje aplikacije pri narobe vnešenih podatkih.

Pri takšnih napravah kot je ta, ki smo jo testirali, pa je še posebej pomembno testiranje varnosti. Takšne naprave so namreč do zdaj bile načrtovane za uporabo v zasebnih omrežjih in zato se varnosti ni posvečalo toliko pozornosti. S potencialnim odpiranjem teh sistemov v javna omrežja pa te naprave IoT postajajo zelo ranljive, kar pomeni, da je varnosti treba posvetiti veliko več pozornosti. Zato pričakujemo, da bosta razvoj in testiranje varnostnih mehanizmov naprav IoT eden ključnih izziviv v prihodnosti IoT.

Zahvala

Raziskava je nadgradnja projektnega sodelovanja s podjetjem Iskraemeco. Delo je podprla Agencija za raziskovalno dejavnost Republike Slovenije v okviru raziskovalnega programa ≫Decentralizirane rešitve za digitalizacijo industrije ter pametnih mest in skupnosti≪.

Literatura

[1] Why is IoT important?, https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[2] What is the internet of things (IoT)?, https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[3] What is IoT?, https://www.oracle.com/internet-of-things/what-is-iot

[4] Internet of Things (IoT) Testing: Why Is It So Important?, https://relevant.software/blog/iot-testing-importance/

[5] Internet stvari in analiza dodane vrednosti pametne naprave za končnega uporabnika, https://dk.um.si/Dokument.php?id=86036&lang=slv

[6] Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges, https://ieeexplore.ieee.org/abstract/document/6424332

[7] Internet of Things: Architectures, Protocols, and Applications, https://www.hindawi.com/journals/jece/2017/9324035/

[8] A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications, https://ieeexplore.ieee.org/abstract/document/7879243

[9] Research on the architecture of Internet of Things, https://ieeexplore.ieee.org/abstract/document/7879243

[10] What is an API?, https://aws.amazon.com/what-is/api/

[11] HTTP, https://developer.mozilla.org/en-US/docs/Web/HTTP

[12] MQTT Vs. HTTP for IoT, https://www.hivemq.com/blog/mqtt-vs-http-protocols-in-iot-iiot/

[13] Common application layer protocols in IoT explained, https://www.techtarget.com/iotagenda/feature/Common-application-layer-protocols-in-IoT-explained

[14] IOT Testing Framework, https://www.clariontech.com/blog/iot-testing-framework

[15] Internet Of Things (IoT) Testing: Challenges, Tools And Testing Approach, https://www.softwaretestinghelp.com/internet-of-things-iot-testing/

[16] Towards an open framework of online interoperability and performance tests for the Internet of Things, https://ieeexplore.ieee.org/document/8016248

[17] A Comprehensive Guide to IoT Security Testing, https://www.getastra.com/blog/security-audit/iot-security-testing

[18] Understanding IoT Interoperability Testing, https://www.einfochips.com/blog/understanding-iot-interoperability-testing/

[19] IoT-TaaS: Towards a Prospective IoT Testing Framework, https://ieeexplore.ieee.org/document/8281514

[20] How IoT Enables the Smart Grid - Applications, Benefits, and Use Cases, https://www.particle.io/iot-guides-and-resources/iot-smart-grid-applications-benefits-and-use-cases/

[21] What Is the Smart Grid and How Is It Enabled by IoT?, https://www.digi.com/blog/post/what-is-the-smart-grid-and-how-enabled-by-iot

[22] What Is The Internet Of Energy?, https://www.nesfircroft.com/blog/2019/05/what-is-the-internet-of-energy?source=google.com

[23] The Internet of Energy: Challenges and Purpose, https://justenergy.com/blog/internet-of-energy-what-is-it-why-important/

[24] Internet of Energy (IoE), https://www.investopedia.com/terms/i/internet-energy-ioe.asp

[25] Data Concentrator, http://www.subnet.com/resources/dictionary/data-concentrator.aspx