

# *Potisna obvestila in njihova uporaba v sodobnih sistemih obveščanja*

Marko Novak

Oddelek za Računalništvo in informatiko

Univerza na Primorskem

Fakulteta za Matematiko, Naravoslovje in Informacijske Tehnologije (FAMNIT)

Koper, Slovenija

**Abstract**—Sodobni informacijski sistemi nudijo uporabnikom vse več in več koristnih informacij na vseh področjih. Prav zaradi tako velike količine podatkov so sistemi obveščanja ključni za učinkovito komunikacijo z uporabniki. Vsi si želimo, da bi podatki, ki jih prejemo bili kar se da uporabni in pravočasni, hkrati pa tudi nemoteči. Potisna obvestila so eden od sodobnih načinov obveščanja, ki take lastnosti v veliki meri imajo. V tem dokumentu je predstavljena tudi “push” tehnologija na kateri potisna obvestila temeljijo in njene prednosti ter razlike s “pull” tehnologijo. Opisani so arhitektura sistema, dobre implementacijske prakse in lastnosti učinkovitih potisnih obvestil. Navedene so tudi glavne prednosti in slabosti takega načina obveščanja ter kateri so glavni ponudniki potisnih storitev.

**Keywords**—*push notifications; modern notification systems; push technology; push service providers.*

**Ključne besede**—*potisna obvestila; sodobni sistemi obveščanja; potisna tehnologija; ponudniki potisnih storitev.*

## I. UVOD

Informacijski sistemi oz. sistemi obveščanja vseh vrst nas vedno bolj obkrožajo. Učinkovita komunikacija med slednjimi in uporabniki je ključnega pomena, da bi to sodelovanje bilo istočasno kar se da uporabno in nemoteče. V zadnjih desetih letih smo pri sodobnih informacijskih sistemih videli velik porast uporabe potisnih obvestil kot način obveščanja. To smo lahko zaznali predvsem na področju spleta in aplikacij na priročnih mobilnih napravah. Porast lahko pripišemo tudi dejstvu, da vedno znova iščemo načine, kako uporabnika obveščati o pomembnih dogodkih, o katerih želi biti sam obveščen na njemu čim bolj prijazen in hkrati učinkovit način.

Ta dokument se osredotoča na enega izmed možnih načinov komunikacije med sodobnim informacijskim sistemom ter uporabnikom, in sicer v smeri od sistema k uporabniku. V nadaljevanju je informacijski sistem mišljen kot spletna storitev (spletna stran, spletna aplikacija, ipd.) ali aplikacija za mobilne naprave z operacijskim sistemom Android in iOS.

Zelo pomembno je, da je uporabnik obveščen o pomembnih dogodkih, ki se dogajajo v sistemu, tudi ko ni aktivno vključen v komunikacijo. Lahko se zgodi, da želi sistem uporabniku podati pomemben podatek, ki pa ga slednji ne vidi, ker v tem trenutku sistema ne spremlja aktivno. V takih primerih mora sistem imeti način, s katerim učinkovito,

vendar nemoteče, uporabnika obvesti o zadevah, o katerih želi biti obveščen.

Pri izbiri načina obveščanja, se moramo vprašati o pomembnosti podatkov in se posledično odločiti, s katerim orodjem bomo uporabnika obveščali. Potisna obvestila verjetno niso najbolj primerna za obveščanje najvišje prioritete, ki zahteva takojšen (nujen) odziv. Imajo namreč določene omejitve, ki jim v določenih primerih onemogočajo, da bi uspešno zaključila proces obveščanja. Toda velik delež obvestil ni najvišje prioritete oz. jih lahko uporabnik tudi spregleda in to ne bo imelo nobenih posledic. Kot primer pomislimo na mobilno aplikacijo za novice, ki nas preko potisnih obvestil obvešča o najnovejših novicah. Če uporabnik tega obvestila ne vidi takoj ali ga sploh ne opazi, to najverjetneje ne bo imelo prav veliko negativnih posledic. V tem primeru je zato lahko obveščanje s potisnimi obvestili najprimernejša izbira.

V tem dokumentu sem se osredotočil na potisna obvestila, ker so med uporabniki in ponudniki v zadnjih letih med najbolj priljubljenimi načini obveščanja. Njihova priljubljenost narašča iz različnih razlogov, ki jih bom predstavil v nadaljevanju. Potisna obvestila so v bistvu tehnologija, ki napravi povezani v splet omogoča, da brez njene zahteve prejme obvestila, ki so bila “potisnjena” iz strežnika.

Koncept potisnih obvestil ni nov, saj zelo spominja na SMS (Short Message Service), ki pa ga dobro poznamo že od začetka devetdesetih let. Toda že od samega začetka je storitev SMS ostala nekako izolirana in se večinoma ni integrirala v sisteme za široko uporabo. Po drugi strani pa so se za potisna obvestila ustvarili standardi, ki so jih hitro posvojili velikani informacijskih tehnologij, kot so Apple, Google in Microsoft. To je omogočilo njihov razvoj do današnjega dne, ko jih redno uporabljamo v sodobnih sistemih obveščanja kot učinkovit toda nemoteč način komuniciranja.

V nadaljevanju se bom v naslednjem vrstnem redu dotaknil točk:

- predstavitev osnovne tehnologije potisnih obvestil, ki se imenuje “Push” in za katero lahko rečemo, da je “nasprotje” od “Pull”;
- opis in zgodovina potisnih obvestil;
- dobre prakse tako za uporabo, kot implementacijo potisnih obvestil;

- prednosti in slabosti takega načina obveščanja;
- glavne implementacijske komponente tega orodja ter
- predstavitev nekaj pomembnih ponudnikov potisnih storitev.

## II. "PULL" IN "PUSH" TEHNOLOGIJA

Kot nam samo ime pove, potisna obvestila temeljijo na potisni tehnologiji, ki ima poleg "Pull" tehnologije ključno vlogo pri učinkoviti komunikaciji med dvema elektronskima napravama, ki sta povezani preko spleta. Glavna razlika med "Push" in "Pull" tehnologijo je, da se pri slednji začetna zahteva za izmenjavo podatkov izvede s strani odjemalca, na katero mu nato, v primeru uspešne dvostranske komunikacije, strežnik odgovori. Pri "Push" tehnologiji pa strežnik sam pošlje podatke odjemalcu, na da bi ga ta za to zaprosil.

Za lažje razumevanje delovanja "Push" tehnologije in posledično njenega podsistema potisnih obvestil, bom predstavil najprej "Pull" tehnologijo, s katero lahko tudi simuliramo "Pull" delovanje.

### A. "Pull" tehnologija

Osnove samega mrežnega računalništva temeljijo na "Pull" zahtevah s strani odjemalcev. Ti pošiljajo zahteve centralnim strežnikom, ki nato nanje odgovarjajo. Tak način komunikacije omogočajo tudi HTTP (HyperText Transfer Protocol) zahteve, ki jih opravljajo spletne strani.

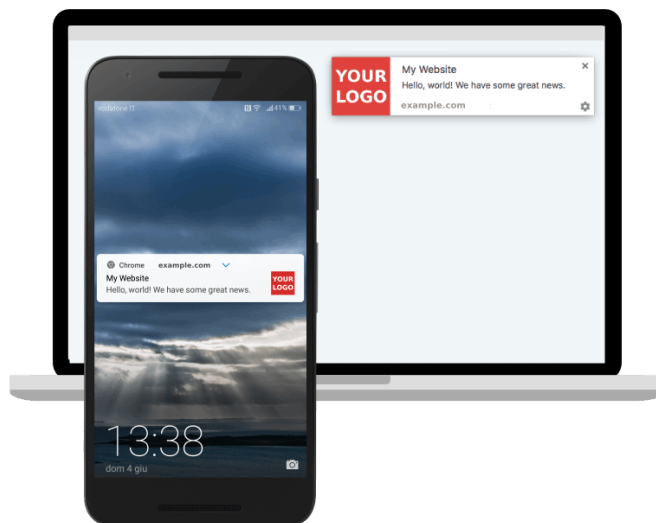
Z uporabo "Pull" tehnologije lahko navidezno simuliramo delovanje "Push" tehnologije. To poteka tako, da odjemalec zelo pogosto pošilja zahteve strežniku, z namenom preverjanja ali ima strežnik nove podatke, ki jih želi poslati odjemalcu. Na tak način se odjemalcu zdi, kakor da so novi podatki, ki jih prejema, dejansko potisnjeni iz strežnika. Če so odjemalčeve zahteve namreč dovolj pogoste, je lahko komunikacija skoraj v realnem času. Ta metoda se imenuje "Polling". Hitro pa lahko opazimo, da obstaja možnost, da je tak način komunikacije precej neučinkovit. Razmerje med uporabnimi in vsemi izmenjanimi podatki je namreč lahko zelo majhno, kar nezanesljivo poveča porabo energije in pa zasedenost mrežne pasovne širine. Po drugi strani pa lahko "Polling" metoda zadosti večjim zahtevam po varni komunikaciji. [1]

### B. "Push" tehnologija

O "Push" tehnologiji govorimo, kadar se komunikacija izvede iz smeri strežnika proti odjemalcu, brez da bi slednji to v tistem trenutku zahteval.

Obstaja več vrst "push" tehnologij, ampak tista, ki se danes v največji meri uporablja, temelji na neprekinjeni TCP/IP povezavi med strežnikom in odjemalcem. Taka povezava omogoča, da so podatki dostavljeni v realnem času, vendar na račun večje porabe električne energije. Poraba se skuša minimizirati tako, da se različni sistemi (Android, Apple iOS, itd.) poslužujejo storitev za centraliziran sprejem potisnih sporočil. [24] V takih primerih strežniku rečemo izdajatelj ali glavni strežnik in odjemalcu naročnik ("subscriber/publisher"). Pri tej tehnologiji praviloma velja, da se morajo odjemalci, pred začetkom izvajanja potisnih storitev, registrirati na določen kanal ali vsebino, če želijo prejemati nove podatke, ki jih strežniki objavljajo. Vsakič, ko se pojavi objava na tem kanalu ali objava, ki je povezana s to

vsebino, se podatek potisne vsem registriranim odjemalcem (naročnikom). Odjemalcu se nato podatki praviloma prikažejo v obliki manjšega pojavnega okna. To se zgodi tudi, če je povezana spletna stran ali aplikacija zaprta. [2]



Slika 1: Primer izgleda potisnega sporočila na uporabnikovi napravi

## III. POTISNA OBVESTILA

Danes je uporaba potisnih obvestil za namene obveščanja najbolj prisotna na brskalnikih in aplikacijah na prenosnih ter fiksnih napravah, ki so povezane v splet. Uporabniku se prikažejo v obliki manjšega pojavnega okna na računalniških ekranih ali pa kot sistemsko obvestilo na mobilnih napravah, torej izven samega delovanja programa ali aplikacije. Prikaz potisnih obvestil spominja na pojavna okna ("Pop-up" okna), ki pa nimajo te lastnosti, da se lahko prikažejo izven spletne strani in da so generirana na podlagi potisnjenih podatkov iz strežnika.

Velikokrat lahko opazimo, da se storitev potisnih obvestil deli v dve kategoriji. V prvi so spletna potisna obvestila, v drugi pa potisna obvestila za mobilne aplikacije. Glavna razlika med njima je dosež. Spletna potisna obvestila lahko pošiljamo vsakemu uporabniku, ki ima nameščen brskalnik, ki to tehnologijo podpira (Chrome, Safari, itd.). Zato lahko z njimi dosežemo praktično vsakega uporabnika, ki je povezan v splet. Po drugi strani pa lahko aplikacije ponudniku storitev nudijo več osebnih podatkov o uporabniku, ki so vedno zelo dobrodošli za učinkovitejšo uporabo potisnih obvestil.

### A. Zgodovina potisnih obvestil

Začetek širše uporabe potisnih obvestil, kot jih poznamo danes, lahko pripišemo podjetju Apple Inc. To je leta 2009 lansiralo svojo rešitev Apple Push Notification service (APNs), namenjeno takratnemu Apple operacijskemu sistemu za prenosne naprave iOS 3. Ta storitev omogoča tudi razvijalcem, ki niso del podjetja Apple, pošiljanje sporočil za potisna obvestila svojih aplikacij na uporabniških napravah. Šele kasneje pa so storitev omogočili tudi za spletne strani, in sicer z operacijskim sistemom Mac OS X 10.9 (Mavericks). Omenjena rešitev prejemanje potisnih obvestil združuje v eno samo storitev, ki nato sporočila razporeja posamezni aplikaciji

na uporabnikovi napravi. Taka rešitev je energijsko in procesorsko manj potratna kot pa v ozadju izvajajoče se programske storitve vsake aplikacije, ki se poslužuje na primer "Polling" metode, ki sem jo malo višje že omenil. [3] [4]

Po tem, ko je Apple izdal svojo storitev za pošiljanje potisnih obvestil, sta to storila tudi Google in Microsoft. Google je leta 2010 izdal Google Cloud to Device Messaging (C2DM) za operacijski sistem Android 2.2. Leta 2012 pa ga je zamenjal z Google Cloud Messaging (GCM), ki je imel med drugim izboljšano avtentikacijo in dostavo sporočil ter je omogočal prenos večje količine podatkov v posameznem sporočilu. GCM je omogočal pošiljanje podatkov tako napravam z operacijskim sistemom Android, kot aplikacijam narejenim za internetni brskalnik Google Chrome. Razširjenost Googlovega brskalnika in operacijskega sistema ter dejstvo, da je bila storitev GCM brezplačna, sta omogočila, da se je GCM zelo razširil. Leta 2019 je Google uradno zamenjal GCM s Firebase Cloud Messaging (FCM) podjetja Firebase, ki ga je Google tudi kupil leta 2014. FCM je "cross platform", saj omogoča pošiljanje sporočil in obvestil na operacijska sistema Android ter Apple iOS, kot tudi na spletne aplikacije za razne spletne brskalnike. FCM je neke vrste vmesni člen med strežnikom in odjemalcem pri pošiljanju potisnih sporočil. Tem členom rečemo ponudniki potisnih storitev. Tudi tretji velikan na področju mobilnih naprav in ostalih operacijskim sistemov, Microsoft, je leta 2015 začel ponujati svojo uradno storitev potisnih sporočil, namenjeno operacijskim sistemom Windows 10, Windows 10 Mobile, igričarskemu Xbox-u, itd. [5] [6]

Storitev pošiljanja potisnih obvestil lahko razdelimo na dva glavna procesa: prvi je pošiljanje potisnih sporočil in drugi je prikazovanje obvestil. Prikazno obvestilo se ustvari na podlagi prejetih podatkov iz potisnega sporočila. Nekateri ponudniki potisnih storitev omogočajo, da se prikazovanje obvestil definira že na strežniku, ki je potisno obvestilo poslalo. Tako obvestilo je večinoma omejeno na zelo majhno količino podatkov. Bolj fleksibilen in učinkovit način pošiljanja potisnih obvestil pa je, da se vsebina potisnega obvestila pošlje s potisnim sporočilo in se nato na uporabnikovi napravi, na podlagi uporabniških podatkov, določi način prikazovanja obvestila.

### B. Potisna sporočila

Strežnik je zadolžen, da sproži pošiljanje potisnega sporočila uporabniku. To lahko stori samo, če se je uporabnik registriral na prejemanje njegovih obvestil. Če je uporabnik registriran pri ponudniku potisnih storitev, se med njima vzpostavi povezava na podlagi "websocket-ov". Slednje so nenehne povezave med strežnikom in odjemalcem, ki zagotovijo stabilno dvosmerno komunikacijo preko ene TCP/IP povezave. To omogoča prejemanje potisnih sporočil v realnem času. Podatki, ki se prenašajo v potisnih sporočilih, so gradniki obvestila, ki se bo prikazalo na uporabnikovi napravi.

### C. Prikazovanje obvestila

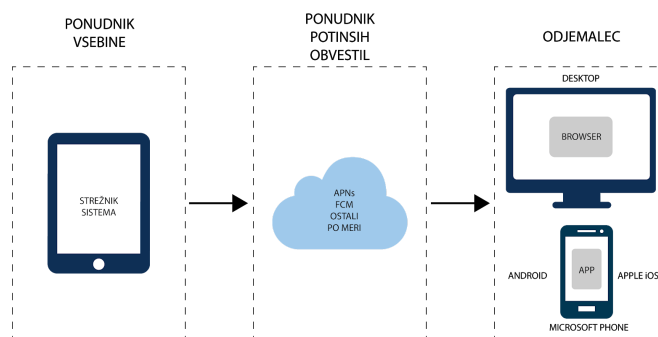
Ko uporabnikova naprava prejme potisno sporočilo, se lahko iz tega generira obvestilo, ki se nato prikaže na ekranu. Obvestila so tekstovna in slikovna sporočila, ki se pojavijo na uporabniškem vmesniku naprave, praviloma izven same aplikacije, ki je obvestilo generirala. S takimi obvestili ponavadi želimo uporabnike pravočasno obvestiti in pritegniti

njihovo pozornost z namenom, da jim nudimo podatke, ki jih želijo prejemati oz. z namenom, da uporabnike zadržimo in jih prepričamo k večji uporabi ponujene storitve s prilagojenimi in ciljnim vsebinami. [7]

## IV. ARHITEKTURA SISTEMA POTISNIH OBVESTIL IN GLAVNE TOČKE NJIHOVEGA DELOVANJA

V osnovi so glavni akterji, ki omogočajo delovanje potisnih obvestil za vse naprave oz. operacijske sisteme zelo podobni. Poglejmo si, kateri so:

- **Ponudnik vsebine:** to je strežnik ali druga naprava, ki hrani jedrno logiko celotnega sistema in je akter, ki ustvari vsebino potisnih obvestil ter sproži njihovo pošiljanje;
- **Ponudnik potisnih obvestil (PPO):** Naloga PPOja je, da izbranemu naslovniku dostavi sporočilo (obvestilo), ki mu ga želi poslati nek ponudnik vsebine. PPO poskrbi tudi za primere, ko naprava ni povezana v splet, saj z dostavo sporočila poskuša večkrat, dokler ne preteče najdaljši možni čas ali je izpolnjen kakšen drug prednastavljen pogoj. Danes to vlogo v veliki meri igrata Apple Push Notification Service (APNs) pri Apple sistemih in pa Firebase Cloud Messaging (FCM) pri napravah z operacijskim sistemom Android ter Chrome brskalniki, saj sta ta dva operacijska sistema tudi med bolj razširjenimi.
- **Odjemalec:** odjemalec je lahko namizni računalnik ali druga mobilna naprava končnega uporabnika, ki podpira prejemanje potisnih obvestil preko brskalnika ali mobilne aplikacije. Naprava odjemalca mora biti seveda povezana v splet, če želimo, da se obvestilo dostavi. Ko podatki prispejo na napravo odjemalca, jih mora storitev za prestrezanje takih sporočil pravilno usmeriti glede na njihovo vsebino: obvestilo se lahko samodejno prikaže glede na v njem (na strežniku) prednastavljene parametre ali pa se podatki posredujejo povezani aplikaciji na napravi, ki jih nato obravnava na osnovi svoje logike za prikazovanje obvestil.



Slika 2: Arhitektura sistema za pošiljanje potisnih obvestil: puščice označujejo smer potovanja podatkov pri pošiljanju potisnega obvestila iz strežnika do uporabnika.

Lahko rečemo, da skoraj vsaka naprava, bodisi mobilna naprava ali namizni računalnik, ki je povezana v splet, podpira možnost prejemanja takih obvestil. To pa zato, ker imajo vse te naprave ponavadi že kot privzeto prednaložen vsaj en

brskalnik. Tisti, ki so najbolj razširjeni, pa že vsi podpirajo tehnologije, ki omogočajo uporabo potisnih obvestil. [11]

Zaradi razširjenosti potisnega obveščanja na brskalnikih, bom v tem delu na grobo predstavil primer njihove implementacije. Slednja predvideva uporabo jezika JavaScript, ki je med spletnimi rešitvami zelo razširjen.

Tehnologijo spletnih potisnih obvestil lahko delimo na potisni del in del, ki upravlja s samim prikazovanjem obvestil. Ta dva dela uporabljata različna API-ja, ki pa se dopolnjujeta. Push API spletni aplikaciji omogoča prejetje sporočil, ki so bila potisnjena iz strežnika, ne glede na to ali je aplikacija v ospredju sistema (aktivna). Notification API obvestila, namenjen prikazovanju obvestil, pa spletnim stranem omogoča, da nadzirajo prikazovanje sistemskih obvestil na napravah končnih uporabnikov. [12] [13] [14]

Poglejmo najprej potisni del, pri katerem so za implementacijo ključni trije koraki:

### 1. Prijava uporabnika na potisna obvestila

Najprej je potrebno od uporabnika pridobiti dovoljenje in nato izvesti prijavo na prejetje potisnih obvestil. Ko imamo dovoljenje, lahko s strežniškimi ključi, ki so unikatni za vsak strežnik, od ponudnika potisnih obvestil zahtevamo "PushSubscription" objekt.

Ta objekt med drugim vsebuje dva zelo pomembna podatka, ki sta URL povezava in šifrirni ključ. Oba sta nujna za pošiljanje podatkov. Vse lastnosti tega objekta so:

- PushSubscription.endpoint (samo branje) - URL povezan z aplikacijo/spletno stranjo, ki bo potisna sporočila prejela.
- PushSubscription.expirationTime (samo branje) - vsebuje rok trajanja registracije na prejetje potisnih sporočil.
- PushSubscription.options (samo branje)
- Objekt, ki ima naslednje lastnosti:
  - userVisibleOnly - določa ali bodo prejeta sporočila vedno vidna uporabniku.
  - applicationServerKey - javni šifrirni ključ, ki ga bo strežnik uporabljal za pošiljanje sporočil uporabnikom preko ponudnika potisnih sporočil.
- PushSubscription.subscriptionId (samo branje) - ID povezan s to registracijo na prejetje potisnih obvestil.

Naj dodatno izpostavim tudi "PushSubscription.expirationTime" vrednost, ki določa čas veljavnosti registracije na potisna obvestila. To vrednost lahko določijo brskalniki. Toda tisti, ki so najbolj razširjeni tega ne naredijo, kot to dovoljuje "Push API". Ta vmesnik pravi, da morata tako brskalnik kot ponudnik potisnih obvestil, v primeru preklicane registracije, podatke povezane s tem takoj izbrisati. Zgodi se lahko, da se informacija o preklicu izgubi (na primer zaradi slabe internetne povezave) ali se brskalnik preneha uporabljati. Tako se lahko pri ponudniku potisnih storitev hranijo zelo stare in nepotrebne registracije na prejetje potisnih obvestil. Iz teh razlogov bi bilo smiselno, da bi brskalniki pri registraciji določili čas njenega trajanja.

Google je za svoj brskalnik Chrome napovedal, da bo začel s testiranjem omejenega časa trajanja registracij in sicer najprej za obdobje 90 dni. [5] [20]

"PushSubscription" objekt si lahko predstavljamo kot osebno izkaznico uporabnika.

Podatke za prijavo (PushSubscription objekt) pridobimo tako, da izbranemu ponudniku potisnih storitev pošljemo zahtevo za prijavo na potisna obvestila z uporabo HTTP POST metode. Ko nam ponudnik potisnih storitev pošlje PushSubscription objekt, se ta iz brskalnika pošlje strežniku, ki si ga mora shraniti, saj vsebuje vse podatke, ki jih bo strežnik sistema potreboval, da lahko sporočilo pošlje določenemu uporabniku. Način posredovanja "PushSubscription" objekta strežniku je v domeni razvijalcev spletne strani/aplikacije. Pomembno je zagotoviti visoko stopnjo varnosti, saj omenjeni objekt vsebuje občutljive podatke. Večinoma se za to uporabljata HTTPS POST metoda in privzeta vrata 443. [2]

### 2. Pošiljanje potisnih obvestil

Ko želimo iz strežnika uporabniku poslati potisno sporočilo, mora strežnik priklicati njegov "PushSubscription" objekt, ki si ga je predhodno shranil. Nato mora narediti HTTP POST klic k ponudniku potisnih obvestil, kjer smo uporabnika prijavi na njihovo prejetje potisnih sporočil. Ta klic mora vsebovati podatke, ki jih želimo prikazati prejemniku obvestila, naslovnika in ostale pogoje pod katerimi želimo sporočilo ali obvestilo poslati in prikazati. Ko ponudnik potisnih storitev prejme omrežno zahtevo, jo validira in nato, glede na podatke v zahtevi, pošlje potisno sporočilo izbranemu uporabniku. Brskalnik poskrbi, da je v ozadju vedno vzpostavljena povezava s ponudnikom potisnih storitev. Ta praviloma deluje s "socket-i" na TCPju.

V primeru, da brskalnik naslovnika ni povezan v splet, si ponudnik potisnih storitev sporočilo shrani v vrsto za pošiljanje, dokler se brskalnik ne poveže v splet oz. pride do izpolnitve katerega izmed pogojev zahteve. Ti so na primer čas življenja (TTL - Time to live) sporočila, prioriteta sporočila (če sporočilo nima visoke prioritete se ga, z namenom ohranjanja baterije odjemalca, lahko pošlje kasneje) in pogoj, ki določa ali se mora sporočilo prepisati oz. se združiti z ostalimi še ne dostavljenimi. Brskalnik se lahko poslužuje kateregakoli ponudnika potisnih storitev. Na to razvijalci nimajo vpliva. To pa ne predstavlja težave, saj je ta komunikacija standardizirana in zato vsi glavni ponudniki pričakujejo iste API klice.

Dobra praksa je, da so podatki, ki jih pošiljamo s potisnim sporočilom, kriptirani. Razlog za to je pravkar omenjeno dejstvo, da razvijalec brskalnika določi, katerega ponudnika potisnih storitev bo uporabljal in lahko se zgodi, da slednji ne izpolnjuje pričakovanih varnostnih zahtev.

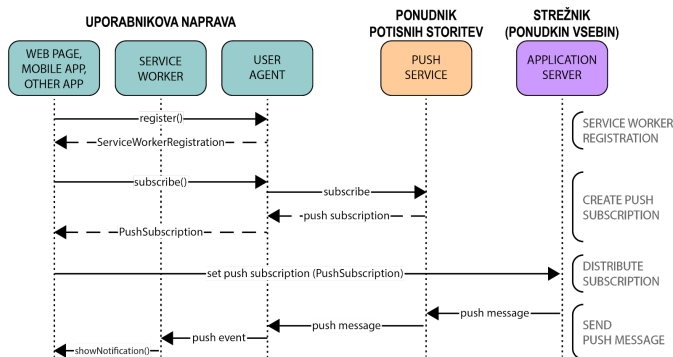
### 3. Obdelava podatkov in prikaz obvestila na uporabnikovi napravi

Ko so podatki potisnega sporočila dostavljeni na uporabnikovo napravo, bodo najprej dekriptirani in prestregel jih bo storitveni delavec ("service worker"). Slednji nato sproži potisni dogodek. Storitveni delavec je JavaScript datoteka, ki jo brskalnik lahko sam zažene, brez da bi bila povezana spletna stran aktivno odprta, oz. tudi brez da bi bil

sam brskalnik aktivno v uporabi. To je seveda ena od glavnih lastnosti in prednosti potisnih obvestil. V logiki storitvenega delavca se lahko opravi različna opravila v ozadju, kot so analitika in seveda tudi prikazovanje obvestil na ekranu. [15]

Slika 3 prikazuje sekvenčni diagram procesa prijave na prejetje potisnih obvestil in njihove dostave. Tak proces se izvede, ko uporabnik obišče spletno stran, ki mu želi pošiljati potisna obvestila. V nadaljevanju bom bolj podrobno opisal omenjeni sekvenčni diagram.

- **Service Worker Registration:** Obiskana spletna stran namesti na uporabnikovo napravo (v brskalnik) JavaScript datoteko, ki ima obliko storitvenega delavca, ki bo povezan samo s to eno prijavo na prejetje potisnih obvestil.
- **Create Push Subscription:** Uporabniku se prikaže prošnja za pošiljanje potisnih obvestil. Če uporabnik dovoli njihovo prejetje, brskalnik ponudniku potisnih storitev pošlje zahtevo za prijavo na prejetje potisnih obvestil. Ponudnik odgovori s podatki, ki so shranjeni v PushSubscription objektu in jih bo strežnik spletne strani potreboval za pošiljanje potisnih obvestil uporabniku.
- **Distribute Subscription:** Brskalnik preda PushSubscription objekt spletni strani, ki ga pošlje njenemu strežniku. Ta si objekt shrani.
- **Send Push Message:** Ko želi strežnik poslati potisno obvestilo, pošlje zahtevo ponudniku potisnih storitev. Ta nato posreduje podatke obvestila brskalniku, s katerim ima vedno vzpostavljeno povezavo. Slednji jih preda storitvenemu delavcu, ki je nato zadolžen za sprožitev prikaza obvestila uporabniku. [5]



Slika 3: Sekvenčni diagram delovanja potisnih obvestil. [5]

Zdaj pa si pogledajmo še Notification API, ki prejeto sporočilo dejansko prikaže končnemu uporabniku na ekran v obliki systemskega obvestila. Najprej je potrebno pridobiti dovoljenje uporabnika, da mu v prihodnosti lahko prikazujemo obvestila. Notification API ponuja priročno metodo, ki uporabniku v manjšem pojavnem oknu prikaže prošnjo za pošiljanje potisnih obvestil.

Samo prikazovanje obvestil lahko razdelimo na dva dela. Prvi poskrbi za način prikaza obvestila (izgled, zvok/vibracija, ipd.), drugi pa določa, kako se bo sistem odzval pri interakciji uporabnika s prikazanim obvestilom. Torej, ko uporabnikova

naprava prejme potisno sporočilo in ga dekriptira, lahko obvestilo prikažemo z metodo:

```
<ServiceWorkerRegistration>.showNotification(<title>, <options>);
```

Kjer je <title> naslov obvestila in <options> je JSON objekt, ki definira izgled in obnašanje obvestila, ki se bo prikazalo kot posledica prejetega potisnega sporočila. <options> objekt ima lahko naslednje pare ključ-vrednost:

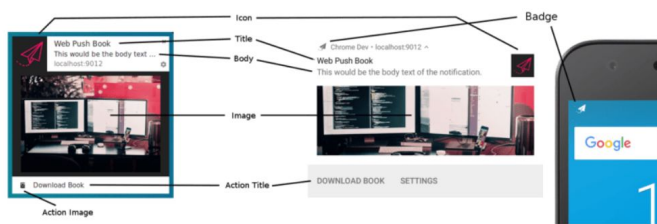
```
{
  // Visual Options
  "body": "<String>", // Tekst obvestila.
  "icon": "<URL String>", // Lokacija slike za ikono obvestila.
  "image": "<URL String>", // Lokacija večje slike, ki jo lahko prikažemo v obvestilu.
  "badge": "<URL String>", // Lokacija do manjše enobarvne slike, ki predstavlja pošiljatelja obvestila.
  "vibrate": "<Array of Integers>", // Vzorec načina vibriranja naprave.
  "sound": "<URL String>", // Lokacija zvočne datoteke, ki se predvaja ob prejemu obvestila.
  "dir": "<String of 'auto' | 'ltr' | 'rtl'>", // Smer pisanja in poravnava besedila.

  // Behavioral Options
  "tag": "<String>", // ID za združevanje vsebinsko podobnih obvestil.
  "data": "<Anything>", // Poljubni podatki, ki jih želimo poslati z obvestilom.
  "requireInteraction": "<boolean>", // Določa, ali mora obvestilo ostati prikazano, dokler ga uporabnik ne zapre ali klikne na gumba obvestila.
  "renotify": "<Boolean>", // Določa, ali bo novo obvestilo sprožilo vibriranje/zvok obvestila.
  "silent": "<Boolean>", // Določa, ali se bo novo obvestilo pojavilo brez vibracije, zvoka in vklopa ekrana na mobilnih napravah.

  // Both visual & behavioral options
  "actions": "<Array of Strings>", // Objekti za definiranje morebitnih "gumbov", ki omogočajo hiter odziv na prejeto obvestilo. Parametri takih objektov so "action" (ID ukaza, ki se mora izvesti), "title" (naziv gumba) in "icon" (URL lokacije slike za ikono gumba).

  // Information Option. No visual effect.
  "timestamp": "<Long>" // Datum generiranja obvestila na strežniku.
}
```

Slika 4 nam pomaga razumeti, kaj definira večino parov <options> objekta, ki vizualno definirajo obvestilo.



Slika 4: Prikaz obvestil in njihovih elementov na različnih napravah. [14] [7]

## V. LASTNOSTI DOBRIH POTISNIH OBVESTIL

Sodobnim sistemom lahko znatno povečamo dodano vrednost s tem, ko jim dodamo storitev potisnih obvestil, vendar je njihova pravilna uporaba lahko ključna za uspeh celega sistema. Verjetno si noben uporabnik ne želi prejemati obvestil, ki nimajo relevantnih podatkov, niso pravočasna ali so prepegosta. Sama implemetacija pošiljanja potisnih obvestil je danes relativno preprosta. Kljub temu pa zagotavljanje dobre uporabniške izkušnje, zahteva dobro poznavanje problema, ki ga želimo rešiti in potreb uporabnikov. V nadaljevanju si bomo pogledali, kakšne so lastnosti dobrih potisnih obvestil.

### A. Pravočasnost

Zagotoviti, da so potisna obvestila, ki jih uporabnik prejema pravočasna, je ključnega pomena, saj s tem zelo povečamo njihovo uporabnost. Poglejmo primer aplikacije z novicami, ki nudi možnost prejemanja potisnih obvestil za izredne novice. Če se obvestilo na uporabnikovi napravi pokaže uro ali več prepozno, je uporabnik novico lahko videl že drugje in je posledično v njegovih očeh aplikacija izgubila vrednost. Tako obvestilo postane moteče, saj ni več relevantno. Ali pa primer, ko uporabniku pošljemo nepomembno obvestilo sredi noči. Veliko je primerov, ki bi jih lahko našli, da bi nakazali, kako pomembna je pravočasnost potisnih obvestil. Seveda je lahko slednja v določenih primerih od nekaj sekund do nekaj minut kot tudi od nekaj ur do nekaj dni. To je odvisno od ponudbe aplikacije oz. potreb uporabnikov. Velja pa splošno pravilo, da se potisna obvestila uporabljajo za časovno občutljive dogodke. Dober primer takih dogodkov so prejeta sporočila v aplikaciji za klepet. Pogovori namreč potekajo v realnem času, saj so sinhrona vrsta komunikacije. Drug primer je koledar. Pri tej obliki časovno občutljivega dogodka želimo, da uporabnik v pravem trenutku vidi podatek in se nato lahko pravočasno odloči, kako bo ukrepal glede tega. [8]

### B. Relevantnost

Če želimo, da so potisna obvestila relevantna, naj prikazujejo podatke, ki so za uporabnike uporabni v nekem kontekstu. Če je možno, naj bodo ti podatki osebno povezani z njimi in z njihovimi potrebami in ne preveč splošni. Slednje dosežemo z vsebinami, ki so čim bolj ciljno usmerjene za vsakega uporabnika.

Nikakor ni dobro prikazati obvestila s podatki, ki jih ima uporabnik pravkar na zaslonu. Pred prikazom obvestila je torej

pomembno, da preverimo ali ga je primerno prikazati glede na to, kaj je v tistem trenutku na uporabnikovem zaslonu. Če želimo izpostaviti določene podatke, a se uporabnik že nahaja v aplikaciji/na spletni strani, moramo to storiti v uporabniškem vmesniku same aplikacije/spletne strani in ne s potisnimi obvestili. Kot primer pomislimo na aplikacijo trgovine, ki je izdala in objavila nov digitalni katalog izdelkov. Če si uporabnik pravkar ogleduje katalog v aplikaciji, mu seveda ne smemo pokazati obvestila, da je bil ta katalog na novo objavljen in ga povabiti, da si ga ogleda. V kolikor se uporabnik nahaja v aplikaciji, ga lahko obvestimo o novem katalogu tako, da na primer razdelek s katalogi obarvamo z drugo barvo ali mu dodamo značko ("badge") s številko 1, kar pomeni, da je bil objavljen en nov katalog, ki je na voljo za ogled. Če pa si katalog pravkar ogleduje, pa obvestilo o tem ni ustrezno v nobeni obliki.

### C. Pogostost potisnega obveščanja

Ker imajo potisna obvestila tako velik potencial pri doseganju uporabnikov, se nam hitro lahko zgodi, da jih želimo pošiljati prepegosto. Četudi smo mnenja, da ponujamo dovolj kvalitetne in pravočasne vsebine, to ni ustrezno. Za uporabnika lahko postane preveč moteče, kar poveča odjave od prejemanja potisnih obvestil in celo izbris aplikacije. Težko je točno določiti število potisnih obvestil, ki jih lahko pošljemo uporabnikom, saj je slednje odvisno od panoge, v kateri deluje pošiljatelj obvestil, želje in potrebe uporabnikov ter drugih dejavnikov, ki jih je potrebno dobro proučiti.

### D. Natančnost

Vsebina potisnih obvestil naj bo natančna in specifična. Dobro je, da se lahko uporabnik samo na podlagi vsebine prejetega obvestila odloči, kako želi, glede na prejeta informacijo ukrepati, ne da bi moral poiskati dodatne informacije na spletni strani ali aplikaciji, ki je obvestilo poslala. Seveda se velikokrat zgodi, da ne moremo vseh pomembnih podatkov umestiti v tako omejen prostor, vendar mora biti to naša težnja. V primeru, da ena izmed trgovin s potisnim obvestilom obvesti svoje uporabnike, da je izšel nov katalog izdelkov, nas mora klik na obvestilo takoj pripeljati na omenjeni katalog in ne, na primer, na vstopno stran spletne strani.

Splošno velja, da morajo biti obvestila kratka in jasna. Uporabniki smo čez dan zelo obremenjeni z različnimi obvestili v taki ali drugačni obliki. Zato mora vedno biti naš cilj, da to obremenitev zmanjšamo v čim večji meri. V tem primeru velja slediti naslednjim napotkom, če želimo ponuditi kvalitetna obvestila, ki so vredna uporabnikovega časa:

- naslov najprej pritegne uporabnikovo pozornost, zato mora biti specifičen;
- najpomembnejši podatki obvestila morajo biti postavljeni na vrh in poravnani na levo;
- poudariti je potrebno, kaj želimo, da uporabnik stori, ko obvestilo prebere;
- če je možno, obvestilo opremimo z ikono, podobna obvestila pa smiselno prikazujemo z istimi ikonami, tako da lahko uporabnik hitreje razume, za kakšno vrsto obvestila gre;
- obvestilom dodamo slike;

- obvestilom dodamo gumbe za hitre nadaljne akcije povezane s prejetimi podatki obvestila;
- itd.

## VI. DOBRE PRAKSE PRI IMPLEMENTACIJI POTISNIH OBVESTIL

V nadaljevanju sem predstavil nekaj pomembnih implementacijskih napotkov, ki pripomorejo k ponujanju kvalitetnejših potisnih obvestil.

### A. Dovoljenje za pošiljanje potisnih obvestil

Ena izmed najbolj pomembnih funkcionalnosti potisnih obvestil je tista, ki uporabniku omogoča, da dovoli prejemanje takih obvestil. To je seveda tudi edini pravi način, da pristopimo k pošiljanju slednjih. V nasprotnem primeru je zelo velika verjetnost, da bo uporabnik preklical prejemanje obvestil in ponavadi tudi zapustil spletno stran ali odstranil aplikacijo, ki ta sporočila pošilja. Pri tej funkcionalnosti se pojavi vprašanje, kdaj je pravi trenutek, da uporabniku ponudimo možnost prejemanje potisnih obvestil. Če to storimo prehitro, bo imel premalo informacij, da se pravilno odloči. Če pa to storimo prepozno, ko uporabnik naše storitve uporablja zelo malo ali sploh ne, pa bomo izgubili možnost, da mu ponudimo storitev prejemanja potisnih obvestil in s tem tudi zelo močno orodje, da ga znova in znova privabljam na našo spletno stran ali aplikacijo.

- Ko smo uporabnika prvič vprašali za dovoljenje in ga je ta odklonil, je dobra praksa, da ga o tem več ne sprašujemo. Torej je tak uporabnik iz tega vidika izgubljen. Če pa uporabniki pridobijo informacije, ki jim omogočajo, da razumejo koristi in prednosti potisnih obvestil, ki jim jih želimo pošiljati, pa bo seveda veliko večji odstotek takih, ki bodo v to privolili.
- Poglejmo si nekaj primerov situacij, v katerih je primerno uporabnika zaprositi za dovoljenje za pošiljanje potisnih obvestil:
- ko uporabnik ureja del nastavitve za komunikacijo z aplikacijo ali spletno stranjo, se lahko potisna obvestila ponudijo kot ena od možnosti za obveščanje (seveda ob primerni razlagi koristi takega načina obveščanja);
- ko uporabnik izvede aktivnost, ki bo pravilno delovala samo, če bo storitev imela možnost dostavljanja potisnih obvestil. Na primer, ko v koledarju nastavimo nov dogodek, o katerem želimo biti pravočasno obveščeni. Ali pa, ko na spletni strani opravimo nakup. V tem primeru lahko uporabnika vprašamo ali želi biti s potisnimi obvestili obveščen o odpremi izdelka, ipd.;
- ko se uporabnik pogosto vrača k naši storitvi, mu lahko ponudimo možnost prejemanja potisnih obvestil. Tak uporabnik je očitno pokazal svojo naklonjenost naši storitvi, verjetno jo je tudi bolje spoznal in se bo zato lažje odločil za njihovo prejemanje;
- uporabnike lahko povabimo k prejemanju potisnih obvestil, tako da jim ob tem ponudimo kakšno dodatno ugodnost (na primer popust ob nakupu

naslednjega izdelka), vendar jih moramo v vsakem primeru dobro seznaniti o prednostih prejemanja takih obvestil;

- itd.

### B. Upravljanje s številom potisnih obvestil

Glede na to kako učinkovita so potisna sporočila za doseganja uporabnikov, se hitro lahko zgodi, da se njihovo uporabo zlorabi v smislu, da se uporabnikom pošilja preveliko število obvestil. Seveda ne trdim, da v nobeni situaciji ne smemo poslati veliko obvestil, ampak želim povedati, da moramo biti zelo pozorni, da so vsa sporočila relevantna in obvladljiva za uporabnika, ter da jih želi prejemati v takem številu. Če imamo storitev, ki mora za pravilno delovanje uporabniku pošiljati veliko obvestil (kot je na primer storitev za klepet), potem je ključno, da se poslužujemo dobre prakse združevanja več vsebinsko podobnih v eno skupno obvestilo. Dober primer tega je storitev klepeta, kjer lahko skupek obvestil istega pošiljatelja združimo v eno samo.

### C. Odstranitev nerelevantnih obvestil

Ko kliknemo na neko obvestilo, je dobra praksa, odstraniti tudi vsa ostala obvestila istega pošiljatelja, ki so povezana med seboj in jih uporabnik še ni pregledal. Praviloma mora imeti uporabnik možnost pregledati vse vsebine, ki so bile na odstranjenih obvestilih na spletni strani ali v aplikaciji, v katero ga je klik obvestila pripeljal.

Če neko obvestilo postane nerelevantno po tem, ko je bilo dostavljeno na uporabnikovo napravo in ga ta še ni prebral, ga seveda tudi moramo odstraniti. Obvestila, ki za uporabnika nimajo nobene dodane vrednosti, so seveda nepotrebna.

### D. Upravljanje z obvestili na strežniku

Zgodi se lahko, da naprava uporabnika, ki mu želimo poslati potisno obvestilo, ni povezana v splet. Obvestilo, ki mu ga želimo poslati, pa je relevantno samo, če ga uporabnik prejme v roku določenega časa. Tak primer je storitev za obveščanje o izrednih novicah. Če uporabnik tako novico prejme dan kasneje, je ta skoraj sigurno nerelevantna zanj. Pri implementaciji potisnih obvestil moramo tako predvideti tudi takšno situacijo. Ta problem lahko rešimo na več načinov. Eden izmed njih je ta, da potisno obvestilo opremimo z informacijo o tem, do kdaj je sporočilo, ki ga obvestilo nosi, še relevantno oz. pod katerimi pogoji ga je še smiselno prikazovati. Storitve pošiljanja potisnih obvestil omogoča tudi vstavev vrednosti "čas življenja obvestila" (Time to Live - TTL), v podatke, ki sestavljajo obvestilo. S to funkcijo lahko definiramo čas, v katerem je še smiselno, da se obvestilo pošlje uporabniku. Če ponudnik potisnih storitev slednjemu ne more dostaviti obvestila do TTL časa, se podatki tega obvestila izbrišejo. V veliko primerih že vnaprej vemo, koliko časa je za nekega uporabnika informacija potisnega obvestila relevantna, zato se je te funkcionalnosti dobro posluževati v čim večji meri.

### E. Upravljanje s podvojenimi obvestili

Sodobni sistemi lahko uporabniku nudijo možnost, da isto storitev sočasno uporablja na različnih napravah. Na primer v aplikaciji na mobilni napravi in na spletni strani na

računalniku. V takih primerih se lahko pojavi podvojevanje potisnih obvestil. Za lažje razumevanje te problematike si pogledimo primer storitve pošiljanja in prejemanja elektronskih sporočil (e-mail), ki jo lahko uporabljamo tako na spletu kot v mobilni aplikaciji. Ko prejmemo elektronsko sporočilo, nam sistem obvestilo o tem dogodku pošlje na spletno stran in na aplikacijo na mobilni napravi. Če uporabnikova mobilna naprava ni povezana v splet, si lahko prejeto sporočilo prebere na spletni strani. Ko mobilno napravo ponovno poveže v splet, se na slednji pojavi obvestilo o prejetem elektronskem sporočilu, ki ga je že prebral na spletni strani na namiznem računalniku. Seveda si ne želimo, da se morajo uporabniki z istim obvestilom ukvarjati več kot enkrat. Zato moramo poskrbeti, da je tudi v takih situacijah uporabniška izkušnja primerna. V omenjenem primeru bi bilo najbolje, da se v aplikaciji na uporabnikovi napravi, preden se obvestilo generira in prikaže, preveri status sporočil (prebrano, neprebrano, itd.) na oddaljenem strežniku, ki hrani elektronska sporočila. Če je uporabnik sporočilo že videl, potem mu ga aplikacija ne sme prikazati. Iz tega primera lahko razumemo tudi, kako pomembna je logika na uporabnikovi napravi pri oceni relevantnosti posameznega potisnega obvestila, z namenom zagotovitve višje stopnje uporabnosti storitve. [7]

## VII. PREDNOSTI IN SLABOSTI POTISNIH OBVESTIL

Kot vsaka storitev, imajo tudi potisna obvestila prednosti in slabosti. Potrebno je dobro poznavanje problema, ki ga želimo rešiti in ugotoviti, ali prednosti potisnih obvestil ustvarijo dovolj veliko dodano vrednost, ki opravičuje integracijo te tehnologije v naš sistem. Obstaja nekaj tehničnih slabosti, ki jih imajo potisna obvestila, ampak je vseeno bistveno več prednosti, v kolikor se sledi vsem dobrim praksam za njihovo pravilno uporabo in implementacijo.

V nadaljevanju bom predstavil nekaj njihovih glavnih prednosti in slabosti.

### PREDNOSTI

#### A. Učinkovito obveščanje

Potisna obvestila nudijo zelo učinkovit način obveščanja uporabnikov o časovno občutljivih vsebinah, kjer ima pravočasnost pomembno vlogo. Primerna so tudi za relativno pogosto obveščanje uporabnika, saj od njega ne zahtevajo veliko pozornosti pri njihovem spremljanju. Potisna obvestila tudi niso preveč moteča, če se upoštevajo dobre prakse njihove uporabe in implementacije. Seveda, pa je potrebno poudariti, da vsebine takih sporočil praviloma ne smejo biti najvišje prioritete. Za njihovo pravilno delovanje mora biti naprava namreč povezana v splet. Poleg tega pa smo tudi uporabniki razvili neko vrsto neobčutljivosti nanje in jim nemalokrat namenimo pozornost šele nekaj časa po samem prejetju obvestila na napravi.

#### B. Ohranjanje višje stopnje zvestobe med uporabniki

Želimo si, da ima naša storitev čim bolj visoko stopnjo zvestobe med uporabniki. To pomeni, da slednji redno uporabljajo naše storitve. Ko opazimo, da jo nek uporabnik v določenem obdobju majnkart obišče, lahko s pomočjo potisnih obvestil pritegnemo njegovo pozornost. To je temeljni korak,

če želimo našo storitev znova izpostaviti v morju ostalih, s katerimi smo vsi preplavljeni. Seveda, če za uporabnika naša storitev ni uporabna, se bomo težko izognili temu, da ga izgubimo.

#### C. Večja uporaba storitve

Potisna obvestila so lahko zelo učinkovita pri povečanju same uporabe storitve. Če uspemo uporabnika prepričati, da dovoli prejemanje potisnih obvestil, smo si omogočili zelo močno orodje za povečanje uporabe ponujene storitve. Potisna obvestila so namreč zelo dober kanal za učinkovito doseganje uporabnikov. Vsaka dodatna interakcija slednjega z našo storitvijo nezanemarljivo poveča verjetnost konverzije. Torej se, če zagotovimo visoko stopnjo zvestobe in istočasno tudi dejavnosti med uporabniki, možnost doseganja ciljev naše storitve (na primer prodaja izdelkov) zelo zviša.

#### D. Segmentiranje uporabnikov

Potisna obvestila in orodja povezana z njimi nam omogočajo možnost zbiranja podatkov za segmentacijo uporabnikov. Vsekakor je ključno, da slednje segmentiramo v različne skupine glede na lokacijo, vedenje, nakupovalne vzorce in navade, demografske lastnosti, itd. Vse te podatke lahko pridobimo tudi, ko uporabniki uporabljajo potisna obvestila. Z upoštevanjem tega je lahko pošiljanje relevantnih potisnih obvestil veliko bolj ciljno usmerjeno in posledično tudi bolj učinkovito. Pri uporabnikih lahko z zelo prilagojenimi in ciljno usmerjenimi potisnimi obvestili ustvarimo tudi občutek direktne komunikacije med ponudnikom storitve oz. podjetjem in stranko. To izboljša uporabniško izkušnjo in ustvari boljše pogoje pri direktni (prodaja izdelka, storitve, ipd.) ali indirektni prodaji (podajanje informacij o izdelku, ki ga prodajamo). Velja omeniti, da je segmentiranje uporabnikov s pomočjo potisnih obvestil veliko bolj učinkovito na mobilnih aplikacijah, saj lahko preko njih dostopamo do večjega števila podatkov o uporabniku, kot so na primer zelo natančni podatki o lokaciji, merilcu pospeška, kompasu, seznamu kontaktov, kameri, itd. Seveda moramo za dovoljenja, za dostopanje do njih, zaprositi uporabnike pred njihovo uporabo.

#### E. Povečanje stopnje konverzije

Večina informacijskih storitev (aplikacije, spletne strani) ima cilj, da svojim uporabnikom nekaj proda. Pomemben podatek, ki ga ponudniki teh storitev spremljajo, je stopnja konverzije, ki v tem primeru velikokrat pomeni razmerje med številom ogledov/uporabe/obiska spletne strani ali aplikacije s strani uporabnika (stranke) in številom opravljenih nakupov. Izkušnje kažejo, da uporaba potisnih obvestil zelo poveča stopnjo konverzije. Orodja, ki jih ponujajo ponudniki potisnih storitev, ponavadi omogočajo zbiranje in pregledovanje statističnih podatkov povezanih s potisnimi obvestili, kot so razmerje med prikazi in kliki na potisno obvestilo ali odstranitev obvestila brez ogleda njegove vsebine in pa stopnja konverzije.

#### F. Pomoč uporabnikom

Potisna obvestila lahko uporabimo tudi za namene vodenja in pomoči uporabnikom pri različnih aktivnostih, ki jih lahko izvajajo v aplikaciji ali na spletni strani. Če spremljamo uporabnikovo obnašanje v nekem informacijskem sistemu,



lahko s tem ugotovimo tudi, kdaj ima težave pri doseganju svojih ciljev. S potisnimi obvestili mu lahko nato ponudimo pomoč ali ga usmerjamo. Uporabnike lahko tudi usmerimo na določene dele naše spletne strani ali aplikacije, ki dodatno povečajo uporabnost storitve in uporabniško izkušnjo. [9]

### G. Cena

Zelo pomembna prednost uporabe tehnologije potisnih obvestil je tudi cenovna ugodnost njihovega pošiljanja.

## SLABOSTI

### A. Omejeno število znakov

Potisna obvestila so praviloma krajše narave in ne predvidevajo veliko prostora za daljša besedila. To je lahko v nekaterih primerih slabost. V slednjih tako izbira potisnih obvestil kot kanal obveščanja ni najbolj primerna. Poslužujemo se jih predvsem takrat, ko želimo uporabnike opozoriti o nečem, kar zahteva njihovo pozornost. V sami vsebini obvestila mu na kratko, vendar čim bolj jasno, nakažemo o čem gre in ga povabimo, da si podrobno pogleda v naši aplikaciji ali na spletni strani.

### B. Poraba baterije

Vsaka komunikacija med (oddaljenima) napravama, četudi kratka, porabi nekaj električne energije. Pri tistih, ki so vedno priklopljene na električno omrežje (npr. namizni računalniki), je ta problem manjši, medtem ko je pri mobilnih napravah potrebno upoštevati tudi ta faktor. Sprejemanje velike količine potisnih obvestil namreč ni energetskega zanemarljivo. Sodobni sistemi so se morali prilagoditi veliki količini potisnih sporočil (med katerimi so tudi obvestila). Najbolj pogosta rešitev, ki se je poslužujejo danes, je centralizirano sprejemanje potisnih sporočil in obvestil na nivoju naprave. To pomeni, da je na vsaki napravi samo ena storitev, ki ima nalogo, da vsem aplikacijam in programom na napravi posreduje potisna sporočila, ki jih prejme od ostalih naprav (oddaljenih strežnikov, ipd.). Na tak način se poraba energija zmanjša na minimum, a je možnost sprejemanja potisnih sporočil s strani naprave povezane v splet kljub temu neprekinjena. Tako logiko delovanja imajo tudi zelo pogosto uporabljene storitve Firebase in Apple Push Notification Service.

### C. Poseganje v uporabnikov osebni prostor

Dobre prakse narekujejo, da damo uporabniku vedno možnost, da se sam odloči ali želi prejeti potisna obvestila in da lahko v kateremkoli trenutku to prekine na preprost način. Vseeno pa storitev pošiljanja potisnih obvestil razvijalcem omogoča, da do neke mere ne upoštevajo želj uporabnikov in jim tako obvestila vseeno pošiljajo. V takih primerih lahko uporabnik občuti, kot da se posega v njegov osebni prostor, kar je seveda nedopustno. Tudi v primerih, ko storitvi dovoli, da mu dostavlja potisna obvestila, velikokrat sam ne more izbrati njihovega števila. Sodobni pametni sistemi omogočajo, da se sprejemanje potisnih obvestil onemogoči na nivoju operacijskega sistema naprave za vse ali le za posamezne aplikacije in programe. Toda uporabnik, ki ni več njihove uporabe, se lahko kaj hitro počuti nemočen pri obvladovanju vseh teh informacij in to ga lahko odvrne od nadaljnje uporabe. [10]

## VIII. GLAVNI PONUDNIKI POTISNIH OBVESTIL

V tem poglavju bom predstavil nekaj glavnih ponudnikov potisnih storitev. V zadnjih letih se je uporaba potisnih sporočil/obvestil zelo razširila in posledično se je povečalo tudi število ponudnikov potisnih storitev.

V smislu načina delovanja pošiljanja potisnih obvestil lahko rečemo, da večina glavnih ponudnikov temelji na protokolu "Web Push Protocol". Ta definira način komunikacije med akterji pri obveščanju s potisnimi storitvami. To so strežnik (spletne strani/aplikacije ali mobilne aplikacije) nekega podjetja, ponudniki potisnih storitev in uporabnikove naprave. Omenjeni protokol se uporablja predvsem za spletne strani in ne za mobilne aplikacije. [5] [21] Za pošiljanje potisnih obvestil na mobilne naprave z operacijskima sistemoma Android in iOS, pa se vedno uporablja lastniške rešitve podjetij Google in Apple. Pri Androidu se uporablja Firebase Cloud Messaging (FCM), pri Apple napravah pa Apple Push Notifications service (APNs). To pomeni, da bodo tudi v primeru uporabe drugih ponudnikov potisnih storitev, kot so OneSignal ali Airship, tej za pošiljanje potisnih podatkov na omenjene naprave uporabljali FCM ali APNs. To večinoma ne predstavlja velike težave, saj se po začetni nastavitvi razvijalcem s tem ni potrebno več veliko ukvarjati. Tak način delovanja je praviloma velika prednost. Če pride namreč do spremembe delovanja pri FCM ali APNs, ponudnik potisnih storitev sam poskrbi za potrebne implementacijske spremembe pri sami dostavi obvestil. [22] Glavna prednost uporabe storitev, ki niso FCM ali APNs, je v tem, da lahko nudijo možnost hkratnega pošiljanja obvestil na obe vrsti mobilnih operacijskih sistemov (Android, iOS). To se izvaja preko prilagojenih in enostavnih vmesnikov, ki so prijaznejši do uporabnika. Taki vmesniki zelo olajšajo uporabo vseh orodij, ki jih je možno uporabljati v povezavi s potisnimi obvestili. To so orodja za pripravo vsebine obvestila, za analitiko, za izboljšanje uporabniške izkušnje, za optimalno pošiljanje obvestil (čas pošiljanja, izgled obvestila, ipd.) in enostavne knjižnice za delo z različnimi tehnologijami, itd. Podjetja, kot so OneSignal in Airship, veliko truda vložijo tudi v pisanje dobre dokumentacije. Na tak način povečajo uporabnost njihove storitve med svojimi strankami. Slednje je za njih ključno, ker so, za razliko od Googla in Appla, to njihovi glavni prodajni produkti. [23]

Ponudnike, ki bodo predstavljeni v nadaljevanju, sem izbral po moji manjši raziskavi člankov o tem, kateri so najbolj uporabljani na dan pisanja tega članka.

### A. Firebase Cloud Messaging (FCM)

Firestore Cloud Messaging ponuja zanesljivo storitev, ki je energetske učinkovita in je na dan pisanja tega članka brezplačna. Omogoča tako pošiljanje, kot sprejemanje potisnih sporočil na in iz naprav povezanih v splet. Uporablja se predvsem z operacijskimi sistemi Android in iOS ali brskalnikih, kot je Chrome. Sporočila lahko vsebujejo do 4 kB podatkov. Ključne funkcionalnosti te storitve so:

- pošiljanje potisnih obvestil ali podatkovnih sporočil,

- ciljno pošiljanje sporočil z možnostjo ciljanja izbranih naprav, skupin naprav ali naprav, ki so se prijavile na neko vsebino in
- pošiljanje sporočil iz naprave na ostale naprave.

Firestore je del velike skupine produktov pod skupnim imenom Firebase, ki ga je leta 2014 prevzelo podjetje Google. Ostale pomembne storitve družine Firebase, ki se med seboj zelo dobro integrirajo, so:

- Cloud Firestore: podatkovna baza v oblaku;
- Hosting: zanesljivo in z zelo nizko zakasnitvijo ("latency") spletno gostovanje spletnih aplikacij z brezplačnim SSL certifikatom;
- Authentication: ponuja enostavno možnost za avtentikacijo uporabnikov s pomočjo Google ali Facebook prijave kot tudi prijave s katerikoli elektronskim naslovom;
- Analytics: močno orodje za analizo uporabe in delovanja spletnih ali mobilnih rešitev;
- itd.

[16]

### B. Apple Push Notification Service (APNs)

Apple push notification service je centralizirana storitev za pošiljanje potisnih obvestil v oblaku, ki jo je leta 2009 začelo ponujati podjetje Apple. To je tako postalo prvi večji ponudnik potisnih obvestil za pametne mobilne telefone. APNs je namenjen pošiljanju takih obvestil mobilnim aplikacijam in ostalim programom, ki temeljijo na operacijskih sistemih podjetja Apple. Za slednje je APNs tudi edina možna rešitev, če želimo sistem nadgraditi z obveščanjem s potisnimi obvestili. Tudi Firebase cloud messaging, ki ponuja možnost pošiljanja obvestil na Appleove naprave, se mora v vsakem primeru posluževati APNs-ja za uspešno delovanje potisnega obveščanja. [17]

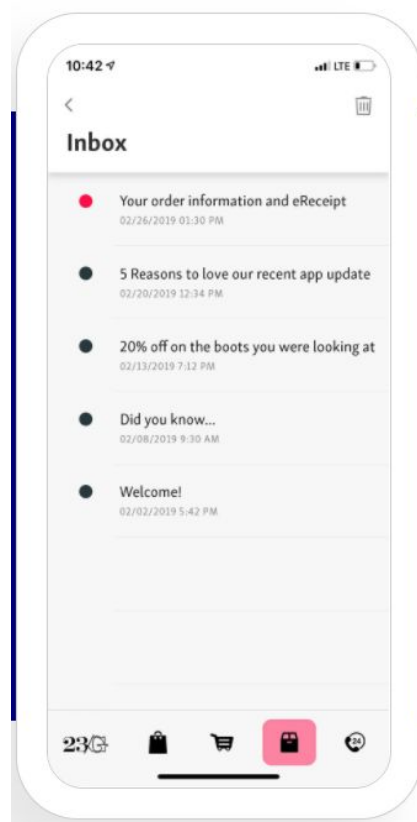
### C. OneSignal

OneSignal je podjetje, ki ponuja tako spletne kot mobilne potisne storitve. Trdijo, da so vodilni na trgu na področju zvišanja uporabe storitve med uporabniki ("customer engagement"), saj ponujajo programsko razvojno orodje, ki ga uporablja največ razvijalcev. Poleg potisnih storitev ponujajo tudi orodja za email marketing in interno obveščanje v aplikaciji. Med glavnimi funkcionalnostmi, ki jih ponujajo, so tudi segmentiranje uporabnikov, A/B testiranje, poročanje v živo in storitev pametne dostave sporočil, ki optimizira čas njihove dostave glede na navade uporabnikov. [18]

### D. Airship

Airship, ki se je do nedavno imenoval Urban Airship, izpostavlja, da je platforma številka ena za povečanje uporabe storitve med uporabniki namenjene predvsem uveljavljanju znamka podjetji. Poleg potisnih obvestil ponuja tudi SMS in email obveščanje. Pri Airshipu je zanimiv njihov inovativen pristop za nemoteče obveščanje zvestih uporabnikov preko mobilnih aplikacij. Prav slednji pa so tisti, ki podjetju praviloma prinašajo največji zaslužek. To storitev so poimenovali "Message center" in je namenjena hrambi vseh pomembnih obvestil, ki jih uporabnik prejme v povezavi z

določeno aplikacijo nekega ponudnika, vendar jih ne vsiljuje na način, kot se to dogaja s potisnimi obvestili. Njena največja prednost pred potisnimi obvestili je, da ni potrebno pridobiti dovoljenja za prejetje takih obvestil, saj so zelo nemoteča. Velika prednost te rešitve je tudi ta, da so obvestila še vedno na voljo tudi po tem, ko smo si jih enkrat že ogledali. Ponudnik lahko sam določi katera niso več koristna in jih odstrani tako, da se uporabnikova pozornost ne "razprši" preveč. Po moji oceni je slabost te rešitve ta, da so taka obvestila manj opazna (in zato so tudi manj moteča) in je posledično uporaba same aplikacije/spletne strani med uporabniki manjša.



Slika 5: Primer Airship Message centra v katerem so vidno vsa prejeta obvestila. [19]

## IX. ZAKLJUČEK

Ta dokument nudi pregled načina obveščanja z uporabo potisnih obvestil v sodobnih informacijskih sistemih. S tako veliko količino informacij, ki so nam na voljo danes, ocenjujem, da je lahko uporaba potisnih tehnologij za namen njihovega posredovanja zelo primerna. Med pomembnejšimi lastnostmi "push" tehnologije, ki omogoča potisna obvestila, je prejetje podatkov v realnem času. To je tudi glavna razlika s "pull" tehnologijo. V določenih primerih je lahko "pull" metoda energijsko varčnejša [25], toda ta faktor igra manjšo vlogo, ker je vse bolj pomembna tudi pravočasnost ponujenih podatkov. Porabo električne energije se tako skuša minimizirati z uporabo centraliziranih storitev za prejetje potisnih sporočil.

Vsi glavni akterji, kot so Google, Apple, Microsoft, itd., so spoznali uporabnost te tehnologije in omogočili njeno masovno rabo. Ker se s storitvami, ki nam jih ponujajo omenjena podjetja, srečujemo zelo pogosto, je poznavanje predstavljenega načina obveščanja lahko uporabno. Spoznanja v tem dokumentu so lahko zelo dobrodošla tudi vsem, ki se ukvarjajo z implementacijo te storitve. Spoznajo lahko njene začetke, kot tudi najboljše prakse njene uporabe, ki so ključne za to, da se uporabnikom ponudi kvalitetnejšo storitev obveščanja. S temi spoznanji pridobijo širši pogled na to področje in lažje razumejo, na kaj je potrebno biti pozoren že od samega začetka ter v katero smer želijo usmeriti svojo programsko rešitev. Ugotovijo lahko tudi, ali je opisana rešitev sploh primerna za problem, ki ga želijo rešiti.

Seznanijo se lahko s tem, da obstaja več ponudnikov potisnih storitev. Slednji se med seboj razlikujejo predvsem na podlagi ponudbe orodij za pripravo, pošiljanje in analizo potisnih obvestil ter interakcije z njimi. Na področju zanesljivosti, varnosti in tehnologij, ki podpirajo delovanje potisnih storitev v ozadju, pa so si zelo podobni.

Ocenjujem, da so potisna obvestila še vedno novost na področju sistemov za obveščanje uporabnikov. Kljub temu, da se je v smeri preprečevanja njihove zlorabe že nekaj storilo, pa se za zdaj še dogaja, da prihaja do tega. To bi lahko privedlo do povečanja uporabnikovega odpora do njih, kar bi lahko na dolgi rok celo ogrozilo njihov obstoj. Nekatere raziskave namreč nakazujejo na to, da lahko prepogosto prekinjanje uporabnika pri njegovem delu s potisnimi obvestili, pripomore k zmanjšanju stopnje njegove pozornosti [26]. Potrebni bo še veliko raziskav, ki bodo pokazale, kako je najbolje integrirati tak način obveščanja z današnjim stilom življenja. Na tak način bi lahko povečali verjetnost uporabe potisnih obvestil tudi v prihodnosti, saj je njihova uporabnost nesporna.

#### VIRI

- [1] "Pull technology", Wikipedia, 13. februar 2020, URL: [https://en.wikipedia.org/wiki/Pull\\_technology](https://en.wikipedia.org/wiki/Pull_technology)
- [2] "Generic Event Delivery Using HTTP Push draft-ietf-webpush-protocol-12", M. Thomson (Mozilla), E. Damaggio, B. Raymor, Ed. (Microsoft), 22. oktober 2016, URL: <https://tools.ietf.org/html/draft-ietf-webpush-protocol-12>
- [3] "APNs Overview", Apple Inc., 4. junij 2018, URL: <https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html>
- [4] "Apple Push Notification service", Wikipedia, 21. september 2020, URL: [https://en.wikipedia.org/wiki/Apple\\_Push\\_Notification\\_service](https://en.wikipedia.org/wiki/Apple_Push_Notification_service)
- [5] "Push API", World Wide Web Consortium (W3C), 4. februar 2020, URL: <https://www.w3.org/TR/push-api>
- [6] "Android Cloud to Device Messaging (C2DM) - Tutorial", Lars Vogel (Vogella GmbH), 25. oktober 2016, URL: <https://www.vogella.com/tutorials/GoogleCloudMessaging/article.html>
- [7] "Introduction to Push Notifications", Google Inc., 1. maj 2019, URL: <https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications>
- [8] "Push Notifications Tutorial: Top Tips for Better Push Notifications", Emma Mullan, 16. september 2016, URL: <https://blog.hurree.co/blog/push-notifications-tutorial>
- [9] "What is a Push Notification? - Definition and Types", SendPulse, 12. februar 2020, URL: <https://sendpulse.com/support/glossary/push-notification>
- [10] "Push Notifications vs. SMS vs. Email For Connecting with Your Customers", Dinarys GmbH, 16. marec 2020, URL: <https://dinarys.com/blog/pushnotificationssmssemail>
- [11] "Push API", Can I Use, URL: <https://caniuse.com/push-api>
- [12] "Web Push Notifications: Timely, Relevant, and Precise", Joseph Medley, 12. februar 2019, URL: <https://developers.google.com/web/fundamentals/push-notifications>
- [13] "Push API", Mozilla, 16. april 2020, URL: [https://developer.mozilla.org/en-US/docs/Web/API/Push\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Push_API)
- [14] "Notifications API", Mozilla, 23. julij 2020, URL: [https://developer.mozilla.org/en-US/docs/Web/API/Notifications\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API)
- [15] "How Push Works", Matt Gaunt, 4. junij 2019, URL: <https://developers.google.com/web/fundamentals/push-notifications/how-push-works>
- [16] "Firebase Cloud Messaging", Google Inc., 27. oktober 2020, URL: <https://firebase.google.com/docs/cloud-messaging>
- [17] "APNs Overview", Apple Inc., 4. junij 2018, URL: [https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple\\_ref/doc/uid/TP40008194-CH8-SW1](https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW1)
- [18] "Predstavitvena stran", OneSignal, URL: <https://onesignal.co>
- [19] "Message Center", Airship, URL: <https://www.airship.com/platform/channels/mobile-app/message-center/>
- [20] "Google Chrome shipping new feature that limits push notification subscriptions to 90 days", avgust 2020, URL: <https://www.thewindowsclub.com/google-chrome-shipping-new-feature-that-limits-push-notification-subscriptions-to-90-days>

- [21] “Web Push Interoperability Wins”, Matt Gaunt, Joseph Medley, 14. januar 2019, URL:  
<https://developers.google.com/web/updates/2016/07/web-push-interop-wins>
- [22] “User Notifications”, Apple Inc., 2020, URL:  
<https://developer.apple.com/documentation/usernotifications>
- [23] “Firebase Cloud Messaging (FCM) Compared to OneSignal”, George Deglin, 12. maj 2020, URL:  
<https://onesignal.com/blog/firebase-vs-onesignal>
- [24] Na Li, Yanhui Du, Guangxuan Chen, “Survey of Cloud Messaging Push Notification Service”, 2013 International Conference on Information Science and Cloud Computing Companion
- [25] Daniel Burgstahler, Ulrich Lampe, Nils Richerzhagen, Ralf Steinmetz, “Push vs. Pull: An Energy Perspective”, 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications
- [26] Kostadin Kushlev, Jason Proulx, Elizabeth W. Dunn, “Silence Your Phones”: Smartphone Notifications Increase Inattention and Hyperactivity Symptoms, maj 2016