



OpenHAB

Profesor:
Dr Dragan Stojanović



Student:
Stefan Marinković



OpenHAB

Open Home Automation Bus

01

Sposobnost

Da integriše mnoštvo drugih uređaja i sistema. OpenHAB uključuje druge sisteme za automatizaciju kuće, (pametne) uređaje i druge tehnologije u jedno rešenje.

02

Obezbedi

Jedinstven korisnički interfejs i zajednički pristup pravilima automatizacije u celom sistemu, bez obzira na broj proizvođača i pod-sistema koji su uključeni.

03

Najfleksibilniji

Dostupan alat za ostvarenje gotovo svake želje za automatizacijom kuće. Sve što možete da zamislite, možete i da ostvarite sa OpenHAB-om.

Prednosti

Istorija

2010

Početak projekta

Programski jezik

Java

Verzije

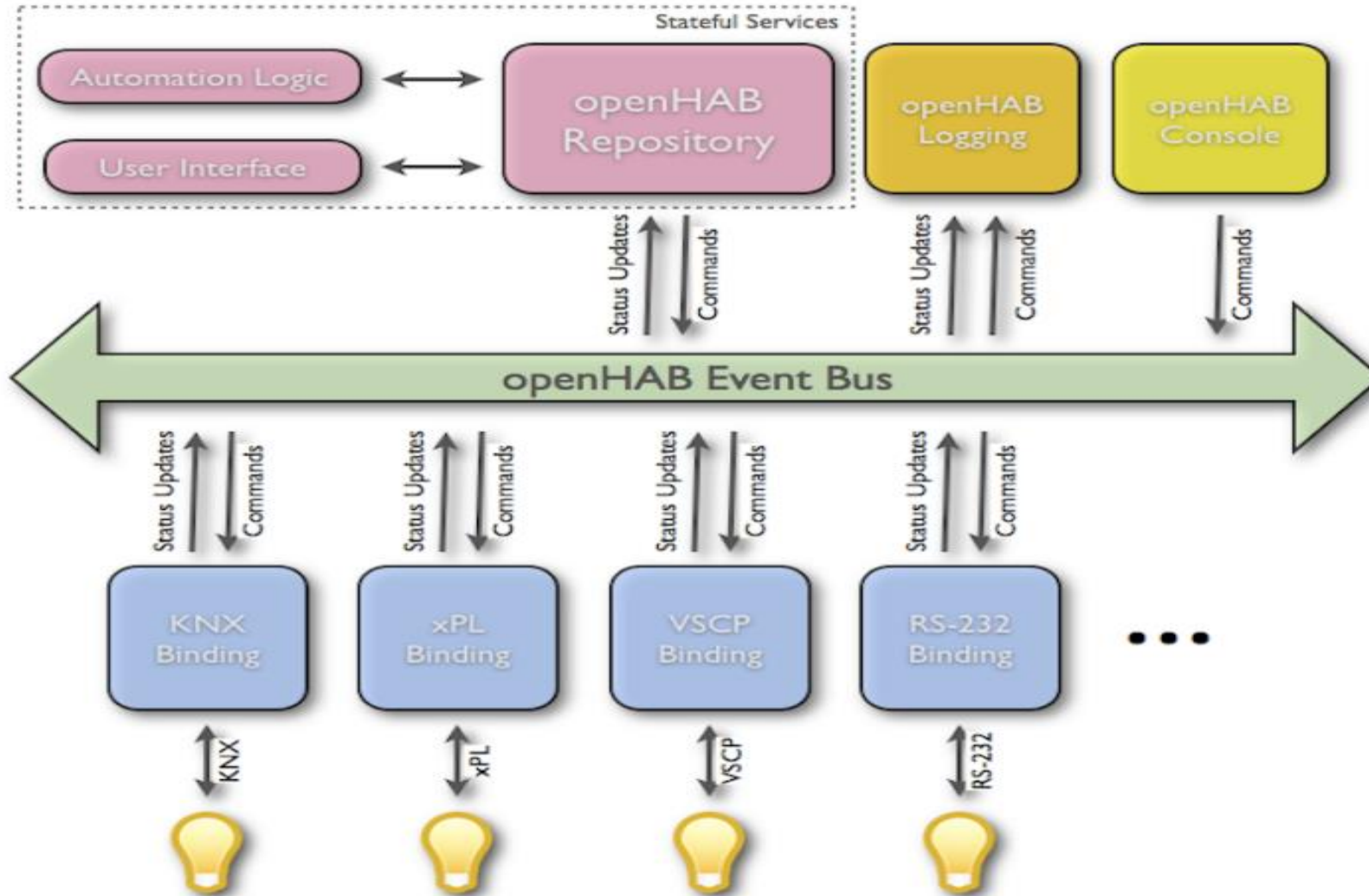
2020. godine kod je podeljen zbog dorade, odvojen na verziju 2.5 od predstojeće 3.0. Osim nekih tehničkih promena koda (npr. Upotreba Java 11), predviđeno je nekoliko funkcionalnih poboljšanja.

2021

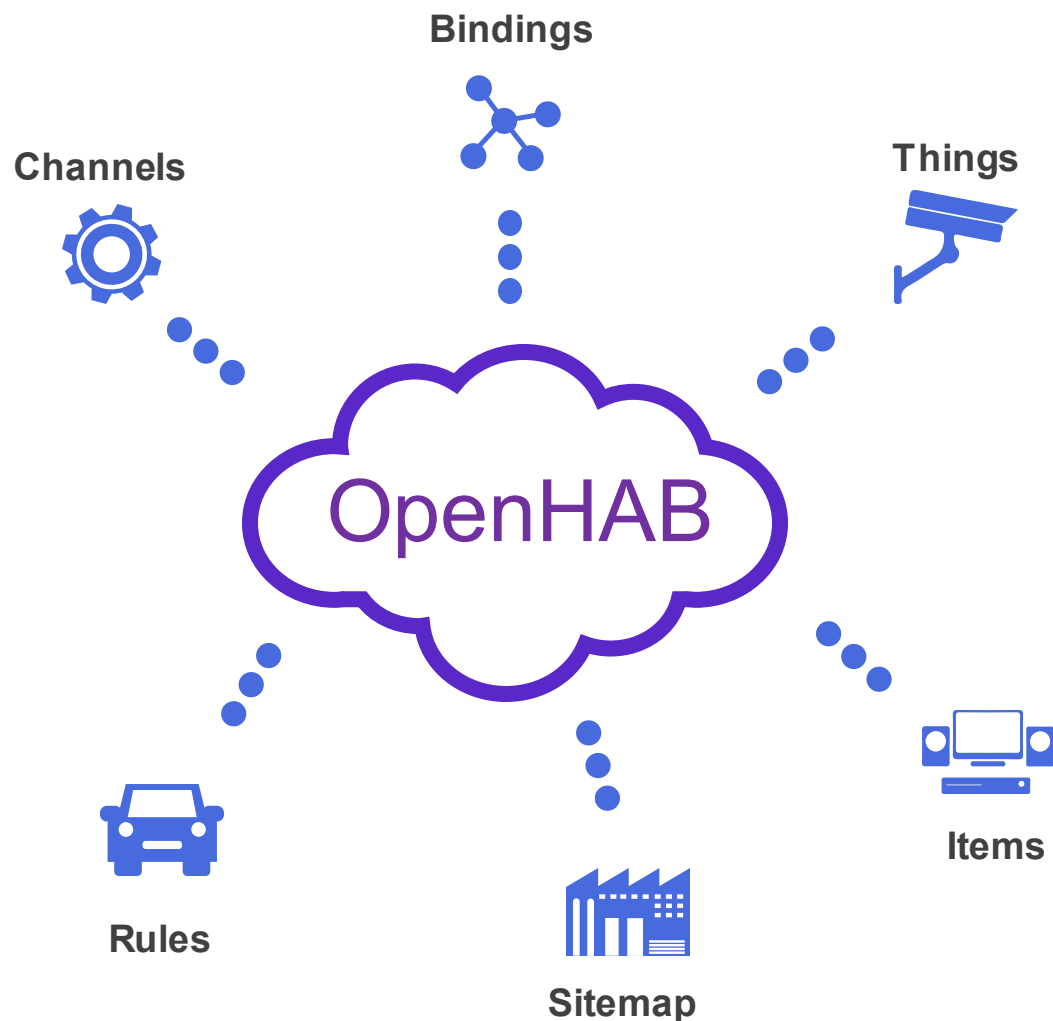
Poslednja stabilna verzija 3.0.2

21 April 2021

Arhitektura OpenHAB-a



Osnovni koncepti



OpenHAB komunicira sa pametnim i ne tako pametnim uređajima, izvodi korisnički definisane radnje i pruža veb stranice sa korisničkim informacijama kao i korisnički definisane alate za interakciju sa svim uređajima. Da bi to postigao, openHAB segmentira i razdvaja određene funkcije i operacije.

Bindings

Softverski paketi koje korisnik instalira u openHAB. Glavna svrha Bindings-a je uspostavljanje veze između vašeg uređaja i vaših stvari (Things). Bindings-i komuniciraju sa vašim uređajem i prevode sve naredbe ka i sa openHAB-a između vašeg uređaja i vaših stvari. Bindings-i integrišu fizički hardver, spoljne sisteme i veb usluge u openHAB.



Bluetooth



Perzistencija
podataka



Eksterni sistemi

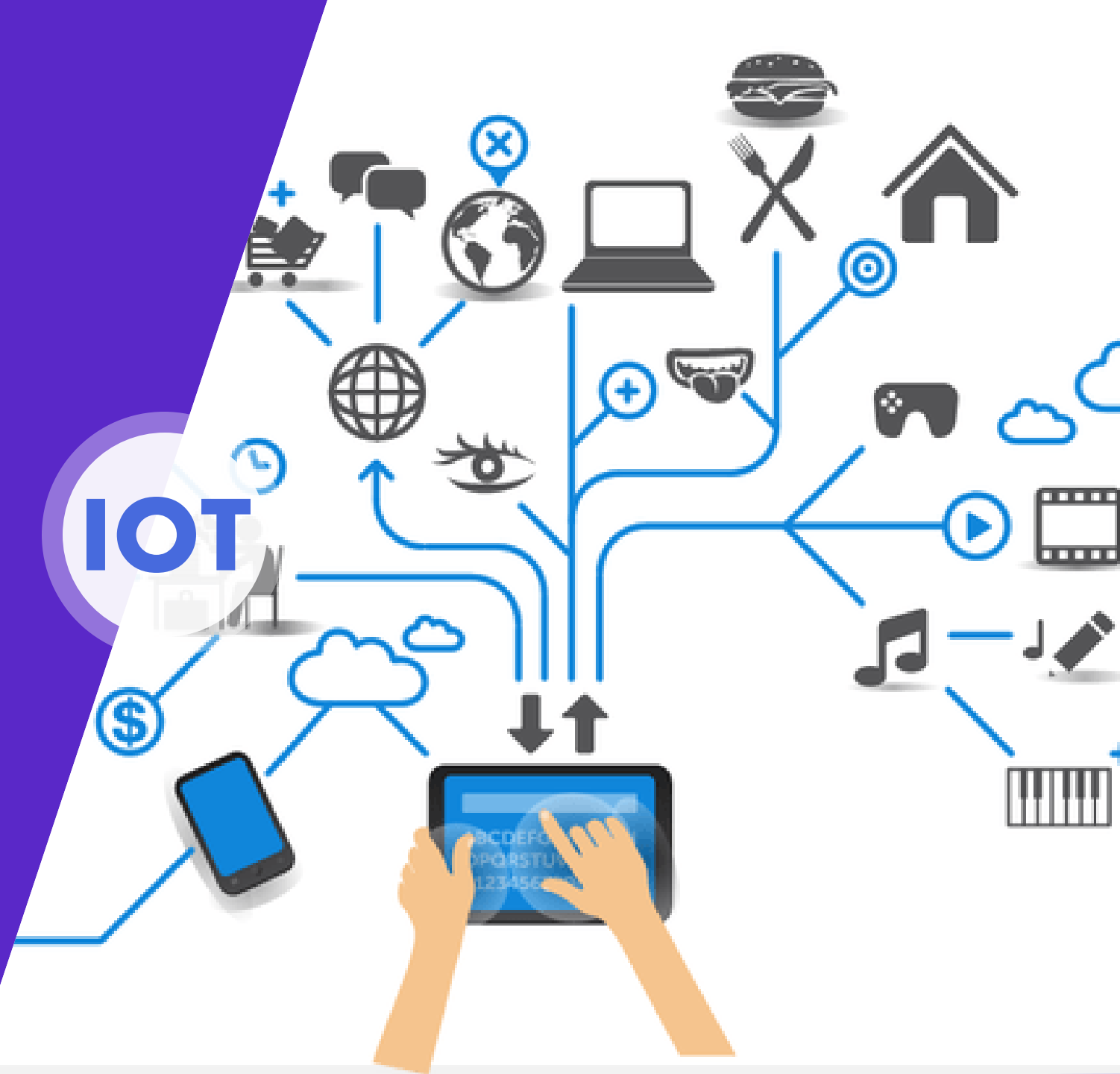


Glas

Things

Stvari su entiteti koji se fizički mogu dodati sistemu i koji potencijalno mogu pružiti mnoge funkcionalnosti u jednom. Važno je napomenuti da stvari ne moraju biti uređaji, već mogu predstavljati veb uslugu ili bilo koji drugi izvor informacija ili funkcionalnosti kojom se može upravljati. Iz perspektive korisnika, stvari su relevantni za postupak podešavanja i konfiguracije, ali ne i za konkretnu operaciju.

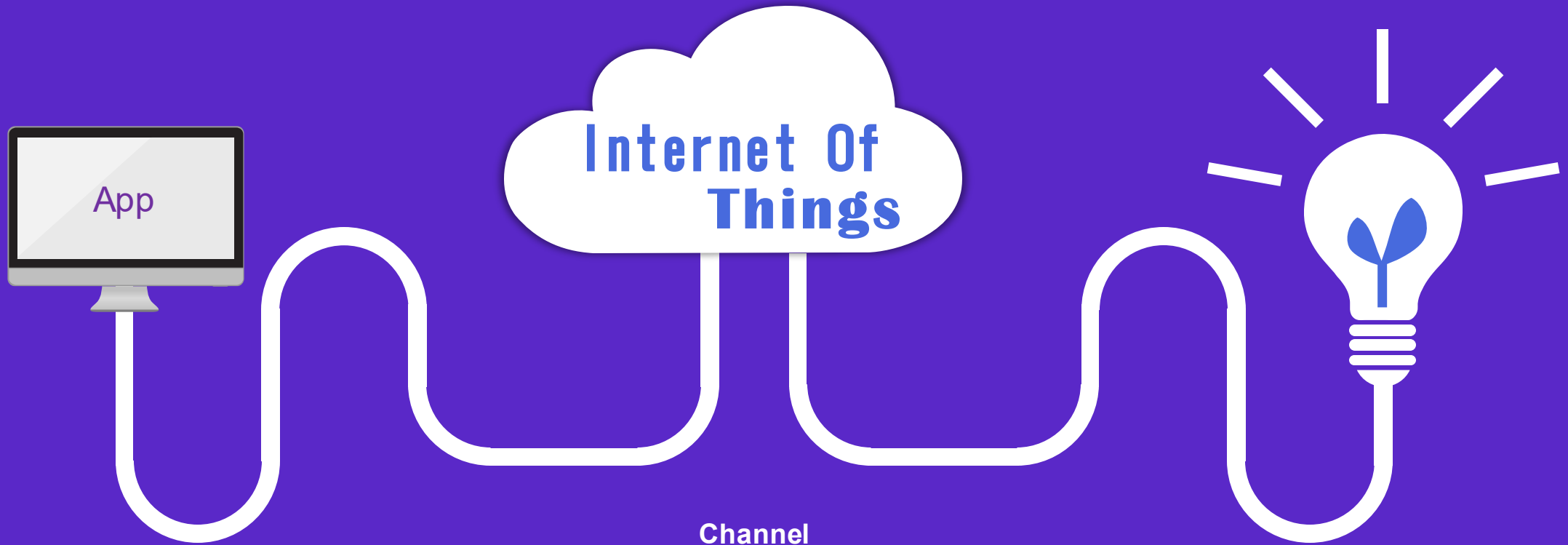
Stvari mogu imati svojstva konfiguracije, koja mogu biti neobavezna ili obavezna. Takva svojstva mogu biti osnovne informacije poput IP adrese, pristupnog tokena za veb uslugu ili specifične konfiguracije uređaja koja menja njegovo ponašanje.



Channel

Kanali su logička veza između Stvari i Stavki (Items). Kanali potiču iz definicije stvari i definišu kako vaša stvar može komunicirati sa stavkom (i obrnuto). Stvorćete kanale kada definišete svoju stvar (Thing).

Tokom definisanja svoje stvari identifikovaćete kanal na koji će vaša stavka biti povezana. Ova dva koraka osiguravaju da openHAB može prenositi informacije sa Thing-a na Item (i obrnuto).



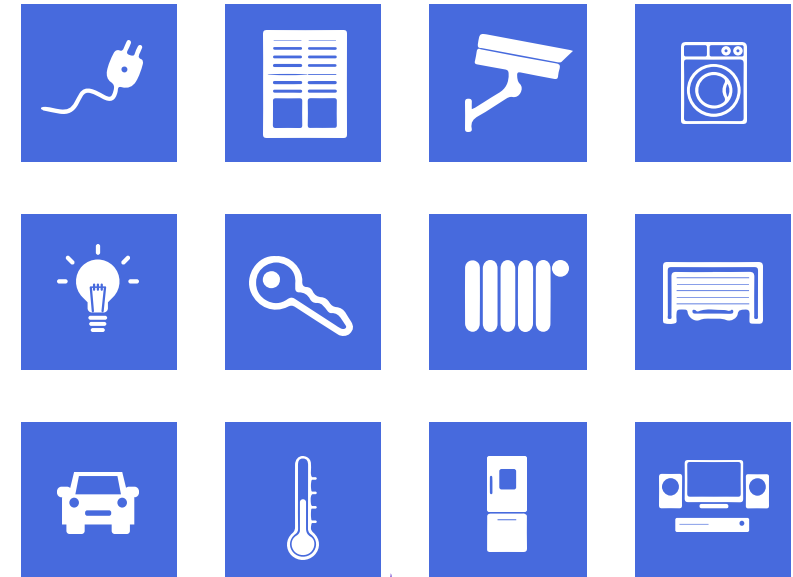
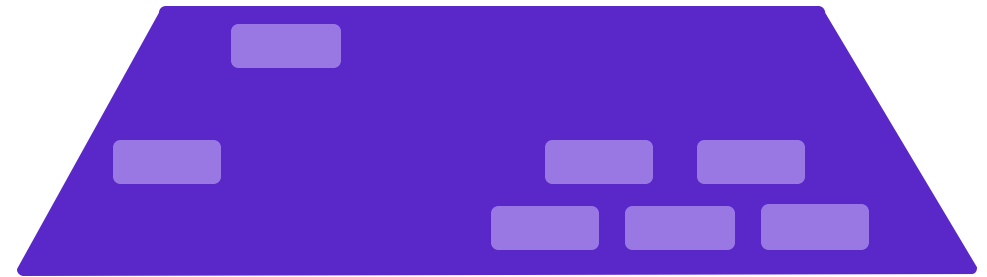
Items

openHAB ima striktno razdvajanje između fizičkog sveta („Things“) i aplikacije koja je izgrađena oko pojma „Items“ (takođe se naziva virtuelni sloj).

Stavke (Items) predstavljaju funkcionalnost koju aplikacija koristi (uglavnom korisnički interfejs ili logika automatizacije), imaju stanje i koriste se kroz događaje.

Tipovi Items-a koji su podržani:

- Color
- Contact
- DateTime
- Dimmer
- Group
- Image
- Location
- Number
- Number:<dimension>
- Player
- Rollershutter
- String
- Switch



RULES



„Pravila“ se koriste za automatizaciju procesa: Svako pravilo se može pokrenuti, što poziva skriptu koja izvršava bilo kakve zadatke, npr. uključite svetla tako što ćete izmeniti svoje predmete, napravite matematičke proračune, uključite tajmere itd.

openHAB ima visoko integrisan, lagan, ali ipak moćan "rule engine".

<RULE_NAME> - Svako pravilo mora imati jedinstveno ime (pod navodnicima).
Preporučuje se da odaberete ime koje ima značenje kada se izgovara.

<TRIGGER_CONDITION> - Pokretački događaj na kojem se izvršava logika pravila.
Pravilo se izvršava kao reakcija na jedan ili više uslova okidača. Više uslova je odvojeno ključnom reči ili.

<SCRIPT_BLOCK> - Sadrži logiku koju treba izvršiti kada je ispunjen uslov okidača.

```
rule "<RULE_NAME>"
when
    <TRIGGER_CONDITION> [or <TRIGGER_CONDITION2> [or ...]]
then
    <SCRIPT_BLOCK>
end
```

Sintaksa

Import:

```
import java.net.URI
```

A few default imports are already done, so classes from these packages do not need to be explicitly imported:

```
org.openhab.core.items  
org.openhab.core.persistence  
org.openhab.core.library.types  
org.openhab.core.library.items  
org.openhab.model.script.actions
```

Deklaracija promenljivih:

```
// a variable with an initial value. Note that the variable type is automatically inferred  
var counter = 0  
  
// a read-only value, again the type is automatically inferred  
val msg = "This is a message"  
  
// an uninitialized variable where we have to provide the type (as it cannot be inferred from an initial value)  
var Number x
```

Pravila:

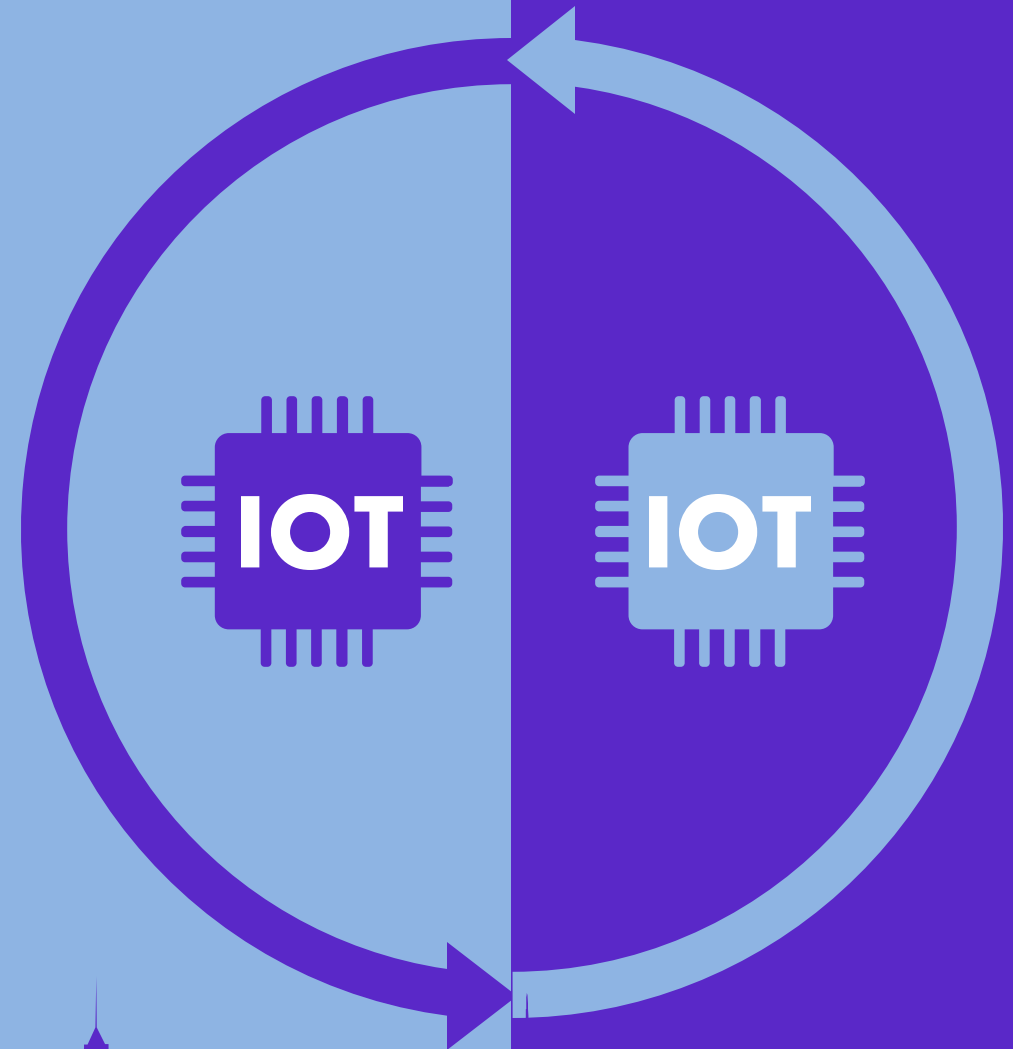


Rule okidači

Pre nego što pravilo počne da deluje, mora se pokrenuti.

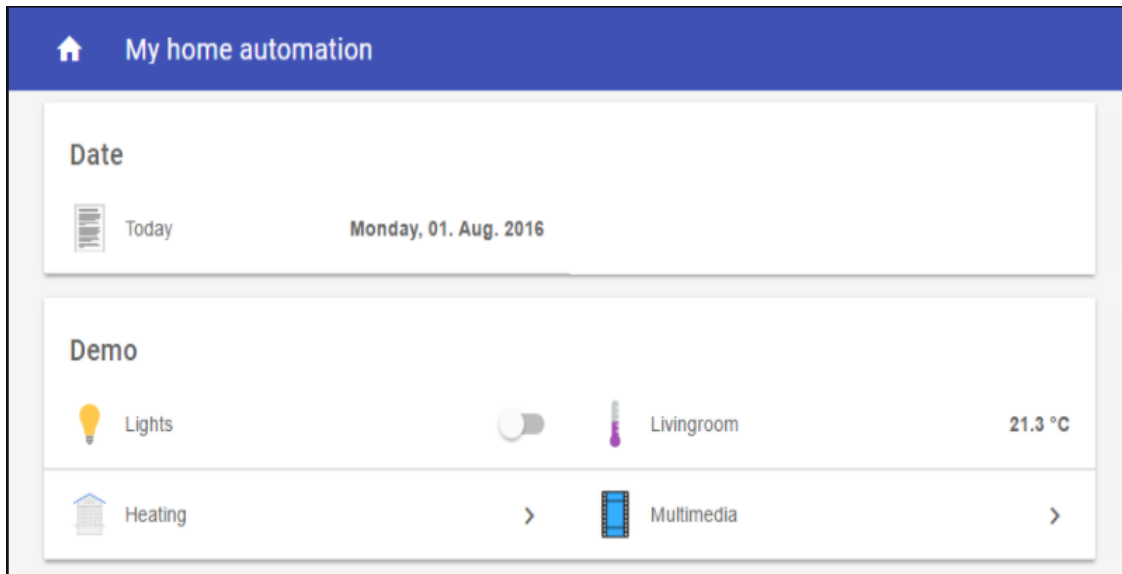
Postoje različite kategorije okidača pravila:

- **Item**(-Event)-based okidači: Oni reaguju na događaje na magistrali događaja openHAB, tj. Naredbe i ažuriranja statusa za stavke
- **Member of**(-Event)-based : Oni reaguju na događaje na magistrali događaja openHAB za stavke koje su članovi određene grupe
- **Time**-based okidači zasnovani na vremenu: Reaguju u posebno vreme, npr. u ponoć, svakog sata itd.
- **Sistemske** okidači: Oni reaguju na određene statuse sistema.
- **Thing**-based okidači zasnovani na stvarima: Oni reaguju na status stvari, tj. prelaze sa ONLINE u OFFLINE.



Sitemaps

Korisnički definisane sajt-mape:



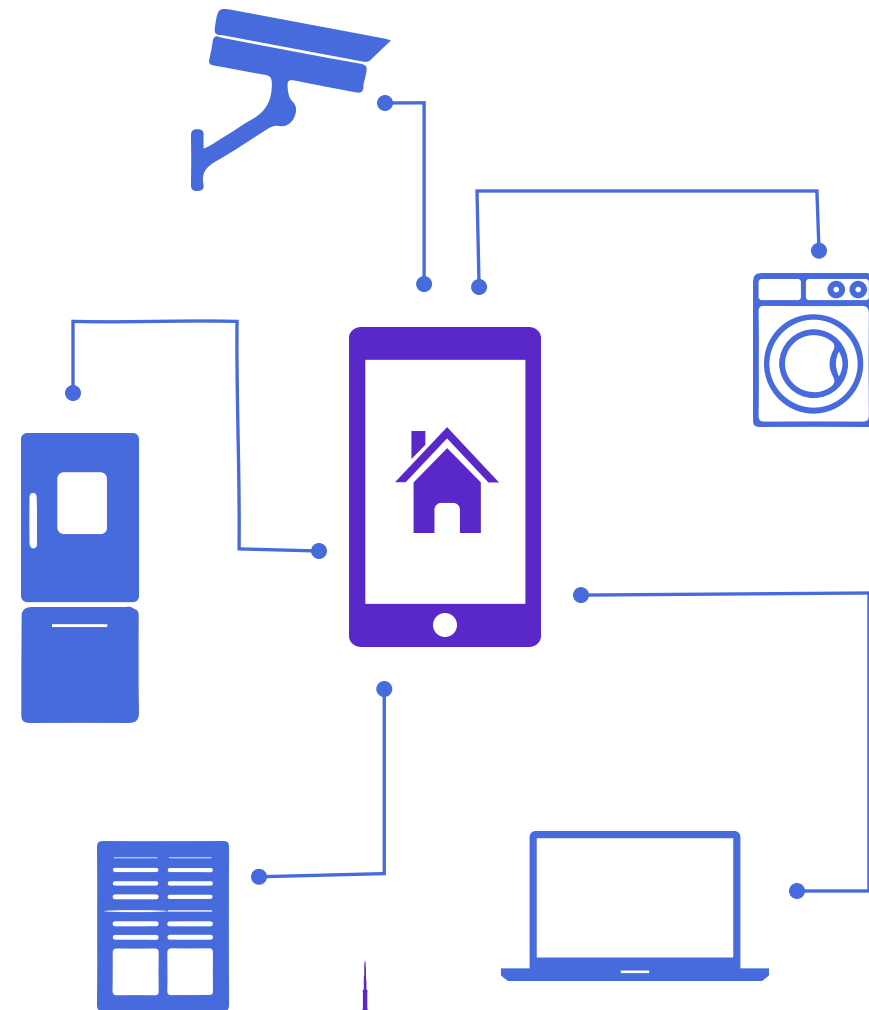
Primer:

```
sitemap demo label="My home automation" {  
  Frame label="Date" {  
    Text item=Date  
  }  
  Frame label="Demo" {  
    Switch item=Lights icon="light"  
    Text item=LR_Temperature label="Livingroom [%.1f °C]"  
    Group item=Heating  
    Text item=LR_Multimedia_Summary label="Multimedia [%s]" icon="video" {  
      Selection item=LR_TV_Channel mappings=[0="off", 1="DasErste", 2="BBC One", 3="Cartoon Network"]  
      Slider item=LR_TV_Volume  
    }  
  }  
}
```



OpenHAB REST API

- ✓ Preuzmite podatke openHAB-a iz spoljnih aplikacija
- ✓ Popunjavanje podataka i pokretanje događaja u openHAB iz spoljnih aplikacija (na primer, neki detektori pokreta ili nadzorne kamere)
- ✓ Pregledajte Bindings OpenHAB-a, stvari (Things) ili stavke (Items) , saznajte više o trenutnim stanjima, parametrima ili problemima
- ✓ Interakcija sa openHAB-om iz drugih programa, mnogi programski jezici i alati za automatizaciju mogu lako da koriste REST API
- ✓ Korišćenje softvera nezavisnih proizvođača na mobilnim telefonima, poput taskera, za otvaranje garažnih vrata



Primeri:

- Switching `My_Item` OFF by issuing an http **POST** request:

```
curl -X POST --header "Content-Type: text/plain" --header "Accept: application/json" -d "OFF" "http://{openHAB_IP}:8080/rest/items/My_Item"
```

sh

- Setting a Contact item `My_Item` to CLOSED by issuing an http PUT request to `My_Item/state` :

```
curl -X PUT --header "Content-Type: text/plain" --header "Accept: application/json" -d "CLOSED" "http://{openHAB_IP}:8080/rest/items/My_Item/state"
```

sh

- Retrieving a list of all Items and Groups by issuing a GET request:

```
curl -X GET --header "Accept: application/json" "http://{openHAB_IP}:8080/rest/items?recursive=false"
```

sh

- Retrieving a list of all sitemaps by issuing a GET request:

```
curl -X GET --header "Accept: application/json" "http://{openHAB_IP}:8080/rest/sitemaps"
```

sh

- Subscription to events:

```
# ThingStatusInfoChangedEvent - The status of a thing changed.  
curl "http://{openHAB_IP}:8080/rest/events?topics=smarthome/things/{thingUID}/statuschanged"  
  
# ChannelTriggeredEvent - A channel has been triggered.  
curl "http://{openHAB_IP}:8080/rest/events?topics=smarthome/channels/{channelUID}/triggered"
```

sh





Hvala na pažnji!