

IoT & Drupal 8

Andor Koza

andor@studiopresent.com

Goran Nikolovski

goran@studiopresent.com



What is this session about?

Drupal 8

Internet of Things

Startup

www.dotlab.rs

Drupal based embedded systems including hardware device.

Why?

We wanted to use benefits of drupal on another way and create smart devices which can replace the old school ones.

Products

Access - access control system

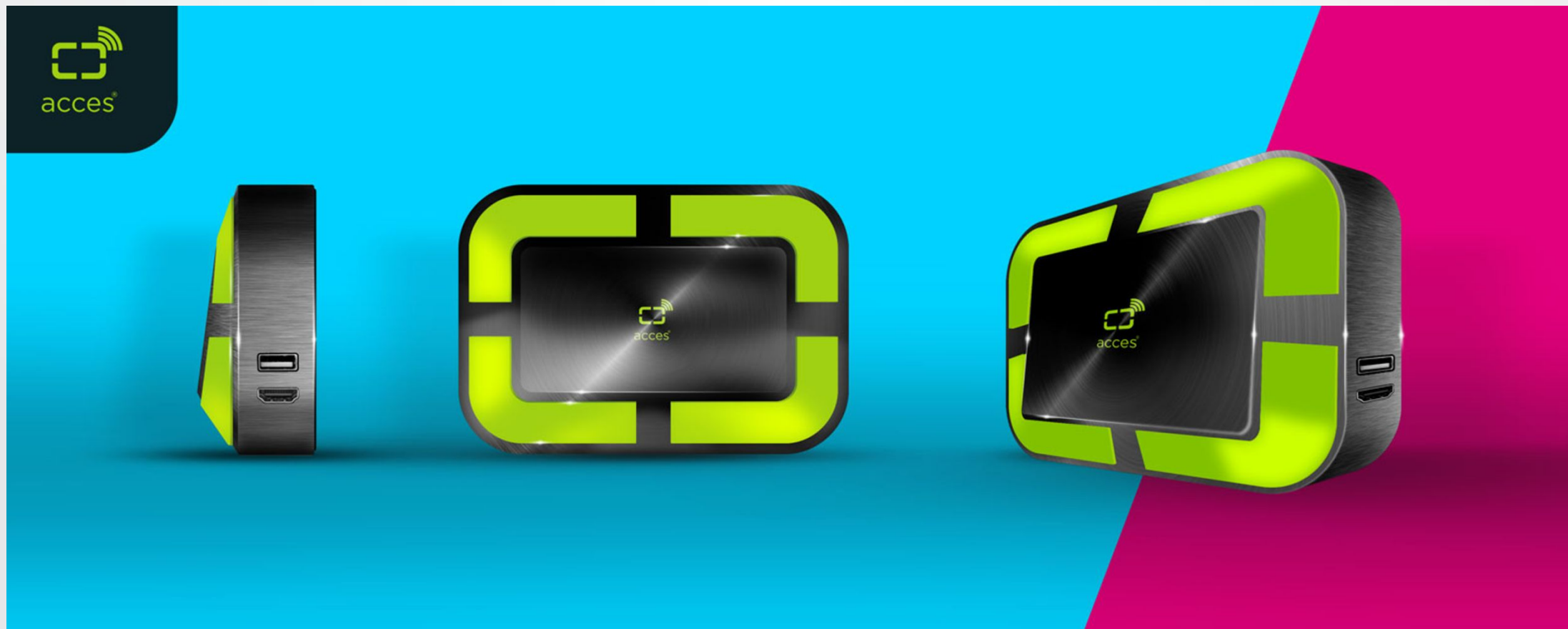
MainStay - smart irrigation system (for now)



Access

Access is cloud based access control system which can be useful everywhere where is need for access controlling.

The main part of hardware is Raspberry PI Zero W onboard computer, which handles the NFC module, communicates with server via websocket and give feedback to the users.



Mainstay

Mainstay is an smart irrigation system.

It can be controlled through mobile application or web interface.

It's smart, it is cool, it has a complete unique hardware.



Mainstay hardware

The hardware is based on ESP8266 microcontroller.

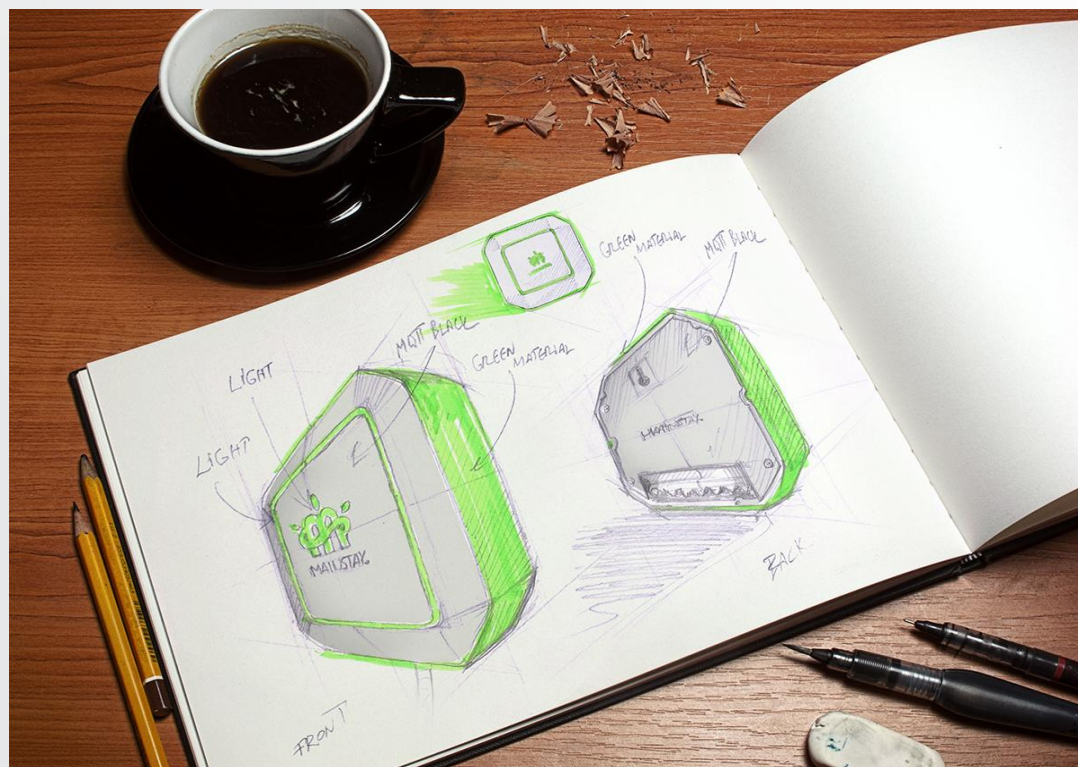
- Onboard WiFi (b/g/n)
- 80MHz (Overclock to 160MHz is possible)
- 32 bit MCU
- 9 GPIO ports (easily expandable)
- Wifi modes: AP, Client, AP+Client
- Integrated ADC
- Supports all wireless security and encryption standards
- Supports all major communication interfaces (SPI, I2C, UART, etc.)
- 4MB Flash memory
- Supports WebSocket



Mainstay usage & future

The device is **scalable** and **modular**, it means that once the core system is ready, it will be very easy to change possible usage. For example it could turn on/off almost anything, get inputs from sensors, build a smart alarm, etc.

After MVP we will start to add sensor inputs (temperature, humidity, rain, hygrometer, water flow, energy consumption), weather forecast API and based on these informations we will embed Fuzzy Logic. Thanks to them, the device can save water and energy.



Next step: Android & iOS app, UX

We are working on mobile app right now. We try to make outstanding UX and design.



Wellcome



Email petar1965@gmail.com

Password



Login

[I forget my password](#)

Sign up

Mode



Tuje

Mo Tu We Th Fr Sa



Tuje2

Mo We Fr



Basta

Weekdays



Prskalice

All days



Ispred

All days



Paulovnije

Mo Tu We Th

Zone 7

Zone 8



Snooze



Manual



Zones



Settings

Add mode

Select zone



A name of this mode



Add a picture

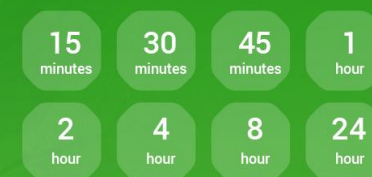


Starting time

3
3
4
5
6
7
8

30
35
40
45
50
55
00
05

Duration



Days



Snooze



Manual



Zones



Settings

Let's see in action

Before we got to this point we created many prototypes, tested a ton of electronic components to find what is the best for our needs.

From software side we also had many challenges (and we will have more in the near future), but this will be presented by Goran.



Intro

- Details of implementation
- Interesting and challenging
- Complex system, various technologies
- Four different components: Websocket server, Drupal module, React JS components and physical devices
- Websocket messages and REST Api (JSON messages)
- Fast, scaled and maintained



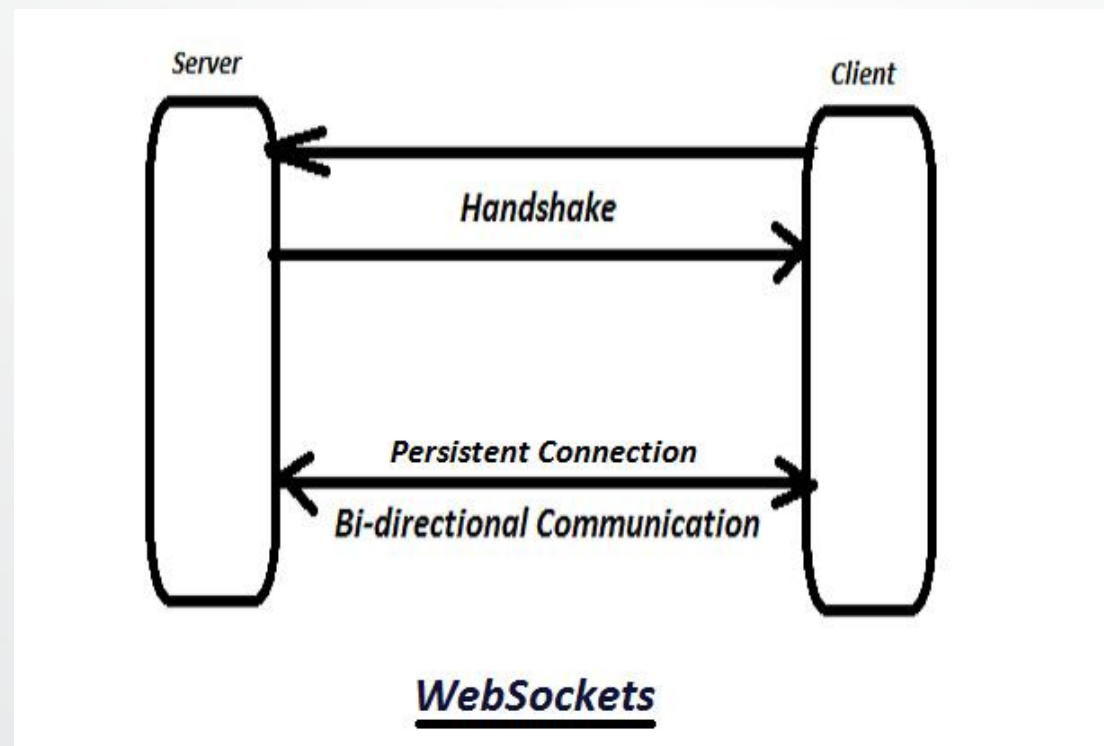
Websocket server

- Using Websockets to speed up communication
- Server is main component
- Written in PHP, used Ratchet library
- Real time, bi-directional messages



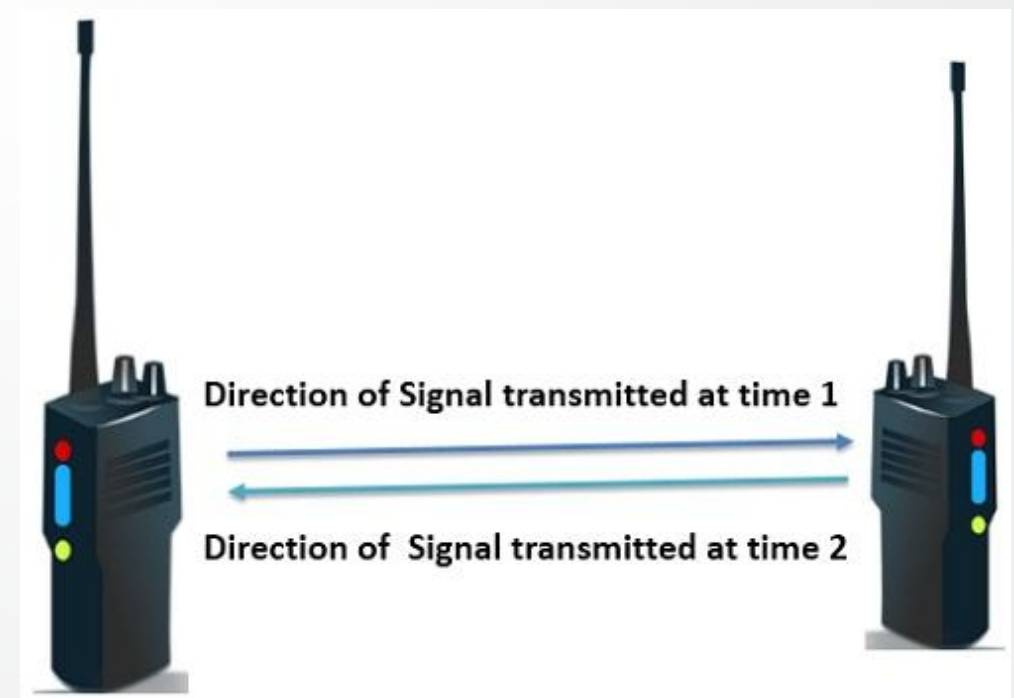
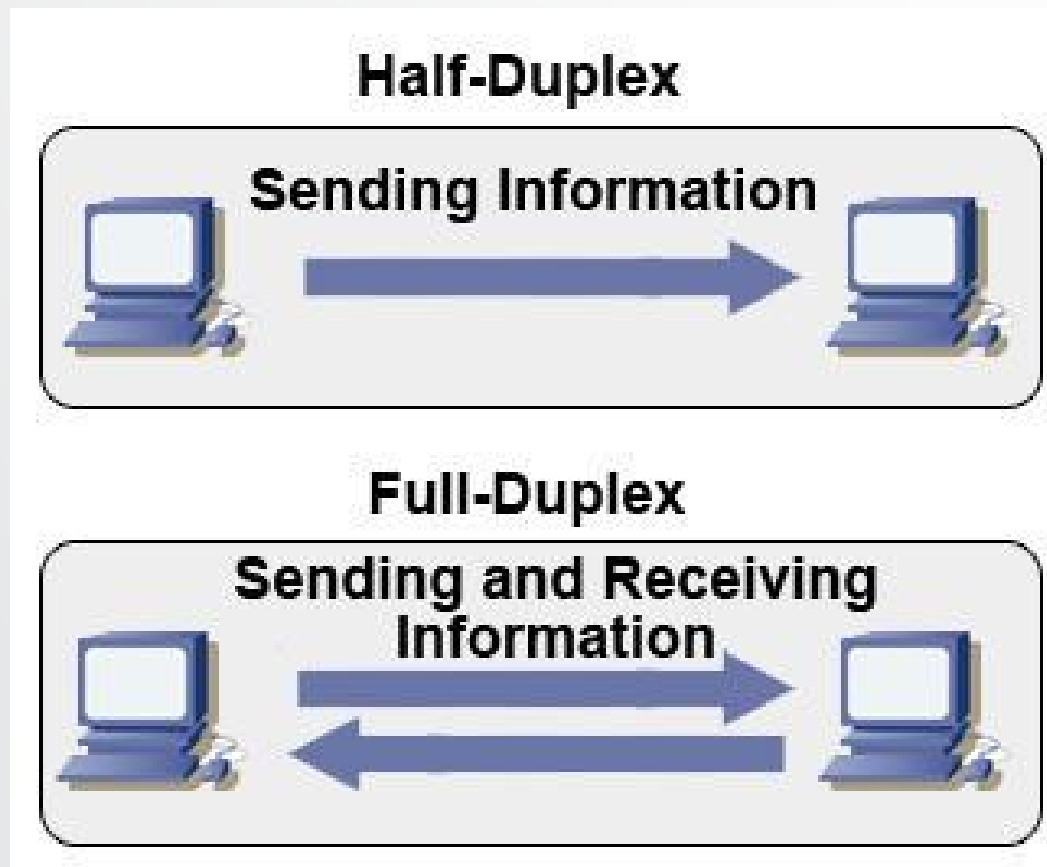
Websocket protocol

- Websocket is a communication protocol
- Full duplex over a single TCP connection
- No request/response
- Both parties can communicate at the same time (telephone)



Websocket protocol

- Half duplex (HTTP)
- Both parties cannot communicate at the same time (walkie-talkie)



Websocket server

- Websocket connection stays open
- Protocol is supported in all major browsers
- Stuck in an infinite loop
- Communication with devices and user interface over Websockets
- Communication with Drupal over REST



Drupal

- Storage system (user profiles and devices)
- Log messages
- Custom entities
- Drupal console
- Performance gain by using custom entities



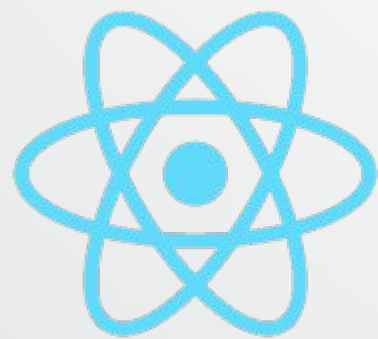
Drupal

- Built in REST services in Drupal 8
- REST for exposing info to React JS user interface
- REST for receiving data from Websocket server
- Custom REST resource plugins
- Basic auth
- Storage system and place to mount React JS. Overkill?



React JS

- User interface for configuring devices
- User dashboard and admin dashboard
- REST for fetching data from Drupal
- React is great for creating user interfaces
- No storing data in DOM
- We had to use Javascript
- Component structure



React

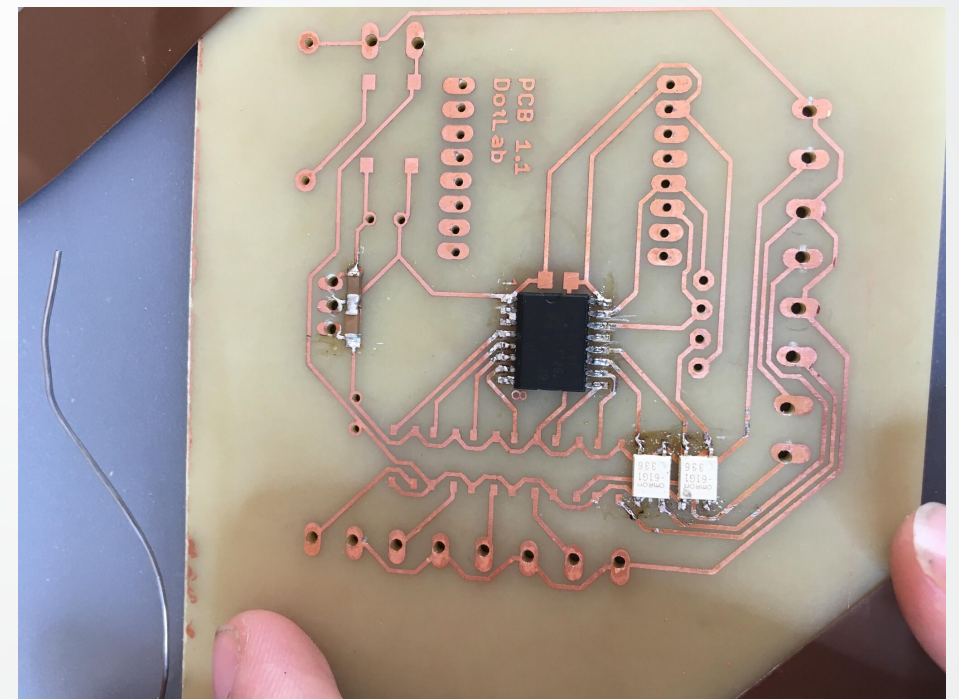
UserDashboard

Device
Interval
Interval
Interval

Device
Interval
Interval
Interval

Devices

- WIFI connection
- Send and receive Websocket messages
- Limited memory and storage
- Several iterations to make it ideal



Further optimization

- WIP
- Optimization to use NoSQL for device config
- Couchbase (key->value storage)
- Device config stored in Drupal

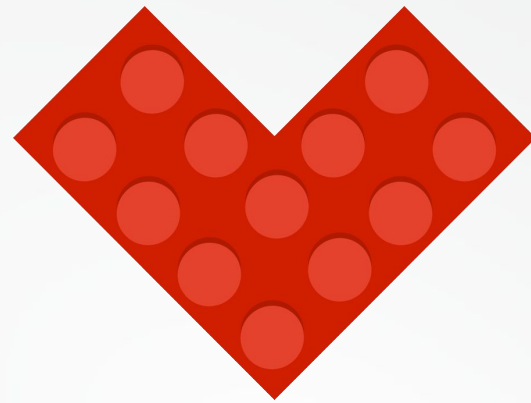


Conclusion

- Fully decoupled
- React Native
- Benefits: smaller sybsystems, not large monolithic system



Thank you



Ask us questions now

Andor Koza

andor@studiopresent.com

Goran Nikolovski

goran@studiopresent.com