

Control all the Things!

Marcus Hellberg / Vaadin

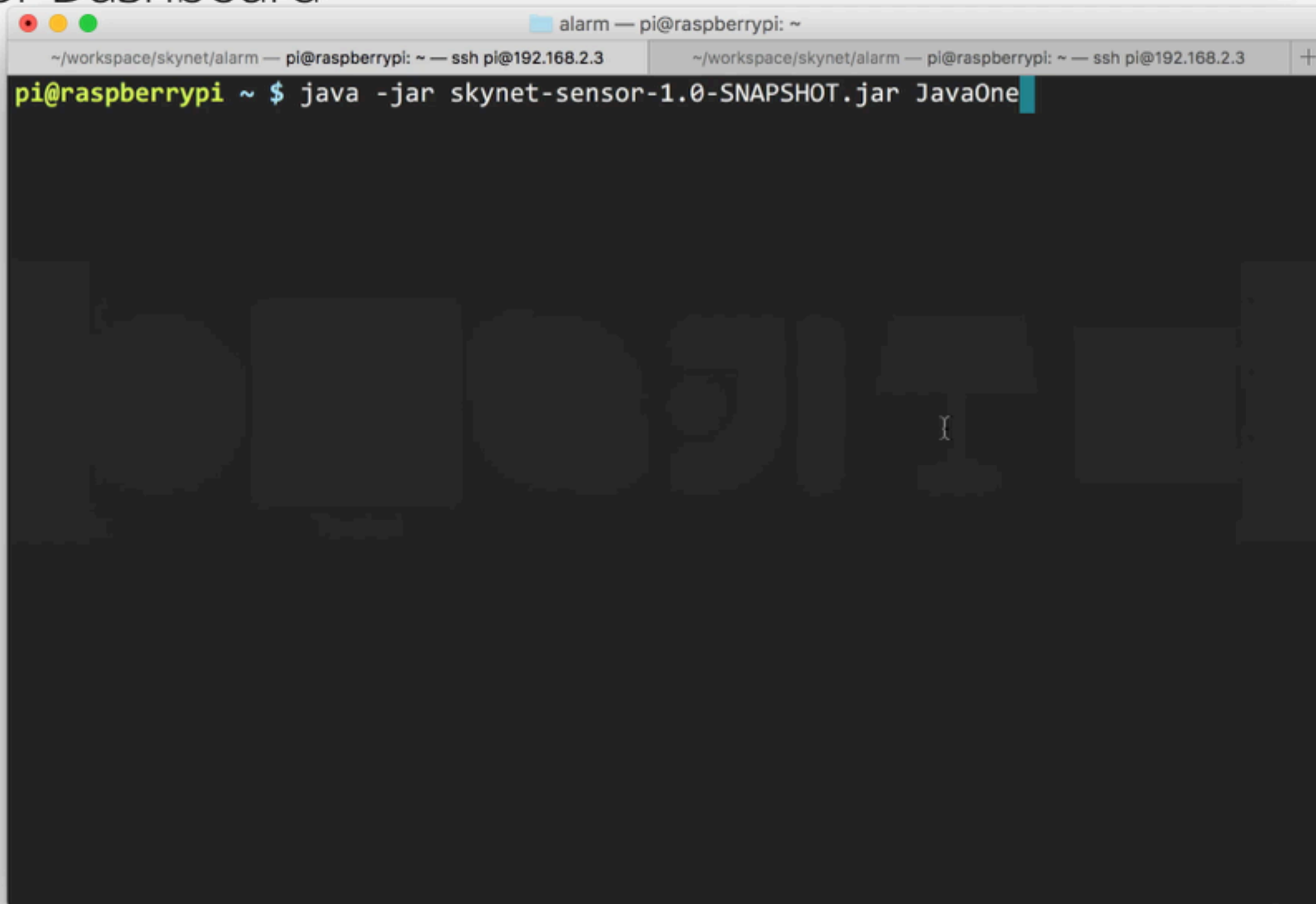
\$ whoami

Sensor/alarm network & dashboard

- Auto discovery of sensors and alarms
- Plot and show raw numeric data
- Configure alarms to get triggered at certain thresholds
- Develop everything in Java

Why Java?

Sensor Dashboard



A terminal window titled "alarm — pi@raspberrypi: ~" is shown. The window has two tabs, both with the same title. The active tab shows the command `java -jar skynet-sensor-1.0-SNAPSHOT.jar JavaOne` being executed. The output of the command is a large, stylized, pixelated text graphic that reads "skynet" in a light gray color against a black background. The cursor is positioned at the end of the command line.

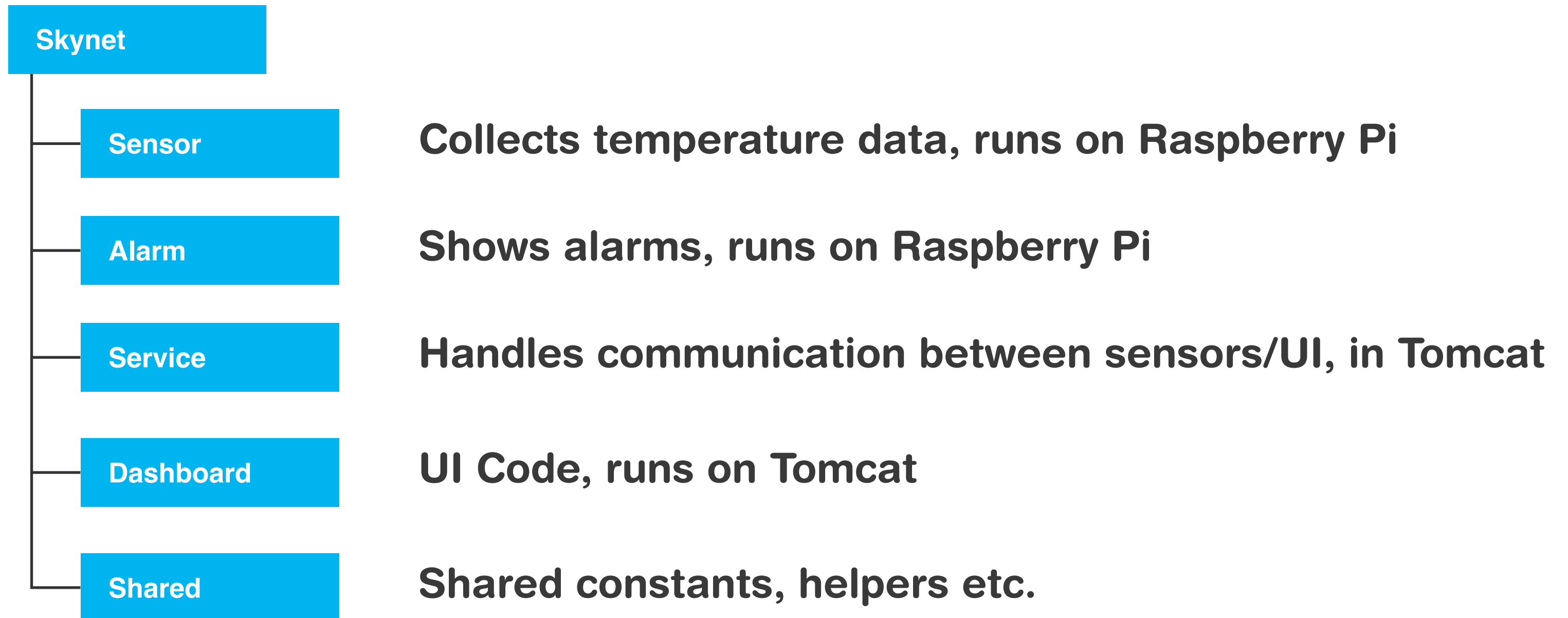
```
alarm — pi@raspberrypi: ~  
~/workspace/skynet/alarm — pi@raspberrypi: ~ — ssh pi@192.168.2.3  
pi@raspberrypi ~ $ java -jar skynet-sensor-1.0-SNAPSHOT.jar JavaOne  
skynet
```

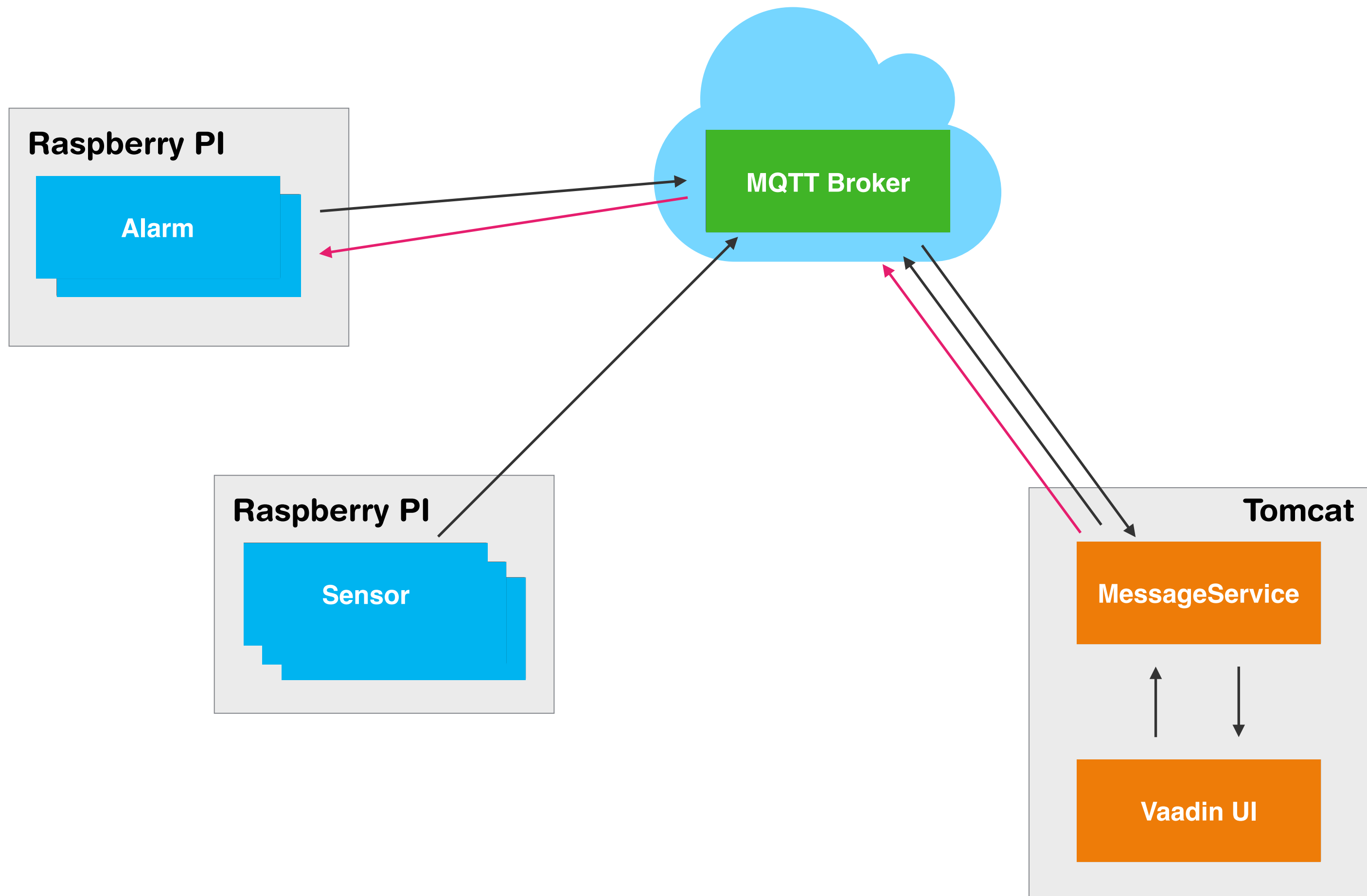
Things we'll need

- **Raspberry PI(s) with temp sensor and GPIO access – Pi4J**
- **A lightweight and robust way of communicating sensor data – Paho**
- **A way of building a web UI in Java – Vaadin**

Overview

Project structure





less talk, more code!

github.com/vaadin-marcus/skynet

Sensor

MQTT

- Light-weight & robust pub-sub protocol over TCP/IP
- Publish and subscribe to different topics (/sensors/temperature/#)
- Central broker, QoS
- OASIS standard, several implementations available



The Paho project provides open-source client implementations of MQTT and MQTT-SN messaging protocols aimed at new, existing, and emerging applications for Machine-to-Machine (M2M) and Internet of Things (IoT).

Latest release: 1.1

Downloads

C client for Windows/Unix/Mac	1.0.3
Java client and utilities	1.0.2
Android service	1.0.2
Python client	1.1
JavaScript client	1.0.1
C/C++ MQTT Embedded clients	1.0.0
.Net and WinRT client (M2Mqtt)	4.0.0.0

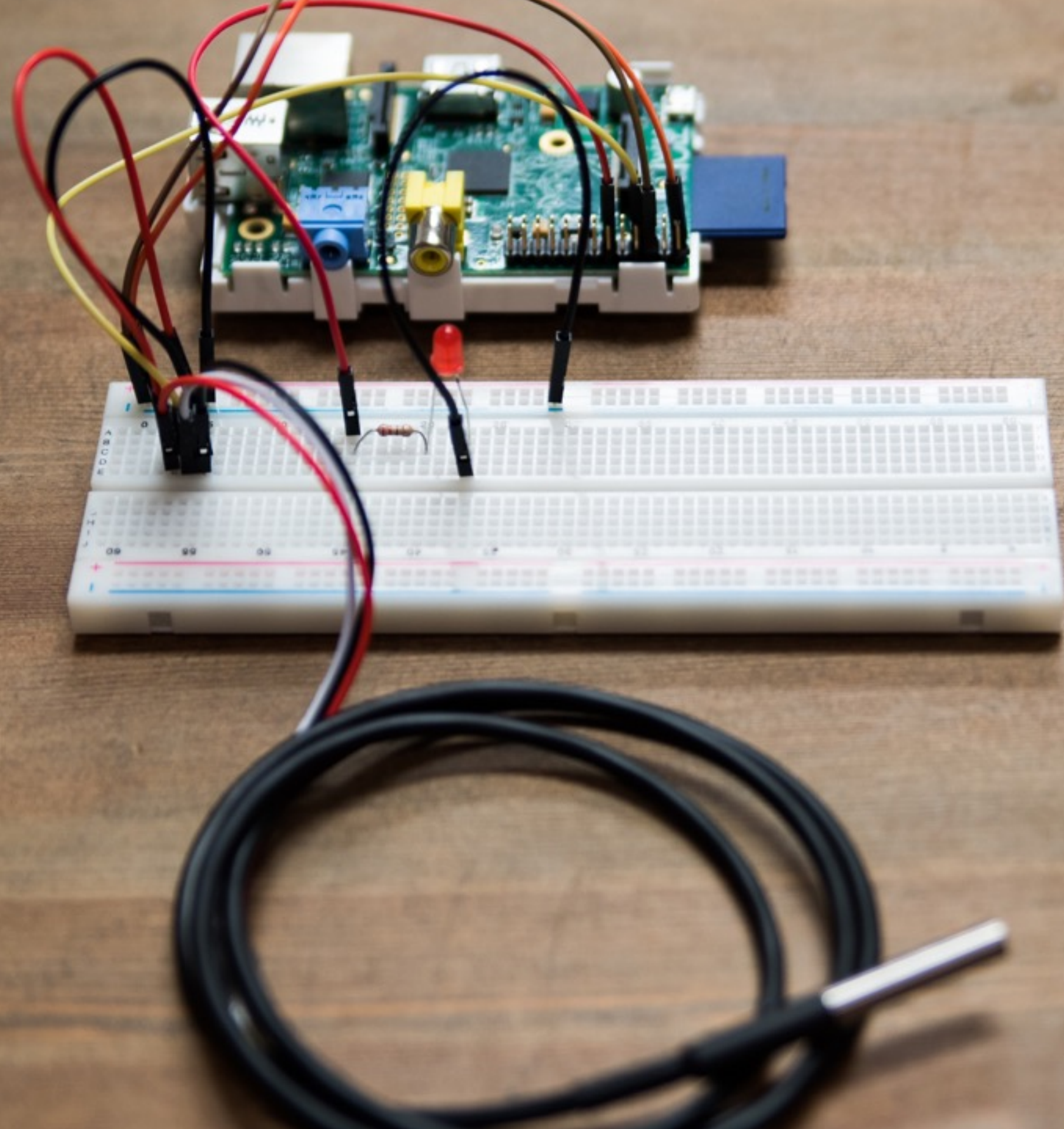
```
MqttClient client = new MqttClient(Skynet.BROKER, MqttClient.generateClientId());
client.connect();
client.subscribe("hello/#");

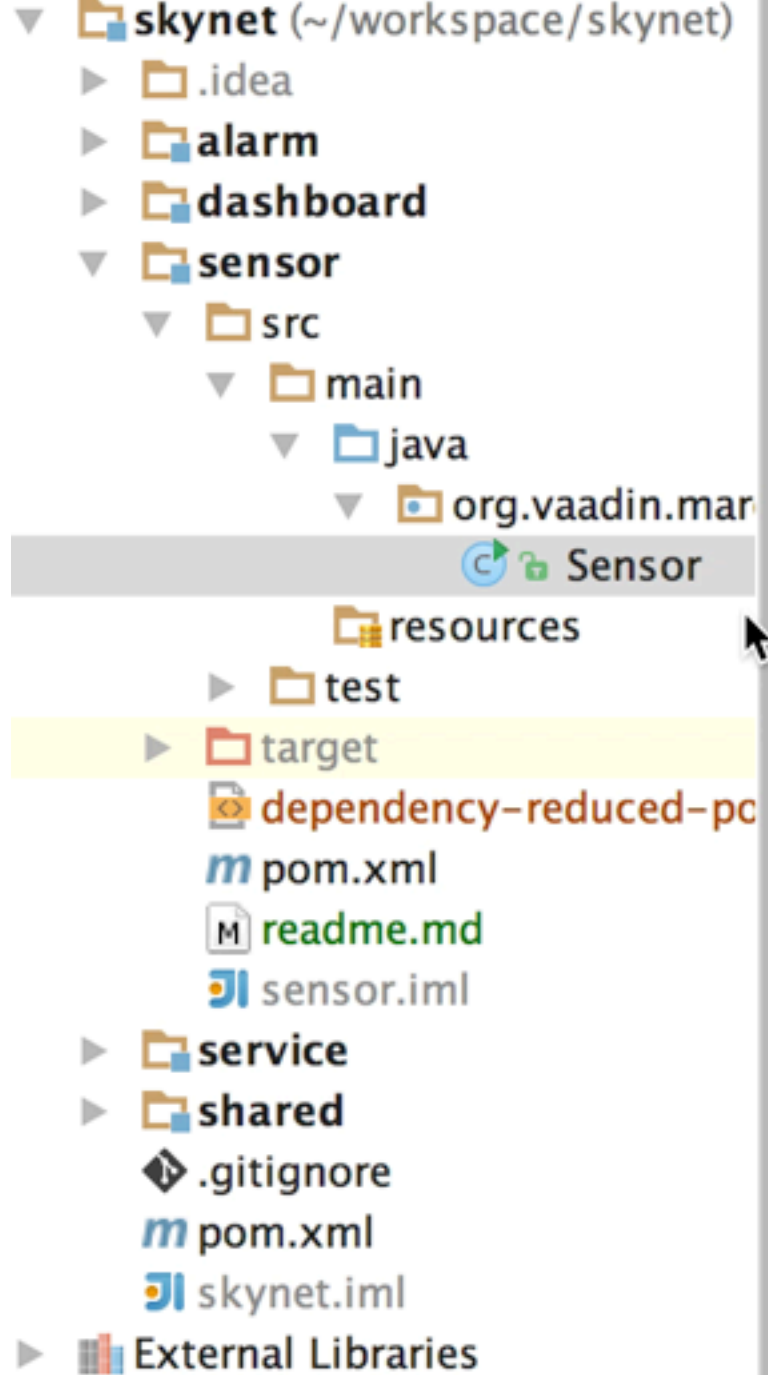
client.setCallback(new MqttCallback() {
    @Override
    public void connectionLost(Throwable cause) {}

    @Override
    public void messageArrived(String topic, MqttMessage message) throws Exception {
        System.out.println(message.getPayload());
    }

    @Override
    public void deliveryComplete(IMqttDeliveryToken token) {}
});

MqttMessage message = new MqttMessage("world".getBytes());
message.setQos(0);
client.publish("hello", message);
```



Search Everywhere [Double](#) [↑](#)

Open a file by name [↑](#) [⌘](#) [O](#)

Open Recent Files [⌘](#) [E](#)

Open Navigation Bar [⌘](#) [↑](#)

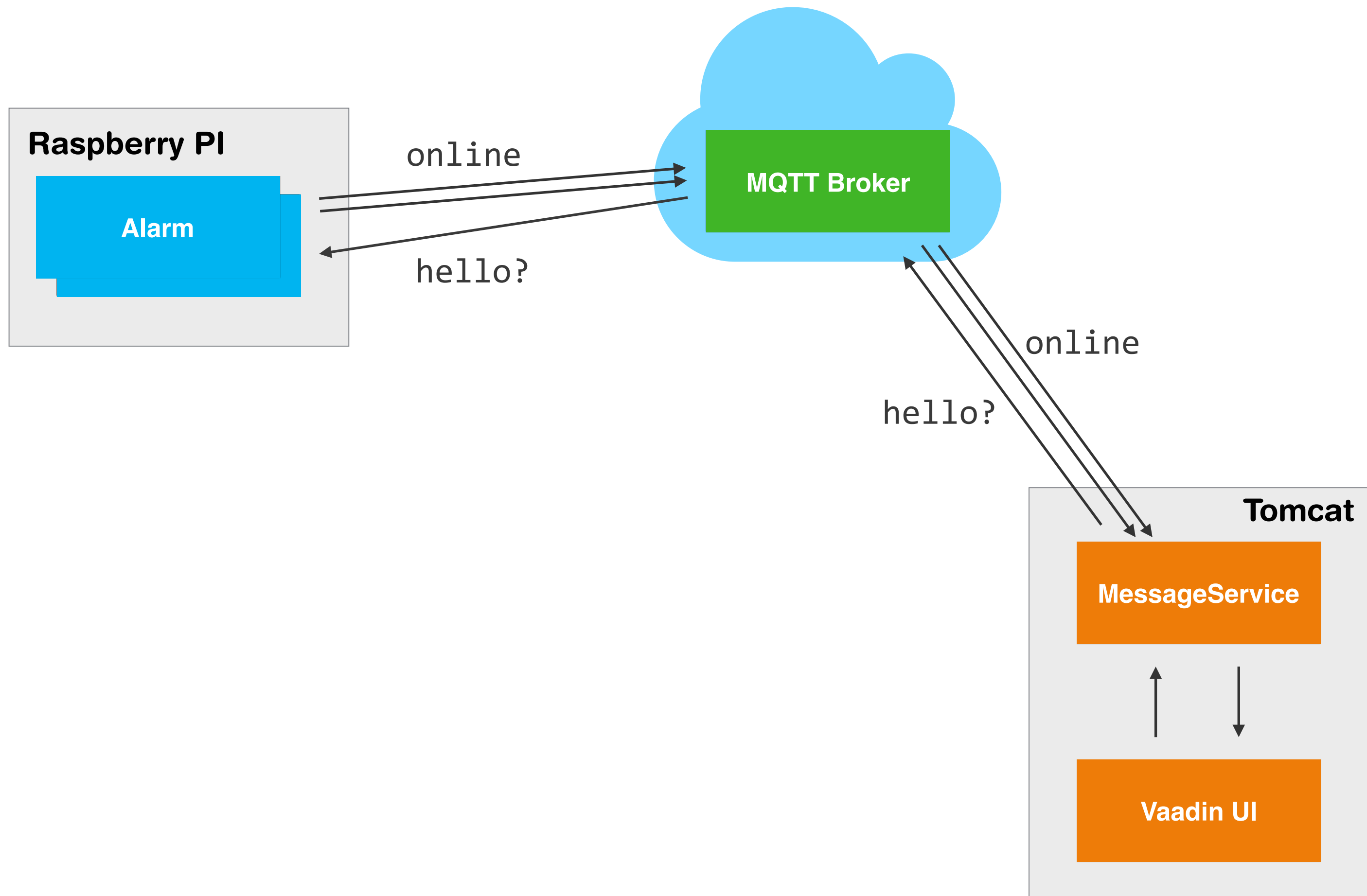
Drag and drop files from Finder

Alarm

Pi4J

- **Object-oriented I/O API in Java**
- **Allows access to GPIO pins**
- **Open Source (LGPL)**

```
GpioPinDigitalOutput pin0 =  
    GpioFactory.getInstance()  
        .provisionDigitalOutputPin(RaspiPin.GPIO_00, PinState.LOW);  
  
pin0.high();  
Thread.sleep(1000);  
pin0.low();
```



skynet (~/.workspace/skynet)

- ▶ .idea
- ▼ alarm
 - ▼ src
 - ▼ main
 - ▼ java
 - ▼ org.vaadin.marc
 - Alarm
 - resources
 - test
 - target
 - alarm.iml
 - dependency-reduced-pom.xml
 - pom.xml- ▶ dashboard
- ▶ sensor
- ▶ service
- ▶ shared
- .gitignore
- pom.xml
- skynet.iml
- External Libraries

Search Everywhere Double ⇧

Open a file by name ⇧ ⌘ O

Open Recent Files ⌘ E

Open Navigation Bar ⌘ ⇧

Drag and drop files from Finder

Message Service

skynet (~/.workspace/skynet)

- .idea
- alarm
- dashboard
- sensor
- service
 - src
 - main
 - java
 - org.vaadin.marcus.s
 - entities
 - events
 - service
 - resources
 - test
 - java
 - org.vaadin.marcus.s
 - entities
 - service
 - target
 - pom.xml
 - service.iml
 - shared
 - .gitignore
 - pom.xml
 - skynet.iml

External Libraries

```
1 package org.vaadin.marcus.skynet.service;
2
3 import ...
4
19
20 public class MessageService implements MqttCallback {
21
22     private static MessageService instance;
23
24     protected MqttAsyncClient client;
25     protected Set<Alarm> alarms = new HashSet<>();
26     protected Set<Trigger> triggers = new HashSet<>();
27     protected EventBus eventBus = new EventBus();
28
29     private MessageService() {...}
30
31
32     public static MessageService getInstance() {...}
33
38
39     protected void connect() {
40         try {
41             client = new MqttAsyncClient(Skynet.BROKER, MqttClient.generateClientId());
42             MqttConnectOptions options = new MqttConnectOptions();
43             options.setCleanSession(true);
44             client.connect().waitForCompletion();
45
46             client.subscribe(Skynet.TOPIC_ALARMS + "/#", 0).waitForCompletion();
47             client.subscribe(Skynet.TOPIC_SENSORS + "/#", 0).waitForCompletion();
48             client.setCallback(this);
49
50             // Send discover message to find any running alarms
51             client.publish(Skynet.TOPIC_ALARMS, new MqttMessage(Skynet.HELLO));
52         } catch (MqttException e) {
53             e.printStackTrace();
54         }
55     }
56 }
```


Dashboard UI

Vaadin

- **Component-based framework for building web apps in Java**
- **Automated server-client communication**
- **No browser plugins required, works on all devices**
- **Open Source (Apache 2)**

Components

Window Caption + ×

Selected Another One more

Subtitle

Normal type for plain text. Etiam at risus et justo dignissim congue. Phasellus laoreet lorem vel

Footer text

OK

Cancel

9/2/14

Option One ▾

Option One
Option Two
Option Three

me@vaadin.com

Drop down ▾

caption	description
Lorem ipsum	Dolor sit amet
Consectetur quid	Securi etiam tamquam
Eu fugiat	Nulla pariatur lorem
Ipsum dolor	Sit amet consectetur
Quid securi	Etiam tamquam eu
Fugiat nulla	Pariatur lorem ipsum
Dolor sit	Amet consectetur quid

Lorem ipsum ×

Dolor sit ×

⌕ A < >

Content for tab 1

Panel caption

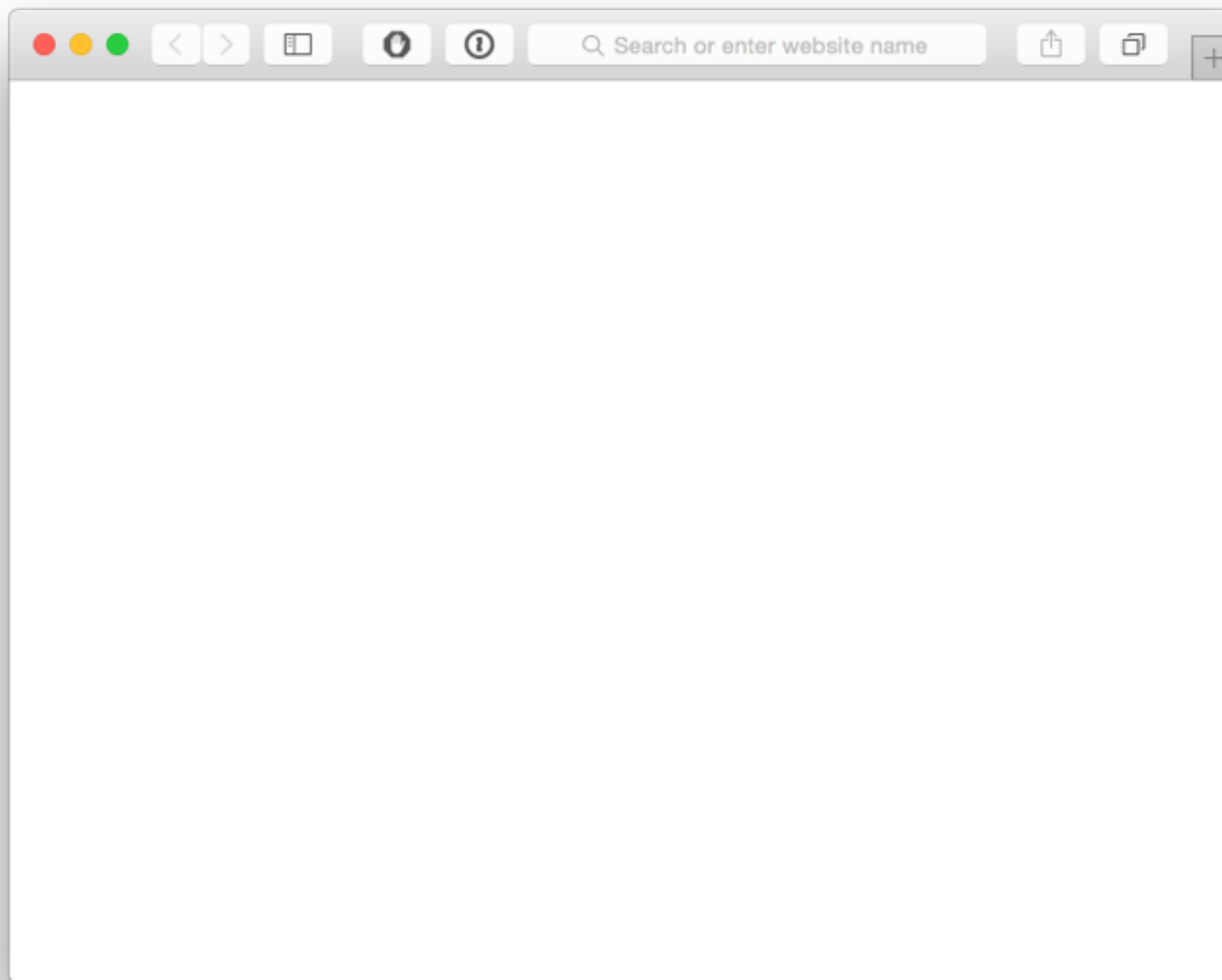
One Two Three

Button

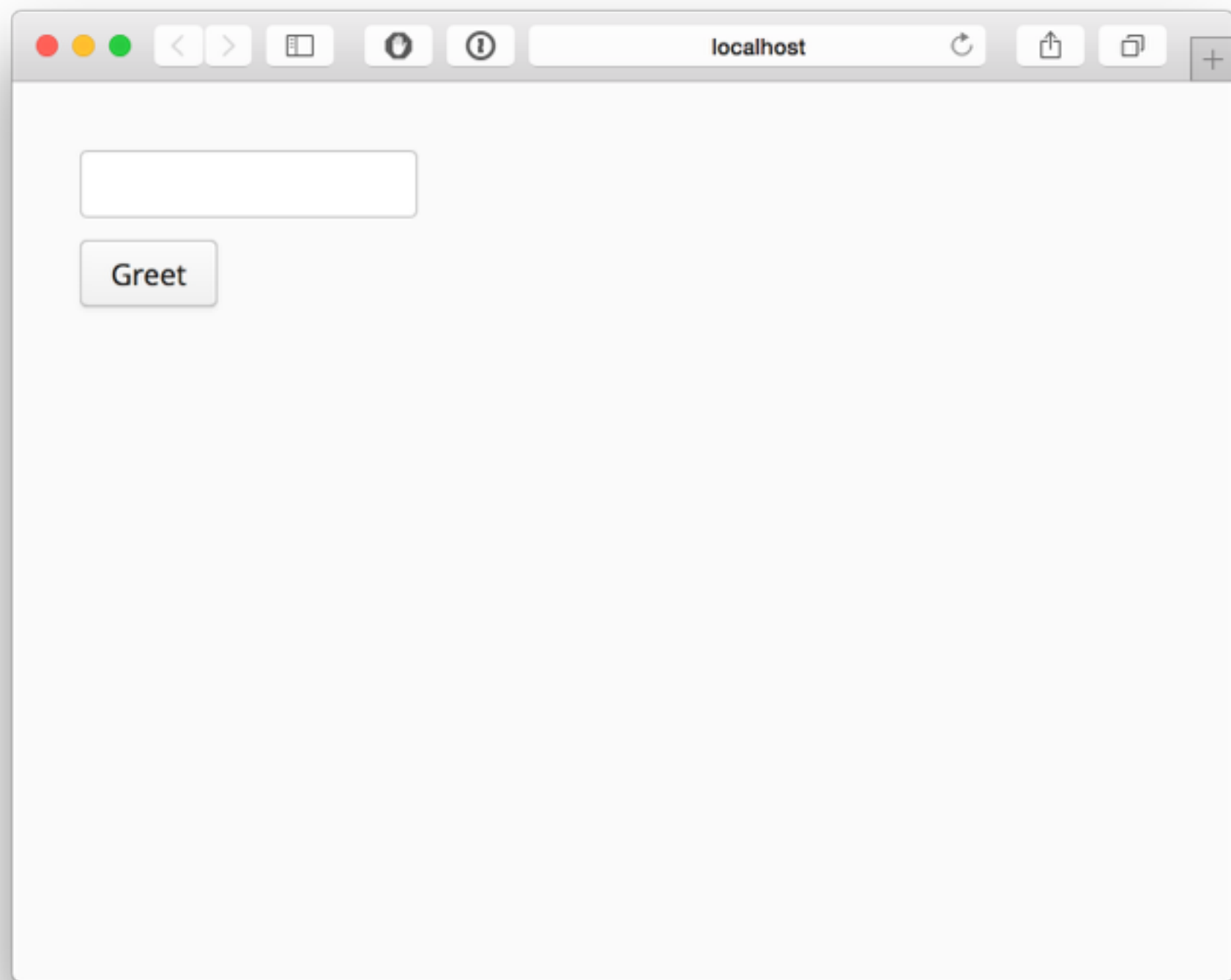
☒ Option One ☐ Option Two ☐ Option Three

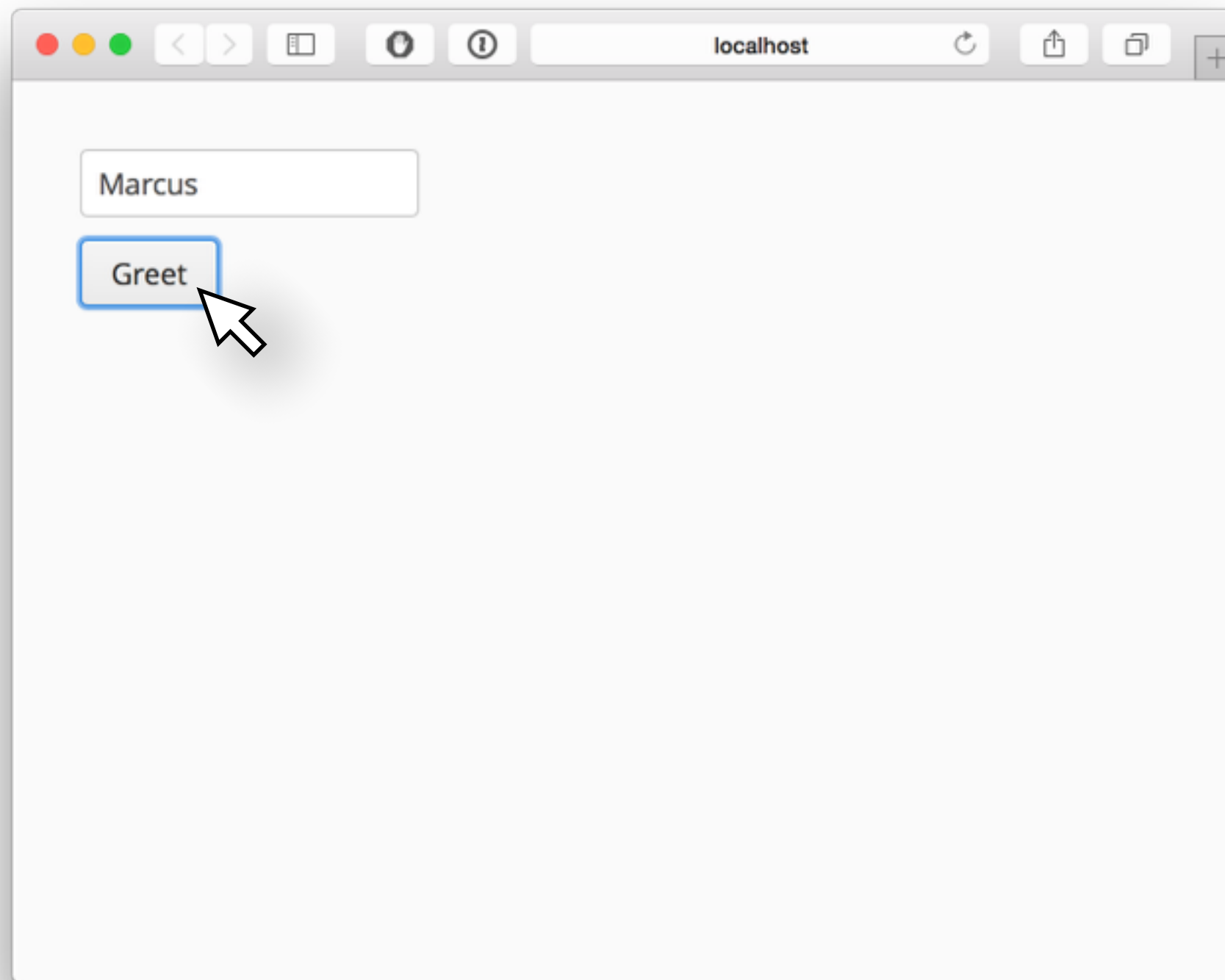
☒ Option One ☐ Option Two ☐ Option Three

```
TextField name = new TextField();  
Button button = new Button("Greet");  
layout.addComponent(name, button);  
  
button.addClickListener(event ->  
    Notification.show("Hi, " + name.getValue()));
```



- ★ Initial HTML
- ★ CSS (theme)
- ★ JavaScript

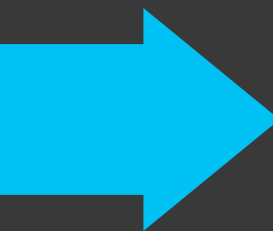




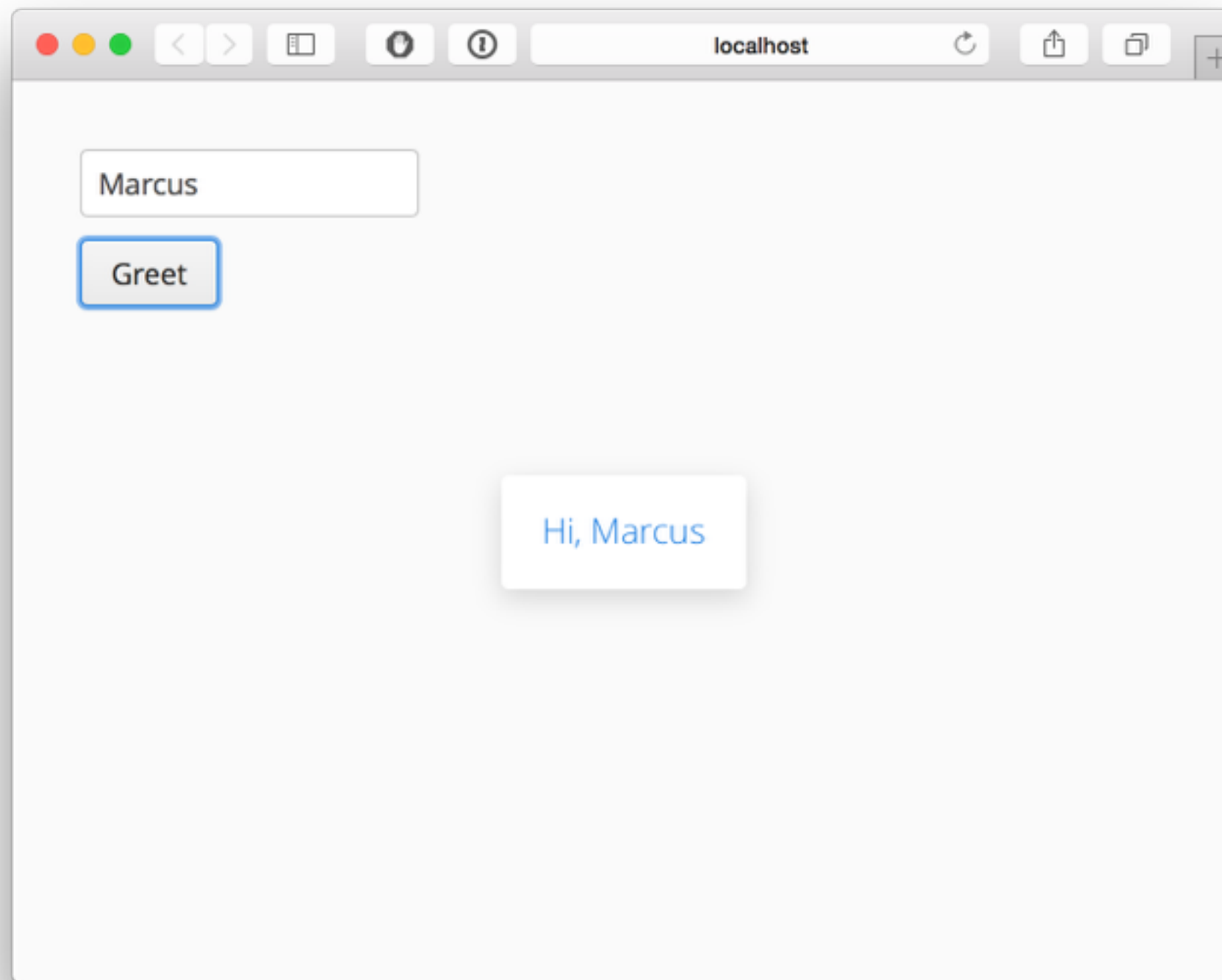
- name="Marcus"
- button clicked


261 bytes

```
TextField name = new TextField();  
Button button = new Button("Greet");  
layout.addComponent(name, button);
```



```
button.addClickListener(event ->  
    Notification.show("Hi, " + name.getValue()));
```

- 
- name="Marcus"
 - button clicked

261 bytes

- 
- Add notification

267 bytes

Main concepts

1. Everything is a **Component**
2. Components are organized by **Layouts**
3. **Data binding** ties Components to your data

Everything is a component

Click

Select your country

No country selected

« < August 2015 > »

Sun Mon Tue Wed Thu Fri Sat

26 27 28 29 30 31 1

2 3 4 5 6 7 8







9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26 27 28 29

30 31 1 2 3 4 5

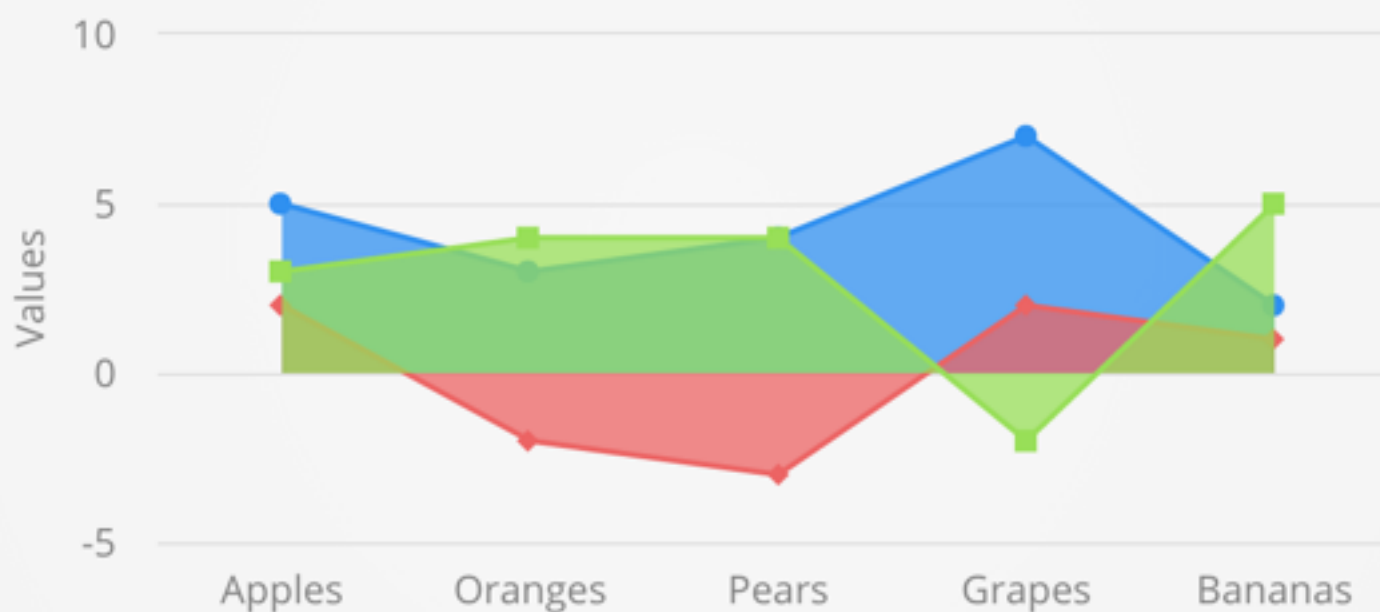
08 : 21 PM

Stock	Price	Price History
QJSH	\$607.59	 -19.90
KVMV	\$741.38	 +13.44
FLTW	\$145.47	 +31.44
FQIY	\$445.36	 +0.77
DLMQ	\$730.66	 -12.32
WNFE	\$352.93	 -57.70

Select the stock to buy/sell

Buy

Sell



Common features

```
setWidth("100%");
```

```
setSizeFull();
```

```
setHeight("50px");
```

```
setSizeUndefined();
```

```
setEnabled(true);
```

```
setVisible(false);
```

Handling user events

```
Button button = new Button("Cancel");
button.addClickListener(new Button.ClickListener() {
    @Override
    public void buttonClick(Button.ClickEvent clickEvent) {
        fieldGroup.discard();
    }
});
```

// Or..

```
button.addClickListener(e -> fieldGroup.discard());
```

**Layouts determine how
Components are shown**

```
Layout layout = new XLayout();  
layout.addComponent(new TextField("First Name"));  
layout.addComponent(new TextField("Last Name"));  
layout.addComponent(new DateField("Birth date"));
```



```
Layout layout = new HorizontalLayout();
```

First Name

Last Name

Birth date



```
Layout layout = new VerticalLayout();
```

First Name

Last Name

Birth date

```
Layout layout = new FormLayout();
```

First Name	<input type="text"/>
Last Name	<input type="text"/>
Birth date	<input type="text"/>

```
Layout layout = new GridLayout(2,2);
```

First Name

Last Name

Birth date



**Nest layouts to create more
complex layouts**

First Name

Marcus

Last Name

Hellberg

Shoe size

10

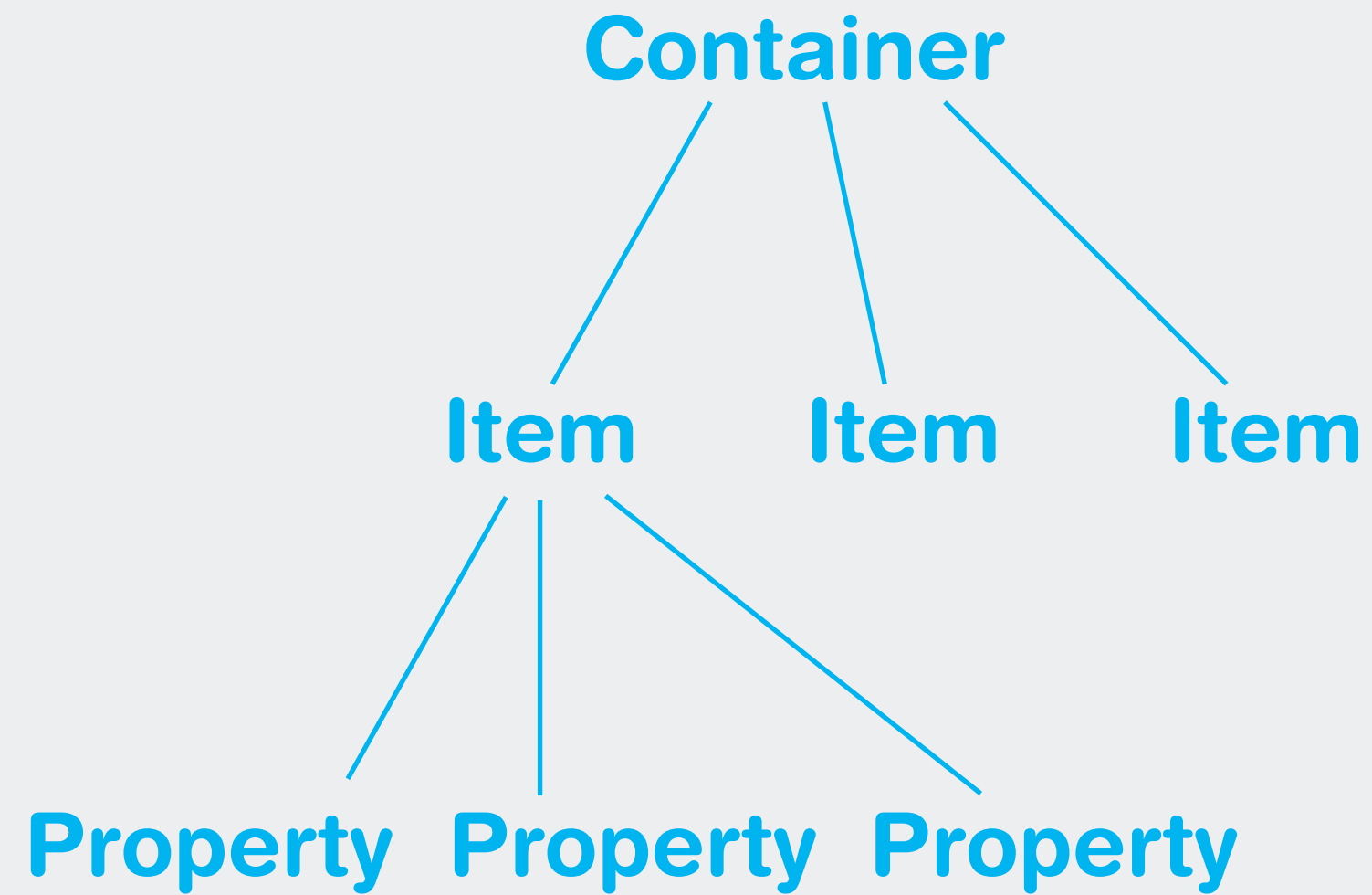
Birth Date



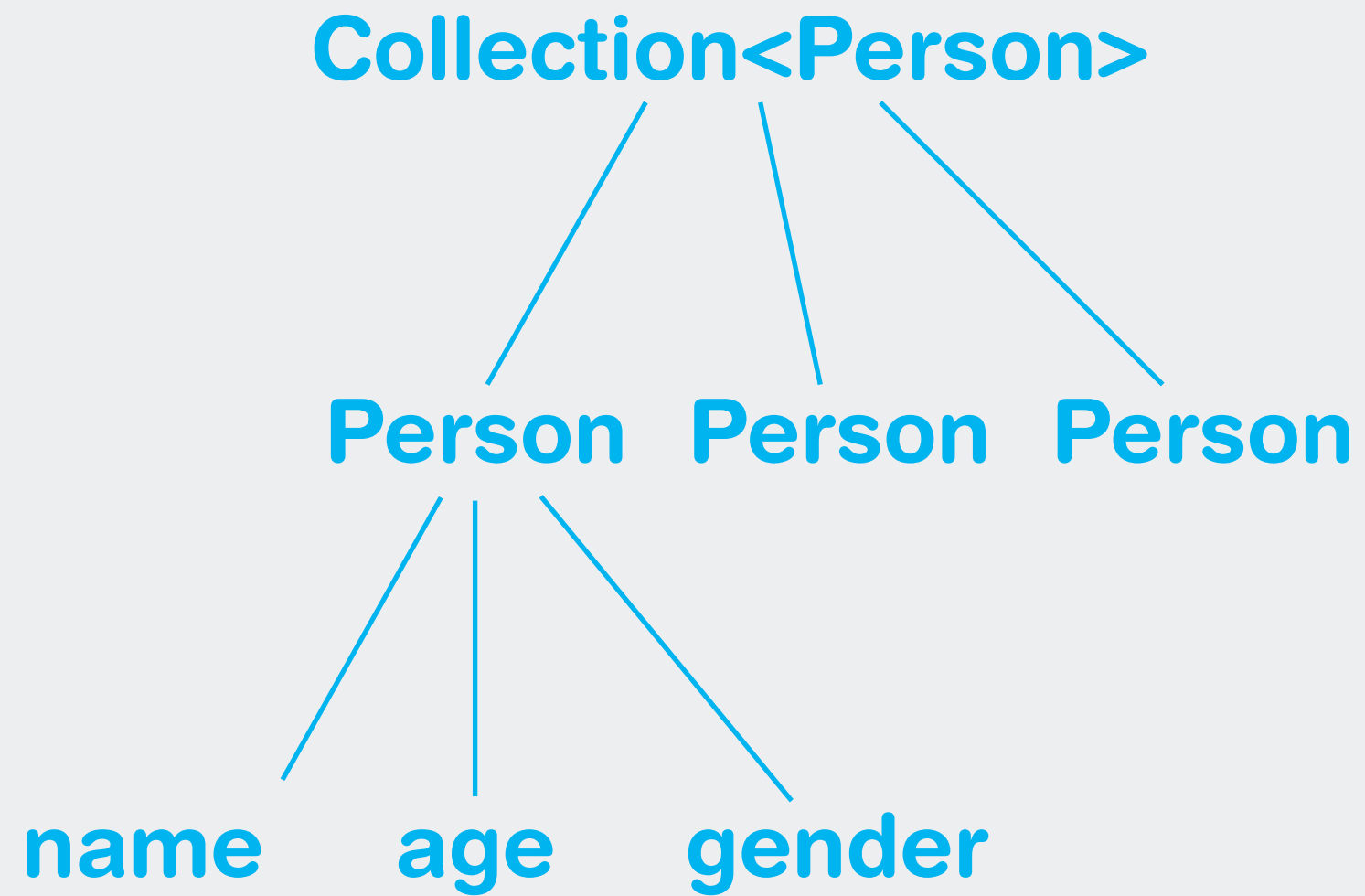
Cancel

 Save

Binding components to data



Basically...



+ listeners

```
@Inject
private PersonService personService;

@Override
protected void init(VaadinRequest request) {

    BeanItemContainer<Person> container =
        new BeanItemContainer<Person>(
            Person.class, personService.findAll());

    Grid grid = new Grid(container);

    setContent(grid);
}
```

First Name	Last Name	Age	Gender
Jonatan	Virkki	57	Male
Minna	Virkki	72	Female
Artur	Ekblad	108	Other
Sauli	Paul	85	Other
Fredrik	Perälä	62	Male
Patrik	Lindström	96	Male
Risto	Murtojärvi	32	Male
Henri	Virtanen	89	Other
Guillermo	Kurki	46	Male
Kim	Mattsson	77	Female
Samuli	Kaitanen	100	Female
Patrik	Paul	105	Female
Artem	Anisimov	89	Female
Dmitrii	Kaksonen	17	Female

```
public class PersonForm extends FormLayout {
```

```
    private final TextField firstName;  
    private final TextField lastName;  
    private final TextField shoeSize;  
    private final DateField birthDate;
```

```
    public PersonForm() {
```

```
        firstName = new TextField("First name");  
        lastName = new TextField("Last name");  
        shoeSize = new TextField("Shoe size");  
        birthDate = new DateField("Birth date");
```

```
        addComponents(firstName, lastName, shoeSize, birthDate);
```

```
    }
```

```
}
```

```
public class Person {
```

```
    private String firstName;  
    private String lastName;  
    private int shoeSize;  
    private Date birthDate;
```

```
    // Getters and setters
```

```
FieldGroup fieldGroup = new FieldGroup(new BeanItem<>(new Person()));  
PersonForm formLayout = new PersonForm();  
fieldGroup.bindMemberFields(formLayout);
```

First name

Last name

Shoe size

Birth date



Main Layout

Project

skynet (~\workspace\skynet)

.idea

alarm

dashboard

src

main

java

org.vaadin.marcus.

AlarmsWindow

DashboardUI

SensorLayout

TriggerForm

resources

webapp

target

dashboard.iml

pom.xml

README.md

sensor

service

shared

.gitignore

pom.xml

skynet.iml

External Libraries

DashboardUI.java x

TriggerForm.java x

AlarmsWindow.java x

3

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

52

53

54

64

65

66

69

70

71

72

73

+

import ...

@Theme("dashboard")

@Title("Skynet dashboard")

public class DashboardUI extends UI {

private

 MessageService messageService = MessageService.getInstance();

private

 Map<Sensor, SensorLayout> sensors = new HashMap<>();

private

 MHorizontalLayout sensorsLayout;

@Override

protected void init(VaadinRequest vaadinRequest) {

}

@Subscribe

public void sensorUpdateListener(SensorUpdatedEvent evt) {...}

@Subscribe

public void sensorOfflineListener(SensorOfflineEvent evt) {...}

@Subscribe

public void sensorTriggeredListener(SensorTriggeredEvent evt) {...}

@WebServlet(value = "/*", asyncSupported = true)

@VaadinServletConfiguration(productionMode = false, ui = DashboardUI.class,

public static class Servlet extends VaadinServlet {

}

Sensor Layout

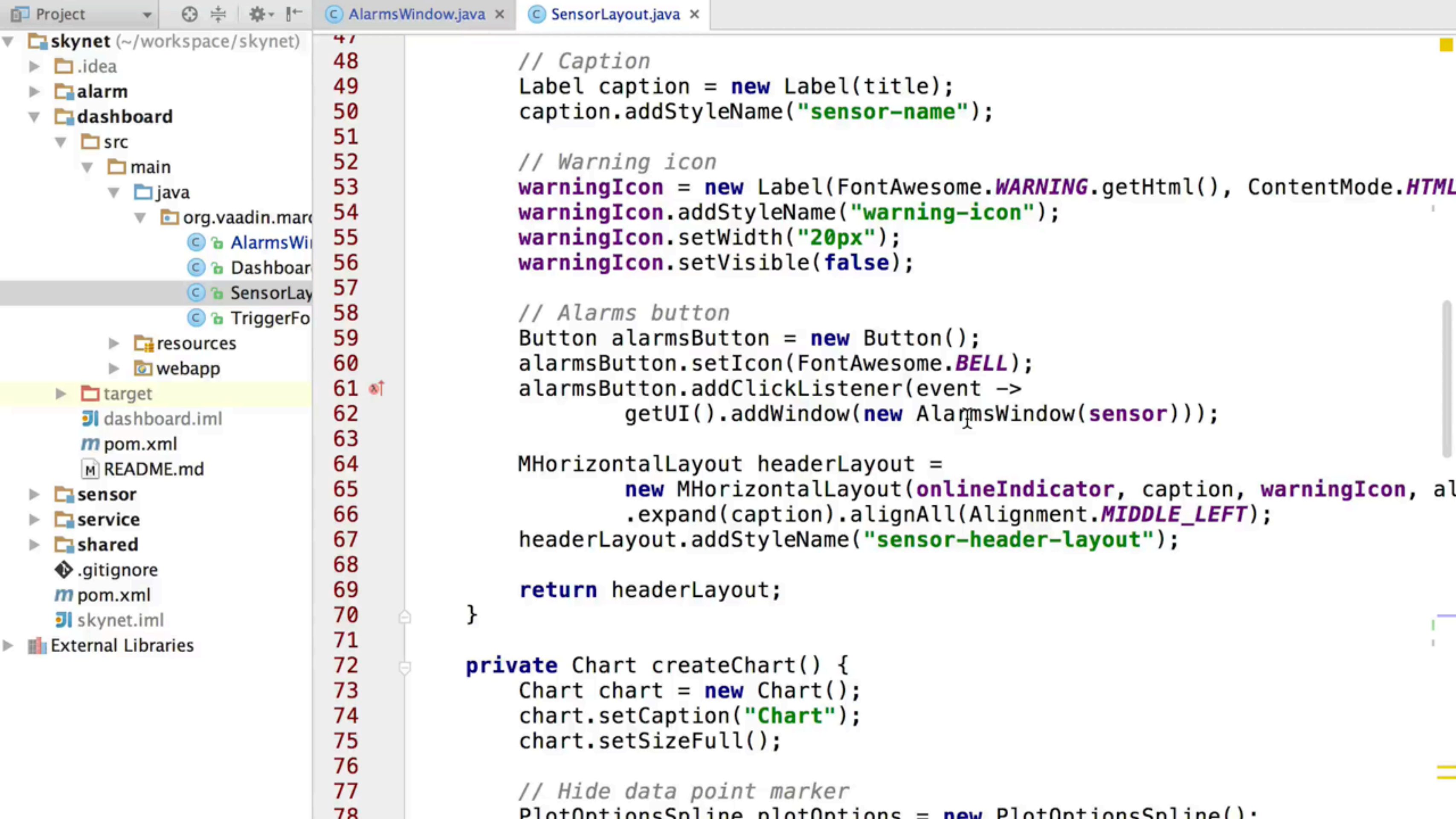
▼ skynet (~/.workspace/skynet)

- ▶ .idea
- ▶ alarm
- ▼ dashboard
 - ▼ src
 - ▼ main
 - ▼ java
 - ▼ org.vaadin.marc
 - ▶ resources
 - ▶ webapp
 - ▶ target
 - dashboard.iml
 - pom.xml
 - README.md
 - ▶ sensor
 - ▶ service
 - ▶ shared
 - .gitignore
 - pom.xml
 - skynet.iml

External Libraries

```
31 private MessageService messageservice = MessageService.getInstance();
32 private Map<Sensor, SensorLayout> sensors = new HashMap<>();
33 private MHorizontalLayout sensorsLayout;
34
35 @Override
36 protected void init(VaadinRequest vaadinRequest) {
37     Label heading = new Label("Sensor Dashboard");
38     heading.addStyleName(ValoTheme.LABEL_H1);
39
40     sensorsLayout = new MHorizontalLayout().withFullWidth();
41     setContent(new MVerticalLayout(heading).expand(sensorsLayout));
42
43     messageService.registerListener(this);
44     addDetachListener(detach -> messageService.unregisterListener(this));
45 }
46
47 @Subscribe
48 public void sensorUpdateListener(SensorUpdatedEvent evt) {
49     Sensor sensor = evt.getSensor();
50
51     if (sensors.containsKey(sensor)) {
52         access(() -> sensors.get(sensor).addDataPoint(sensor.getValue(), sensor.getTime()));
53     } else {
54         SensorLayout sensorLayout = new SensorLayout(sensor);
55         sensorLayout.addDataPoint(sensor.getValue(), sensor.getTime());
56
57         access(() -> {
58             this.sensorsLayout.addComponent(sensorLayout);
59             sensors.put(sensor, sensorLayout);
60         });
61     }
62 }
```

Editing alarms



github.com/vaadin-marcus/skynet

??!

vaadin }> thinking of U and I

marcus@vaadin.com / @marcushellberg