

# Distributed systems with MsgFlo

Flow-Based Programming over message queues

Internet of Things devroom  
FOSDEM 2017

Jon Nordby  
[@jononor](https://twitter.com/jononor)

 flowhub

[msgflo.org](http://msgflo.org)  
[@flowhub\\_io](https://twitter.com/flowhub_io)

# Basic MQTT communication

```
// Motor.cpp
```

```
mqtt.subscribe("motor/start", start_motor);
```

```
// Button.cpp
```

```
mqtt.publish("motor/start", button_pressed);
```

+ Works OK

- Connections hidden in code
- Hardcoded functionality

# MsgFlo participant

```
// Motor.cpp
```

```
mqtt.subscribe("motor/start", start_motor);  
mqtt.publish("msgflo/discover", motor_info);
```

```
// Button.cpp
```

```
mqtt.publish("msgflo/discover", button_info);  
mqtt.publish("button/pressed", button_pressed);
```

- + Devices describe themselves
- + Reusable components
- ! Something need to connect the topics

# MsgFlo discovery message

```
{  
  "component": "fosdem2017/Button",  
  "role": "button",  
  "id": "button",  
  "label": "This is a button, it can be pressed",  
  "icon": null,  
  "outports": [{  
    "id": "pressed",  
    "type": "boolean"  
    "queue": "button/pressed"  
  }],  
  "inports": []  
}
```

# System of participants

```
# motorcontrol.fbp
```

```
button(Button) PRESSED → START motor(Motor)
```

```
...
```

```
motor RUNNING → RED warninglight(Lamp)
```

```
# bind the queues together
```

```
$ msgflo-setup --forever motorcontrol.fbp
```

# Foreign participants

```
# existingdevice.yml
```

```
component: c-base/siri
```

```
label: c-base siri data rescue probe
```

```
inports:
```

```
  openurl:
```

```
    queue: siri/open
```

```
    type: string
```

```
# Send MsgFlo discovery message
```

```
# on behalf of device
```

```
$ msgflo-register-foreign existingdevice.yml
```



# Live programming

```
# start runtime
```

```
$ msgflo --graph motorcontrol.fbp
```

... Open Flowhub in browser

<http://app.flowhub.io/#runtime/endpoint?protocol=websocket&address=ws://localhost:3569>

- + Direct access via URL
- + Only browser needed



# Convenience libraries

`msgflo-nodejs`: JavaScript on Node.js

`noflo-runtime-msgflo`: NoFlo

`msgflo-cpp`: C++11 on Linux

`msgflo-python`: Python 2.x

`msgflo-arduino`: Arduino on ESP8266

! `msgflo-rust`: Rust, *AMQP only*



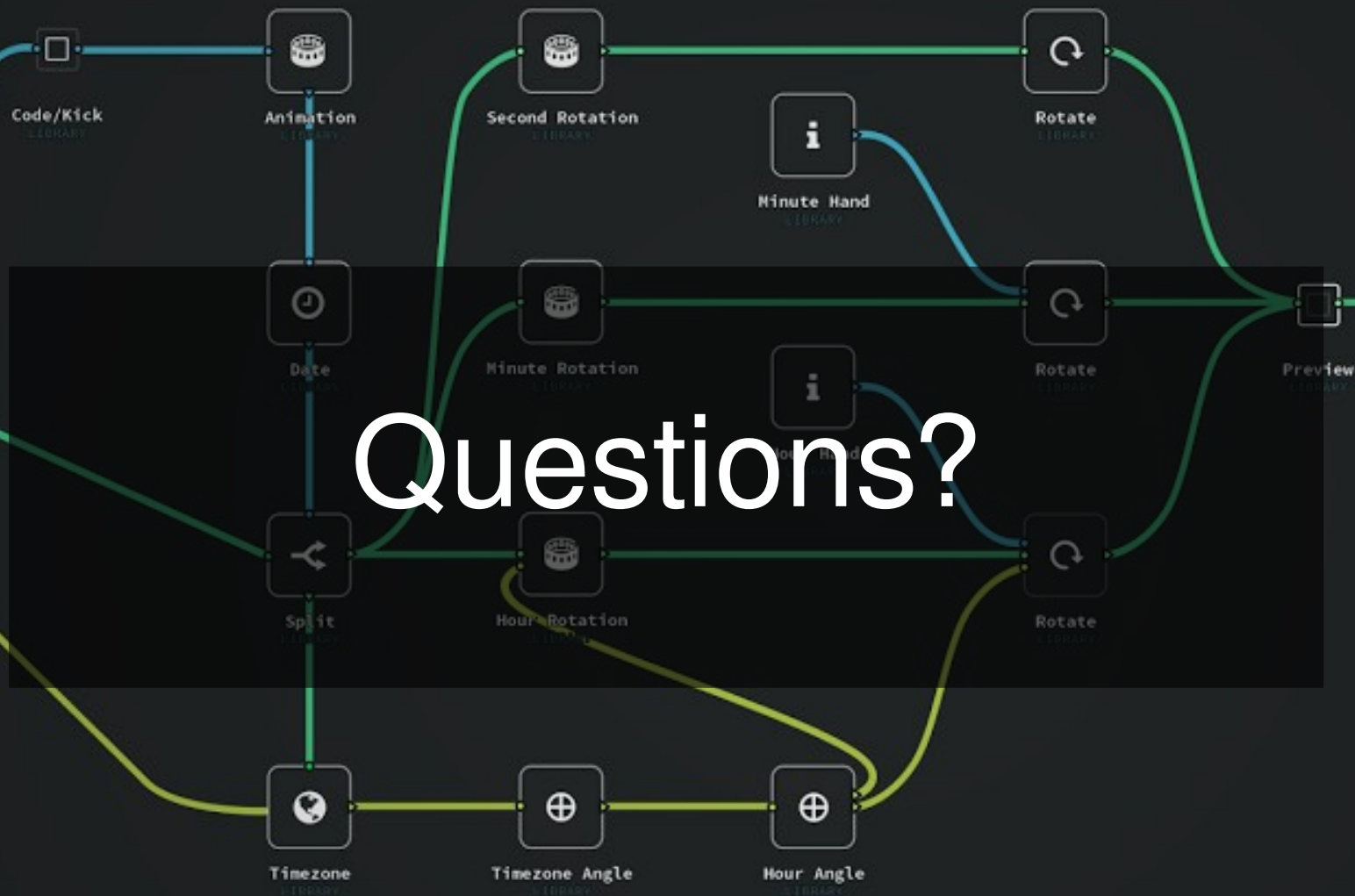
# Current status

- + Programming model, JavaScript libraries, AMQP support = battle tested
- . MQTT deployed in Berlin, Oslo hackerspaces
- ! MQTT SSL support not tested
- ! Live programming not much tested

# Future

- Support RabbitMQ routing on MQTT
- MsgFlo 1.0: Freeze discovery protocol
- Flowhub showing changes from outside automatically (no refresh)
- FBP protocol forwarding. Live-programming all the way down!

Questions?



[msgflo.org](https://msgflo.org)

 flowhub

[@jononor](https://twitter.com/jononor)

Message payloads  
agnostic  
but – discovery message can have info

Participant modelling  
best practices  
FBP + FSM style

# Component library model & tools