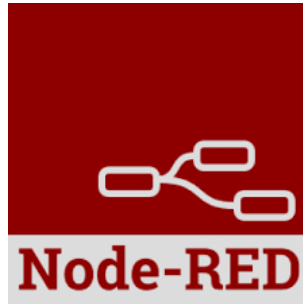




Node-Red (Workshop)

Der Fluss der Dinge

oder

Klick dir dein Internet der Dinge zusammen

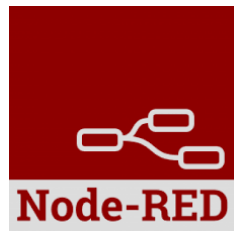
Uwe Berger

`bergeruw@gmx.net`

Uwe Berger



- Beruf: Softwareentwickler
- Freizeit: u.a. mit Hard- und Software rumspielen
- Linux seit ca. 1995
- BraLUG e.V.
- bergeruw@gmx.net

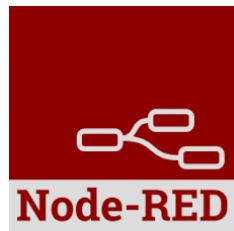


Das Problem...

...Internet of Things (IoT)...:

- Jedes neue „Ding“ bringt eine neue API mit...
- Jeder neue Dienst bringt eine neue API mit...
- Für eine IoT-Lösung müssen diese (vielen) unterschiedlichen Dinge und Dienste verknüpft werden...!

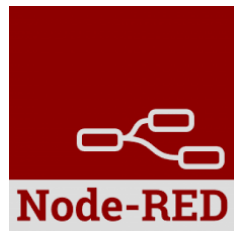
Es kostet Zeit, immer wieder darüber nachzudenken, wie man das Zusammenspiel vieler unterschiedlicher Schnittstellen implementiert!



...eine Lösung

Ein Baukasten...:

- ...in dem die APIs bereits implementiert und untereinander „normalisiert“ sind...
- ...und in dem man die „Dinge“ einfach nur noch zusammenstecken muss...



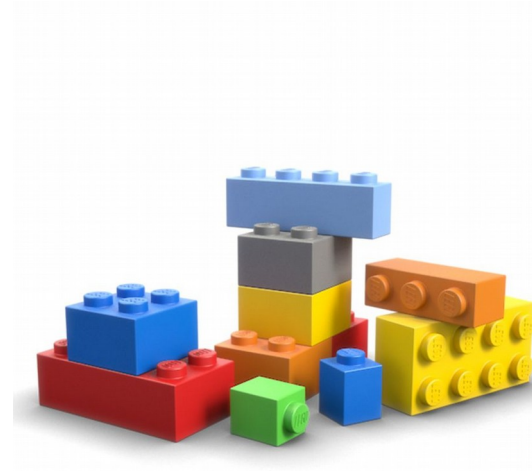
...eine Lösung

Ein Baukasten...:

- ...in dem die APIs bereits implementiert und untereinander „normalisiert“ sind...
- ...und in dem man die „Dinge“ einfach nur noch zusammenstecken muss...

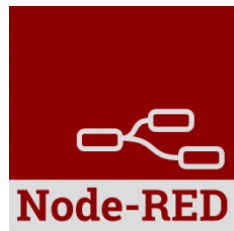
→ Node-Red könnte da eine Lösung sein!

Hääää,... Baukasten...?



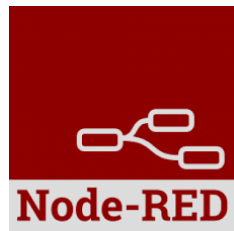
...ich bin doch ein Programmier-
Nerd...?#!.+?!....?





Inhalt

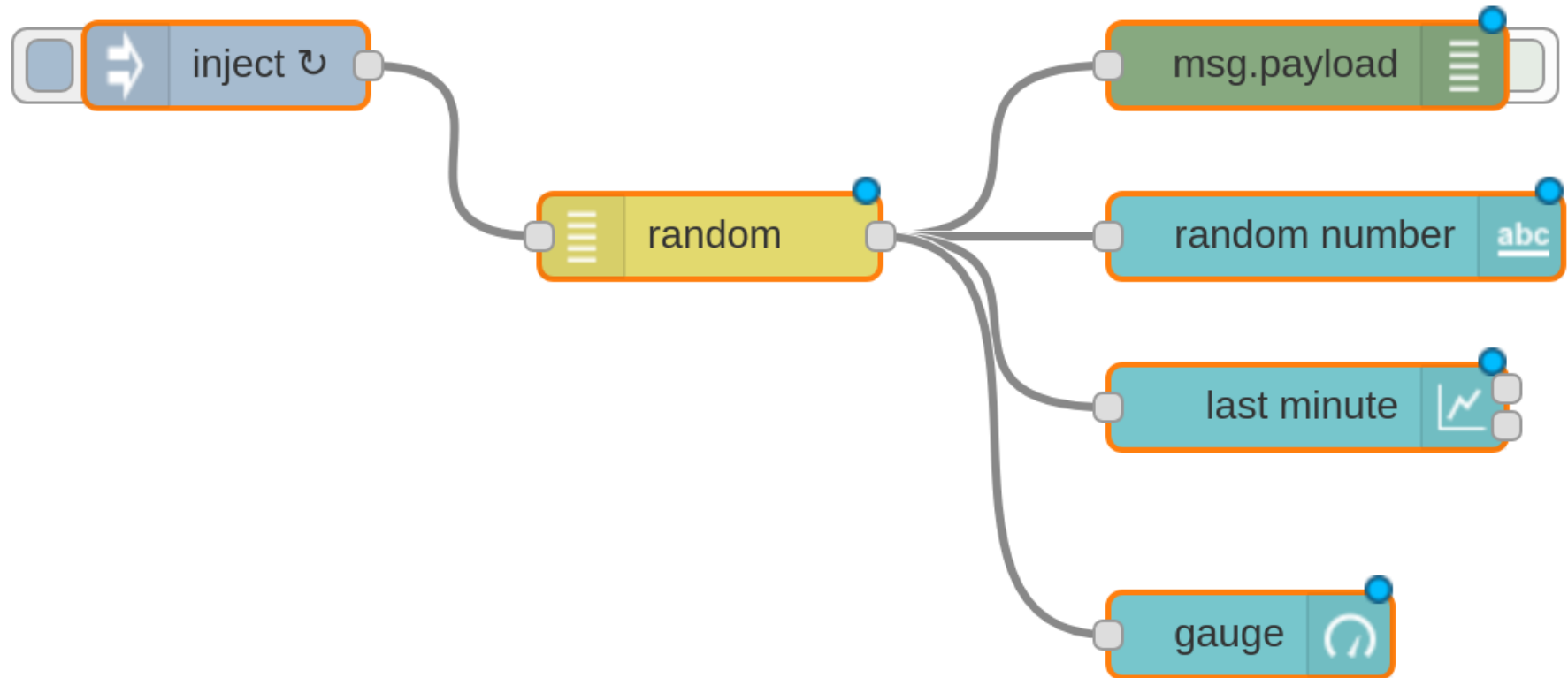
- Was ist Node-Red?
- Installation, Konfiguration von Node-Red
- Arbeiten mit Node-Red...

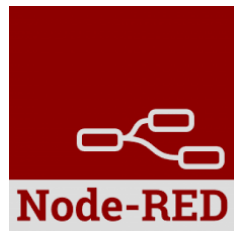


Node-Red („Kennzahlen“)

- Erstes Release: 2013
- IBM; Nick O‘Leary, Dave Conway-Jones
- mittlerweile Open Source (Apache License 2.0)
- Basiert auf Node.js (damit auch plattformübergreifend)
- Browser-basierte Entwicklungsumgebung
- Konzept der Datenflussprogrammierung...

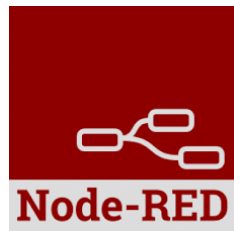
...Datenflussprogrammierung...





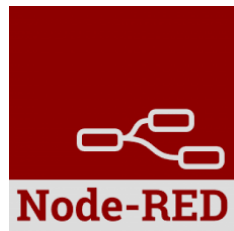
Node-Red ist...

- ein Baukasten zum Zusammensetzen von Anwendungen...
- ...via Drag and Drop
- Ideal für Prototyp-Entwicklung (proof of concept)
- Ideal zur Implementierung von einfachen Abläufen
- einfach zu erweitern
- mehr als ein Baukasten zum Zusammenstellen von Weboberflächen!



Node-Red ist...

- ...**keine** Laufzeitumgebung für unternehmenskritische Anwendungen



Node-Red installieren...

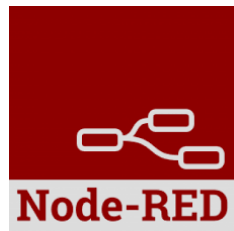
- Node.js und npm (node package manager) via apt installieren
- Node-Red via npm installieren

```
$ sudo apt-get install nodejs npm
```

```
...
```

```
$ sudo npm install -g --unsafe-perm node-red
```

```
...
```



Node-Red konfigurieren...

- Datei **settings.js** (Reihenfolge der Suche)
 - über Kommandozeilenparameter
 - in **/home/<user>/ .node-red/**
 - im Installationsverzeichnis von Node-Red
- Was sollte man konfigurieren:
 - Verzeichnisse
 - Sicherheitseinstellungen(!)
- ...ansonsten ist originale settings.js gut kommentiert...

Node-Red konfigurieren...

User/Passwort für IDE setzen

→ <http://nodered.org/docs/security.html>

- settings.js

```
adminAuth: {  
  type: "credentials",  
  users: [{  
    username: "admin",  
    password: "$2a$08$f1sy5y42uj/YG0iASTE420FQl3tfcPDpf42JPFoa3UMZrTshjRm/q",  
    permissions: "*"   
  }]  
},
```

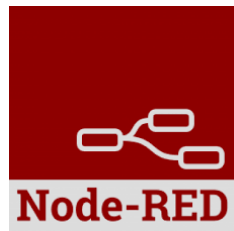
- Passwort-Hash erzeugen:

```
> sudo npm install -g node-red-admin
```

```
> node-red-admin hash-pw
```

```
Password: ...
```

```
$2a$08$f1sy5y42uj/YG0iASTE420FQl3tfcPDpf42JPFoa3UMZrTshjRm/q
```

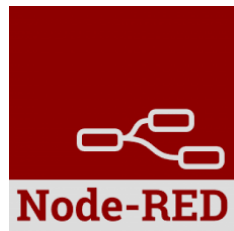


Node-Red konfigurieren...

Verzeichnis für statischen Content (z.B. Bilder, Icons) setzen

- settings.js

```
httpStatic: '/home/bergeruw/node-red-static/',
```



Node-Red starten...

- Kommandozeile:

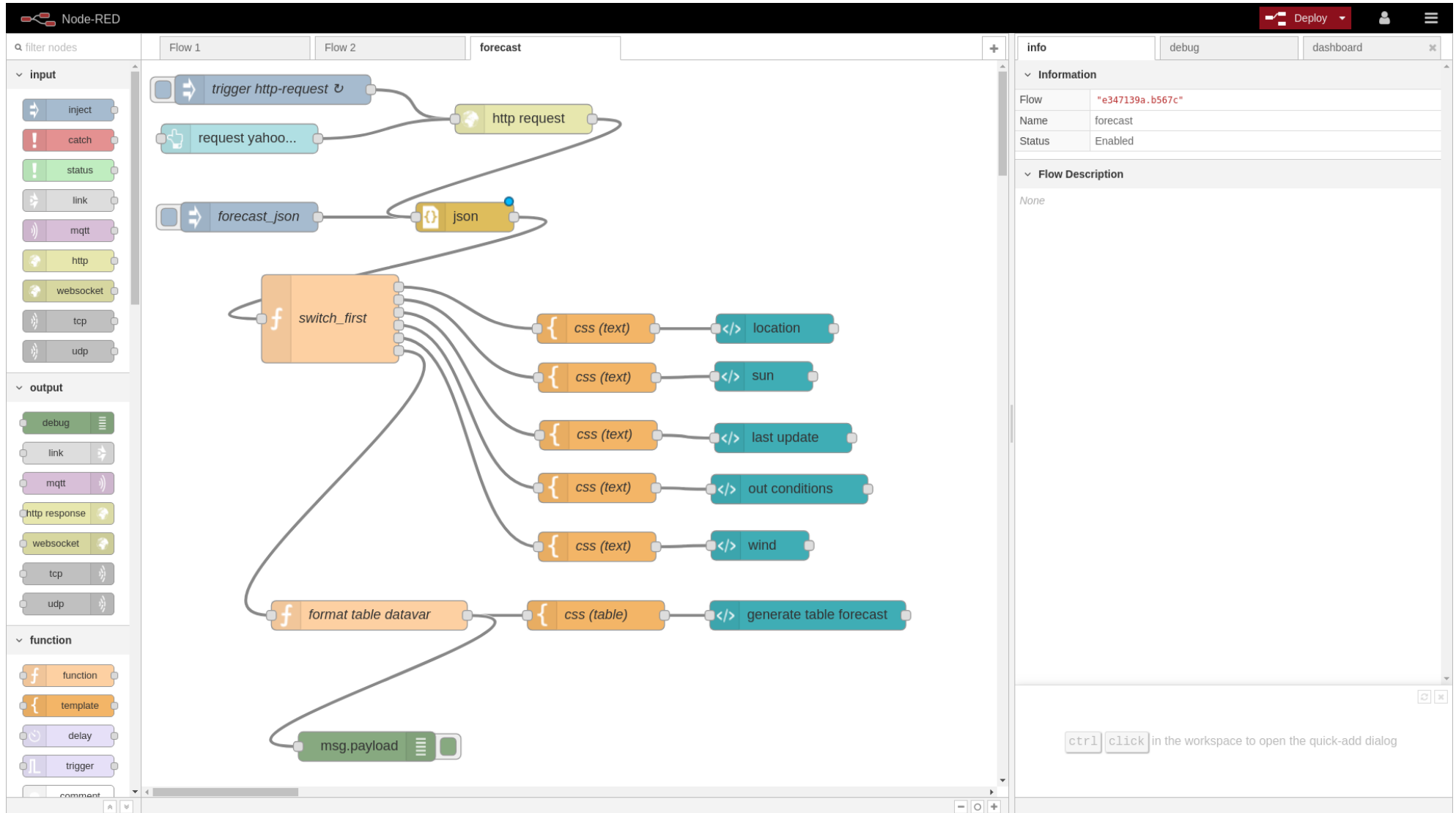
```
$ whereis node-red
node-red: /usr/local/bin/node-red

$ node-red
...
```

- ...oder als Dienst:

- RTFM...
- ...mit systemd z.B.:
→ <https://diyprojects.io/node-red-installation-configuration-ubuntu-16-04-lts/>

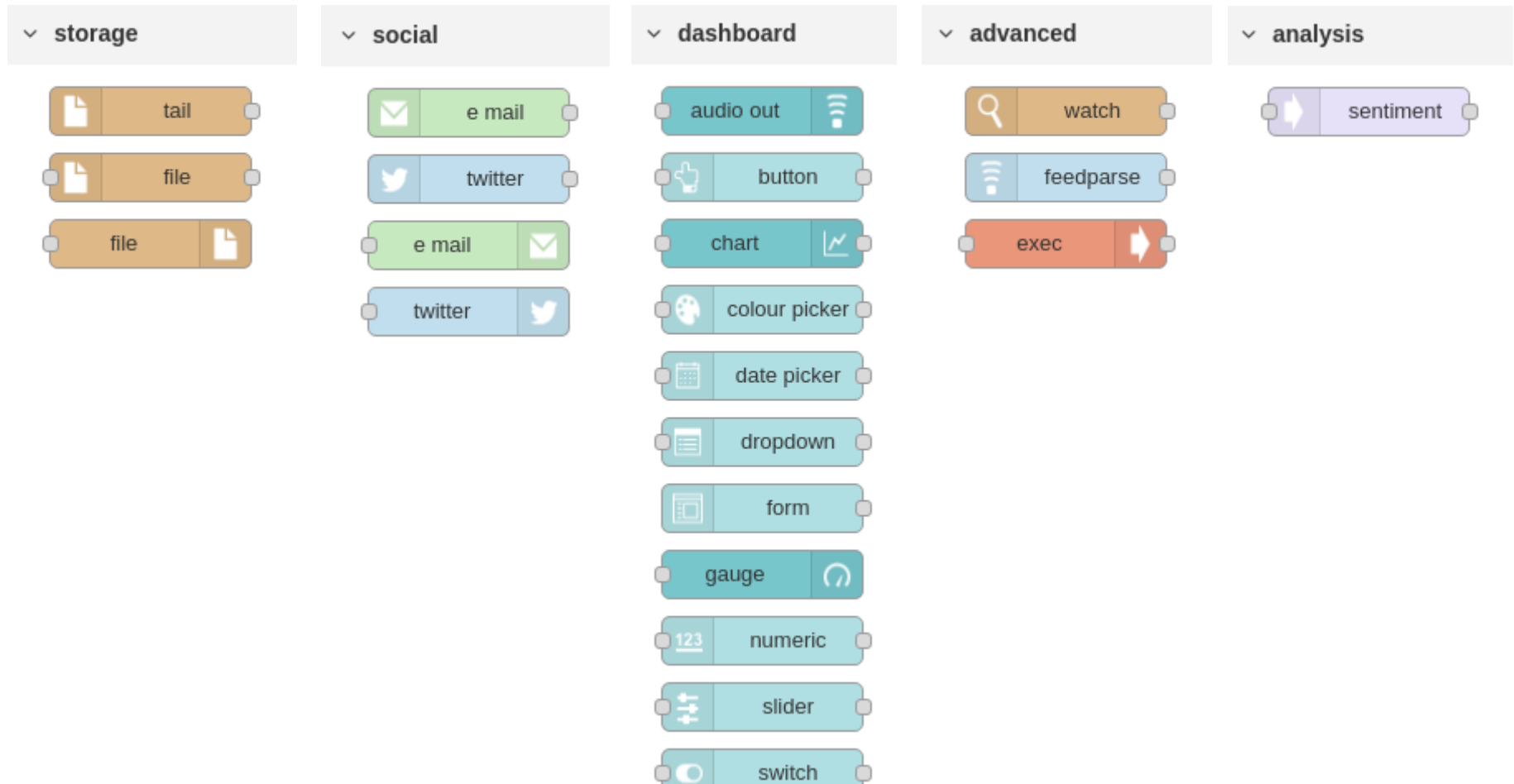
http://localhost:1880/



Nodes (Eingabe/Verarbeitung/Ausgabe)

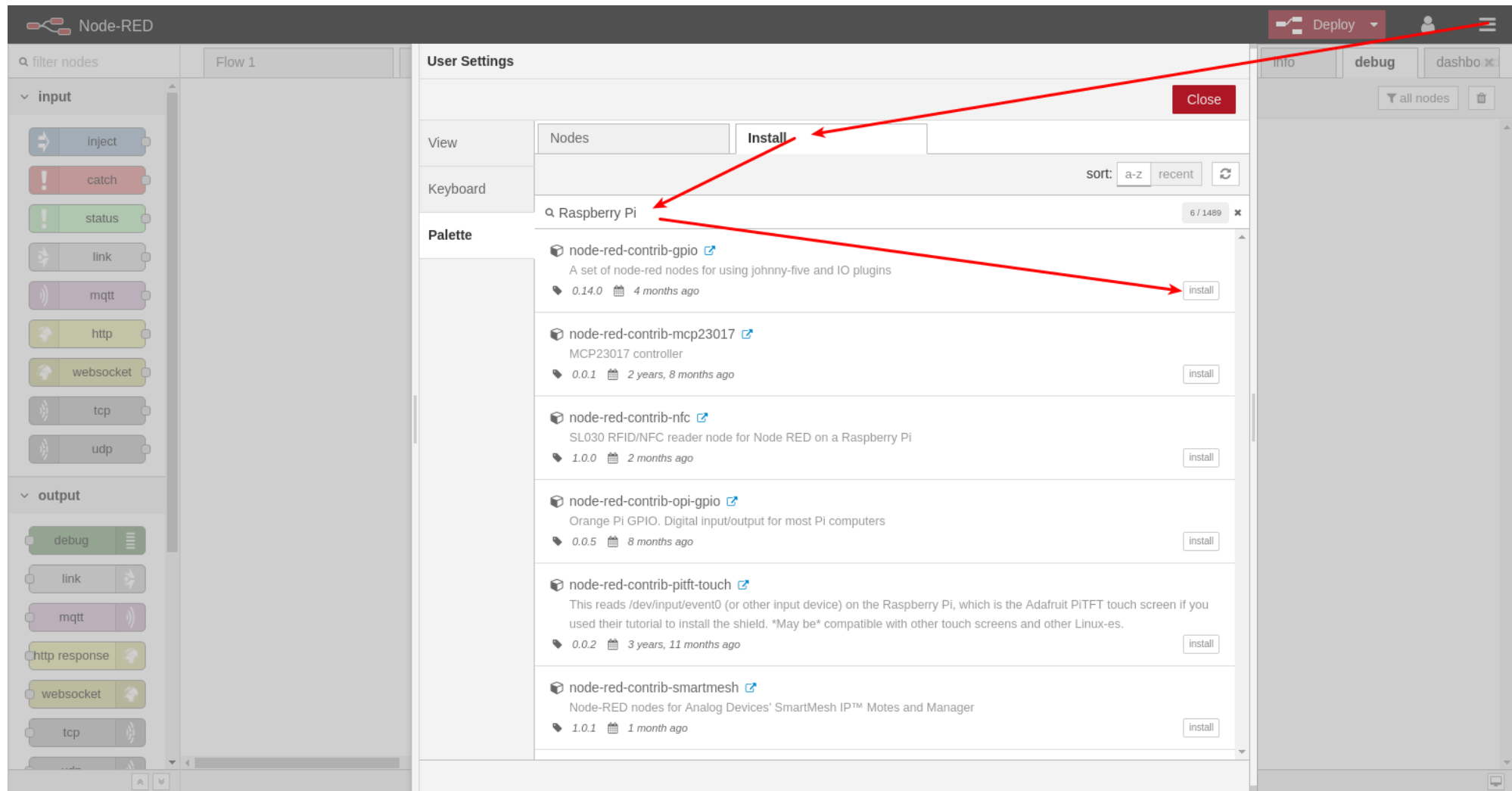


...viele weitere Nodes

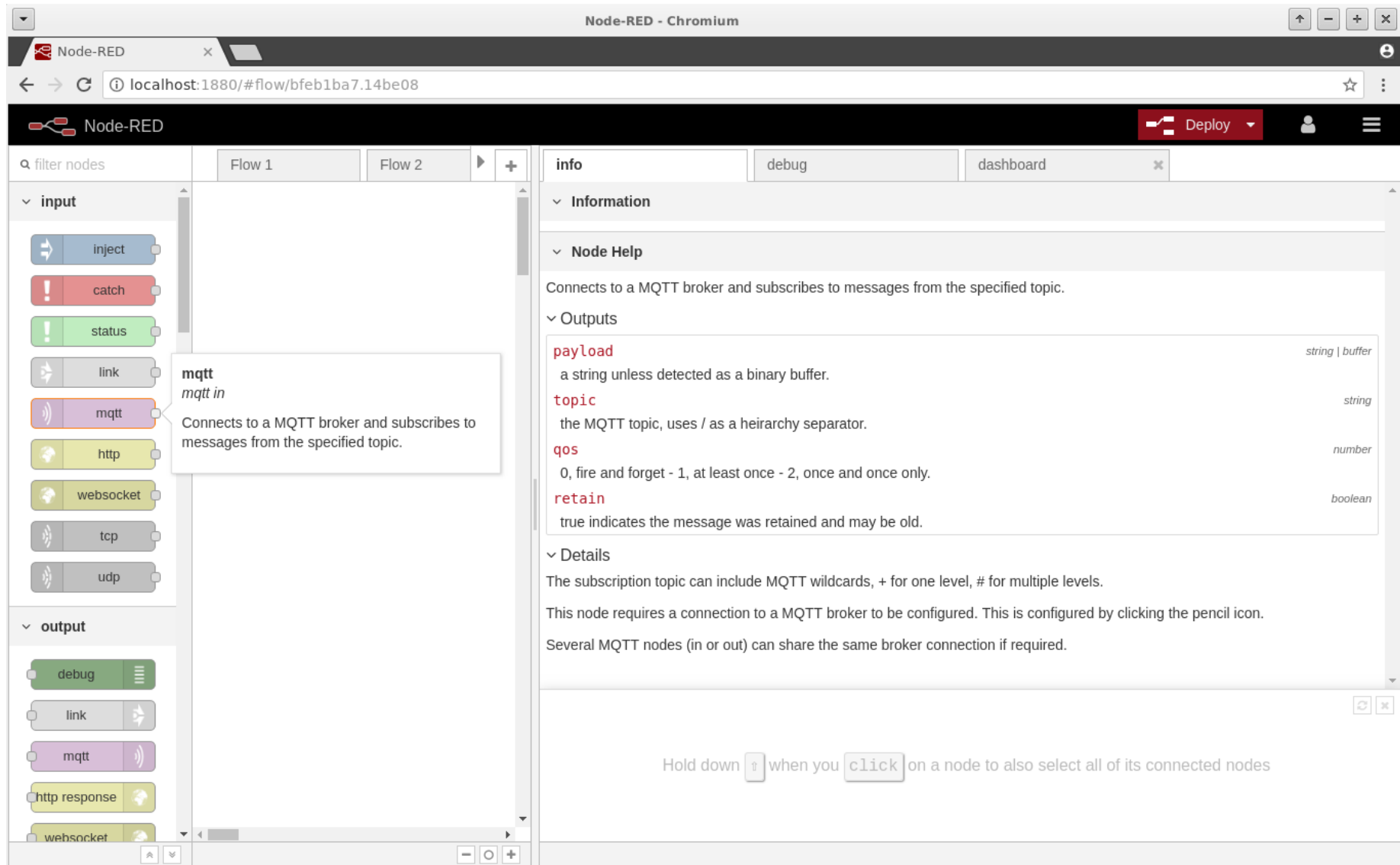


...für Raspberry Pi gibt es da auch was...!

Nodes nachinstallieren...



Hilfe zu Nodes



Message-Objekt...

- In der Regel immer:
 - msg.topic
 - msg.payload
- ...plus node-spezifische Eigenschaften



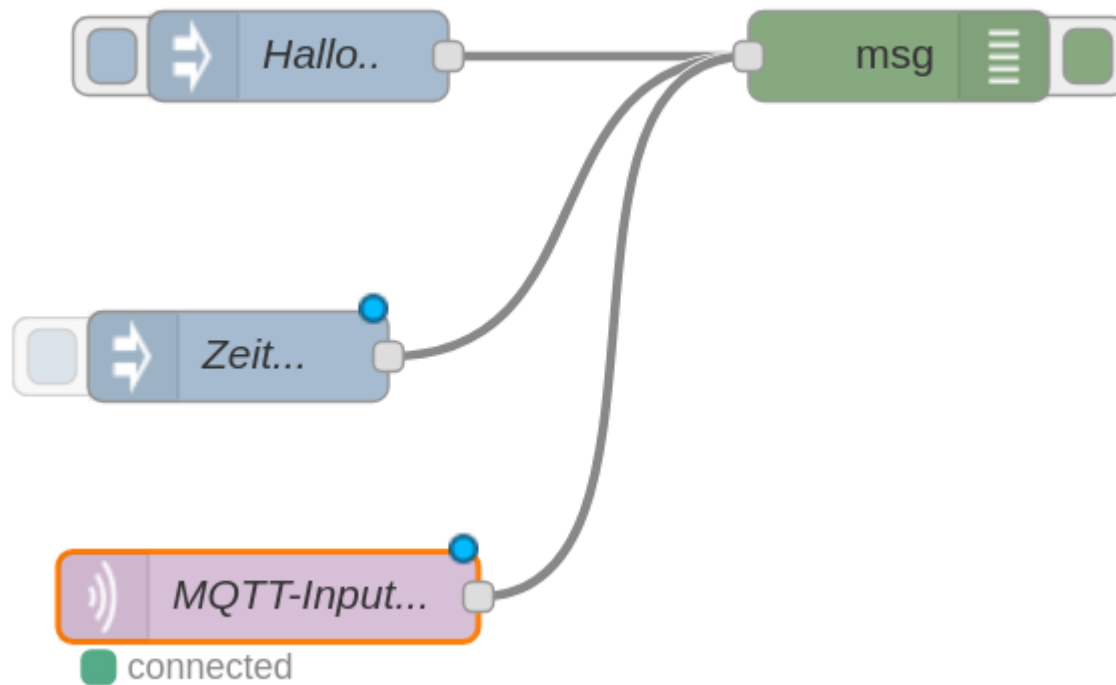
The screenshot shows the Node-RED interface with three nodes on the left: 'Hallo..', 'Zeit...', and 'MQTT-Input...'. The 'MQTT-Input...' node is highlighted with an orange border and a green 'connected' status indicator. On the right, the debug console displays three message objects, each corresponding to one of the nodes. Red arrows point from each node to its respective message object in the console.

```
19.6.2018, 10:05:16 node: 5777616f.3c79a
Hallo-Welt-Nachricht : msg : Object
▶ { _msgid: "3347cf3f.ccb83", topic: "Hallo-Welt-Nachricht", payload: "Hallo Welt!" }

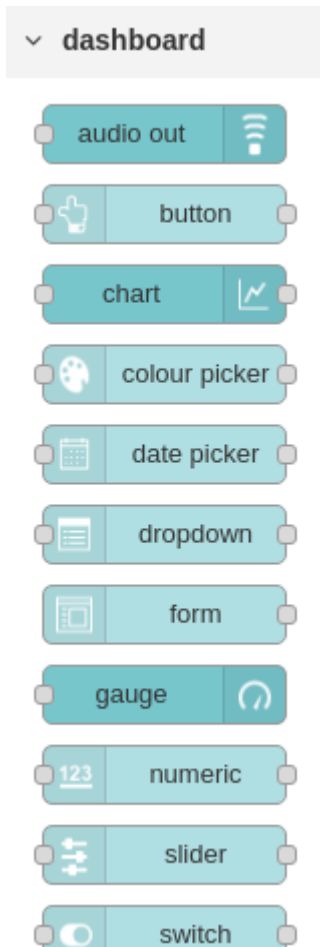
19.6.2018, 10:05:18 node: 5777616f.3c79a
Zeit... : msg : Object
▶ { _msgid: "e414830b.1beb8", topic: "Zeit...", payload: 1529395518250 }

19.6.2018, 10:05:22 node: 5777616f.3c79a
random/ : msg : Object
▶ { topic: "random/", payload: "12345", qos: 0, retain: false, _topic: "random/" ... }
```

Live... - ein „Hello World!“

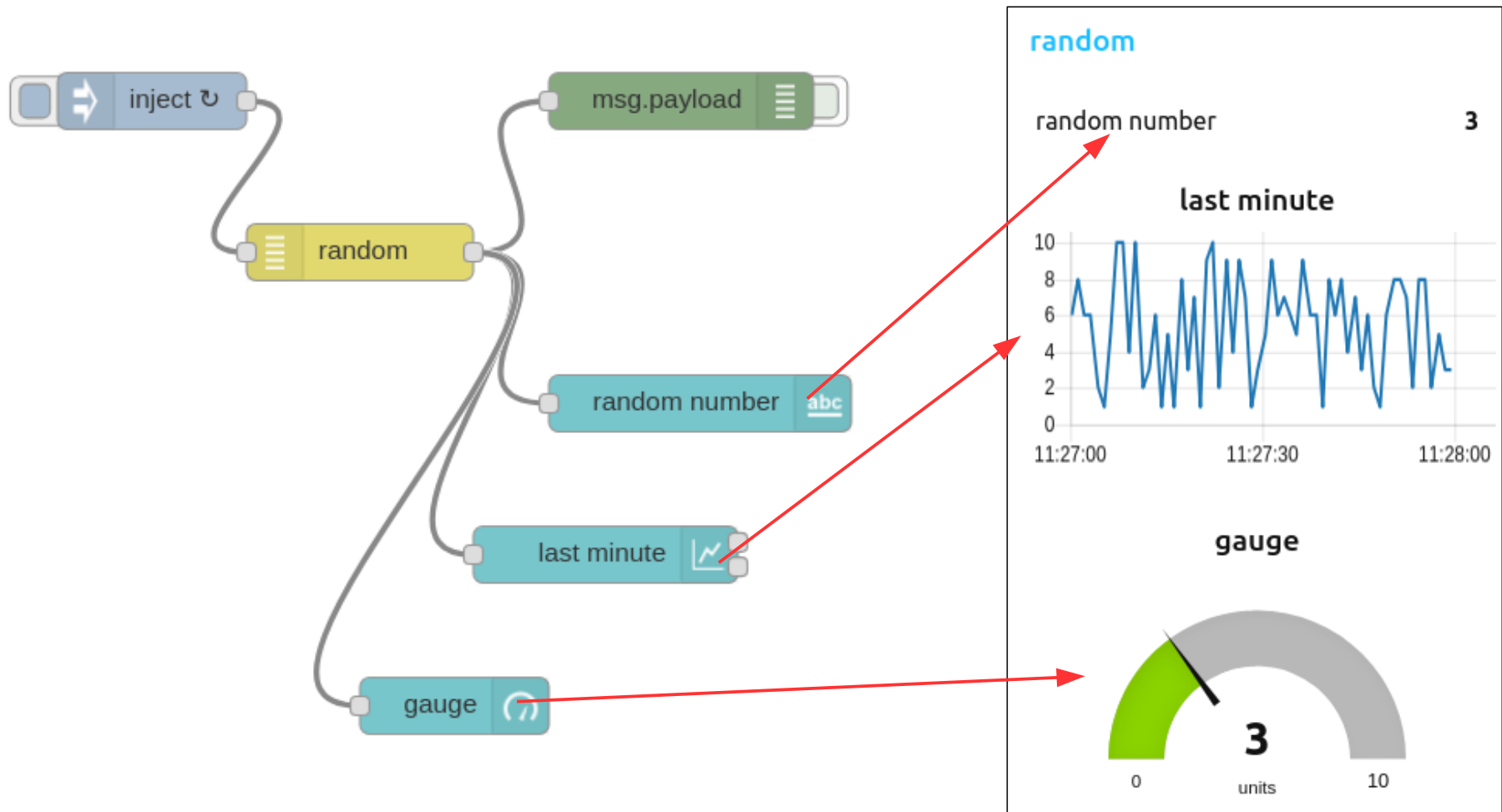


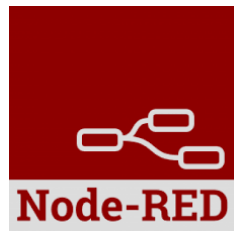
Dashboard



- ...nachinstallieren... (node-red-dashboard)
- u.a. diverse Ein-/Ausgabe-Elemente für Weboberflächen (mit Node-Red)
- <http://localhost:1880/ui/>

Live... ein kleines Dashboard





Stichworte für eigene Experimente...

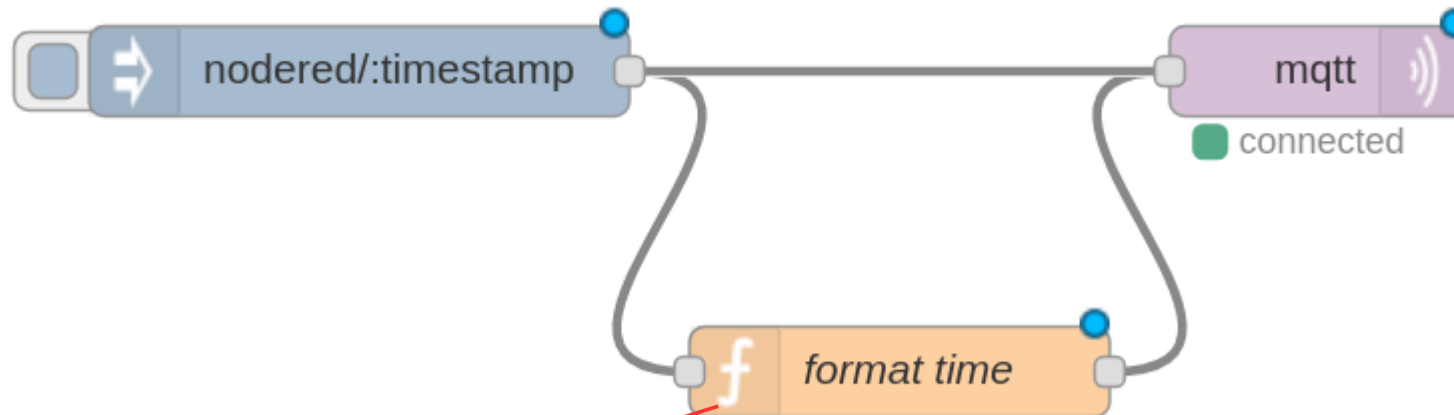
Arbeiten mit Flows

- Export/Import von Flows
- Subflows

Programmieren

- Function-Nodes (die mit eigenem JavaScript-Code „befüllt“ werden können)
- Variablenreichweite:
 - Node-weit: `context.set()`, `context.get()`, ...
 - Flow-weit: `flow.set()`, `flow.get()`, ...
 - Node-Red-weit: `global.set()`, `global.get()`, ...

Function-Node

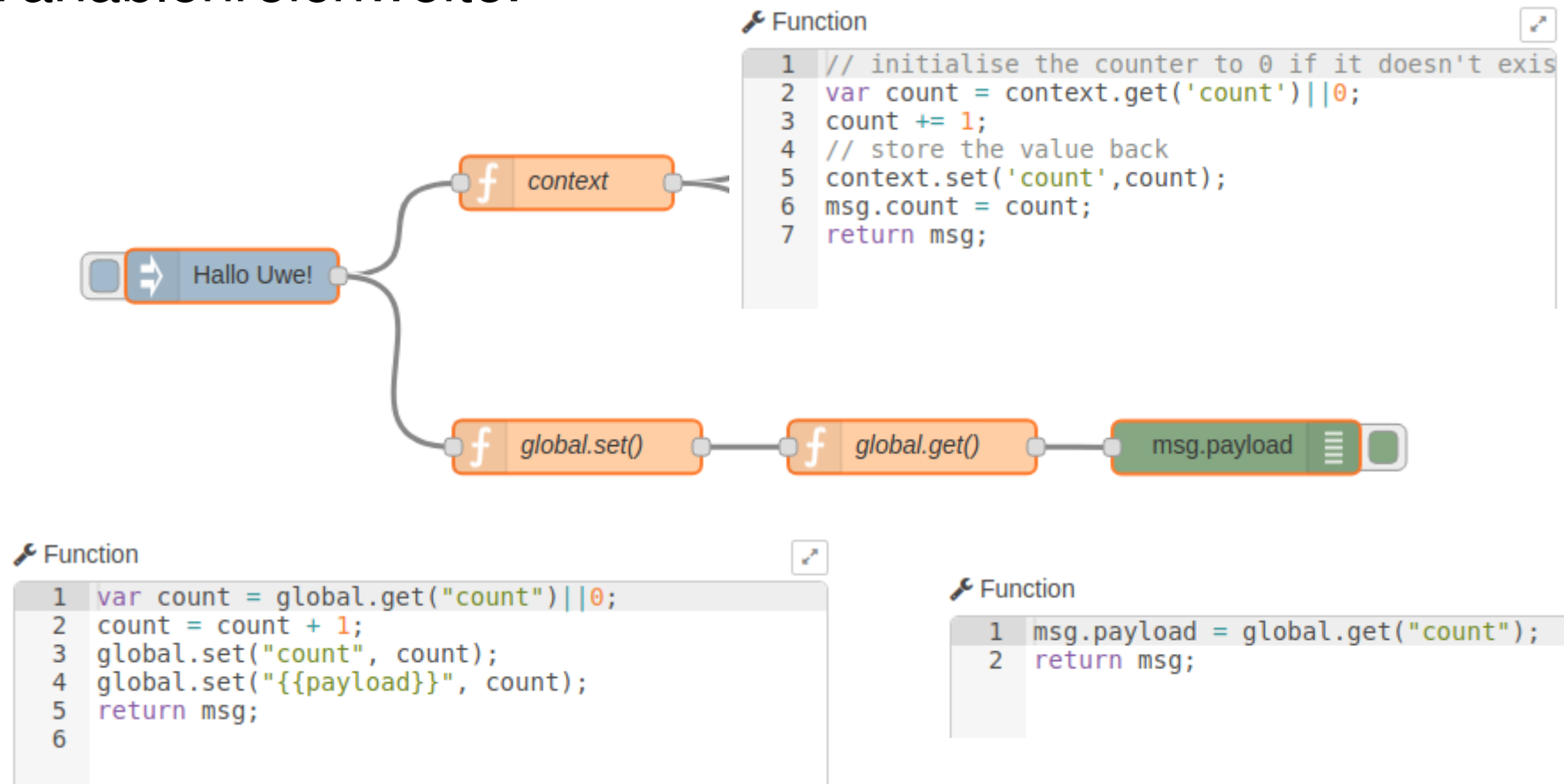


Function

```
1 var date_time = new Date(msg.payload);  
2  
3 msg.payload="Zeitstempel: "+ date_time;  
4 msg.topic=msg.topic+"date_time/";  
5  
6 return msg;
```

Function-Node

Variablenreichweite:



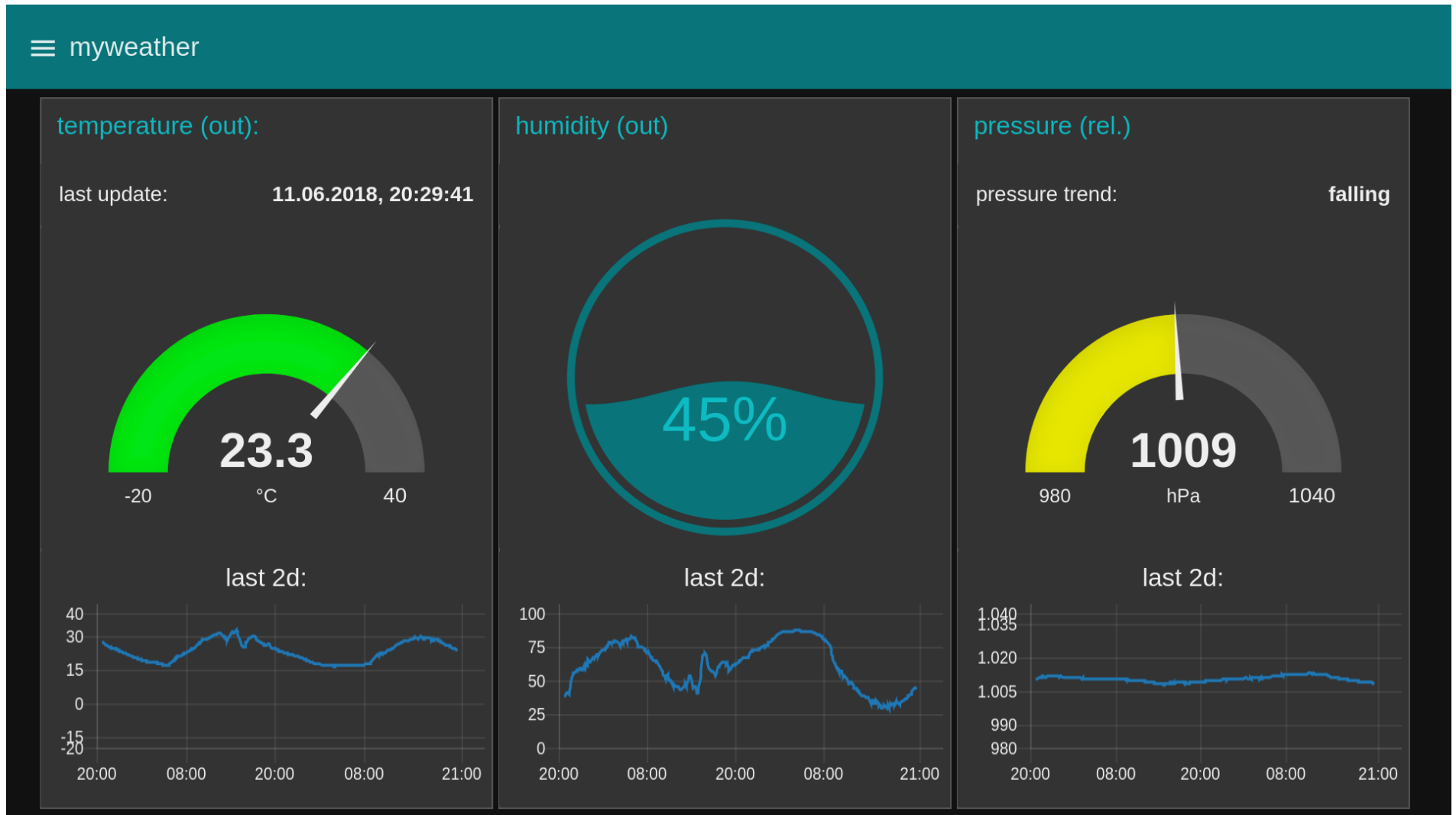
Beispiel: Computer

≡ computers

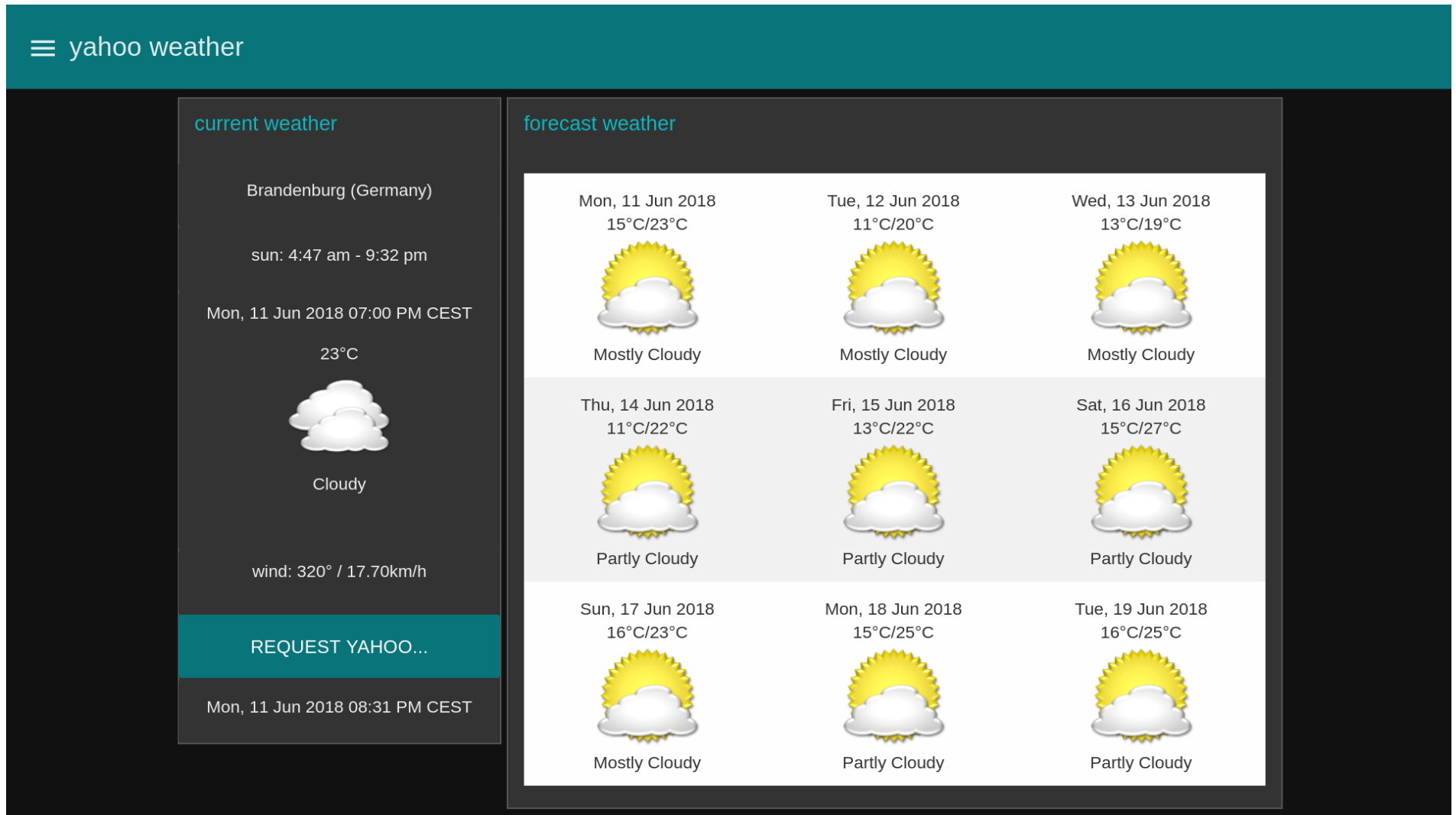
computers

hostname	last seen	uptime	load			free	ram [kByte]			swap [kByte]		processes
			1min	5min	15min		share	buffer	total	free	total	
samsung	2018/06/10 14:46:01	0 days, 1:01:35	0.13	0.12	0.27	203620	47956	167828	2051668	2983348	2988028	397
dockstar	2018/06/11 20:45:01	26 days, 14:15:26	0.09	0.07	0.05	267080	232	140944	996700	102396	102396	123
dellesus	2018/06/08 20:22:01	0 days, 0:51:00	0.25	0.15	0.27	975628	49352	587980	3980192	7807552	7807552	500
nodered	2018/06/11 20:45:02	26 days, 14:15:25	0.06	0.02	0.00	511808	48452	91192	961752	102396	102396	158
rasputin	2018/06/11 20:45:01	26 days, 14:15:27	0.09	0.17	0.15	20216	0	99636	447416	102396	102396	81
banane	2018/06/11 20:45:01	26 days, 14:15:24	0.00	0.01	0.05	589268	0	78316	993780	524284	524284	83
tuxedo	2018/06/11 20:45:01	0 days, 0:22:01	1.02	0.96	0.87	13682224	356468	77836	16301844	8388604	8388604	511
zerow	2018/06/11 20:45:01	9 days, 22:29:50	0.01	0.01	0.00	177572	24996	49180	492620	102396	102396	93

Beispiel: Wetter



Beispiel: Wettervorhersage



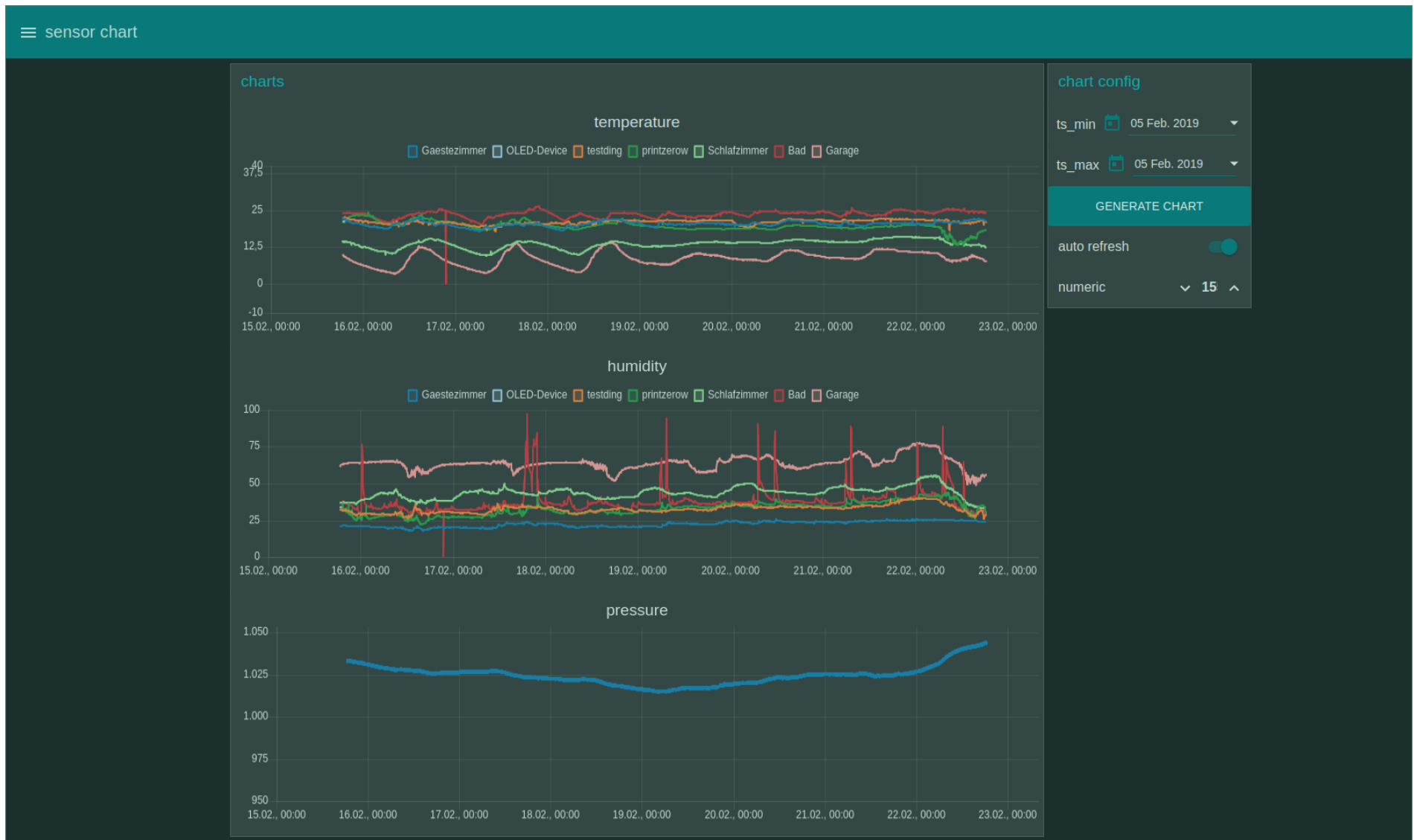
Beispiel: Sensoren

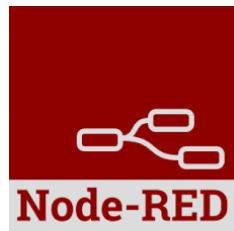
≡ sensors

sensors

name	alias	type	last seen	temp. [°C]	hum. [%]	press. [hPa]	dewp. [°C]
esp8266-9982412	Bad	dht22	2018/06/11 20:39:59	27.9	43.3		
esp8266-10441060	Gaestezimmer	dht22	2018/06/11 20:37:08	26.5	27.8		
esp8266-8671893	OLED-Device	dht22	2018/06/11 20:37:38	26.7	39.8		
esp8266-9981731	Garage	dht22	2018/06/11 20:40:08	27.2	51.4		
esp8266-12130765	Schlafzimmer	dht22	2018/06/11 20:40:18	26.9	41.4		
esp8266-12129880	testding	bme280	2018/06/11 20:37:42	25.8	46.2	1009.8	13.4

Beispiel: Sensoren

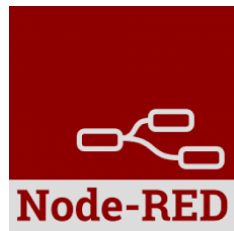




Workshop → was wollen wir machen?

Vorschläge:

- Ein wenig mit der Hardware des Raspberry rumspielen (via MQTT)
- Wetterdaten/-vorhersage von Openweathermap anfordern und visualisieren
- Daten aller Rechner sammeln/visualisieren (via MQTT)
- Einen kleinen Gruppen-Chat bauen (via MQTT)
- ...andere Vorschläge?



Beispiel-Flows für den Workshop

- <https://www.bralug.de/wiki/Benutzer:Bergeruw>
→ „2 CLT2019...” → Link: „Flows“
- Import eines Flows in Node-Red:
 - Dateiinhalt (eines der Beispiele) in Zwischenablage kopieren
 - In Node-Red-IDE:
 - Symbol „drei Striche“ (oben rechts neben „Deploy“)
 - Import → Clipboard → Strg+v → Import

„Workshop-Hardware“

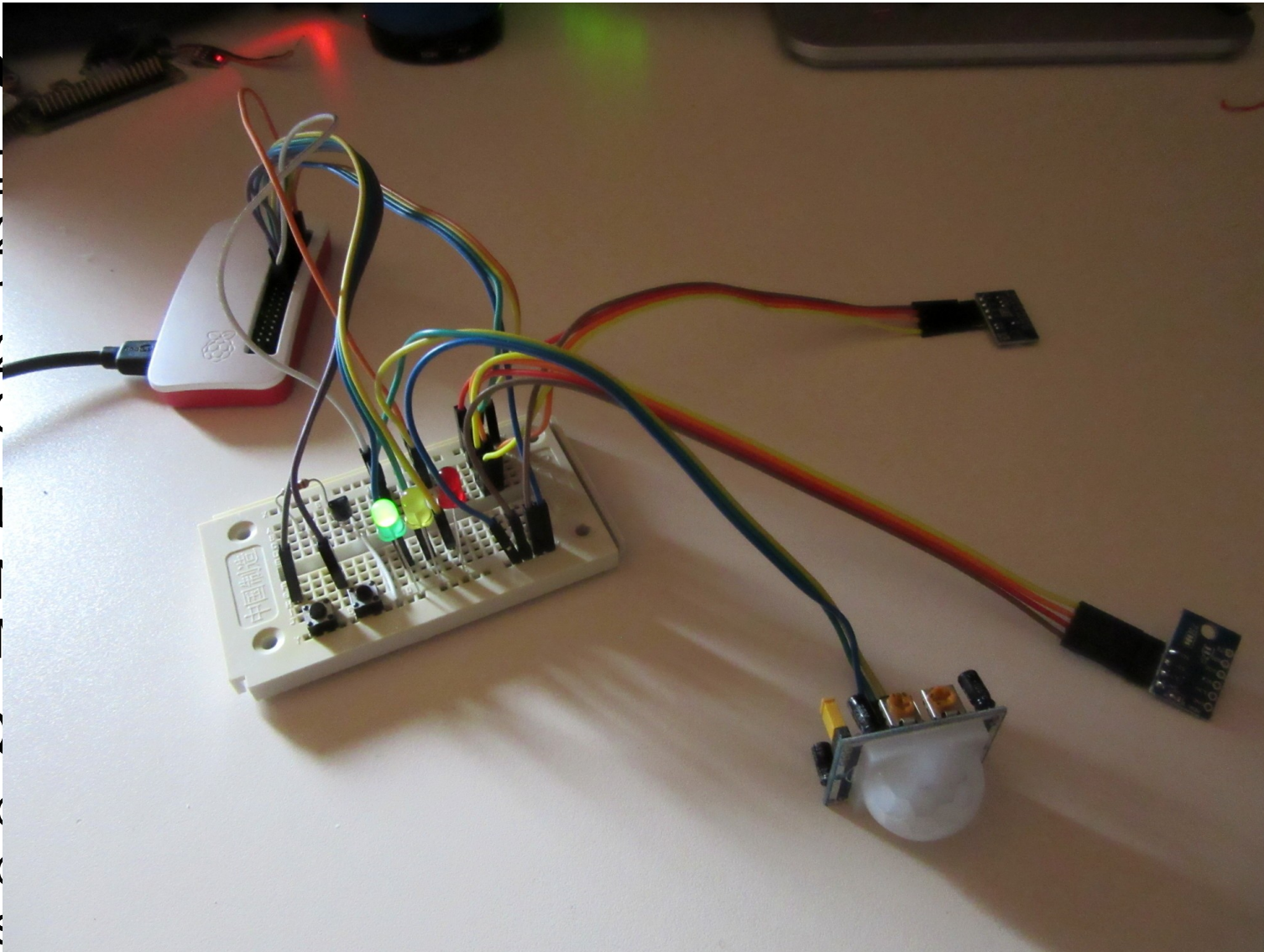
Ein Raspberry Zero W mit:

- Angeschlossene Hardware:
 - 2 BME280-Sensoren (via I2C)
 - 1 PIR-Sensor (Bewegungsmelder)
 - 2 Taster
 - 3 LEDs
- Installierte Software:
 - Node-Red
 - MQTT-Broker (Mosquitto)
- MQTT-Topics (Broker auf printzerow)
 - bme280/<i2c-adr>/json
 - pir/state
 - switch/<nr>/state

„Workshop-Hardware“

Ein Ra

- Ang
 - 2
 - 1
 - 2
 - 2
 - 3
- Inst
 - 1
 - 1
- MQ
 - k
 - p
 - s



Openweathermap (aktuelles Wetter)

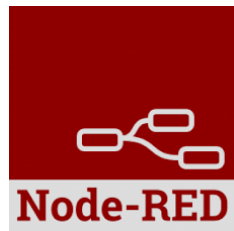
<http://api.openweathermap.org/data/2.5/weather/?q=<city>&units=metric&appid=<api-key>>

```
{
  "coord":{"lon":12.55,"lat":52.42},
  "weather":[{"id":803,"main":"Clouds",
    "description":"broken clouds","icon":"04d"}],
  "base":"stations",
  "main":{"temp":8,"pressure":1019,"humidity":81,
    "temp_min":8,"temp_max":8},
  "visibility":10000,
  "wind":{"speed":2.6,"deg":330},
  "clouds":{"all":75},
  "dt":1546690800,
  "sys":{"type":1,"id":1275,"message":0.0034,"country":"DE",
    "sunrise":1546672742,"sunset":1546701110},
  "id":2945358,
  "name":"Brandenburg an der Havel",
  "cod":200
}
```

Openweathermap (Wettervorhersage)

<http://api.openweathermap.org/data/2.5/forecast/daily?q=<city>&units=metric&appid=<api-key>>

```
{
  "...": "...",
  "list": [
    {
      "dt": 1546686000,
      "temp": { "day": 8.36, "min": 2.41, "max": 8.36,
                "night": 2.41, "eve": 5.37, "morn": 8 },
      "pressure": 1024.2,
      "humidity": 98,
      "weather": [ { "id": 500, "main": "Rain",
                     "description": "light rain",
                     "icon": "10d" } ],
      "speed": 5.95,
      "deg": 328,
      "clouds": 44,
      "rain": 1.22
    },
    ...
  ]
}
```

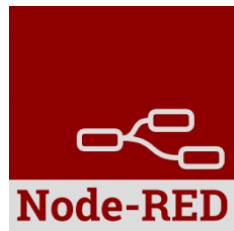


Openweathermap (API-Keys)

- ...nur für diesen Workshop!

35f735cef07611dd808b56c0b62c6880

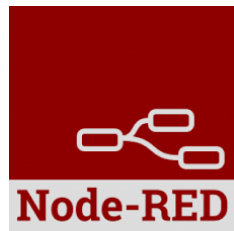
6fb79c2e7542e1127b5956d5d5141810



Systeminformationen sammeln/anzeigen

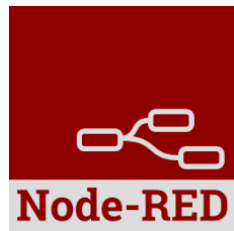
- Node nachinstallieren:
→ node-red-contrib-os
- Vorschlag für MQTT-Topic:
→ sysinfo/<hostname>/json
- Vorschlag für MQTT-Payload:

```
{  
  "loadavg": [0.2314453125, 0.27392578125, 0.212890625],  
  "os": {  
    "hostname": "printzerow", "type": "Linux",  
    "platform": "linux", "arch": "arm", "release": "4.14.73+",  
    "endianness": "LE", "tmpdir": "/tmp"  
  },  
  "uptime": {"uptime": 82023}  
}
```



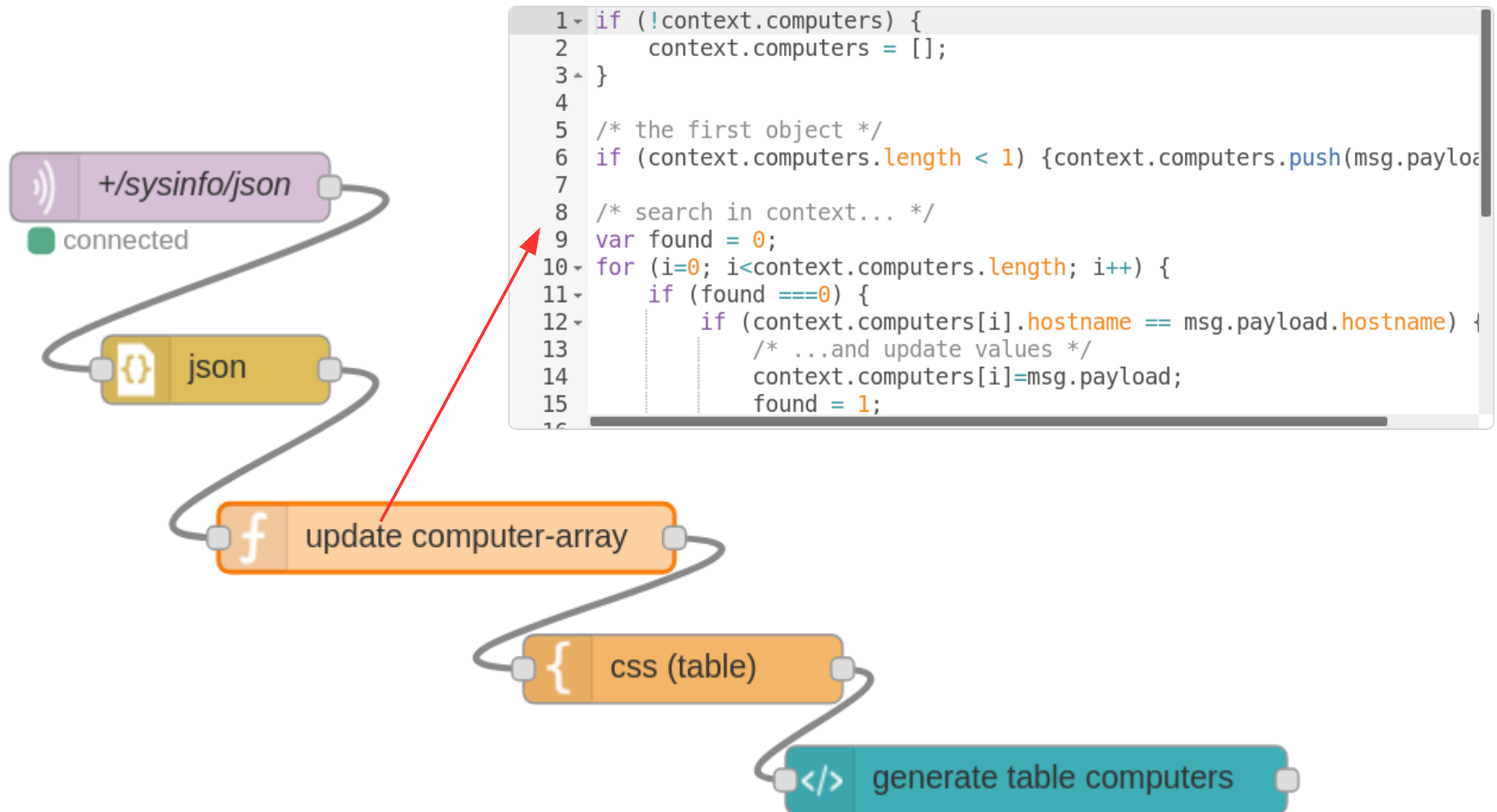
Weiterführende Informationen

- <https://nodered.org/>
- <https://github.com/node-red/node-red>

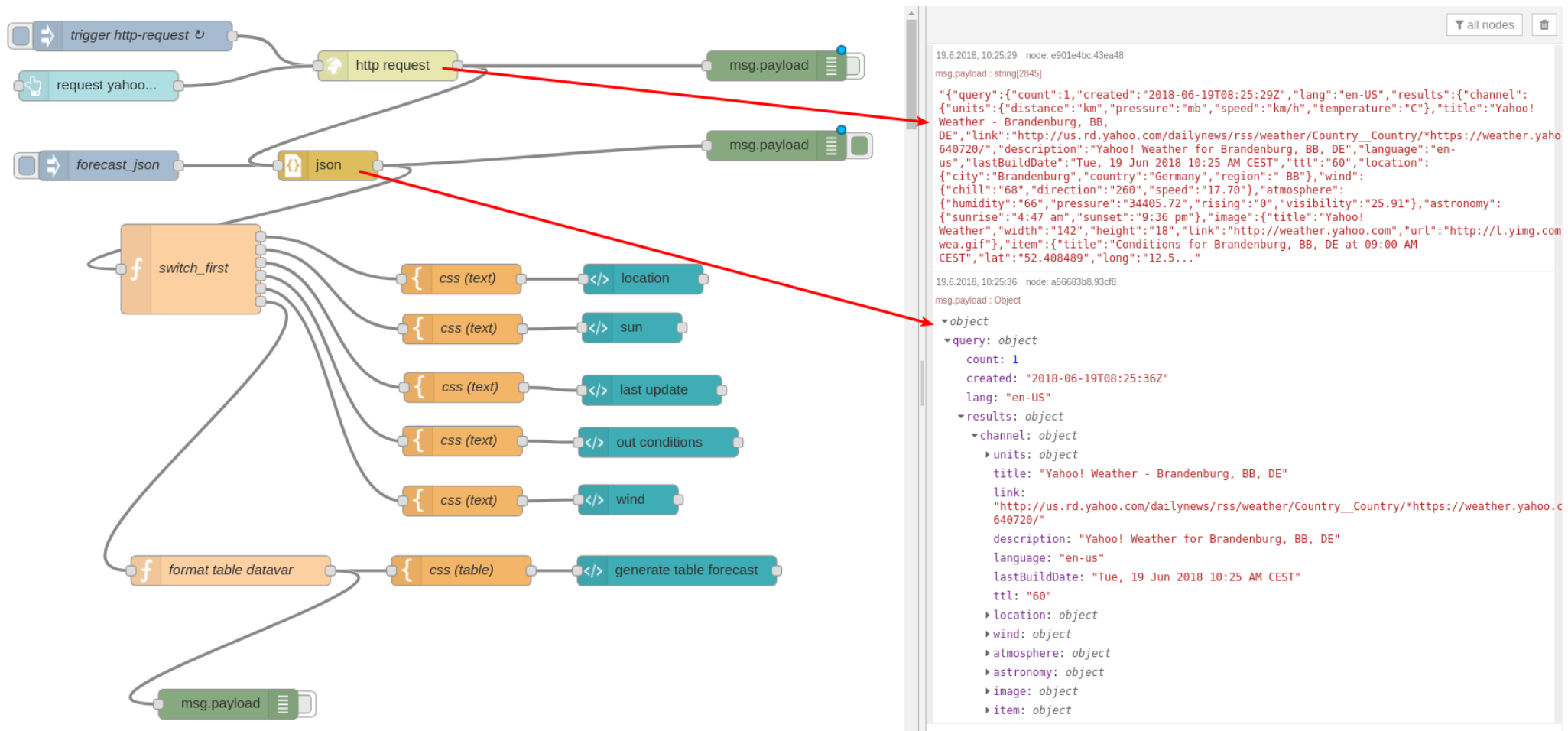


Fragen, Feedback...?

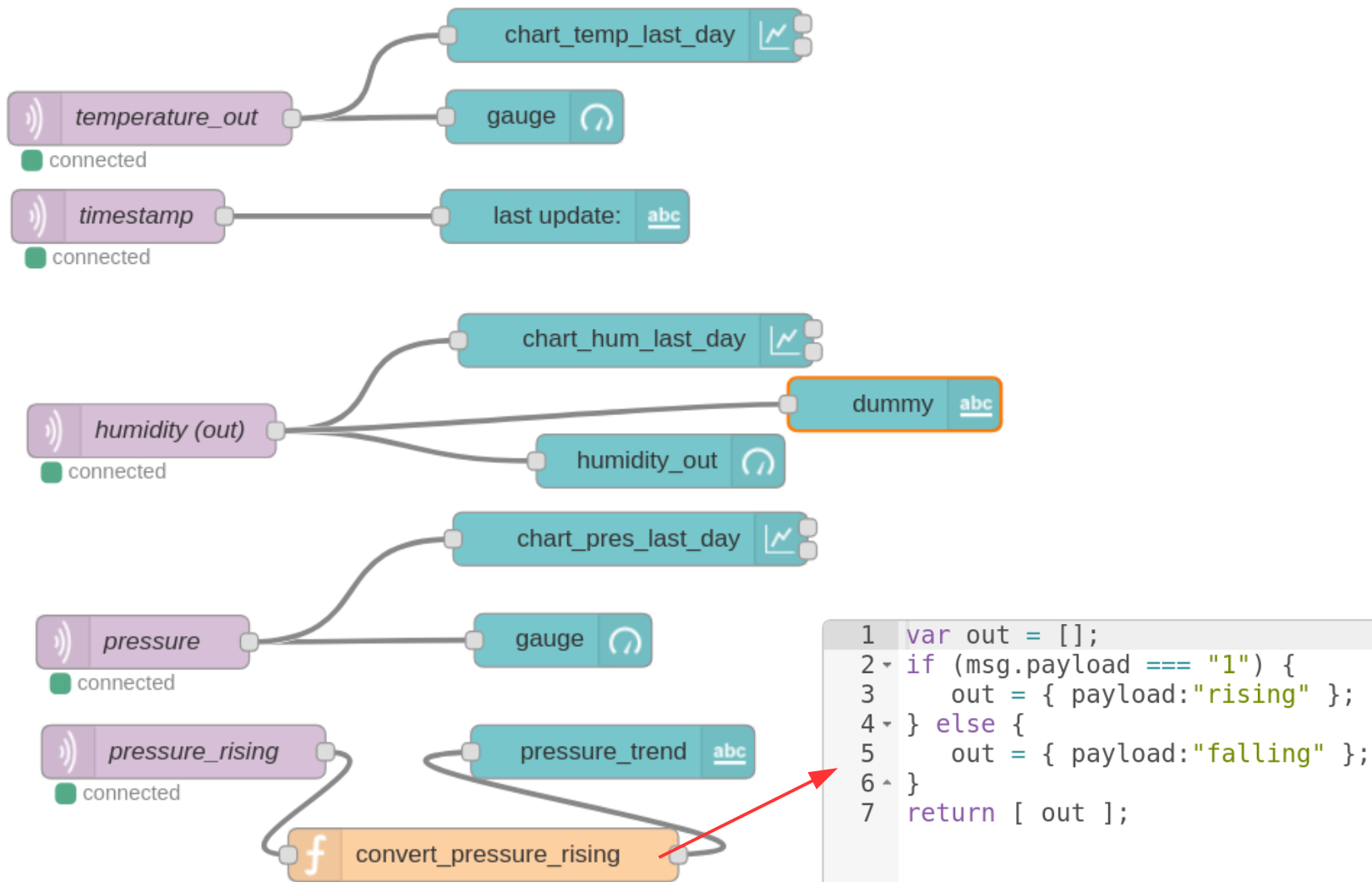
Beispiel: Computer (Flow)



Beispiel: Wettervorhersage (Flow)



Beispiel: Wetter (Flow)



Beispiel: Sensoren (Flow)

