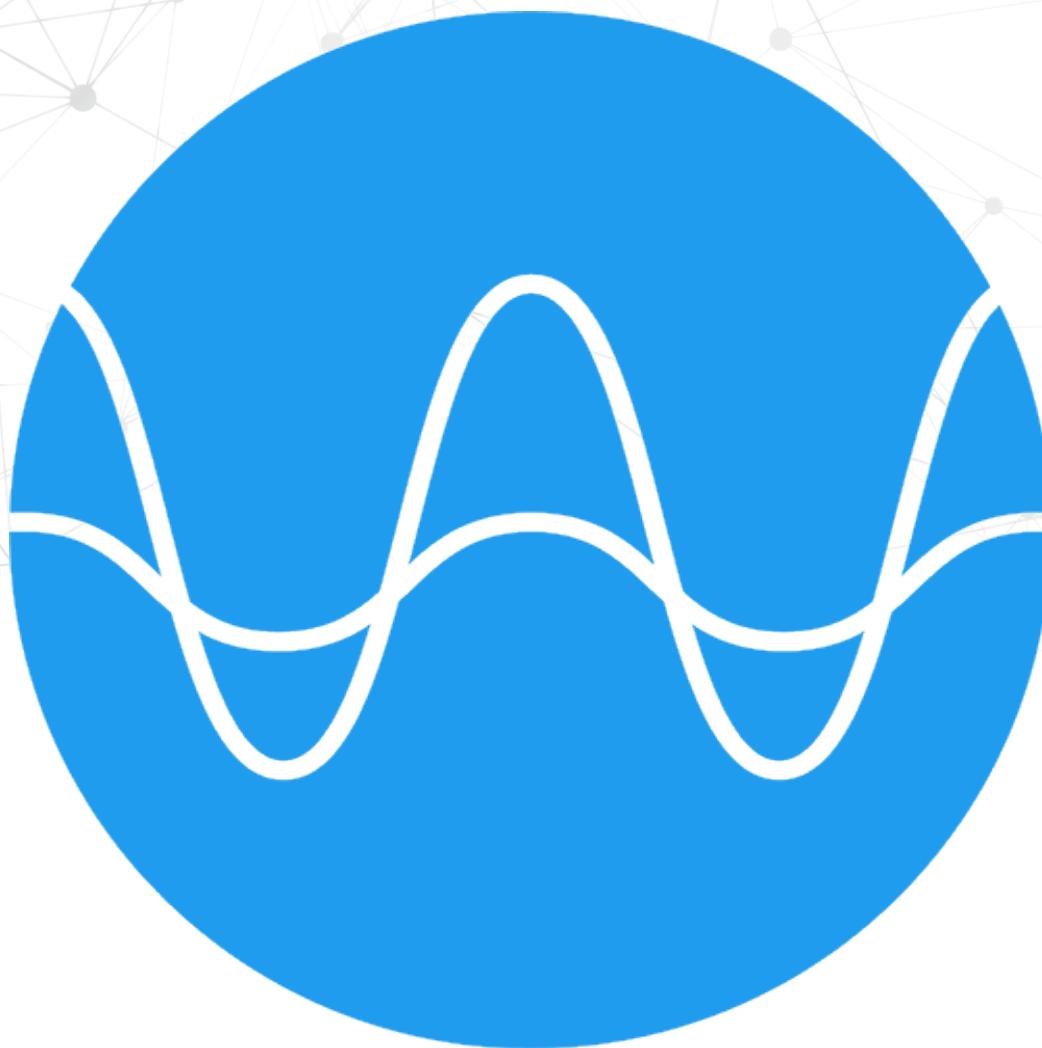


TIMEFLUX

THE API FOR YOUR BODY AND MIND

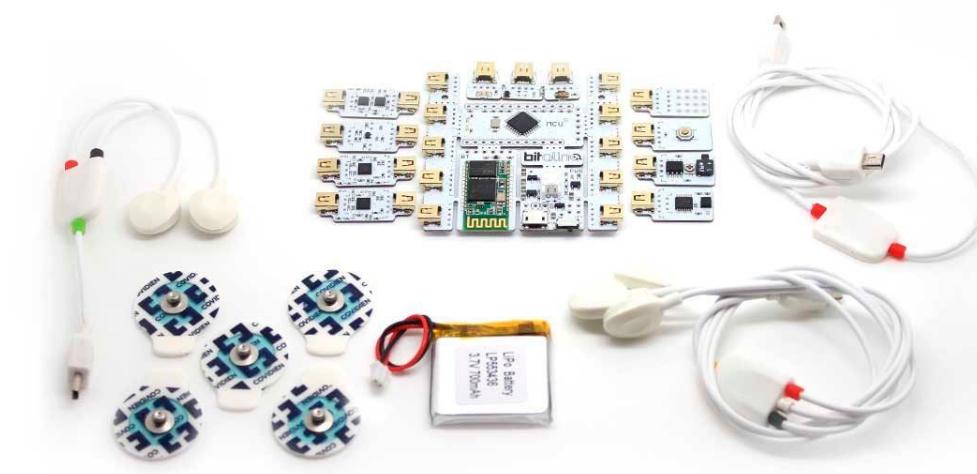
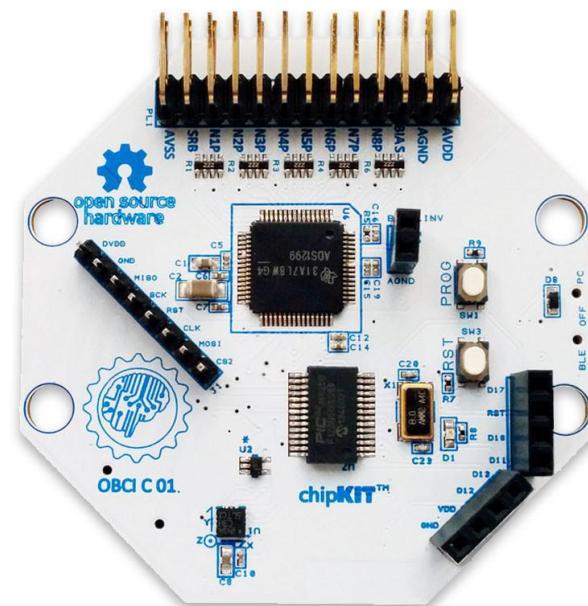


Overview

Use cases

- Data and event **acquisition** from multiple sources
- **Stimulus** presentation
- Bio-feedback
- **Brain-Computer Interfaces**
- Interactive installations
- And more :)

It works with your devices



In a nutshell

- Fits well within the **Python datascience ecosystem**
- Permissive **MIT license**: commercial use authorized
- Works both **offline and online**
- Quick **prototyping**

Easy to learn, use, and extend

- **Familiar concepts:** graphs, nodes, edges
- Relies on **industry standards:** Pandas, Xarray, Scikit-Learn, Lab Streaming Layer
- **Descriptive pipelines:** simple YAML syntax, no coding required
- **Custom nodes:** standard Python classes

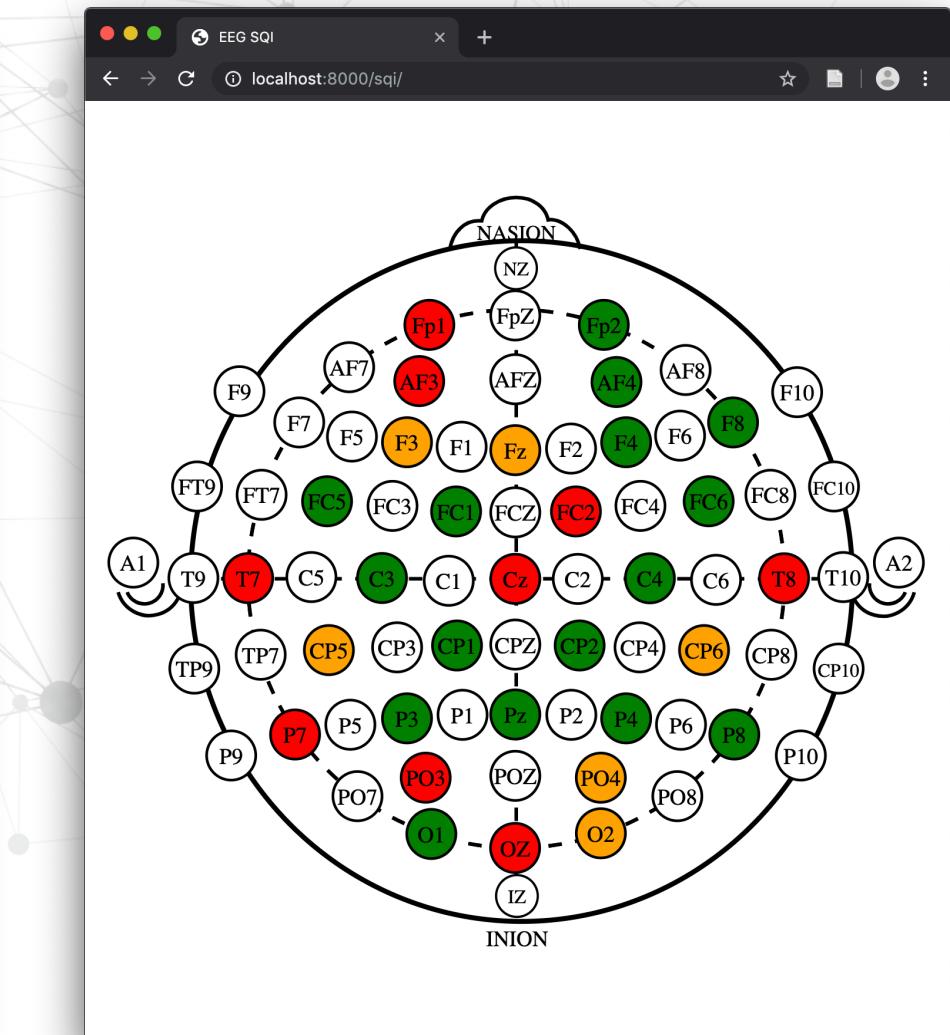
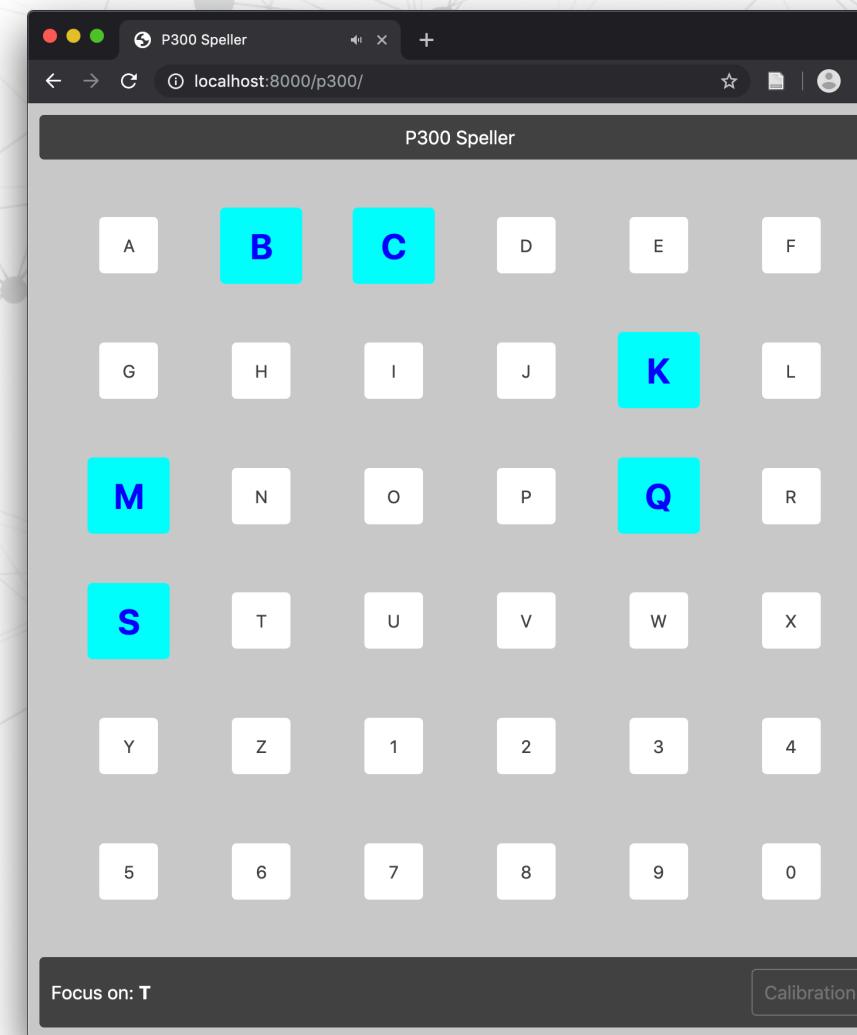
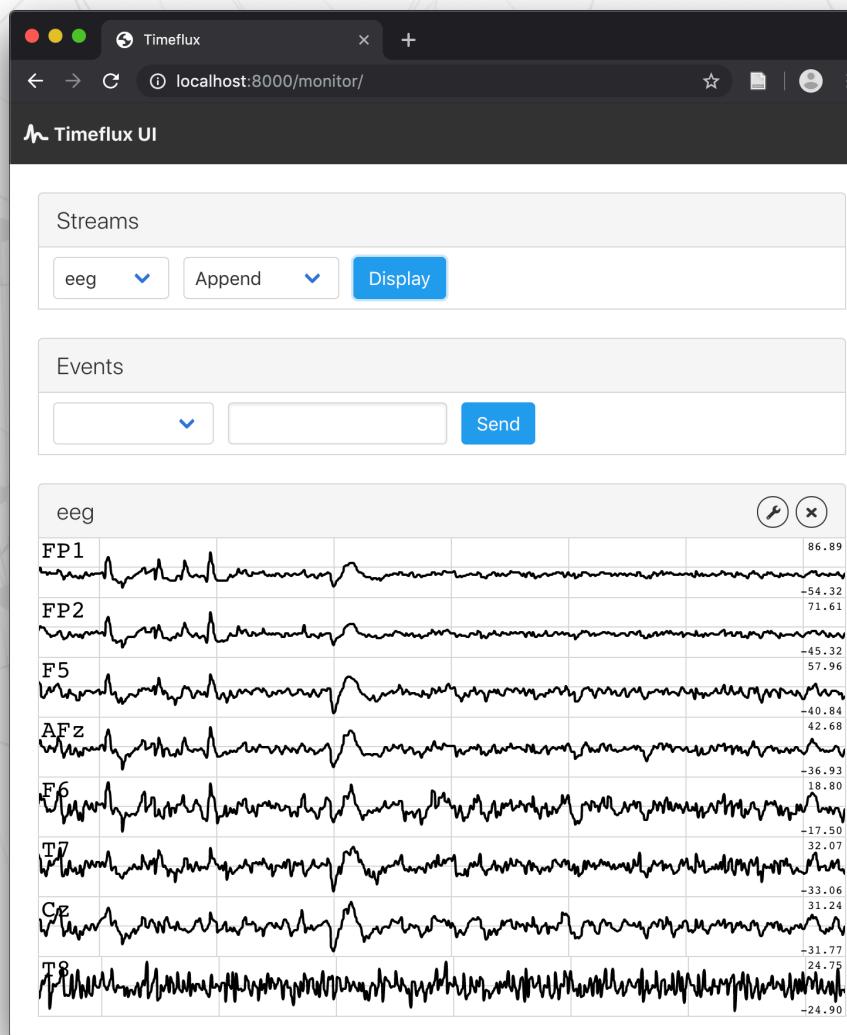
Batteries included

- **Networking:** Pub/Sub, Lab Streaming Layer, OSC, WebSocket
- **Recording and replay:** HDF5 file format
- **Digital Signal Processing**
- **Machine Learning**
- **User interface:** monitoring, web apps

Batteries included

- Multidimensional **matrix manipulation**: queries, transformations, expressions, epoching, windowing
- Native **device drivers**
- Sub-millisecond **synchronization**
- **Debugging tools**
- Pre and post **hooks**

It happens in the browser



Documentation

The screenshot shows the documentation for the "Hello, World!" app. It features a sidebar with navigation links like "GENERAL", "USAGE", "EXTENDING", and "API REFERENCE". The main content area includes a "MyFirstGraph" diagram and a code snippet for YAML configuration.

```
graphs:
- id: MyFirstGraph
  nodes:
  - id: random
    module: timeflux.nodes.random
    class: Random
    params:
      columns: 5
      rows_min: 1
      rows_max: 10
```

The screenshot shows the documentation for a "neurofeedback app". It includes a "MyFirstGraph" diagram and a code snippet for YAML configuration. A note section discusses the processing flow and data streams.

```
graphs:
- id: MyFirstGraph
  nodes:
  - id: random
    module: timeflux.nodes.random
    class: Random
    params:
      columns: 5
      rows_min: 1
      rows_max: 10
```

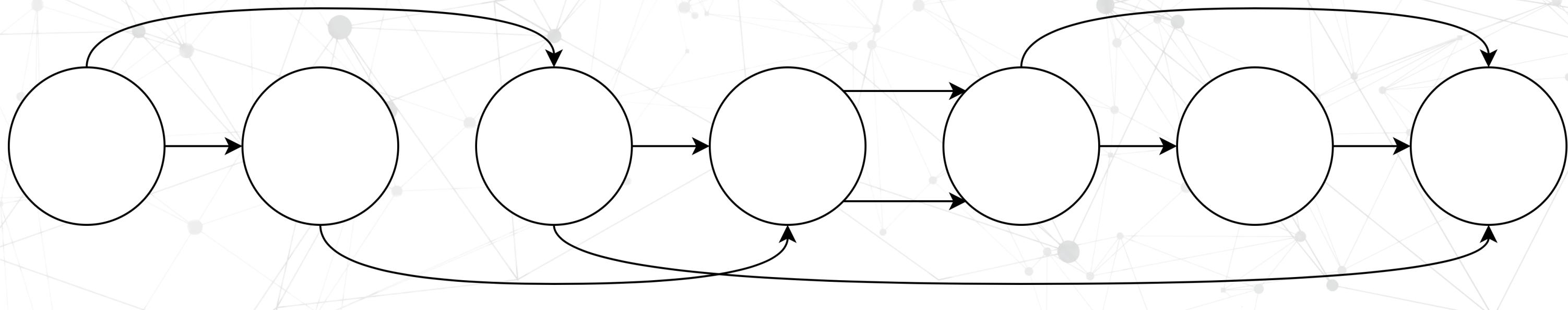
The screenshot shows the documentation for the "timeflux_dsp.nodes.filters" module. It includes a code snippet for YAML configuration, a graph illustrating signal processing, and a line chart comparing offline and online filtered signals.

```
graphs:
- id: MyFirstGraph
  nodes:
  - id: random
    module: timeflux.nodes.random
    class: Random
    params:
      columns: 5
      rows_min: 1
      rows_max: 10
```

Pipelines

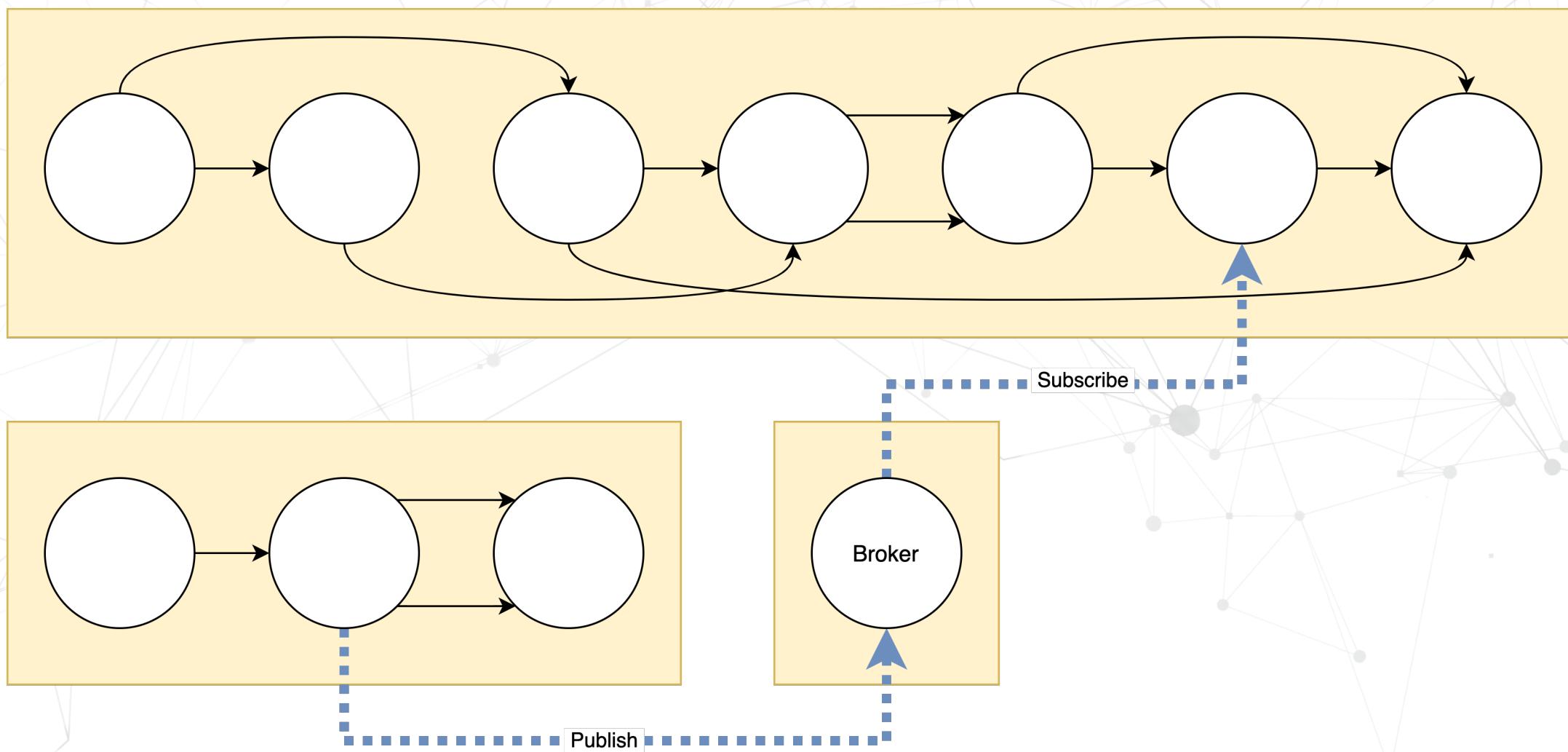
Directed Acyclic Graph (DAG)

A set of **nodes** connected by **edges**, where information **flows** in a given direction, **without any loop**.



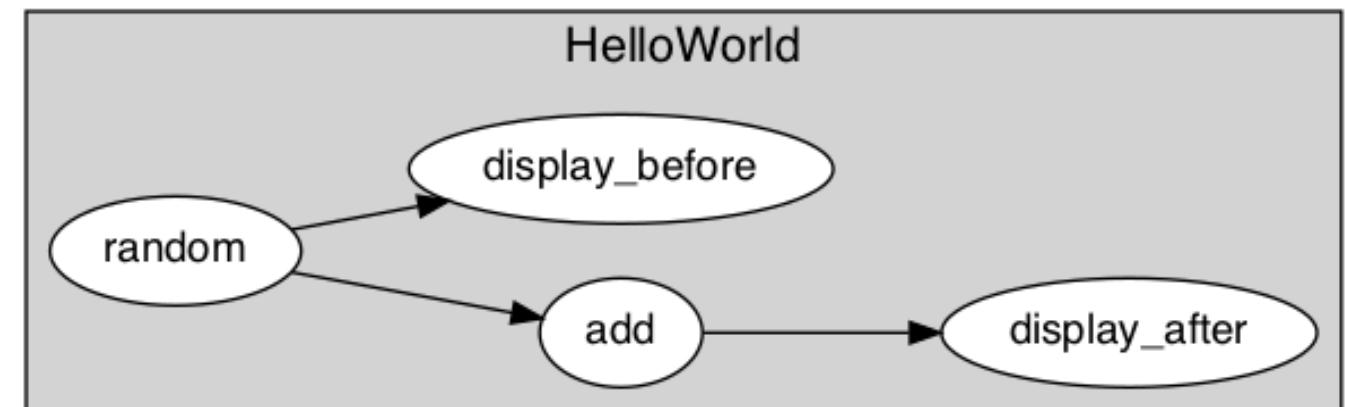
Multiple DAGs

Pub/Sub allows asynchronous loops without breaking anything.



helloworld.yaml

```
graphs:
  - id: HelloWorld
    nodes:
      - id: random
        module: timeflux.nodes.random
        class: Random
      - id: add
        module: timeflux_example.nodes.arithmetic
        class: Add
        params:
          value: 1
      - id: display_before
        module: timeflux.nodes.debug
        class: Display
      - id: display_after
        module: timeflux.nodes.debug
        class: Display
    edges:
      - source: random
        target: add
      - source: random
        target: display_before
      - source: add
        target: display_after
    rate: 1
```



Run it!

`timeflux -d helloworld.yaml`

Interfaces

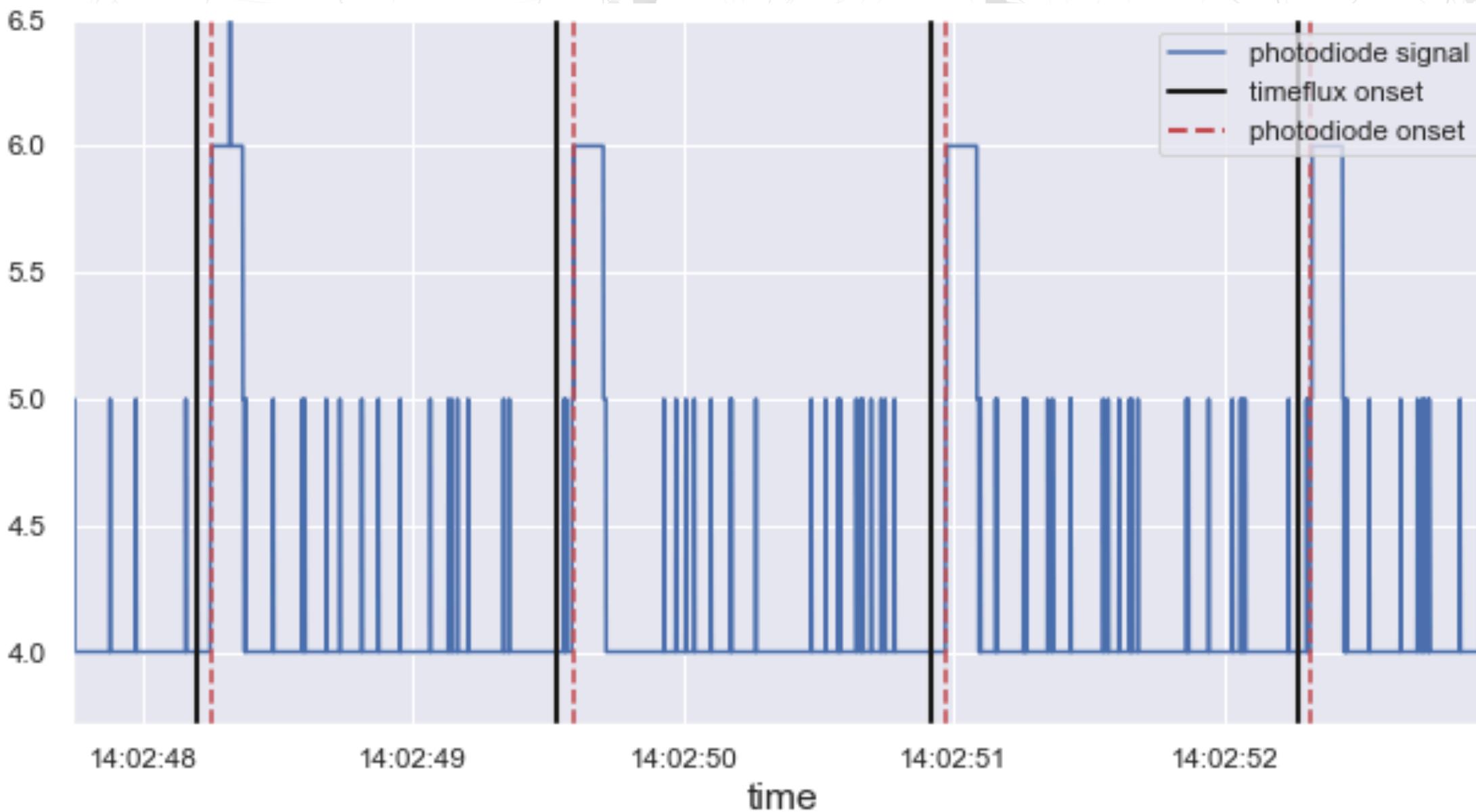
Timeflux.js

- A **JavaScript API** for:
 - Building **user interfaces** available from a browser
 - Receiving and sending data **streams** and **events**
 - Delivering **precisely scheduled** stimuli: suitable for SSVEP and ERP research

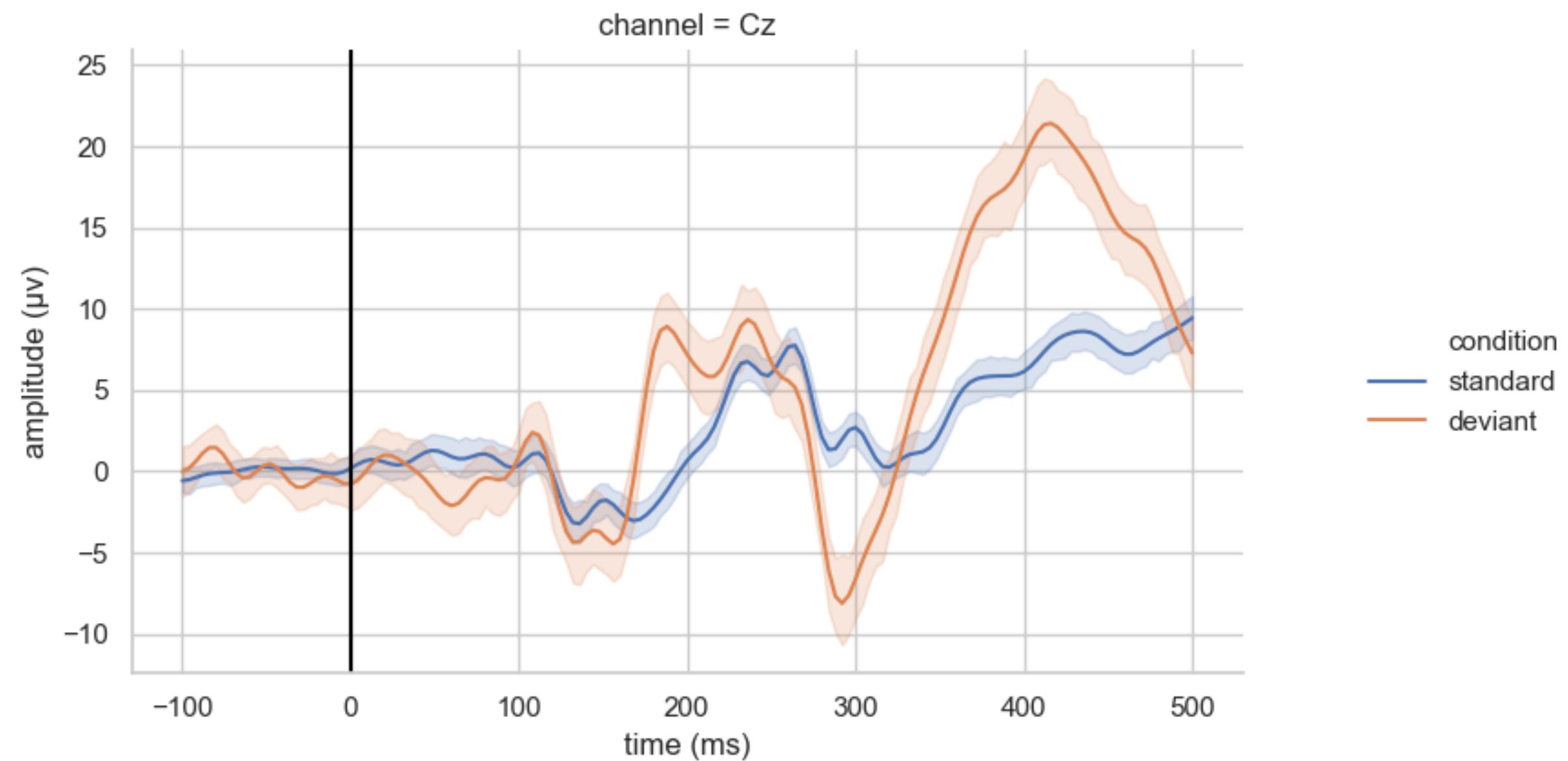
Stimulus presentation

- Perfect timing in a browser is **hard**
- But Timeflux.js makes it **easy!**
- Schedule **repeating** stimuli or **one-time** tasks
- Know **exactly** when the stimulus has been displayed
- Well tested in **Chrome**. Other browsers *may have issues*

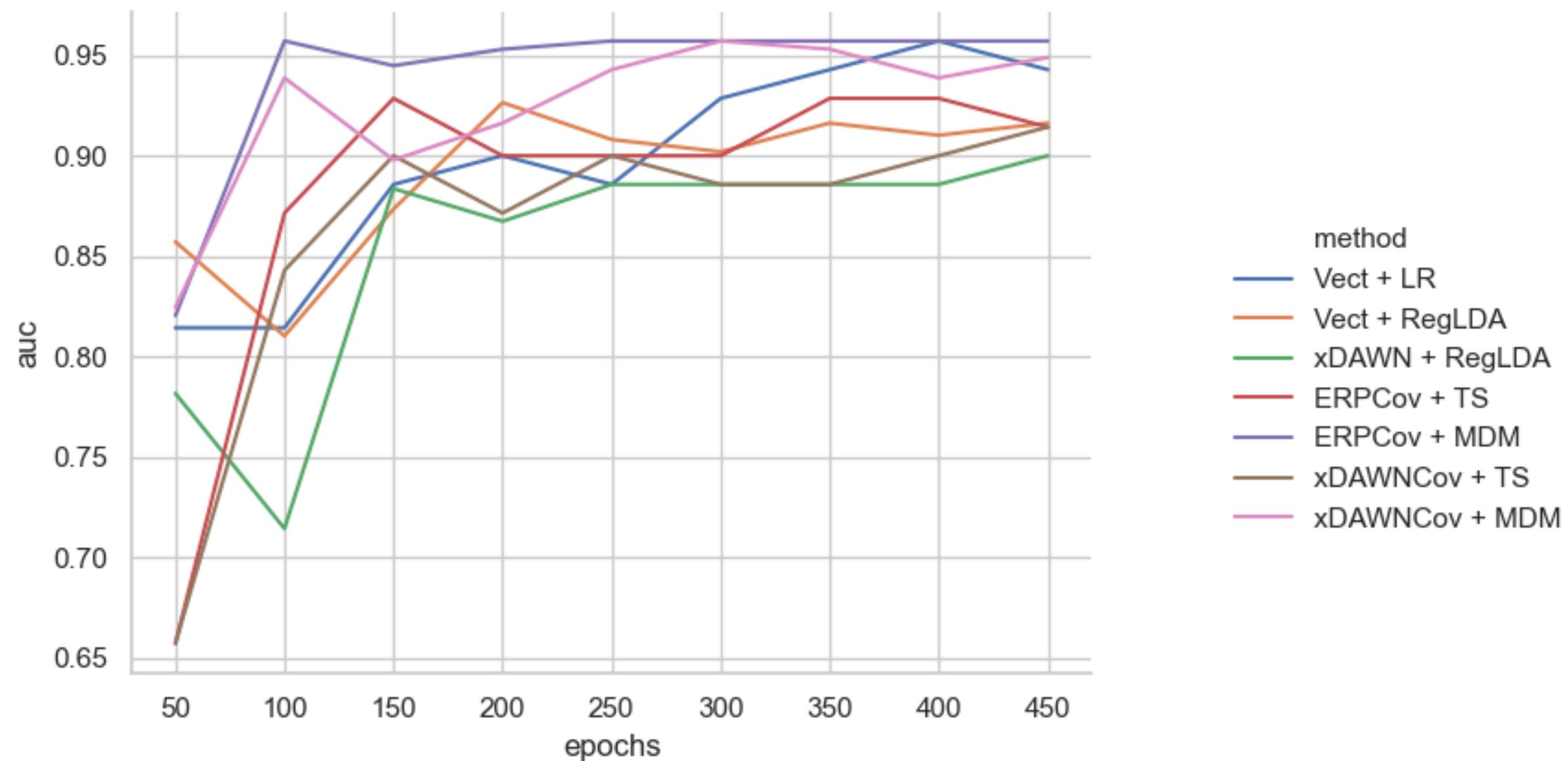
Validation: software event VS photodiode



Validation: evoked potentials



Validation: single-trial ERP classification

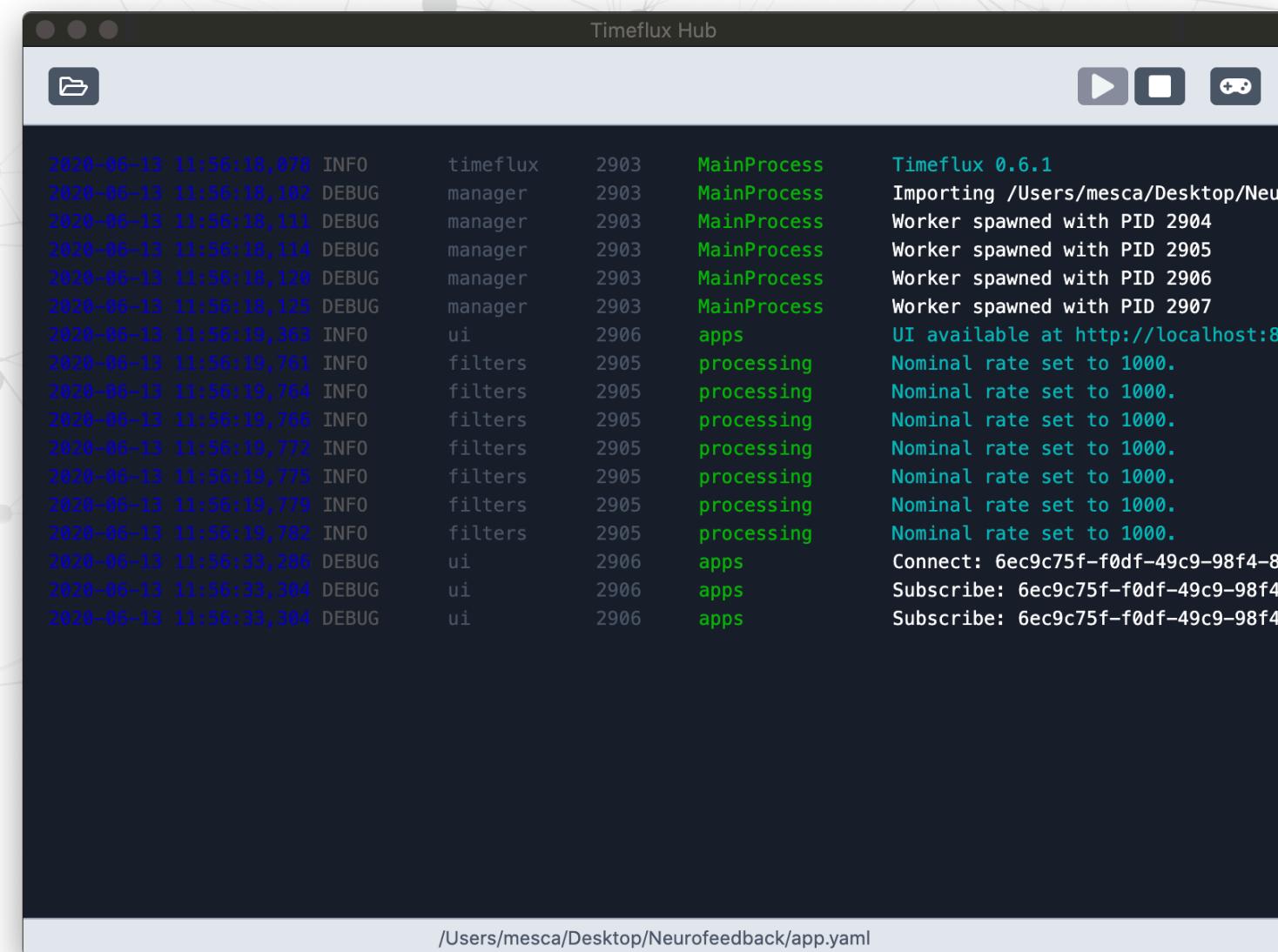


Coming soon!

Plug-and-play pipelines

- Standard **BCI paradigms** (EEG):
 - SSVEP
 - CVEP
 - P300
 - Motor Imagery
- **Neurofeedback** (EEG)
- **Cardiac coherence** (ECG, PPG)
- **Gesture detection** (EMG)

Timeflux Hub



Conclusion

Getting help

- **Website:** <https://timeflux.io>
- **Documentation:** <https://doc.timeflux.io>
- **Bugs:** <https://github.com/timeflux>
- **Slack:** <https://timeflux.io/slack>
- **Email:** contact@timeflux.io

Thanks!