Finnish Institute of Occupational Health

BRAIN·WORK
RESEARCH CENTRE

MIDAS

Framework for Distributed Online Data Stream Analysis

Andreas Henelius andreas.henelius@ttl.fi
Jari Torniainen jari.torniainen@ttl.fi

Finnish Institute of Occupational Health
Brain Work Research Centre
Helsinki, Finland

Symbiotic2015, Berlin, 07.10.2015

OUTLINE

Part I (1h)

- Background
- Architecture (high-level)
- The MIDAS node
- Configuration and usage
- Building analysis nodes
- Installation
- Resources

Part II (2h)

- Practical Exercises

M I D A S

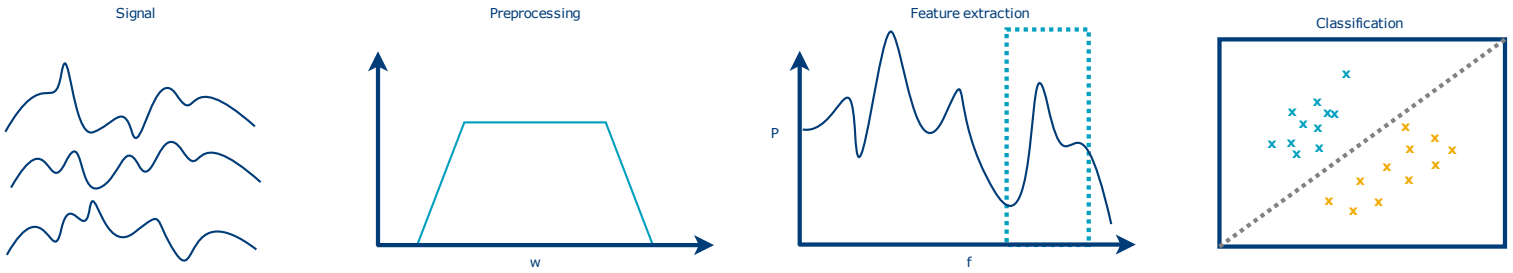
MODULAR INTEGRATED DISTRIBUTED ANALYSIS SYSTEM

WHAT IS MIDAS?

A system for online analysis of streaming signals and allows easy integration of such into machine learning frameworks.

WHY ONLINE ANALYSIS?

- Cognitive State Determination
- Symbiotic Interaction
- IoT

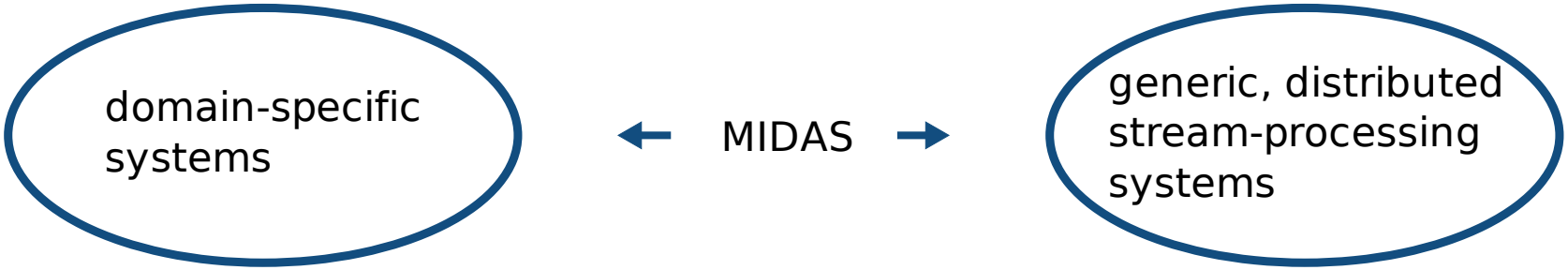


CHALLENGES

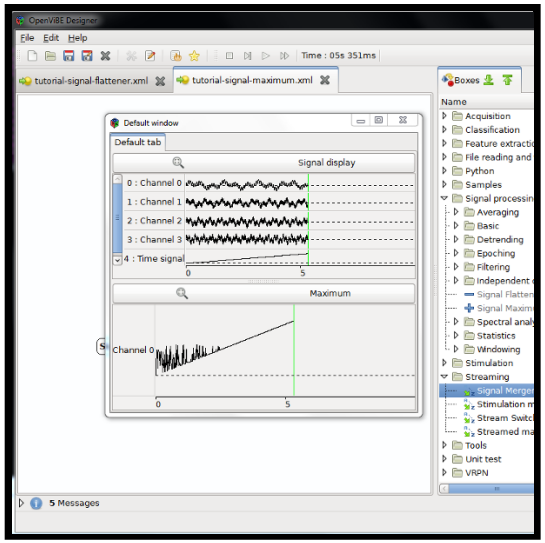
- Sensors and protocols
- Data streams and formats
- Signal fusion and computational load

WHAT IS MIDAS?

Modular Integrated Distributed Analysis System

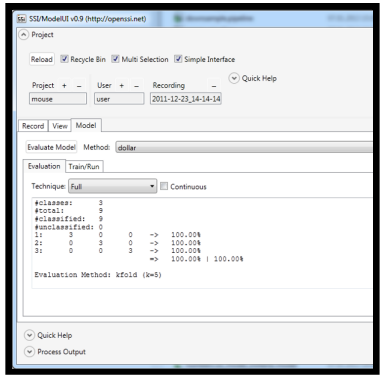


SIMILAR SOLUTIONS



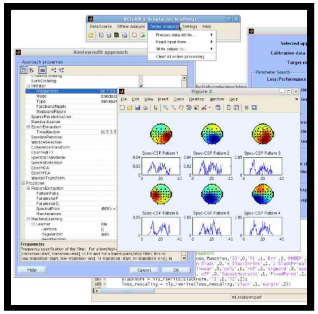
OpenViBE

Renard, Yann, et al. "OpenViBE: an open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments" Presence: teleoperators and virtual environments, 2010



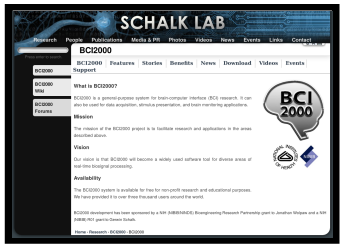
SSI

Wagner, Johannes, et al. "The Social Signal Interpretation Framework (SSI) for Real Time Signal Processing and Recognition" INTERSPEECH, 2011



BCILAB

C Kothe, S Makeig. "BCILAB: A platform for brain-computer interface development" Journal of Neural Engineering, 2013



BCI2000

G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw: BCI2000: A General-Purpose Brain-Computer Interface (BCI) System, IEEE Trans Biomed Eng, 51(6), June 2004.

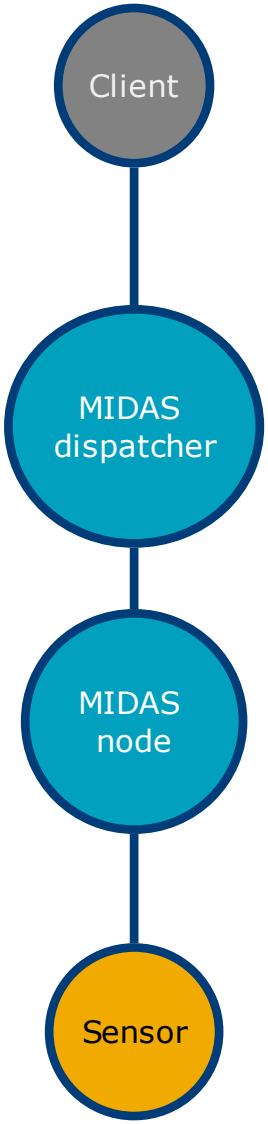
HOW IS MIDAS DIFFERENT?

Easily integrate and combine different sensors into online analyses of streaming data.

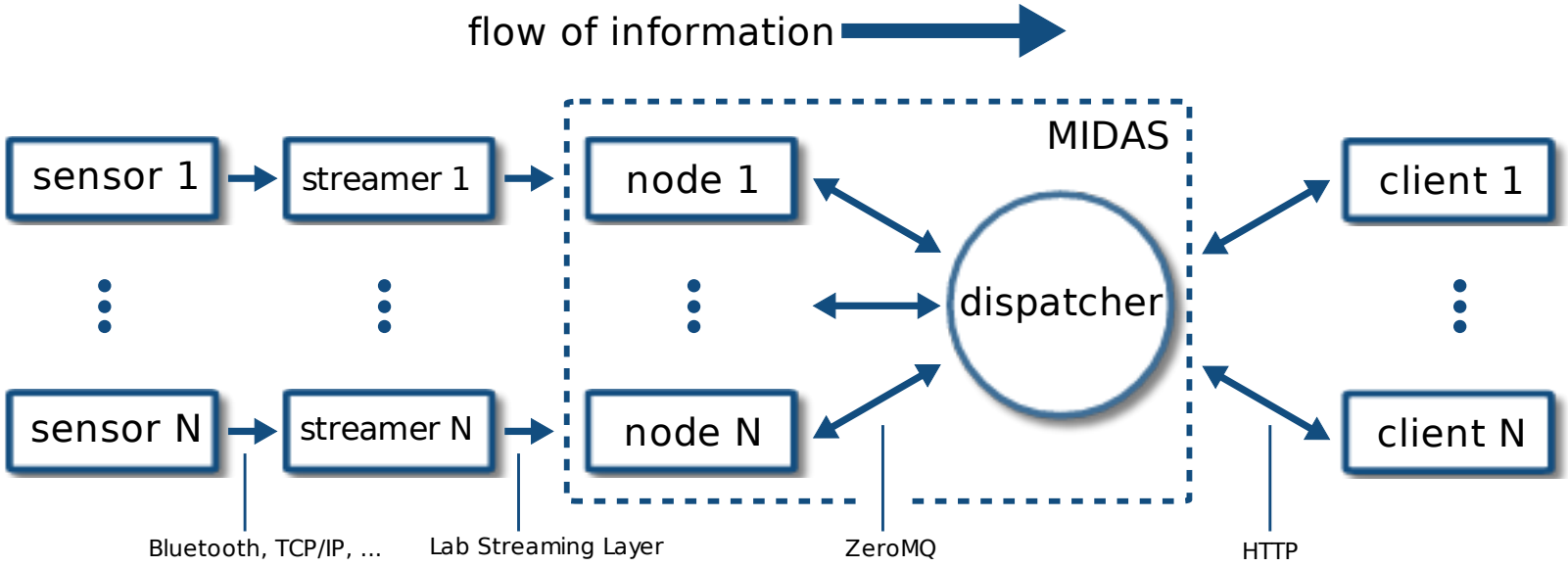
- **Slim:** minimal clutter
- **Modular:** add functionality as needed
- **Distributed:** Runs everywhere

- Written in Python
- Open Source (MIT)

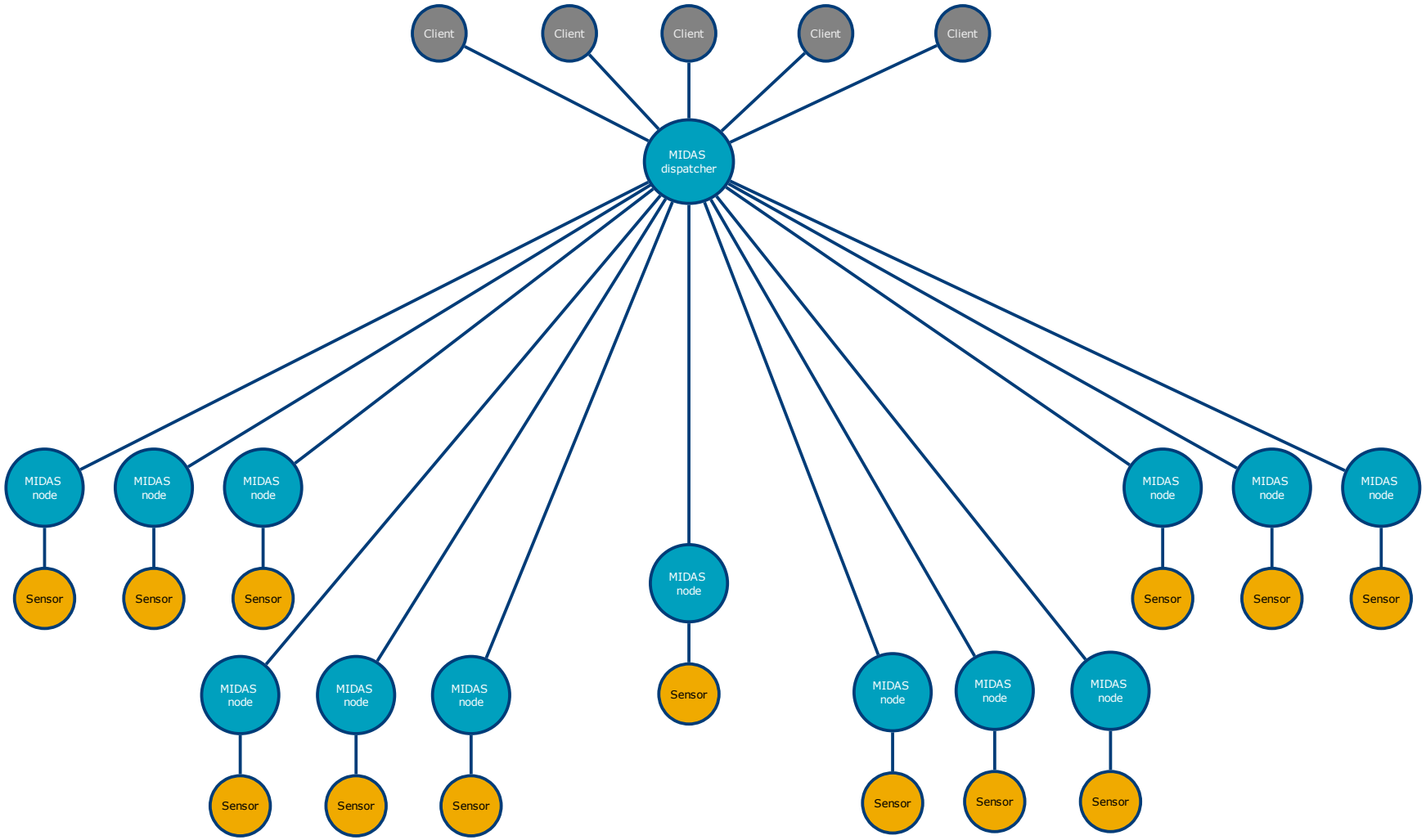
MIDAS ARCHITECTURE



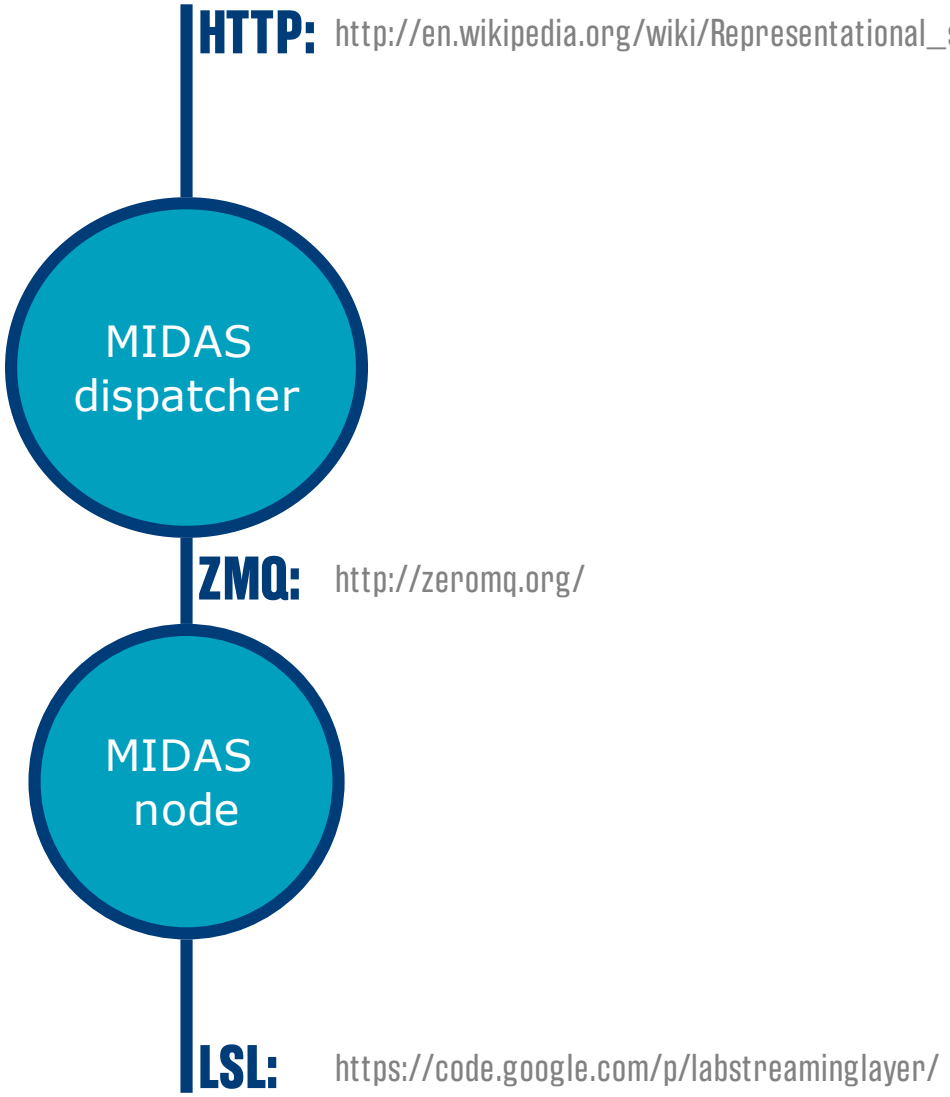
MIDAS ARCHITECTURE



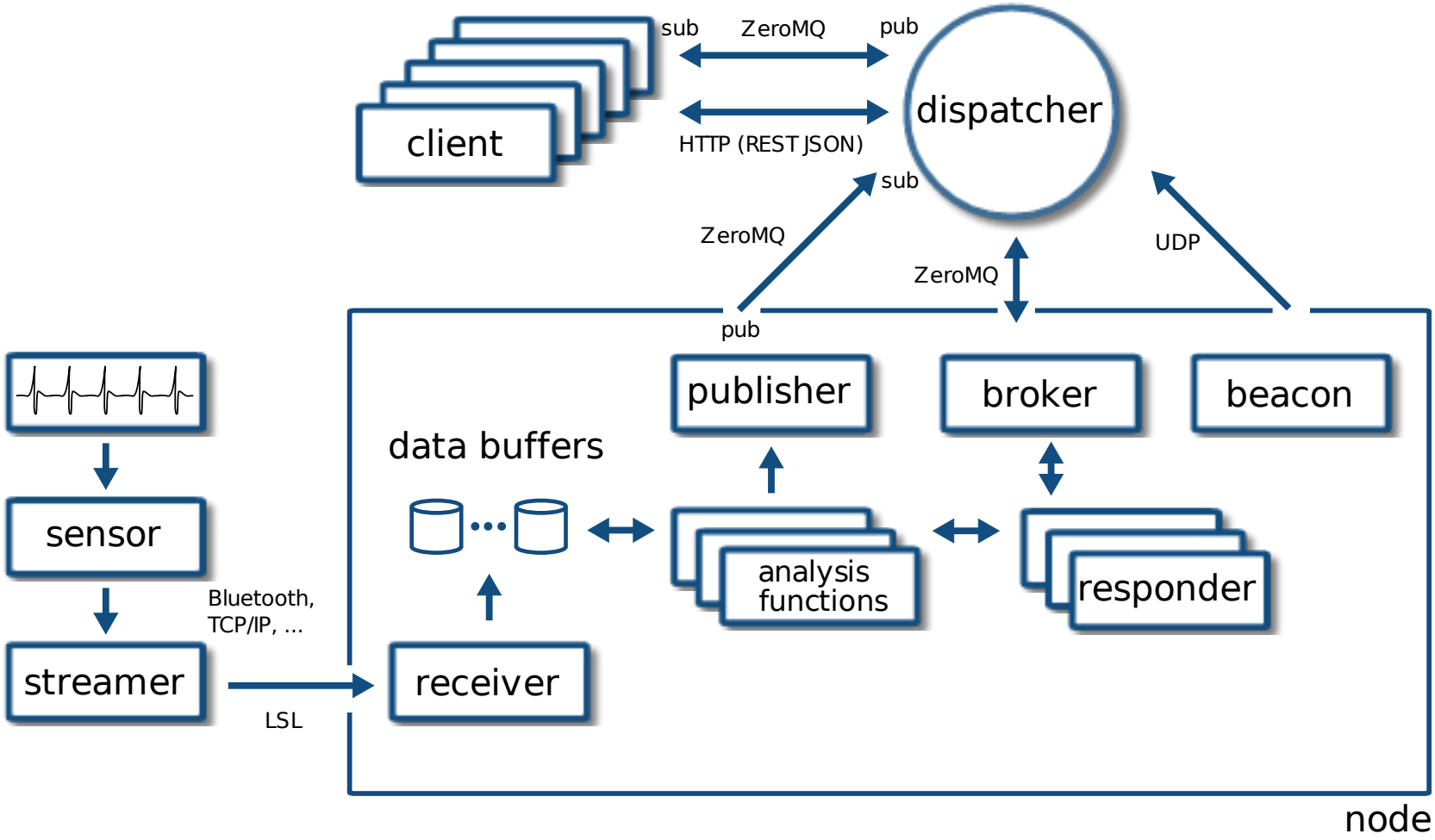
MIDAS ARCHITECTURE



COMMUNICATION PROTOCOLS IN MIDAS



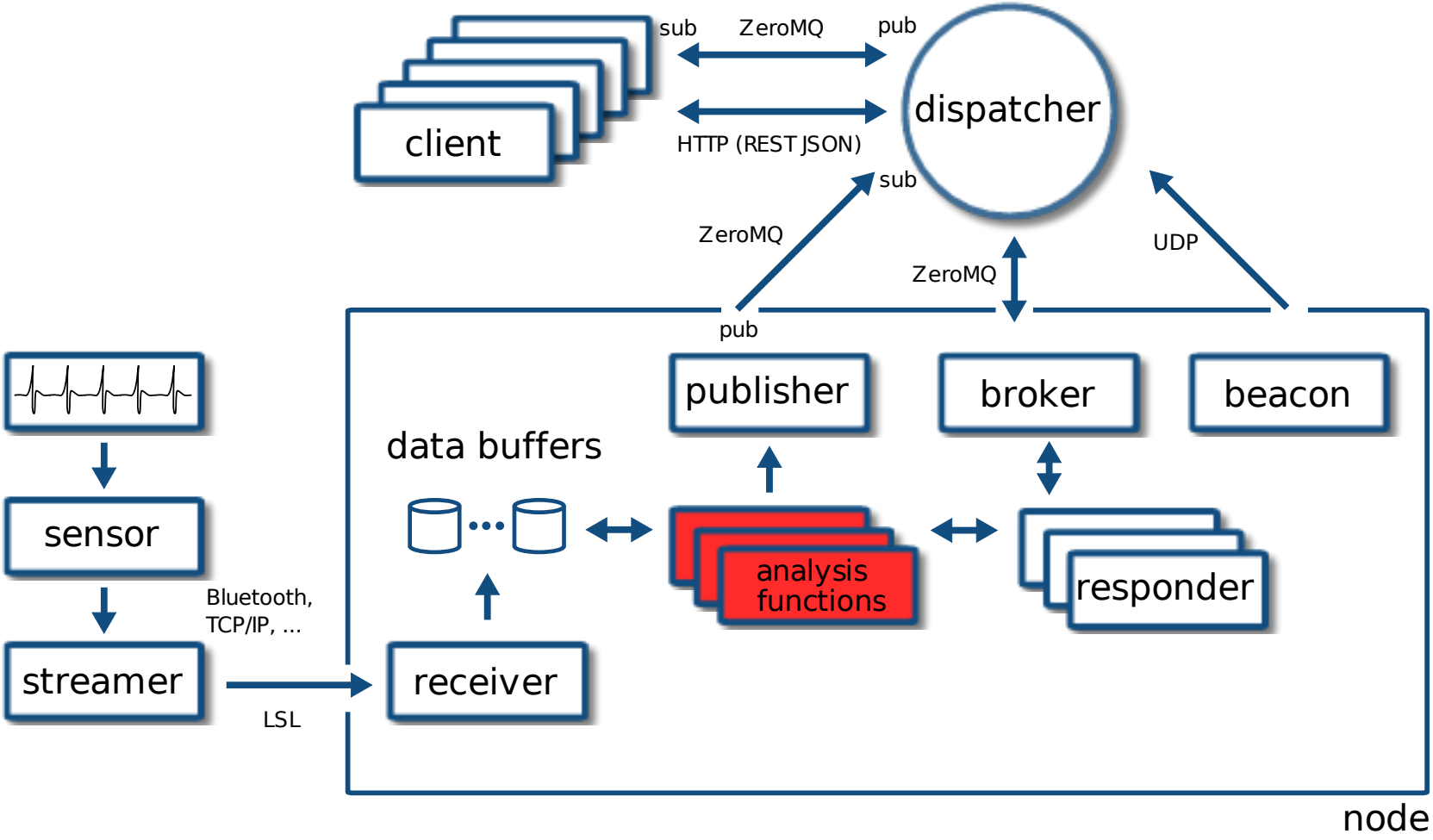
THE NODE


client
sub
ZeroMQ
pub
HTTP (REST JSON)
dispatcher
sub
ZeroMQ
ZeroMQ
UDP
pub
publisher
broker
beacon
data buffers
analysis functions
responder
sensor
Bluetooth, TCP/IP, ...
streamer
LSL
receiver
node

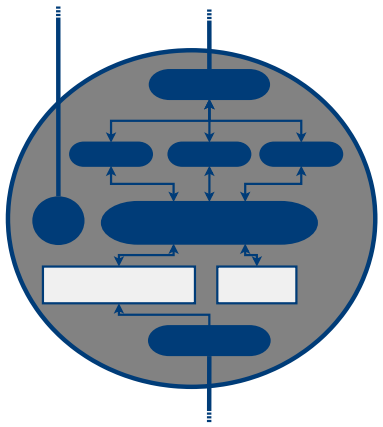
STREAMERS

- Read sensor data and retransmit raw data using the **lab streaming layer**
- Some devices have built-in LSL support (Neuroelectrics Enobio)
- For some devices you can find LSL drivers from **http://code.google.com/p/labstreaminglayer/**
- If no streamer exist, write it yourself! (not difficult)

BUILDING NODES

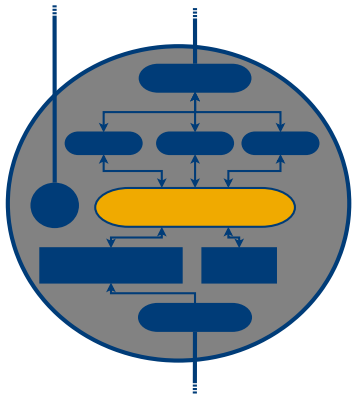


DATA CONTAINERS



- Receiver automatically stores all incoming data into primary data buffers
- Secondary data buffers can be used to store values calculated from the primary data
- Both primary and secondary data can be requested through the dispatcher
- Both primary and secondary data can be used as an input for the analysis functions

ANALYSIS FUNCTIONS



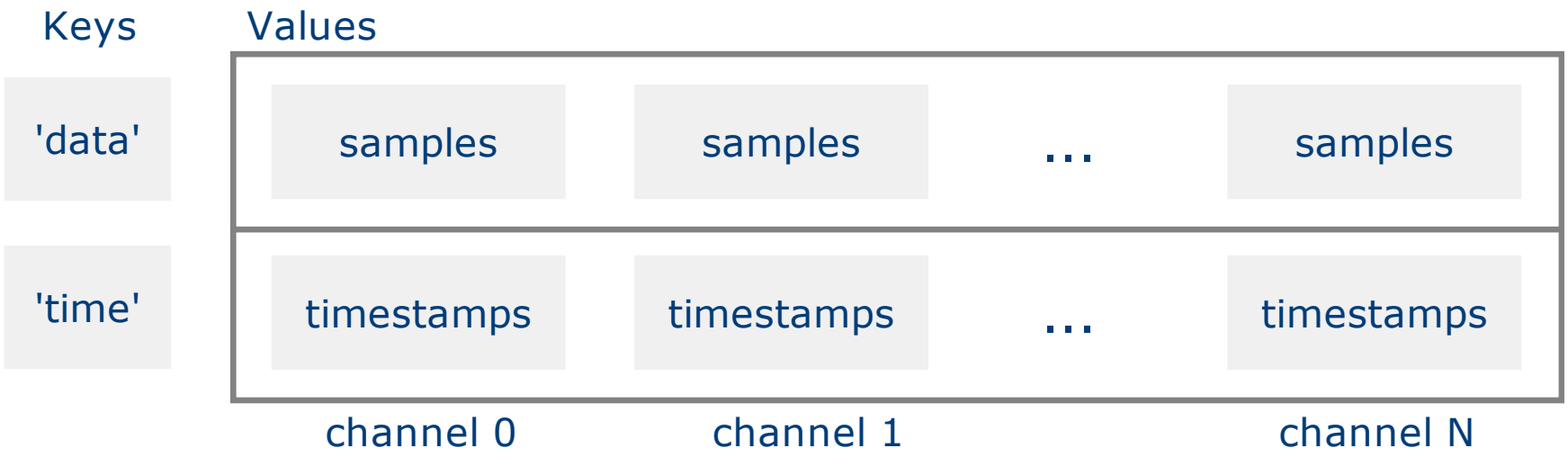
Nodes in the MIDAS framework have two types of analysis functions: **metrics** and **processes**

METRIC FUNCTION

- y = f(x)
- Calculates 'a metric' using data as input
- Input data is specified when the request is made by the client
- Return value can be a scalar or a vector/array
- Can be a method of the Node-class or an external function

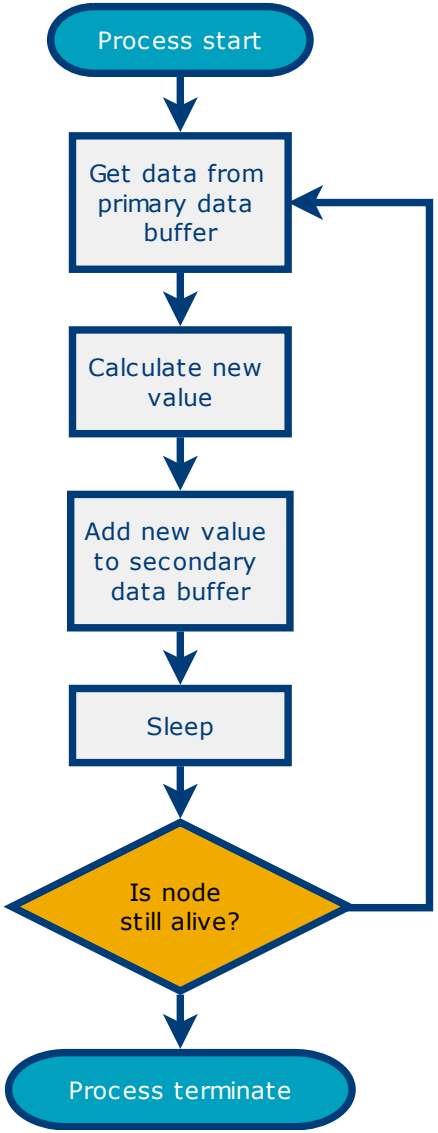
METRIC FUNCTION

- The first positional argument of the metric function always contains the data
- The incoming data is wrapped inside a dict:



PROCESS

- Automatically calculate values from the primary data at set intervals
- Runs in a separate process
- Pushes calculated values into secondary data buffer
- Must be a method of the Node-class


Process start
Get data from primary data buffer
Calculate new value
Add new value to secondary data buffer
Sleep
Is node still alive?
Process terminate

CODE EXAMPLE

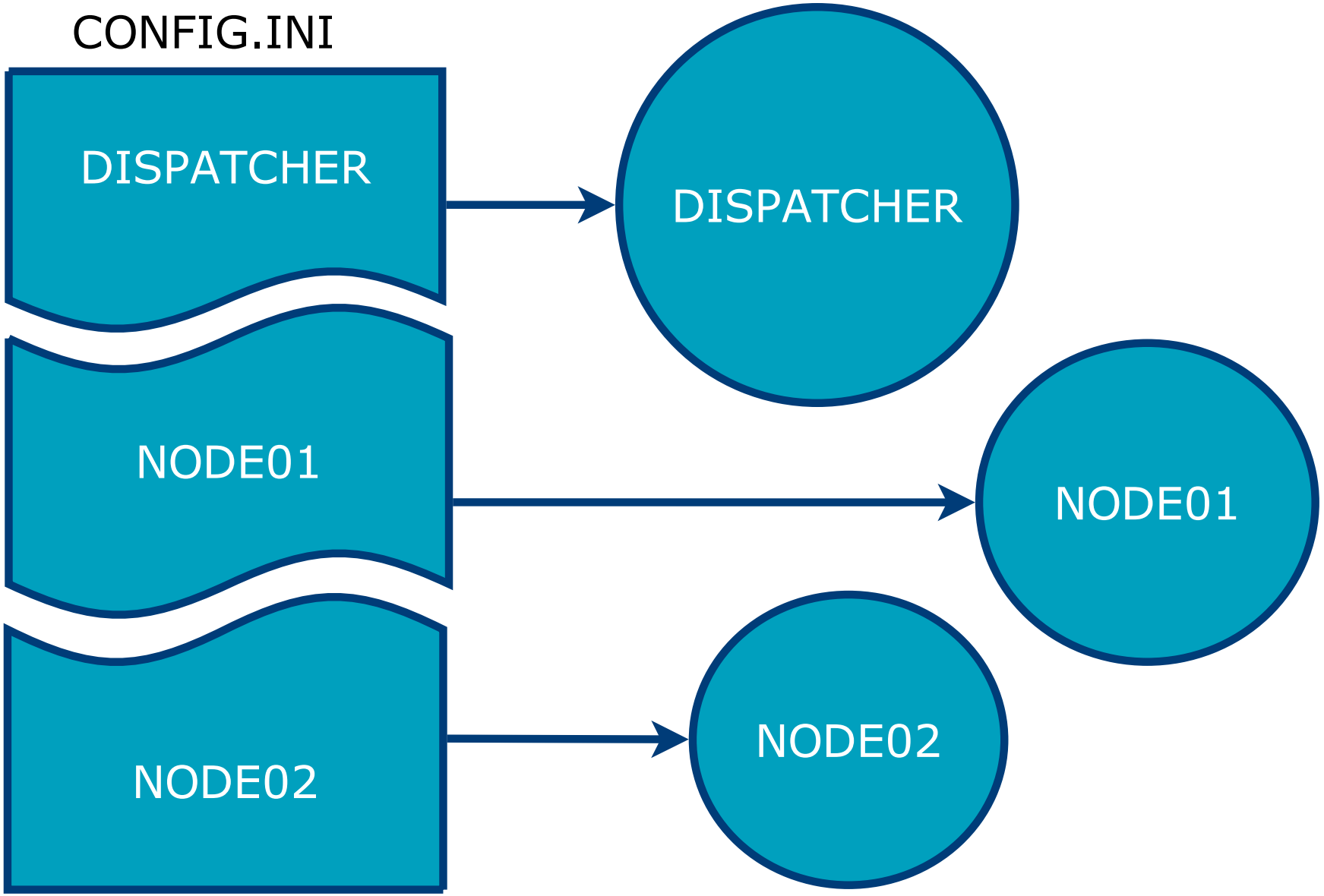
Example from the MIDAS repository.

CONFIGURATION AND USE

1. Configure options in the INI-file

2. Ensure that the data stream is visible
3. Start the node(s)
4. Start the dispatcher
5. Access nodes through the MIDAS API (= client)

CONFIGURATION FILE



CONFIGURATION FILE

Dispatcher

```
# Dispatcher-section
[dispatcher]
port                = 8080
ip                  = localhost
n_threads           = 5
run_pubsub_proxy    = False
proxy_port_in       = 5999
proxy_port_out      = 6000
```

CONFIGURATION FILE

An ECG node

```
# Real-time ECG node
[ecg_rt]
nodename                = ecgnode
nodetype                = ECG
nodeid                  = 05
nodedesc                = ECG Processing node
primary_node            = True
port_frontend           = 5054
port_backend            = 5055
port_publisher          = 5056
run_publisher           = False
n_workers               = 2
n_channels              = 1
channel_names           = ch0
channel_descriptions    = ECG channel
sampling_rate           = 100
buffer_size_s           = 600
lsl_stream_name         = faros_ecg
```

STARTING MIDAS

The components of MIDAS are started from the command line.

Dispatcher

```
midas-dispatcher config.ini dispatcher
```

A node

```
python3 ecg_node.py config.ini ecg
./ecg_node.py config.ini ecg     [if first line is #!/usr/bin/env python3]
```

MIDAS STATUS

Check the status of the MIDAS network

```
http://127.0.0.1:8080
http://127.0.0.1:8080/status/nodes
http://127.0.0.1:8080/status/metrics
```

QUERYING DATA

Python

```
import requests
r = requests.get("http://localhost:8080/example_node/
metric/{"type":"metric_a", "time_window":[5]}")
```

MATLAB

```
r = urlread('http://localhost:8080/example_node/
metric/{"type":"metric_a", "time_window":[5]}');
```

MIDAS also supports **JSONP** for use with JavaScript.

INSTALLATION

Remember to use Python 3!

```
pip install git+https://github.com/bwrc/midas
```

RESOURCES

Source and Wiki

github.com/bwrc/midas

The API

github.com/bwrc/midas/wiki/API

FUTURE

Mobile Applications: Move experiments out of the lab

Internet of Things: Utilize wearable health technology

CONTRIBUTE!

Together we can make MIDAS what it needs to be!

Q & A