

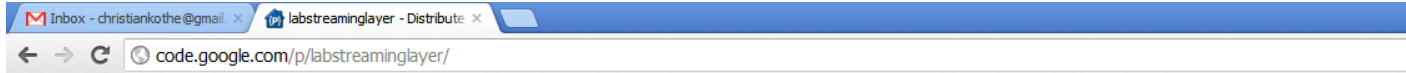
Demo 1: The Lab Streaming Layer


Introduction to Modern Brain-Computer Interface Design

Christian A. Kothe
SCCN, UCSD



The Lab Streaming Layer




 **labstreaminglayer**
Distributed signal transport, time synchronization and collection system for research use


[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#) [Administer](#)

[Summary](#) [People](#)

Tip: Discuss and then document [each teammate's project duties](#).


Project Information

 Recommend this on Google

 Starred by 0 users
[Project feeds](#)

Code license
[MIT License](#)

Labels
[Academic](#), [Interface](#), [Lab](#), [Library](#), [Middleware](#), [Networking](#), [Stream](#), [Research](#)

 **Members**
[christiankothe](#)
[3 committers](#)

Your role
[Owner](#)

Summary

The **lab streaming layer** (LSL) is a system for the unified collection of measurement time series in research experiments and handles both the networking, time-synchronization, (near-) real-time access as well as optionally the centralized collection, viewing and disk recording of the data.

The LSL **distribution** consists of:

- The core transport library (liblsl) and its language wrappers (MATLAB, Python, C, C++). The library is general-purpose and cross-platform (Win/Linux/macOS, 32/64) and forms the heart of the project.
- A suite of tools built on top of the library, including the recording program, a viewer program, importers, and a set of data collection apps that make data from a particular device available on the lab network (for example audio, EEG, or motion capture). The existing tools suite is tailored to the needs of only a small number of labs and should not be considered as general (or production-quality) as the library itself.

Streaming Layer API

The liblsl library provides the following **abstractions** for use by client programs:

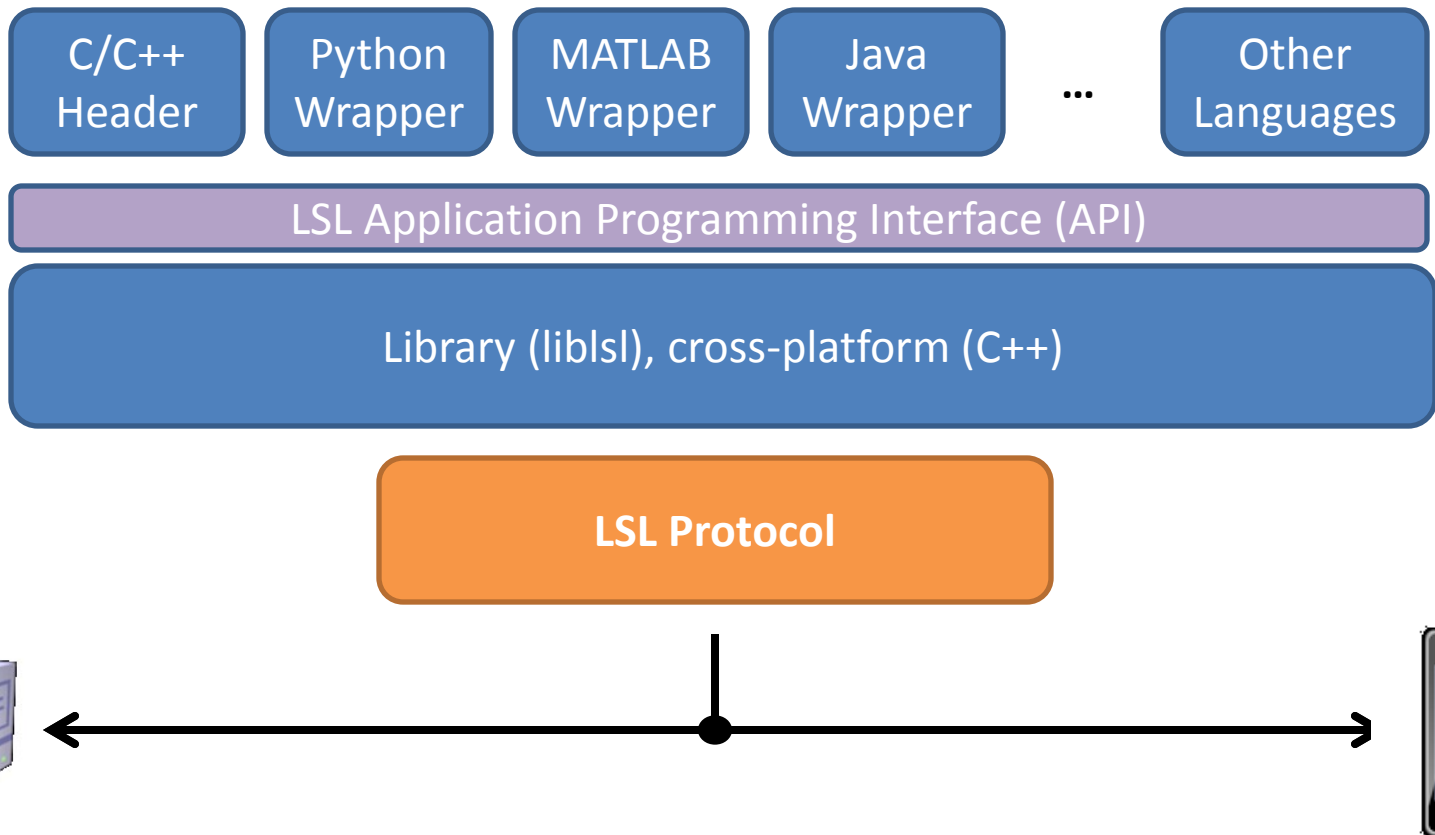
- **Stream Outlets:** for making time series data streams available on the lab network. The data is pushed sample-by-sample or chunk-by-chunk into the outlet, and can consist of single- or multichannel data, regular or irregular sampling rate, with uniform value types (integers, floats, doubles, strings). Streams can have arbitrary XML meta-data (akin to a file header). By creating an outlet the stream is made visible to a collection of computers (defined by the network settings/layout) where one can subscribe to it by creating an inlet.
- **Resolve functions:** these allow to resolve streams that are present on the lab network according to content-based queries (for example, by name, content-type, or queries on the meta-data). The service discovery features do not depend on external services such as zeroconf and are meant to drastically simplify the data collection network setup.
- **Stream Inlets:** for receiving time series data from a connected outlet. Allows to retrieve samples from the provider (in-order, with reliable transmission, optional type conversion and optional failure recovery). Besides the samples, the meta-data can be obtained (as XML blob or alternatively through a small built-in DOM interface).
- **Built-in clock:** Allows to time-stamp the transmitted samples so that they can be mutually synchronized. See Time Synchronization.

Time Synchronization

code.google.com/p/labstreaminglayer

LSL Core Components

- Low-level technology for exchange of time series between programs and computers



LSL Software Distribution

- Includes Documentation, User Guides, Example Programs, Acquisition Programs, Generic Tools
- Everything is open source (MIT-licensed)

Acquisition Programs (EEG, Eye tracking, Human Interfaces, Motion Capture, Multimedia)

Generic Viewers, Recorder

Example Programs

Wiki Documentation

Core Components

C/C+
+ API

Python
API

MATLAB
API

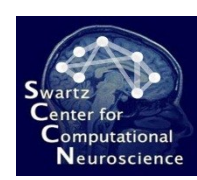
Java
API

...

Future
Languages

Library (liblsl), cross-platform (C++)

LSL Protocol



Supported Hardware

- **EEG:** Biosemi, Cognionics, MINDO , EGI AmpServer, BrainProducts, g.USBamp, Emotiv, Micromed, MindMedia, OpenEEG, TMSi, ANT Neuro ASALAB
- **Eye Tracking:** SR Research EyeLink, custom 2-camera setup
- **Motion Capture:** PhaseSpace, OptiTrack, Kinect, AMTI Force Plates
- **Human-Interface Devices:** Mice, Keyboards, Trackballs, Game Controllers, Wiimote and Expansions
- **Multimedia Devices:** PC-compatible sound cards, DirectShow-compatible video hardware
- **Untested:** ABM B-Alert, Enobio, Neuroscan Synamp, Tobii, SMI iViewX, Mitsar EEG, CTF/VSM

*: most devices currently require Windows

Receiving Data from LSL

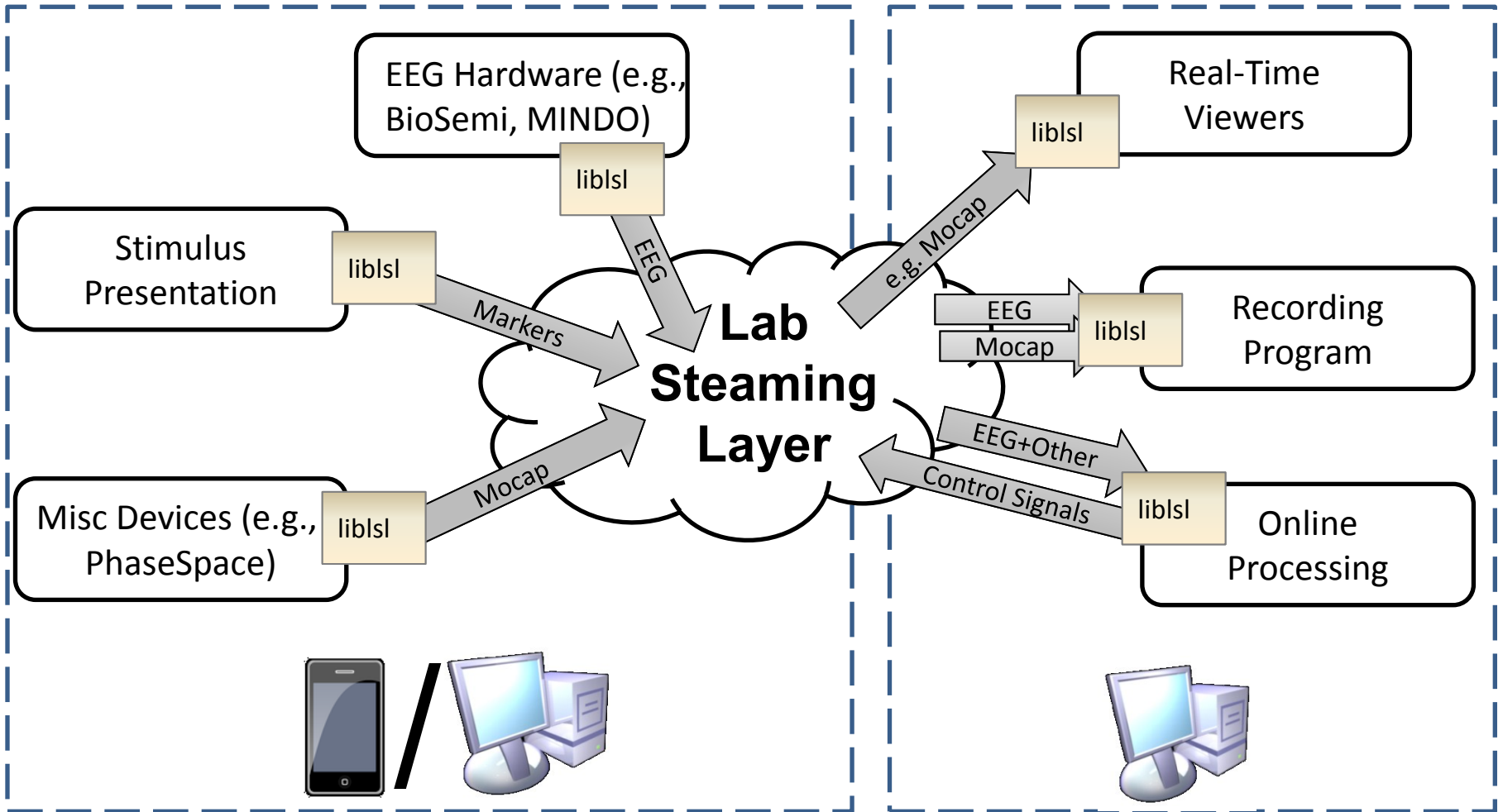
```
% instantiate the library
lib = lsl_loadlib();

% try resolve an EEG stream...
result = {};
while isempty(result)
    result = lsl_resolve_byprop(lib, 'type', 'EEG'); end

% create a new inlet from the first result
inlet = lsl_inlet(result{1});

while true
    % get data from the inlet and print it
    [vec,ts] = inlet.pull_sample();
    fprintf('%.2f\t',vec); fprintf('%.5f\n',ts);
end
```

Network View





Hands-On Section



D1 Questions?