

1 Purpose of this document

This is a discussion document to establish a common understanding, in layman's language, what the Free and Open Source Instrument Server, Froid, does. It is not a technical specification, but will be used to ensure Froid fulfils basic functions.

1.1 Convention

Note the document is written in the present tense, describing the requirement as if it is there already. Lower priority items are designated as **Phase II and III**.

For the purpose of this document a LIMS equals a LIS. But for Patients, Clinical Cases, Patient Privacy measures and regulatory extras added to the latter.

1.2 Document history

Adapted from *BLIS. Bika LIMS Instrument Server v 0.6, 2012*. Technical parts are to be updated from forum discussions under way.

Working title *BLIS* was since dropped, it clashed with BLIS the Basic LIS, and *Froid* Middleware is vendor independent.

2 Laboratory Instrument Middleware. Overview

2.1 Bidirectional Instrument Interfacing

Interface-able laboratory instruments come in all shapes and variants, old and new. Many export results as CSV only that are easily imported by Froid.

Next level, they can also import Sample ID lists corresponding to positions on the sample trays loaded into them.

If the instrument automatically scan barcodes from sample vials, a Froid import interface is redundant.

Many instruments do not do CSV and have more complex live interfaces on serial RS-232 communications, bi-directionally, and Froid resolves this too.

2.2 Goal

Froid is vendor independent for easy interfacing of analysers and instruments to laboratory management Systems such as LIMS.

Froid is vendor neutral, well documented and easy to maintain, to encourage developers and instrument manufacturers alike to add interfaces to the system.

The current Bika LIMS interface module was designed with Froid in mind and can be used as a basis. Wiki:

[Supported Instruments¹](#)

[Creating an Instrument Import Interface²](#)

2.3 Objective

Froid communicates bi-directionally with LIMS and instruments, converting data to and fro between sources and destinations, standards-compliant secure communication protocols with data encryption, provision for images and data blobs, and enough redundancy to take care of future developments.

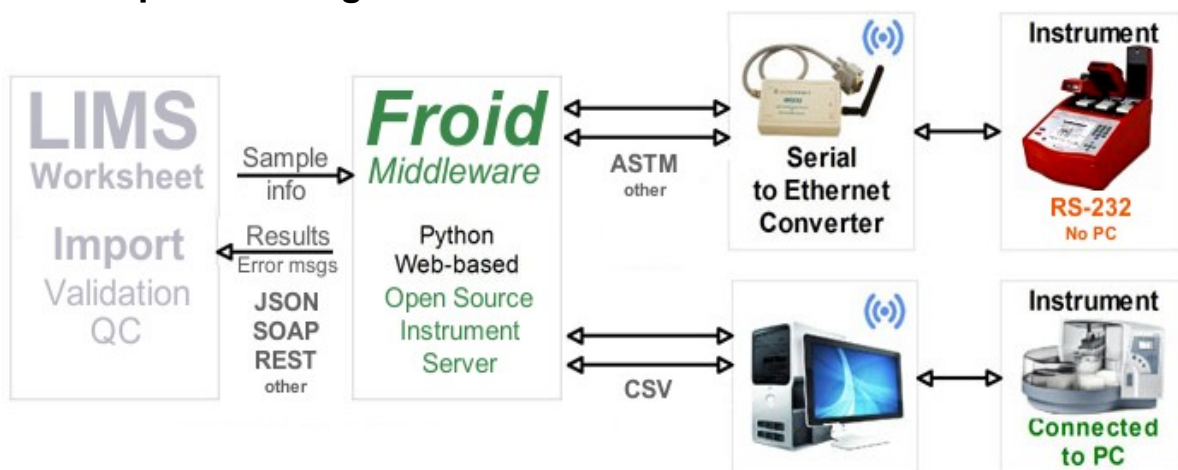
For better load balancing, Froid must be free-standing to allow for it to be deployed on its own hardware in high volume labs. But at the same time, run on low spec PCs or Raspberry Pi.

Customisation in LIMS employing Froid is kept to adjusting IO to one of the available adapters, or adding its own. The Froid user interface for management and maintenance is web based for easy remote access.

2.4 License Considerations

Ideally one would like to see all Froid implementers add their interfaces to the free and open Froid pool. It would be unfair on FOSS contributors if Big Instrument/LIMS uses Froid and available free interfaces without opening their own? If that is the objective, the AGPL might not be enough?

3 Top level Design



For instruments not connected to PCs and capable of serial communication only, Ethernet converters are recommended. They are not only cheaper than providing PCs for all instruments, but also don't require cabling if Wi-Fi with secure encryption is chosen.

They often feature on-board web servers which make remote support possible.

1 <https://github.com/bikalabs/Bika-LIMS/wiki/Supported-instrument-interfaces>

2 <https://github.com/bikalabs/Bika-LIMS/wiki/creating-an-instrument-import-interface>

3.1 Barcode scanning

For instruments reading Sample Information from barcodes, they are produced on the LIMS and affixed to them. The instrument reads them as they go through and there is no need for Froid.

Users may also scan the samples manually with handheld scanners into Instrument UI while loading the samples.

For these instruments, Froid's function is reduced to mono-directional interfacing

4 Technical Development Considerations

4.1 Python

Platform independent, but development in Python and Linux. Because we know how and already told Google so.

4.2 LIMS Protocol

Special attention should be paid to Froid to LIMS communication. It should follow recognised standards with eventual LIS implementation with Patient privacy considerations.

Two different parts of of the Interface needs specification:

- Communication protocol
- Data format

There is, for instance, no point in insecurely serving CSV formatted data via USB storage.

Candidates:

- [JSON API](#)³
- [JSON RPC](#)⁴, Remote Procedure Call Protocol
- [SOAP](#)⁵ Simple Object Access Protocol
- [REST](#)⁶ Representational State Transfer

4.3 Serial RS232 communications

One of the purposes of Froid is to resolve serial RS232 communications with laboratory instruments.

These follow protocols such as [ASTM](#)⁷ [CLSI](#)⁸, [DICOM](#)⁹ etc. Older instrument interfaces are often poorly documented or stray from standards. Others use proprietary protocols.

3 <http://jsonapi.org/>

4 <http://json-rpc.org/>

5 <http://en.wikipedia.org/wiki/SOAP>

6 http://en.wikipedia.org/wiki/Representational_state_transfer

7 <http://www.astm.org/>

8 <http://clsi.org/>

9 <http://en.wikipedia.org/wiki/DICOM>

Examples of popular clinical instruments for resource limited settings are with specifications attached:

[Becton Dickinson FACSCaliber. FACSLink¹⁰](#)

[Becton Dickinson MGIT 960¹¹](#)

[Roche Cobas C 111¹², C 311¹³, E 411¹⁴](#)

[SUIT Sysmex Universal Interface¹⁵](#)

[Thermo Fischer ASTM protocol¹⁶](#)

4.4 CSV

Many instruments simply export results as CSV or TXT files and Froid parses the native file formats and post them in LIMS configured format and protocol to the system.

For Bika's current, see wiki:

- [Instrument Interfacing](#)
- [Supported Instruments](#)

For the quickest solution this infrastructure can be maintained but be modernised.

5 Froid

Froid strives for a single interface per LIMS to keep customisations on the LIMS itself down to single interface which it probably already has.

5.1 Worksheets to instruments for analysis

Users assemble worksheets of IDs for the samples to be analysed in the LIMS, and post them to the selected instrument via Froid, selecting from a drop-down menu of measurement apparatus available.

From here on it depends on how the specific instrument functions. Loosely defined, it could be 'listening' to Froid, and the data simply streamed to it. Either way, Froid writes the data to the instrument when it is ready to receive it.

Froid converts the sample information into the correct Instrument format and dispatch it to the connected destination device using it's configured protocol.

Once the analyst is satisfied the worksheet has been successfully transferred, he/she proceeds with analysing the samples.

LIMS and Instrument manufacturers are expected to add their own channels to Froid, using any of JSON, SOAP, XML etc.

10 <https://jira.bikalabs.com/browse/HEALTH-235>

11 <https://jira.bikalabs.com/browse/HEALTH-234>

12 <https://jira.bikalabs.com/browse/HEALTH-236>

13 <https://jira.bikalabs.com/browse/HEALTH-237>

14 <https://jira.bikalabs.com/browse/HEALTH-238>

15 <https://jira.bikalabs.com/browse/HEALTH-239>

16 <https://jira.bikalabs.com/browse/HEALTH-240>

5.2 Results from instruments

Once analysis is completed, the analyst exports it from the instrument. How this is done will depend on the individual instrument. It might happen automatically or the instrument operator has to action a 'send' function.

Instruments write their results to Froid, in the case of CSV file results to a designated folder Froid polls regularly. Froid converts the data to the LIMS' format and post it per designated protocol.

Froid logs this procedure but data validation is carried out in the LIMS where issues such as unknown IDs or keywords can be highlighted.

The best way to do this is to be determined after technical analysis. It might be easy to reconcile results and sample IDs in the LIMS.

The LIMS is configured for complete QC and that should not be a Froid function.

The process should be automated as far as possible – only when keyword inconsistencies or other import issues arise should user interaction be necessary.

5.3 Fetching results

Some instruments, e.g. the BD MGIT 960 in GND mode, only issue results streams in response to a command from the LIMS

Froid should therefore have the ability to issue commands to the instrument, and to receive data sets from the serial interface in response to these commands.

The user should be able to request the results from the LIMS without having to log into Froid itself. This will mean a “go ask the instrument” function in the LIMS.

5.4 Commanding Instruments

Some LIMS implementations will also require instruments to be managed from the LIMS itself, e.g. started up and initialised, shut down, cleaned etc.

This opens an additional spec for Froid: An open, modular interface structure that allows for easy extension for both commands and data streams.

5.5 Format

Note that instrument imports do not have to match worksheets in the LIMS, the sample and AR IDs are used to match the results to requested analyses regardless of whether they were grouped together on worksheets or not.

Many instruments produce analysis results that are reported with a number of interim values. In the case of the different spectrometers, the result itself is reported, say in ppm, as well as any combination of: dilution factor, reagent volume, peak number, peak height, mean, standard deviation, relative standard deviation, coefficients, etc.

These are then repeated per analyte on the sample while at the same time a differing number of data fields can be reported for the sample itself, e.g. temperature, number of data points (on the curve), starting peak width, peak threshold, etc.

The analysis batch itself could have a number of attributes too, seen before: reagent ID, batch no, electrode ID, serial no, column, etc.

In pseudo format:

Repeat for Import header:

Field title

Field value

Repeat for Samples

Field title

Field value

Repeat for Analytes

Field title

Field value

Repeat for Interim values

Field title

Field value

6 Other LIMS considerations

The results received from Froid enter standard LIMS workflow and QC.

6.1 Correcting problematic imports

The LIMS could use a 2 phased import process and alert users to problematic import files, e.g. the use of non-compliant keywords, and allows for human interaction to fix issues via an easy to use GUI before data are submitted to the DB.

6.2 Referencing import files

For traceability purposes, analysis results imported from an instrument can be linked to the raw input data files where they originated from.

These files can be uploaded centrally, in the Instrument's folder, and then referenced where required on results or worksheet views.

7 Mirth Connect ®

The Swiss Army knife of healthcare integration engines, mature and well supported Open Source [Mirth Connect](http://www.mirthcorp.com/products/mirth-connect)¹⁷, facilitates HL7 messaging for EMR and other medical systems. It could be worth the while to investigate as instrument server too.

Some instruments need low-level harvesting of data, and to receive commands to issue results, there is no streaming as such. It needs to be investigated how Mirth would do this.

¹⁷ <http://www.mirthcorp.com/products/mirth-connect>

Paraphrased from the [Wikipedia](#)¹⁸:

Mirth Connect is a cross-platform HL7 interface engine that enables bi-directional sending of HL7 messages between systems and applications over multiple transports.

Mirth Connect uses a channel-based architecture to connect HIT systems and allow messages to be filtered, transformed, and routed based on user-defined rules.

Channels consist of inbound and outbound connectors, filters, and transformers. Multiple filters and a chain of transformers can be associated with a channel.

Endpoints are used to configure connections and their protocol details. Source connectors are used to designate the type of listener to use for incoming messages, such as TCP/IP or a web service.

Destination connectors are used to designate the destination of outgoing messages, such as an application server, a JMS queue, or a database.

All messages and transactions are optionally logged to an internal database.

Mirth Connect can also be configured to auto-generate HL7 acknowledgement responses (ACK).

Mirth Connect supports messages over a variety of protocols, including:

TCP/MLLP, Database IO, ODBC, File, JMS, SFTP, HTTP, SMTP, SOAP.

Open architecture allows for the easy addition of custom and legacy interfaces.

Types of transforms

- Mapping. Data from incoming message to variables
- Scripts. Execute custom script on message, JS, Python etc.
- HL7 Generator. Construct HL7 messages from data source
- XSLT. Run XSL transformations on incoming HL7 or XML encoded messages

8 Instrument Maintenance

Most labs will manage instrument certifications, calibrations. Maintenance in their LIMS or ERP. Froid needs to pick alerts up from instruments though and report them to listening LIMS. Required dashboard in LIMS. Else just de-activate the Instrument.

¹⁸ [http://en.wikipedia.org/wiki/Mirth_\(software\)](http://en.wikipedia.org/wiki/Mirth_(software))