UMONS
Université de Mons

POLYTECH MONS

R-testbench

A Python library for instruments remote control and electronic test bench automation

Ir. Alexandre QUENON (UMONS)

Remote instrumentation lacks free, open source, reliable and user-friendly tools.

Proprietary tools

such as LabVIEW or MATLAB:

- are generally reliable;
- offer some graphical programming capability;
- but are expensive.

Open source tools

such as Octave or Scilab:

- are not always reliable;
- require (very) low-level programming skills;
- are generally free.

We solved the issue with R-testbench, a Python library for instrument remote control.

Specifications:

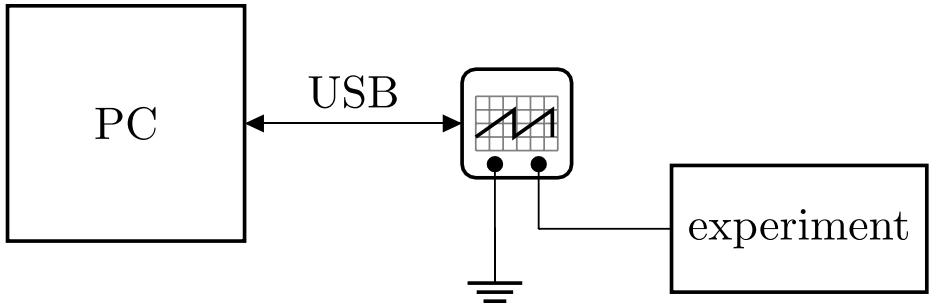
- reliable;
- user-friendly (high-level programming);
- free and open source (OSL v3);
- cross-platform.

How the software architecture makes the tool user-friendly

How the data are managed for later processing

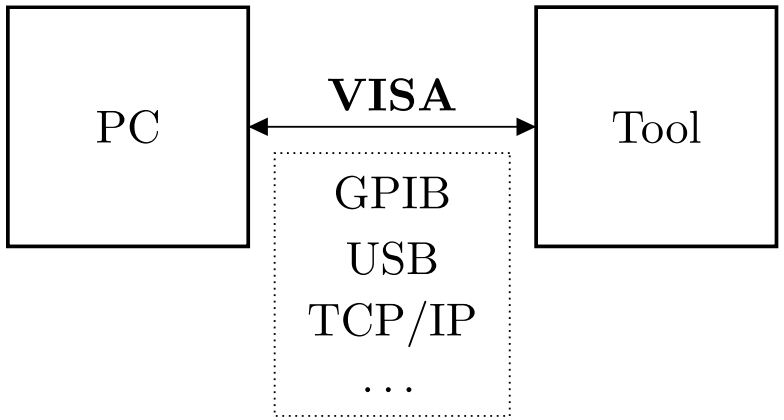
How the software is validated

The design of R-testbench complies with the characteristics of real (critical) use cases.



- One automated test bench controlled from a computer.
- Several types of connections that can change, what should not interfere with the software.
- Several families of instruments, but the same operations within a family, hence a unique interface.

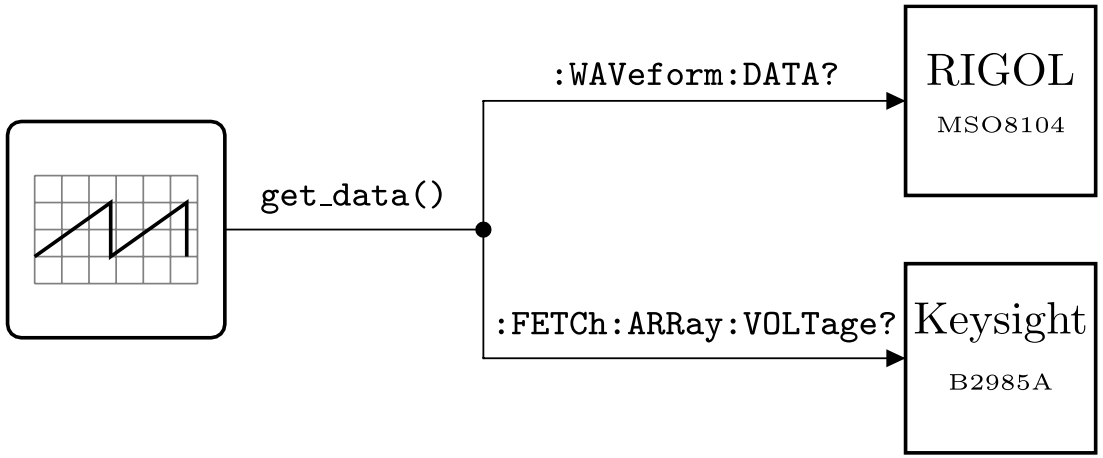
R-testbench makes the type of connection transparent (hidden) from the software.



- VISA (Virtual Instrument Software Architecture) is a reliable and widely implemented standard to communicate with instruments through several interfaces.

- The type of connection is hidden in the software by using a Python wrapper that calls the right VISA libraries: PyVISA.

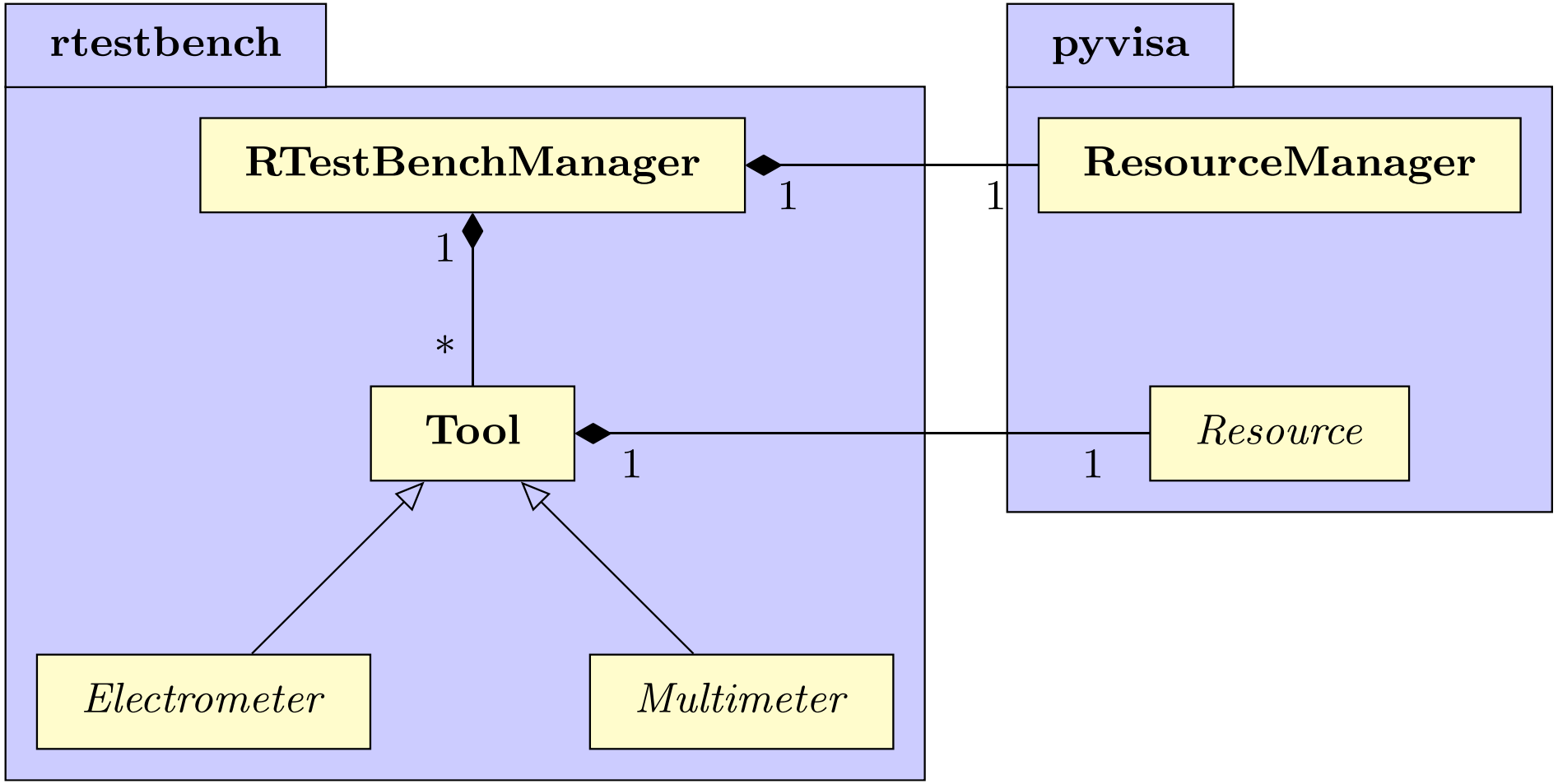
R-testbench identifies automatically the model of instrument to send the right commands.



- On the test bench side, the user can choose any specific model of a family of instruments, which has its own characteristics and commands.

- On the software side, the interface is always the same for a family of instruments, which makes it user-friendly.

The software architecture enables the desired features with the object-oriented paradigm.



How the software architecture makes the tool user-friendly

How the data are managed for later processing

How the software is validated

The management of the data coming from the instruments requires efficiency and flexibility.

This is implemented in three main steps:

- store the data locally → NumPy's N-dimensional array;
- save the data in various files formats → pandas' rich API (csv, hdf5…);
- process the data to extract information → any tool well-known by the user.

How the software architecture makes the tool user-friendly

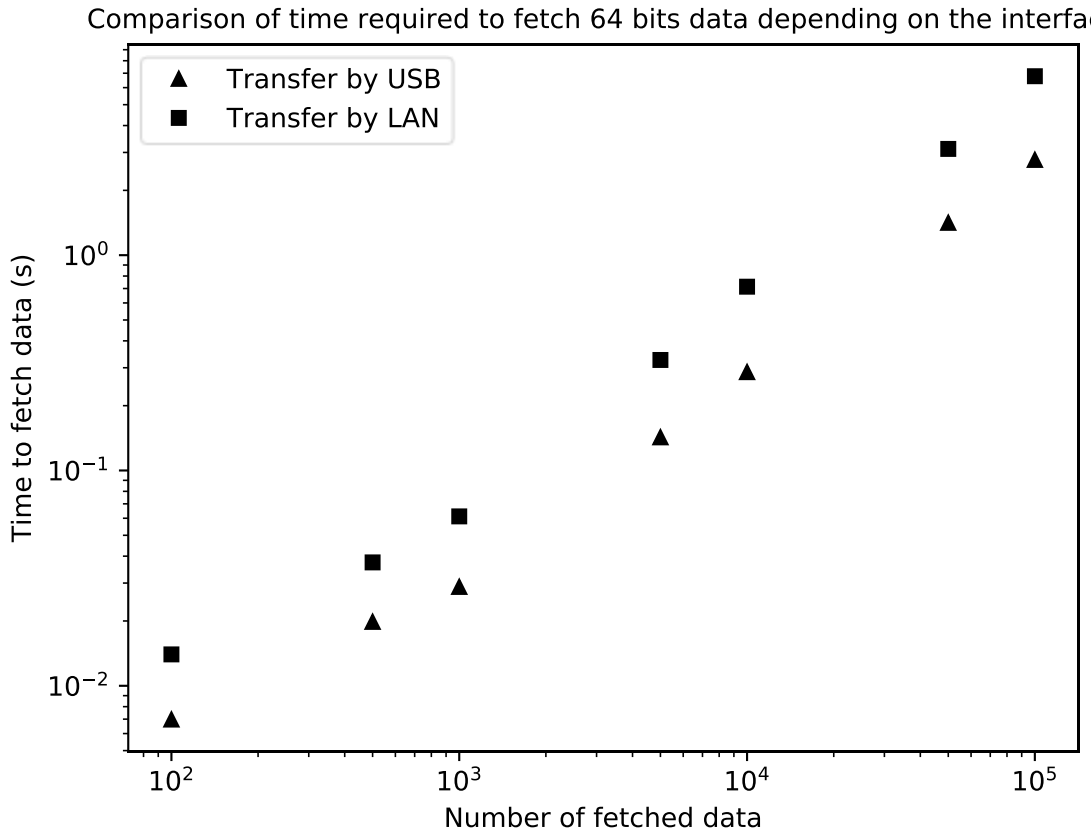
How the data are managed for later processing

How the software is validated

R-testbench has been and is still being validated by four means.

- The library has been extensively characterized.
- R-testbench has been applied to a real use case involving remote characterization of semiconductor devices.
- Tests including functional and coverage tests are run automatically.
- The package is available for download to the open source community (GitHub).

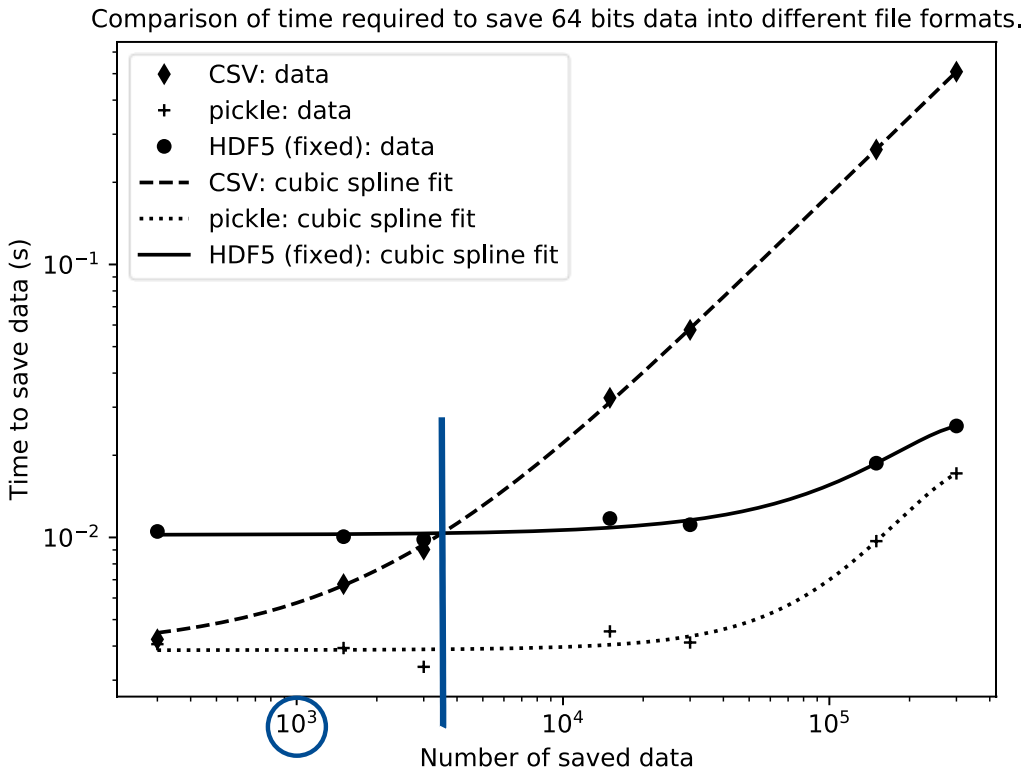
Characterization: the time required to fetch data from an electrometer is linear with the number of data.

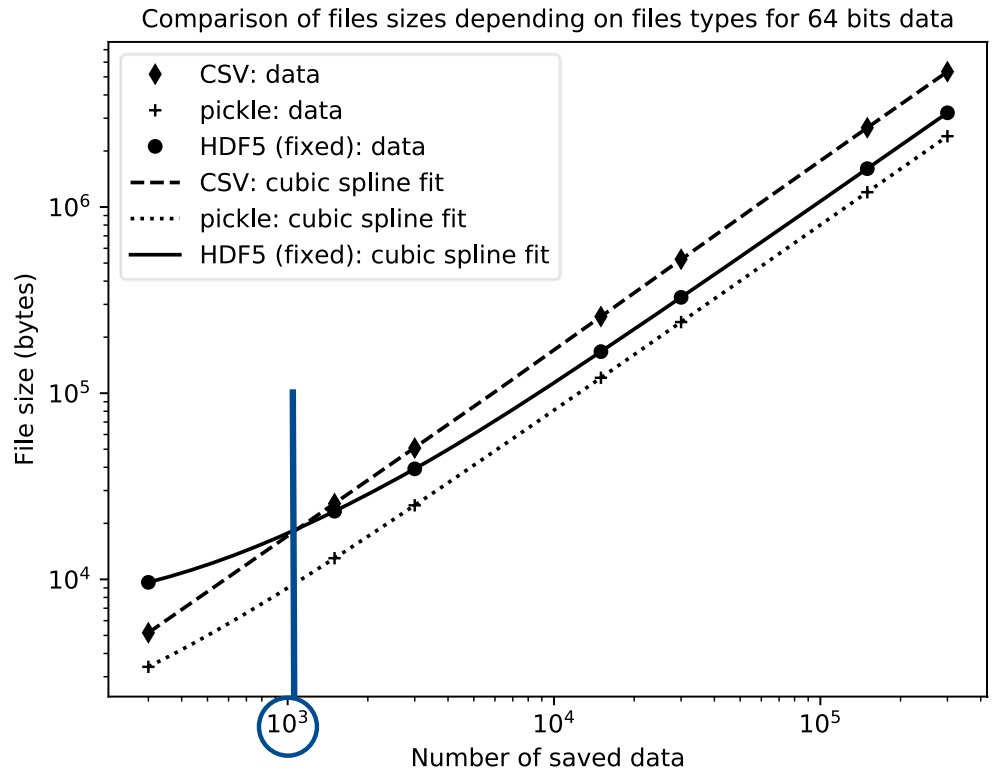


$t_{fetch,\ LAN} = 0.064\ N + 6.7$ (ms)

$t_{fetch,\ USB} = 0.0281\ N + 4.8$ (ms)

Characterization: the binary formats exhibit the best size-speed performance, except for less than 1000 data.





Three things to remember:

- R-testbench is a Python library that offers an open source, user-oriented tool to control instruments remotely and automate test benches;

- the data coming from instruments are embedded into efficient containers and can be saved for later use by any data processing tool;

- the library has been validated and is publicly available on GitHub, waiting for feedback by the scientific community.

  (To use or contribute: https://github.com/Arkh42/rtestbench.)

R-testbench is a new Python library that solves the lack observed in remote instrumentation.

The tool makes the emphasis on:

- the user (high-level programming);
- the reliability;
- the sharing (open source).

UMONS
Université de Mons

POLYTECH MONS

R-testbench

A Python library for instruments remote control and electronic test bench automation

Ir. Alexandre QUENON (UMONS)

Thanks to fnrs FREEDOM TO RESEARCH inforTech for their support.