



Linking data automatically from their origin

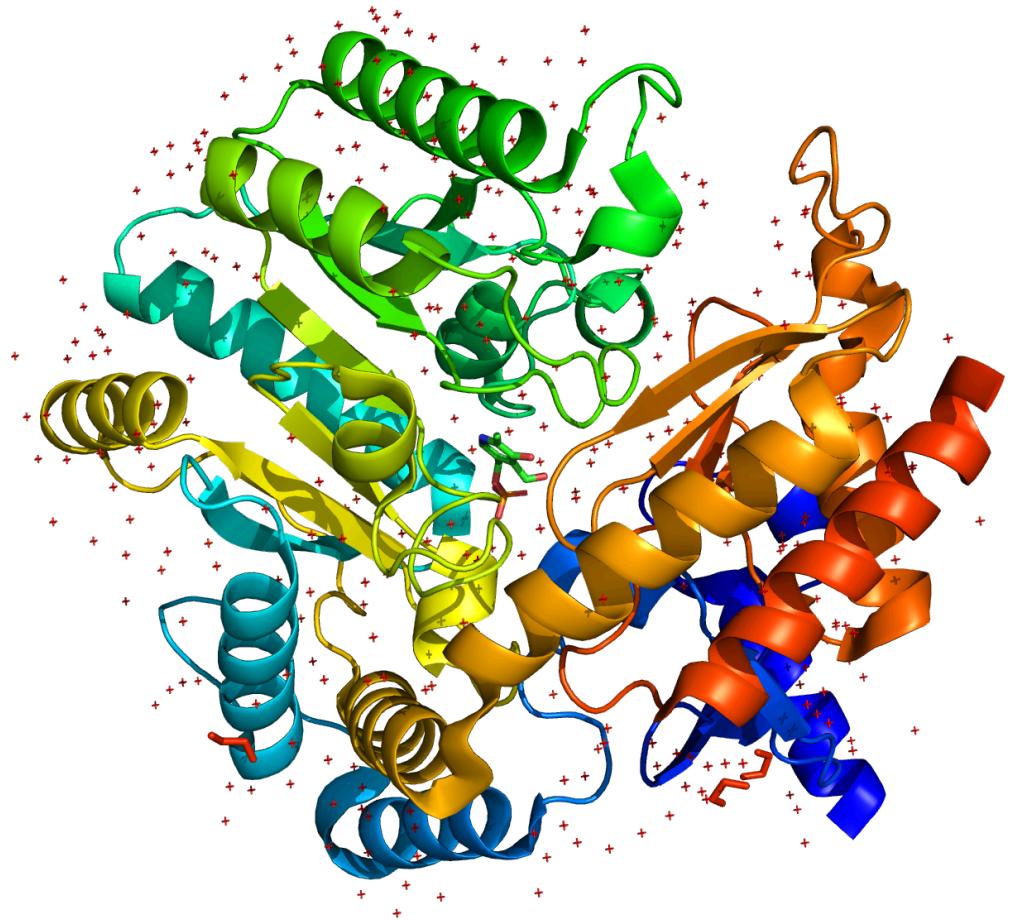
a demonstration of the LARAsuite open data platform

mark doerr, stefan maak, stefan born & uwe bornscheuer

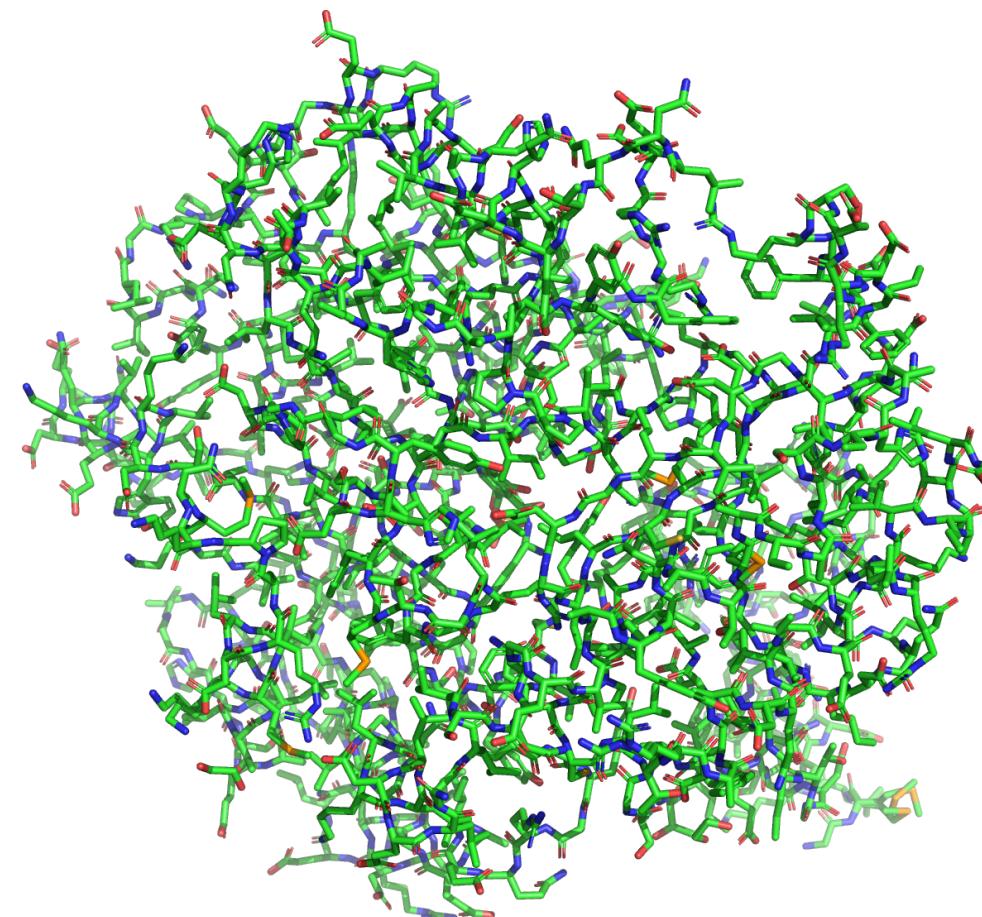
university greifswald / tu-berlin

2024-12-12

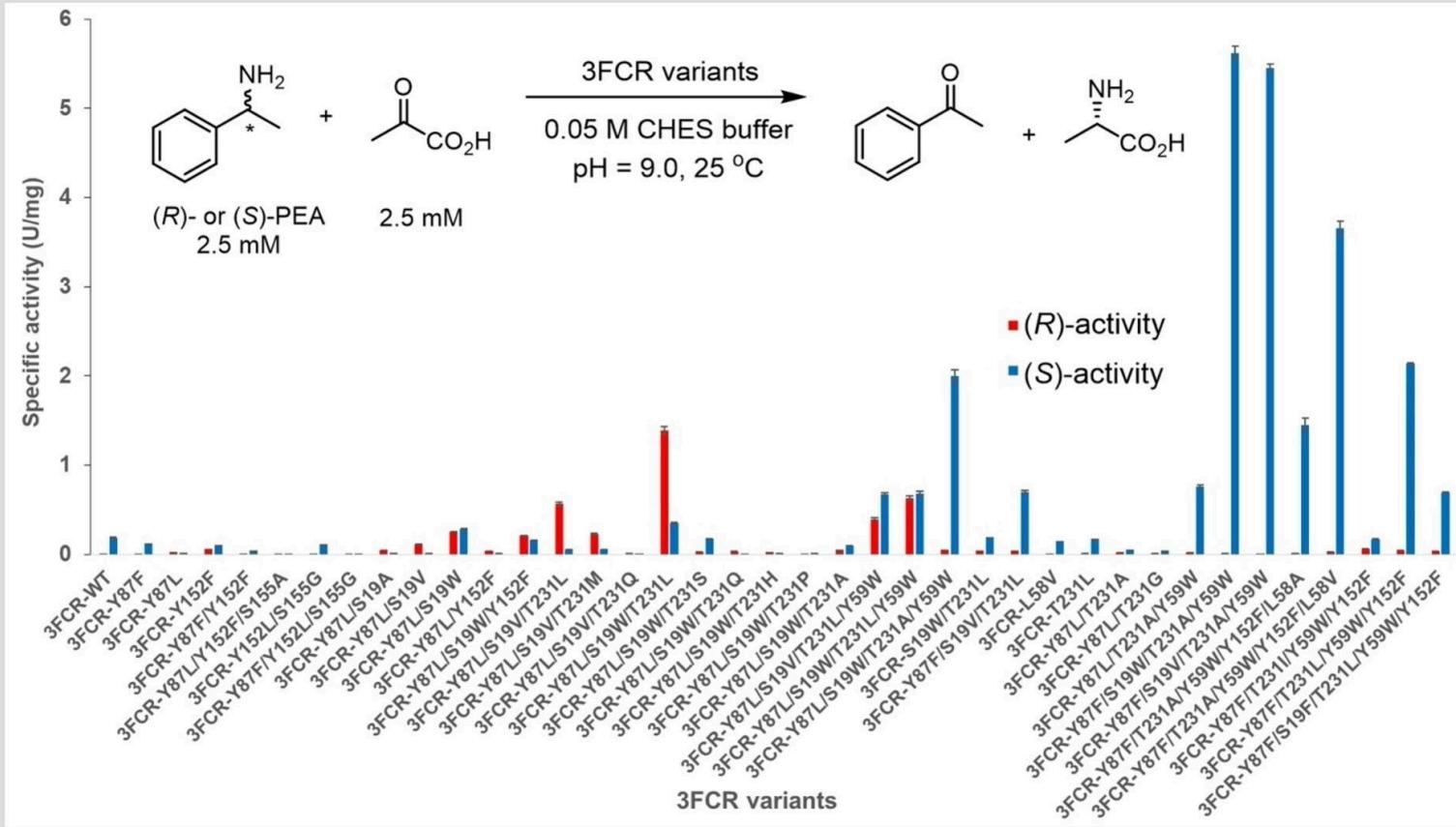
use case/motivation



machine learning

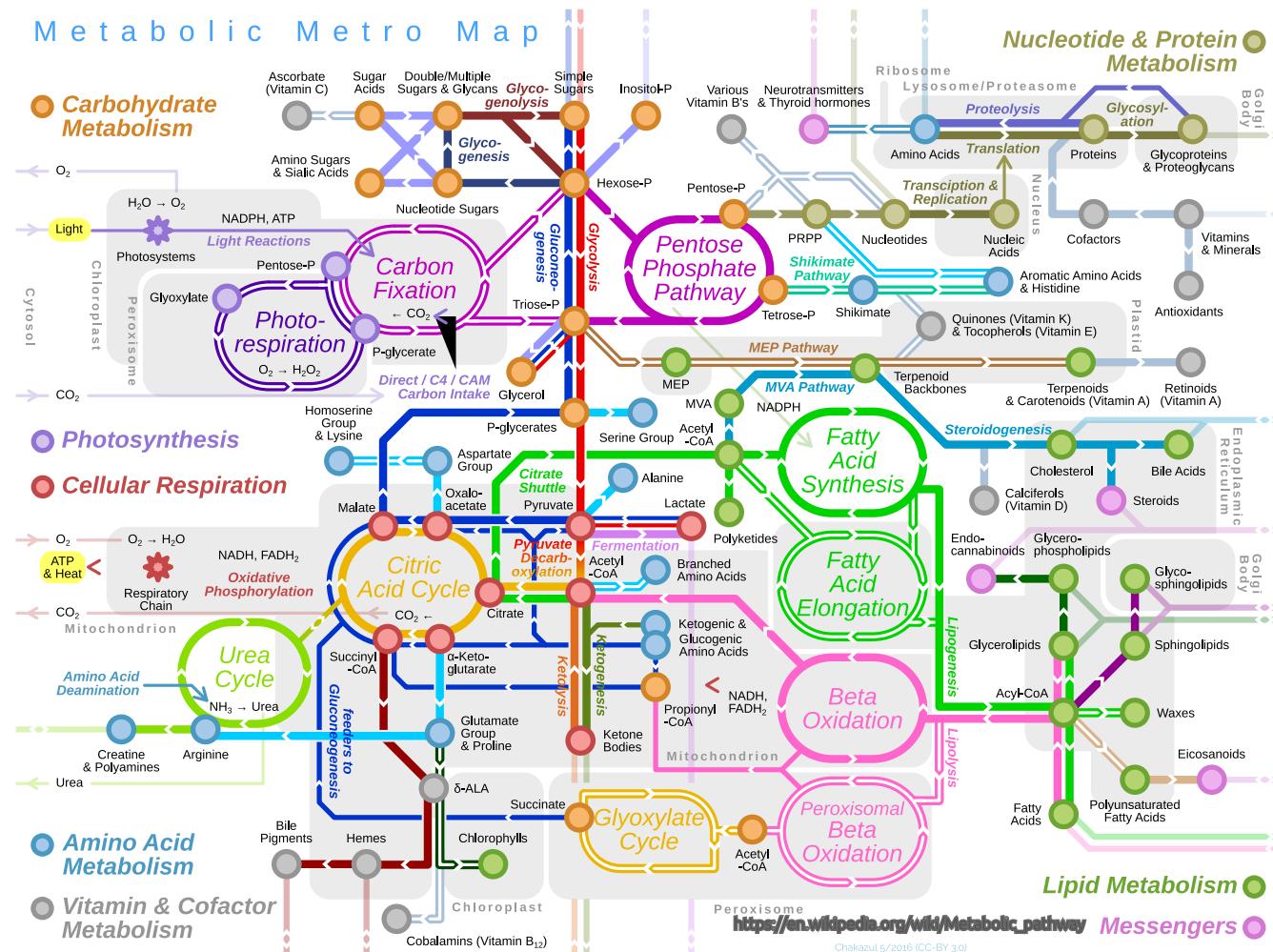


machine learning



Source: Structure- and Data-Driven Protein Engineering of Transaminases for Improving Activity and Stereoselectivity Yu-Fei Ao et. al, Angewandte Chemie 2023. <https://doi.org/10.1002/anie.202301660>

complex expression system



complex instrumentation



process planning

LARA

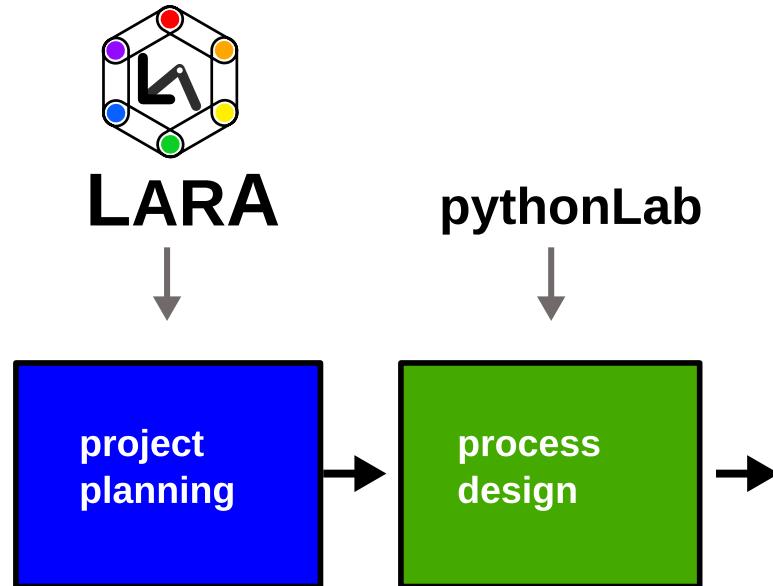


LARA



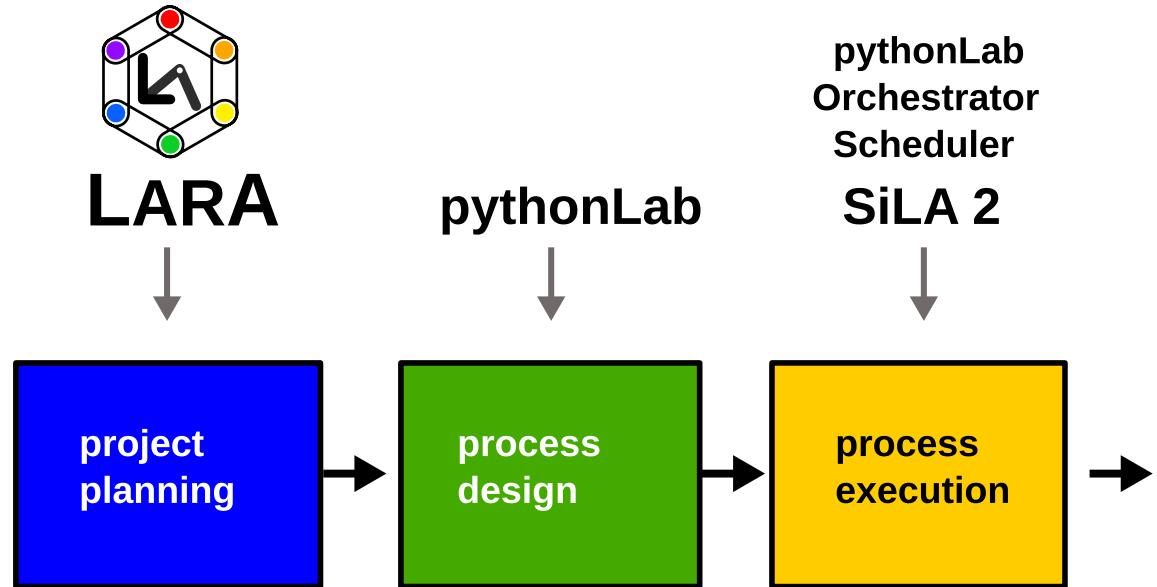
process design

LARA



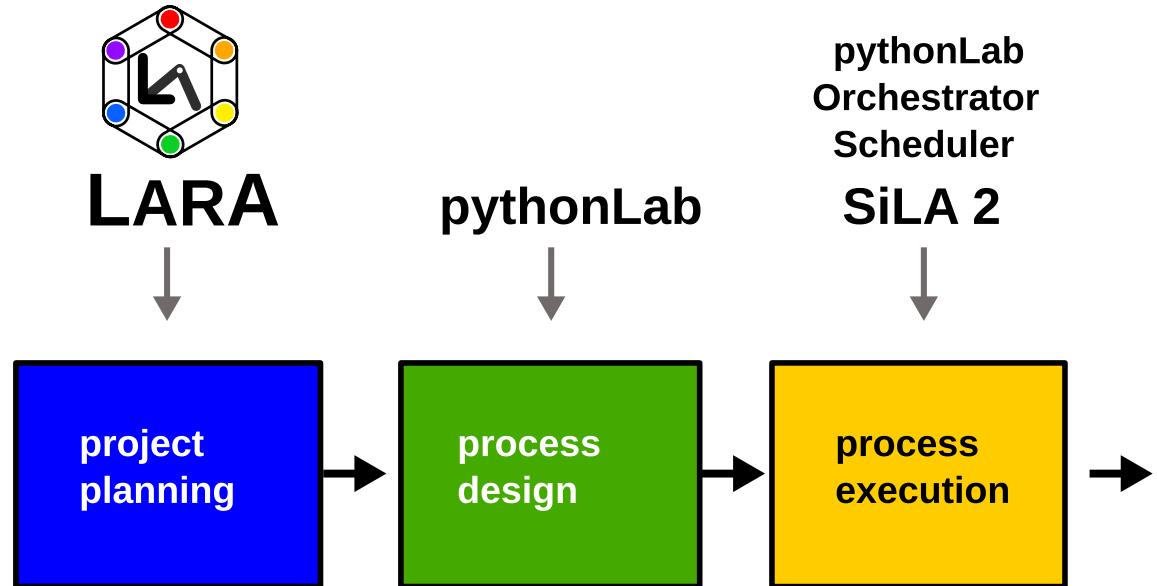
process execution

LARA



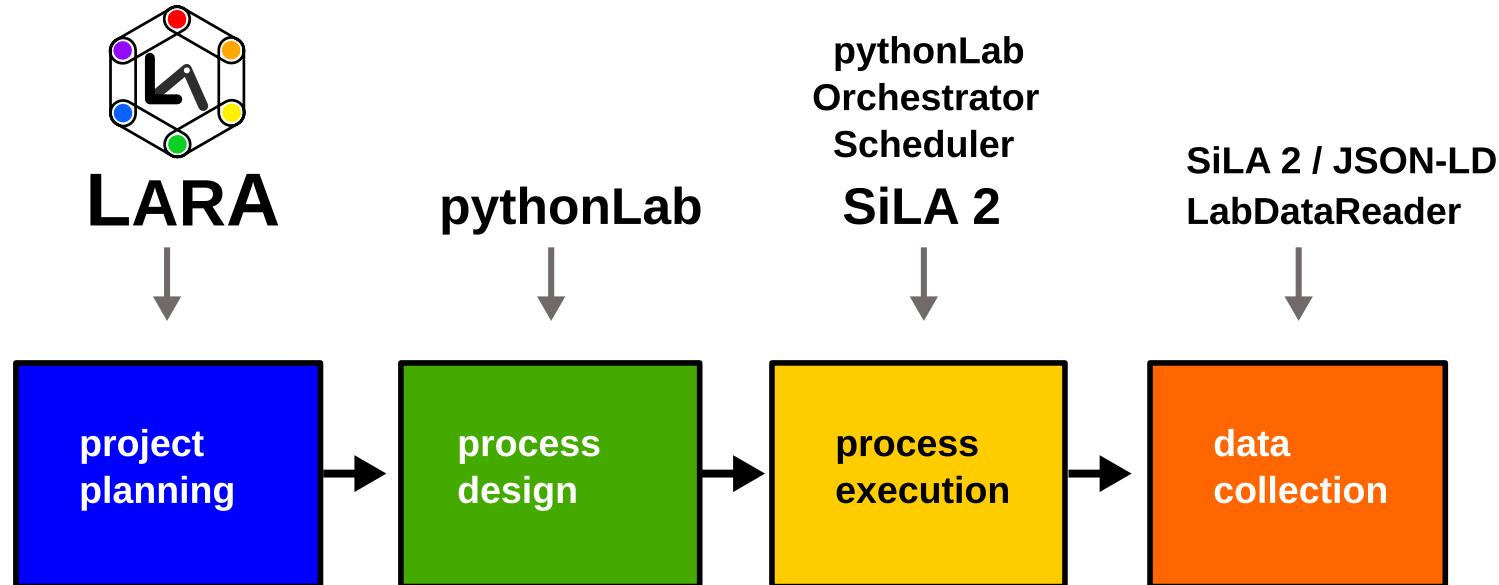
process execution

LARA



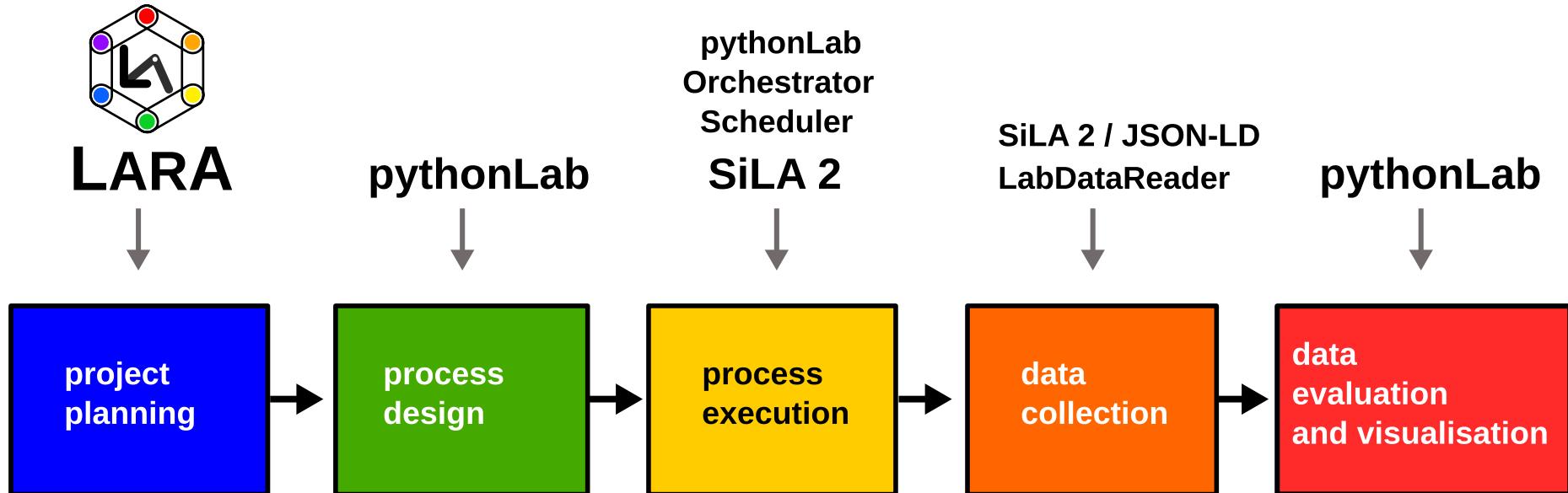
data collection

LARA



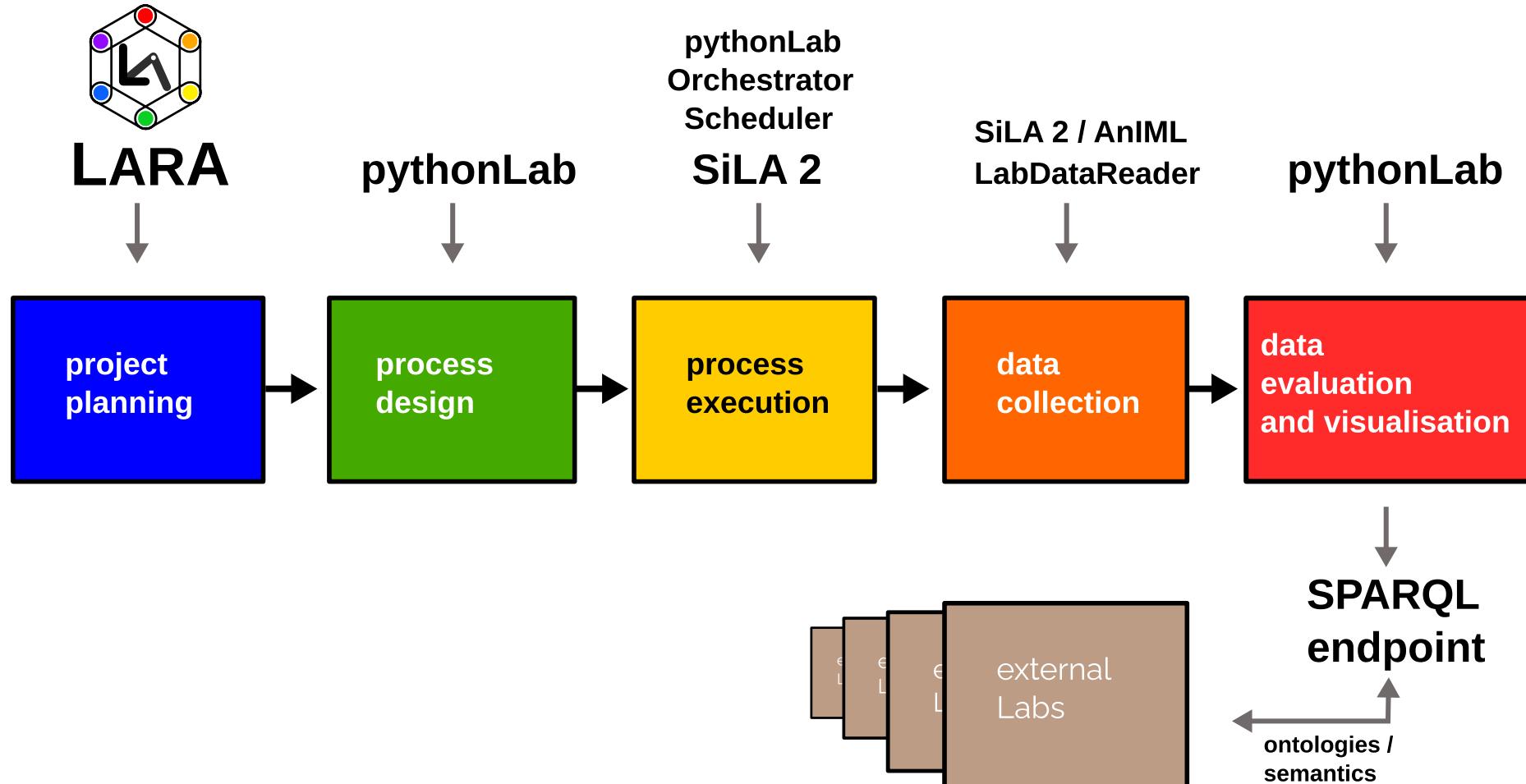
data evaluation / visualisation

LARA



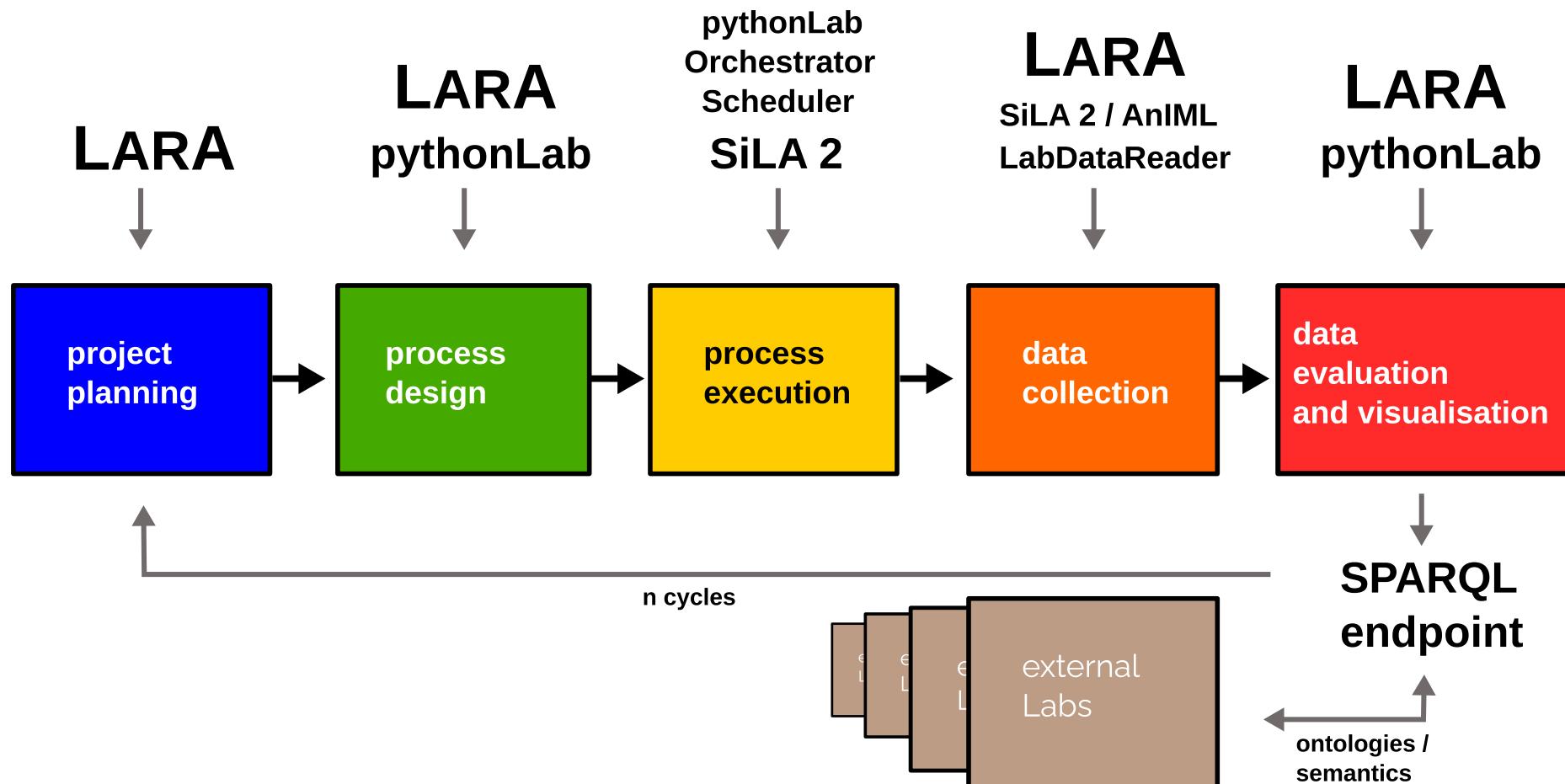
querying of data

LARA



closing the loop

LARA



motivation: why we need full
automation of metadata
aquisition and semantic data ?

- meaning of the data is well defined

- meaning of the data is well defined
- machine readable

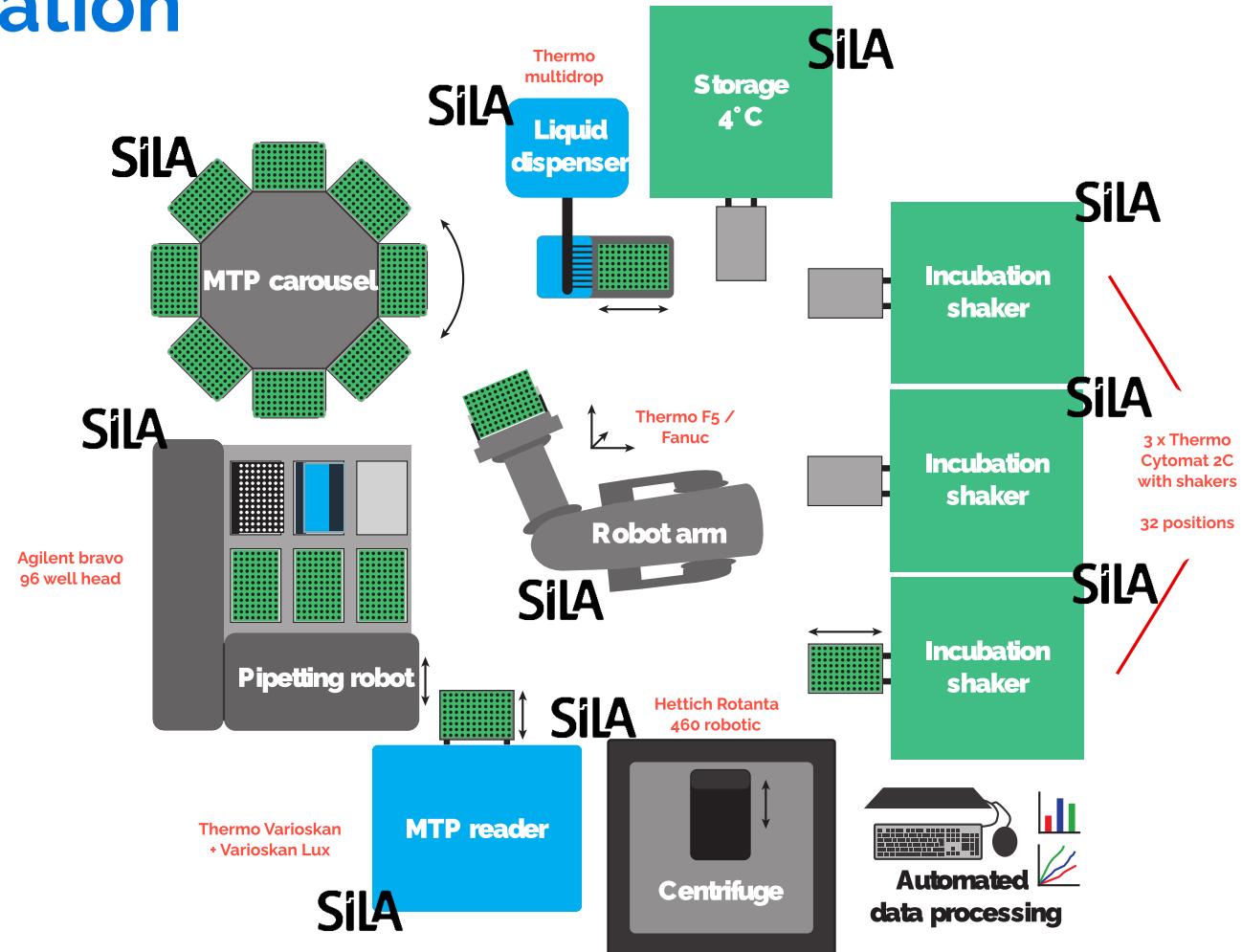
- meaning of the data is well defined
- machine readable
- machine actionable / “understandable”

- meaning of the data is well defined
- machine readable
- machine actionable / “understandable”
- machine reasoning (inference) possible
(reasoners)

https://en.wikipedia.org/wiki/FAIR_data

standardisation of communication

LARA



generic SiLA browser (unitelabs)

LARA

The screenshot shows a browser window displaying a generic SiLA browser interface for the unitelabs repository on GitLab. The URL is `localhost:3000/127.0.0.1:50060`. The page lists various SiLA services and their corresponding URLs:

- SiLA Service** `org.silastandard/core/SiLAService/v1`
- Device Info Provider** `de.unigreifswald.biochemie/device/DeviceInfoProvider/v1`
- Light Intensity Controller** `de.unigreifswald/instruments/LightIntensityController/v1`
- Logging Service** `de.unigreifswald/infrastructure/LoggingService/v0`
- Simulation Controller** `org.silastandard/core/SimulationController/v1`
- Temperature Controller** `de.unigreifswald/instruments/TemperatureController/v1`

A detailed description for the Temperature Controller service is provided:

This is a simple example of a generic Feature for controlling and retrieving the temperature. A new target temperature can be set anytime with the 'Control Temperature' Command. The temperature range has been limited to prevent major damages of a device. In case the first target temperature has not been reached, a ControlInterrupted Error should be thrown.

The interface includes several input fields and controls for the Temperature Controller:

- Current Temperature** (Input field)
- Target Temperature** (Input field)
- Current Temperature JSONLD** (Input field)
- Target Temperature JSONLD** (Input field)
- Metadata** (Input field)
- Control Temperature** (Input field, highlighted with an orange border)
- Defined Execution Error** (Input field)

Each input field has a small circular icon with an upward arrow to its right, indicating it is a dropdown or expandable field.

prefect (www.prefect.io)

LARA

The screenshot displays the Prefect website interface. At the top, there's a navigation bar with links for PRODUCT, SOLUTIONS, RESOURCES, and PRICING. On the right side of the header, there are buttons for SIGNUP / LOGIN and BOOK A DEMO. The main content area features a code editor window titled "flow.py" containing Python code for a data workflow:

```
flow.py
1 from prefect import flow, task
2
3
4 @task(log_prints=True)
5 def say_hello(name: str):
6     print(f"Hello {name}!")
7
8
9 @flow
10 def hello_universe(names: list[str]):
11     for name in names:
12         say_hello(name)
13
14
15 if __name__ == "__main__":
16     # create your first deployment to automate your flow
17     hello_universe.serve(name="your-first-deployment")
```

Below the code editor is a dark-themed dashboard with various performance metrics and charts. The dashboard includes sections for Flow Runs, Task Runs, Events, and Work Pools. The Flow Runs section shows 72 runs. The Task Runs section shows 1.2k task runs with a 6.8% increase. The Events section shows 7.4k events with an 8.2% increase. The Work Pools section shows data for "azure-push" and "azure-worker-b".

prefect 3 robotic workflow

LARA

Runs / papaya-dingo

Completed 2024/10/08 08:58:22 PM 7s 12 Task runs

Flow science-robotic-process

Dashboard

- Runs
- Flows
- Deployments
- Work Pools
- Blocks
- Variables
- Automations
- Event Feed
- Notifications
- Concurrency

Logs Task Runs Subflow Runs Artifacts Details Parameters Job Variables

Level: all Oldest to newest

Oct 8th, 2024

INFO Created task run 'move-2d2' for task 'move' 08:58:22 PM move-2d2 prefect.task_runs

INFO Moving PCR plate to pipetting_robot 08:58:22 PM move-2d2 prefect.task_runs

INFO Created task run 'get_url-633' for task 'get_url' 08:58:22 PM get_url-633 prefect.task_runs

INFO Finished in state Completed() 08:58:23 PM move-2d2 prefect.task_runs

INFO PrefectHQ/prefect repository statistics 😊 08:58:23 PM move-2d2 prefect.task_runs

INFO Stars 🌟 : 16009 08:58:23 PM move-2d2 prefect.task_runs

INFO Forks ⌂ : 1568 08:58:23 PM move-2d2 prefect.task_runs

INFO Finished in state Completed() 08:58:23 PM move-2d2 prefect.task_runs

INFO Created task run 'add_reagent-f46' for task 'add_reagent' 08:58:23 PM add_reagent-f46 prefect.task_runs

INFO Adding 10ul of Te... 08:58:23 PM add_reagent-f46 prefect.task_runs

Ready to scale? Upgrade

Join the Community

Settings

LARAsuite

LARA Main Modules - Apps

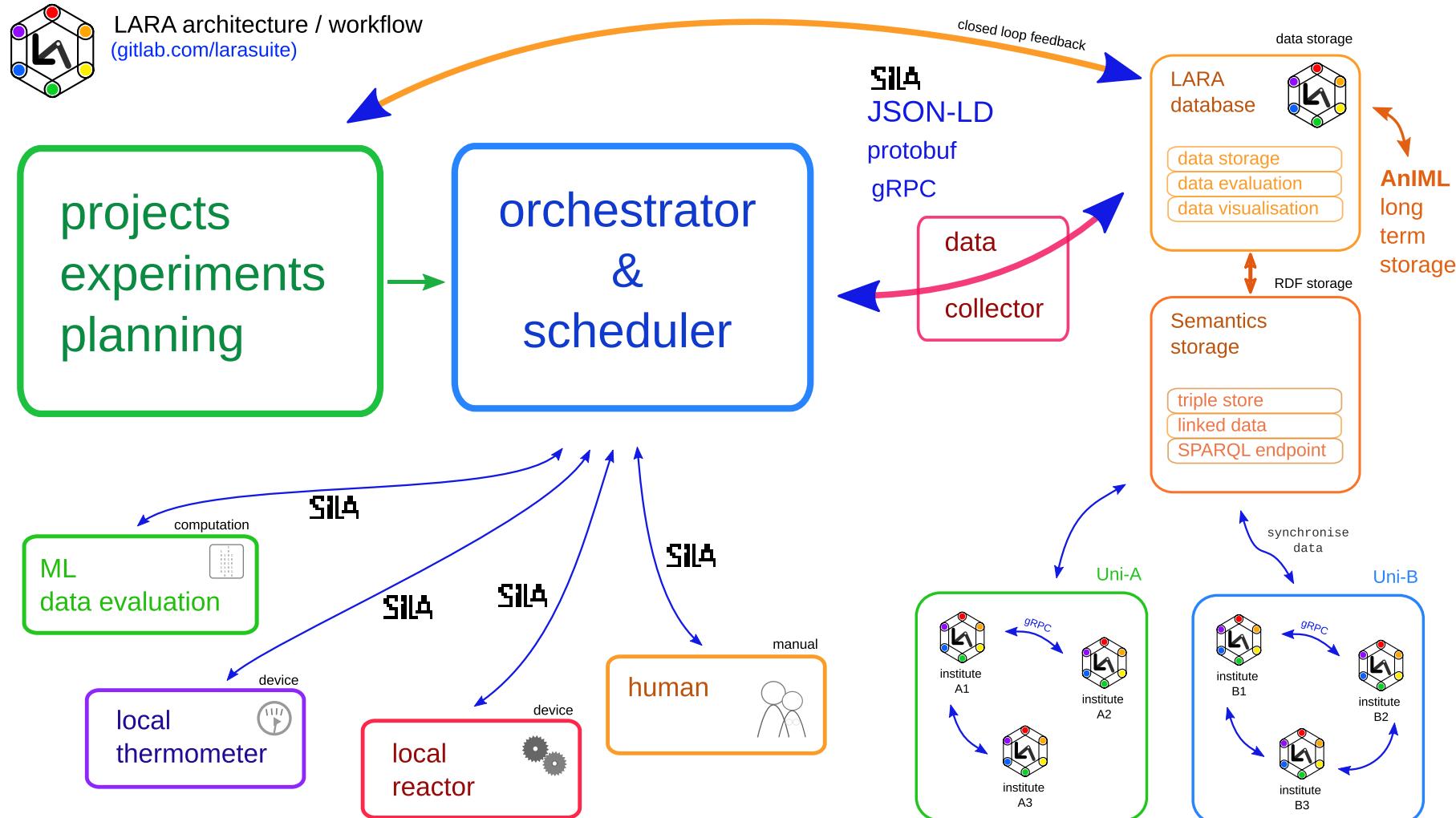
The screenshot displays the LARA Main Modules - Apps interface. On the left is a vertical sidebar with icons for various applications. At the top right is a small 'D.D.T' logo. The main area contains nine colored boxes, each representing a module:

- Projects** (Red): Management of Experiments and Projects. Includes buttons for Projects, Experiments, Procedures, Processes, Methods, Procedure Inst., Process Inst., and Method Inst.
- Processes** (Yellow): Management of Processes and Procedures. Includes buttons for Procedures, Processes, Methods, Procedure Inst., Process Inst., and Method Inst.
- People** (Green): People and Groups database. Includes a People button.
- Data** (Blue): Data database. Includes a Data button.
- Substances, Reactions, Structures** (Blue): Substances, Polymers, Mixtures, Reactions and Structures databases. Includes buttons for Substances, Polymers, Mixtures, Reactions, Substance Inst., Polym.I., Mixtures I., and React I.
- Sequences** (Blue): Sequences database. Includes a Sequences button.
- Material, Parts, Devices and Labware** (Blue): Material, Parts, Devices and Labware databases Includes buttons for Parts, Devices, Labware, Parts I., Devices I., and Labware I.
- Samples** (Blue): Samples database. Includes a Samples button.
- Organisms** (Green): Organism databases Includes buttons for Organisms and Organisms Inst.

At the bottom center, it says 2023 LARAsuite | design by benjamin lear and mark doerr

LARAsuite overview

LARA



SPARQL Query - all

LARA

SPARQL Query Editor About Tables ▾ Conductor Permalink

Default Data Set Name (Graph IRI)
urn:parql:lara:data

Extensions: cxml save to dav sponge User: SPARQL

Query Text

```
SELECT *  
WHERE {  
    ?s ?p ?o .  
}  
LIMIT 30
```

Results Format: HTML

Execute Query **Reset**

Execution timeout: 0 milliseconds

Options:

- Strict checking of void variables
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)

Copyright © 2024 [OpenLink Software](#)
[Virtuoso](#) version 07.20.3240 (a1fd8195b) on Linux (x86_64-ubuntu_focal-linux-gnu) Single Server Edition (47 GB total memory, 117 MB memory in use)

SPARQL Query - all (results)

LARA

SPARQL | HTML5 table

s	p	o
http://w3id.org/lara/demo_data_1b_v0.0.1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#NamedIndividual
http://w3id.org/lara/demo_data_1c_v0.0.1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#NamedIndividual
http://w3id.org/lara/demo_data_1b_v0.0.1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://w3id.org/lara/Data
http://w3id.org/lara/demo_data_1c_v0.0.1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://w3id.org/lara/Data
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/datetime_last_modified	2024-11-06T08:24:08.575Z
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/datetime_last_modified	2024-11-06T08:24:17.634Z
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/version	"v0.0.1"^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/version	"v0.0.1"^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/datetime_created	2024-11-06T09:24:07.308Z
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/datetime_created	2024-11-06T09:24:15.938Z
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/name_full	" https://de.unigreifswald/biochem/akb/demo_data_1b_v0.0.1 "^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/name_full	" https://de.unigreifswald/biochem/akb/demo_data_1c_v0.0.1 "^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/data_id	"48e9088d-9476-4f0e-90f8-c7923939266e"^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/data_id	"627d3639-f41a-428b-87a0-a208fa8150a2"^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1b_v0.0.1	http://w3id.org/lara/title	"Demo Data 1b"^^< http://www.w3.org/2001/XMLSchema#string >
http://w3id.org/lara/demo_data_1c_v0.0.1	http://w3id.org/lara/title	"Demo Data 1c"^^< http://www.w3.org/2001/XMLSchema#string >

SPARQL Query - datetime

LARA

SPARQL Query Editor About Tables ▾ Conductor Permalink

Default Data Set Name (Graph IRI)
urn:parql:lara:data

Extensions: cxml save to dav sponge User: SPARQL

Query Text

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX emmo: <http://emmo.info/emmo-inferred#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX lara: <http://w3id.org/lara/>

SELECT ?data_title ?date ?data_id
WHERE
{
    ?data lara:datetime_created ?date .
    ?data lara:data_id ?data_id .
    ?data lara:title ?data_title .
    FILTER (?date > "2024-11-01T00:00:00Z"^^xsd:dateTime)
}
LIMIT 30
```

Results Format: HTML

Execute Query Reset

Execution timeout: 0 milliseconds

Options:

- Strict checking of void variables
- Log debug info at the end of output (has no effect on some queries and output formats)
- Generate SPARQL compilation report (instead of executing the query)

Copyright © 2024 OpenLink Software
[Virtuoso](#) version 07.20.3240 (a1fd8195b) on Linux (x86_64-ubuntu_focal-linux-gnu) Single Server Edition (47 GB total memory, 117 MB memory in use)

SPARQL Query - datetim (result)

LARA

SPARQL | HTML5 table

data_title	date	data_id
"Demo Data 1b"^^< http://www.w3.org/2001/XMLSchema#string >	2024-11-06T09:24:07.308Z	"48e9088d-9476-4f0e-90f8-c7923939266e"^^< http://www.w3.org/2001/XMLSchema#string >
"Demo Data 1c"^^< http://www.w3.org/2001/XMLSchema#string >	2024-11-06T09:24:15.938Z	"627d3639-f41a-428b-87a0-a208fa8150a2"^^< http://www.w3.org/2001/XMLSchema#string >

labDataReader

- generic reader of proprietary data (e.g. HPLC, plate readers)
- primary output : pandas data frame and *JSON-LD* (metadata)

<https://gitlab.com/opensourcelab/scientificdata/labDataReader>

SciDat

- packing tabular data / data frames into *parquet* files, including *JSON-LD* metadata

<https://gitlab.com/opensourcelab/scientificdata/scidat>

LARA is a tool ...

LARA is a tool ...

- for planning, executing and evaluating experiments as complex as BioCatalytic / Enzymatic processes

LARA is a tool ...

- for planning, executing and evaluating experiments as complex as BioCatalytic / Enzymatic processes
- leverages automation in many steps

LARA is a tool ...

- for planning, executing and evaluating experiments as complex as BioCatalytic / Enzymatic processes
- leverages automation in many steps
- generates a knowledge graph on the fly

LARA is a tool ...

- for planning, executing and evaluating experiments as complex as BioCatalytic / Enzymatic processes
- leverages automation in many steps
- generates a knowledge graph on the fly
- synchronises with external data sources (like Dataverse)

acknowledgements

- Stefan Maak
- Uwe Bornscheuer's group (Univ. Greifswald)

project partners

- Stefan Born (TU Berlin)
- Peter Neubauer's group (TU Berlin)
- Johannes Kabisch's group and associates (Uni Trondheim)
- Egon Heuson (Uni Lille)
- Lukas Bromig and Julian Willand (unitelabs)

KIWI-biolab team, NFDI4Cat teams, SiLA team & AnIML team

This work was supported by the German Federal Ministry of Education and Research through the Program "International Future Labs for Artificial Intelligence" (Grant number 01DD20002A) as well as the NFDI4Cat grant

We are grateful to the Deutsche Forschungsgemeinschaft (DFG, INST 292/118-1 FUGG) and the federal state Mecklenburg-Vorpommern for financing the robotic platform.

