2024 Student Internship Topics for VLIZ OpSci

About VLIZ OpSci

The Flanders Marine Institute (VLIZ) Open Science Team takes a technical approach to facilitating practical Open Science Data sharing. Located in Oostende Belgium, open access and data sharing principles have been an integral part of the VLIZ DNA since its inception.

The challenge is to translate that vision into scalable solutions using solid and broadly-shared collaborative methods and standards. We take our cues from the many topical “open” movements (open source, open data, open access); the well-known Findability, Accessibility, Interoperability, and Reuse of digital assets (FAIR) Principles; and the European Open Science Cloud (EOSC) Association.
Day-to-day interactions with local scientists, sharing end-user expertise, international collaborations in the context of EU Horizon 2020 projects, and interactions with Research Infrastructures such as European Marine Biological Resource Centre (EMBRC) ERIC and LifeWatch ERIC, help us develop practical solutions.

The outset of our approach is data-agnostic, which allows us to collaborate, share, and borrow across domains and disciplines. In practice, VLIZ being in the Marine domain, a lot of the data we handle gravitates around biodiversity, genomics, climate, ecology, habitat changes and invasive species.

Our technical toolchain blends typical data-science elements with industrial software engineering techniques, and takes in many influences from core (semantic) web architecture, linked-open-data, and knowledge graph technologies.

About the team

As Part of the VLIZ Data Centre, the OpSci team are a mixed bunch of nationalities, experiences, ages. We work as a team, but with individual responsibilities for our projects and products. We have weekly collaborative meetings where we discuss and question in order to solve issues and fine-tune the direction of our work. Every member of the team is as valid as the next :-). We expect our interns to enjoy the same: to know to do the work assigned to you, but to also know that we are open to questions about what it is you are doing and how it fits into the bigger picture. Helping each other is a natural part of that.

Advisory team

We believe the described internships below are challenging and are likely to contain a lot of concepts and technical references that "are not known yet". Acknowledging the daunting

character of that, we want to assure that we consider the goals achievable to student profiles(*), and that we are offering the environment to guide you through unknown territory. Next to the expertise provided by our own team we have also engaged an international group of experts. You will present them regularly with the status of your progress, giving them a chance to nudge your work further into the good direction.

(*) We assume future Master level students in computer science oriented programs that have passed their bachelor level.

Relevant Resources

From generic to detailed

- General Open Science Resources
  - Open Science at United Nations
  - EU Open Science Policy
  - The EOSC (European Open Science Cloud)-Association
  - The FAIR Guiding Principles for scientific data management and stewardship
- VLIZ Open Science Team
  - website: https://open-science.vliz.be/
  - github space: https://github.com/vliz-be-opsci
  - data management for the EMO-BON biodiversity project: https://data.emobon.embrc.eu/
  - recent presentation about semantics in open science
  - older general intro into RDF
- Semantic web information
  - let me google that for you: wikipedia: Semantic Web - All About The Semantic Web - Getting Started with Semantic Technologies - Introduction to the Semantic Web - Ruben Verborgh: Ugent course : Web Fundamentals : Semantic Web & Linked Data - …
- UGent KNoWS team
  - team website: https://knows.idlab.ugent.be/
  - Ruben Verborgh - Pieter Colpaert
  - linked data fragments: https://linkeddatafragments.org/
  - comunica: https://comunica.dev/
  - tree-cg: https://w3id.org/tree/specification
  - LDES: https://w3id.org/ldes/specification

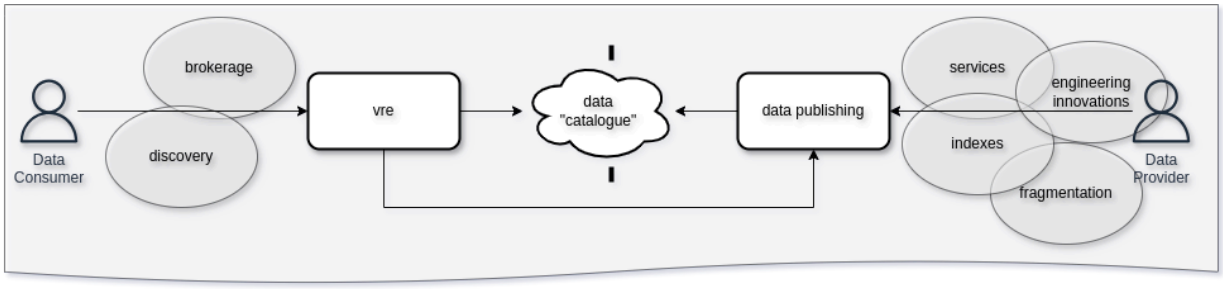
Student Internship Topics 2024

- 1 | UDAL for FAIR-EASE
- 2 | Term Label Translation Management Support
- 3 | Connecting the Science Knowledge Graph

1 | UDAL for FAIR-EASE

Intro

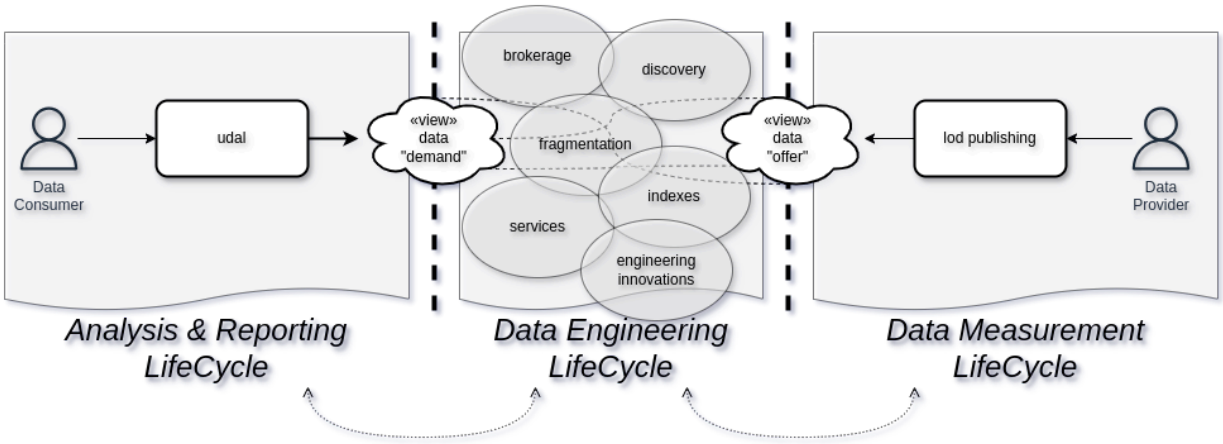
UDAL stands for "Uniform Data Access Layer": it is a software engineering attempt to isolate data-reporting and visualisation front-ends from the various ways the data they rely on are provided. The aim is to allow continued working with the data-providers, while accommodating inevitable changes in the data landscape.
The source of such changes is manyfold: new data capturing devices and platforms; better quality assurance and data cleaning intermediaries; smart recombination strategies of increasingly linkable data sources; innovative technical delivery protocols….



One LifeCycle, impacted by all influences:
Analysis, reporting, engineering, measurements, ...

The challenge and main goal for UDAL is to introduce an abstraction layer that allows for alternatives and innovations to be instantly interchangeable and pluggable into existing reporting and visualisation platforms (typically known as VREs: Virtual Research Environments) - thus effectively reducing the cost of engaging and evaluating these alternatives and innovations.

On a cultural / social level the introduction of UDAL is challenging the research community to look at research data *as a commodity*. Instead of always controlling the complete chain themselves, they should rely and reuse the efforts from others and obtain the data they need for analysis and reporting from an open and flexible data marketplace. This implies a change of mind on both sides: on the consumption it requires trusting the data provided by others, and on the providing side to make sure data offered is self-explaining and linked to a detailed provenance trail that can vouch for its trustworthiness. This will obviously require effort. Still, the motivation to take these steps is clear: the current tight data flows from data provider to policy report do not scale well, resulting in a cycle of continuous redoing all parts to accommodate for a change at either end. A well-designed abstraction layer between the sides would extend the applicability of all components involved and vastly improve the smoothness of the data flows. .

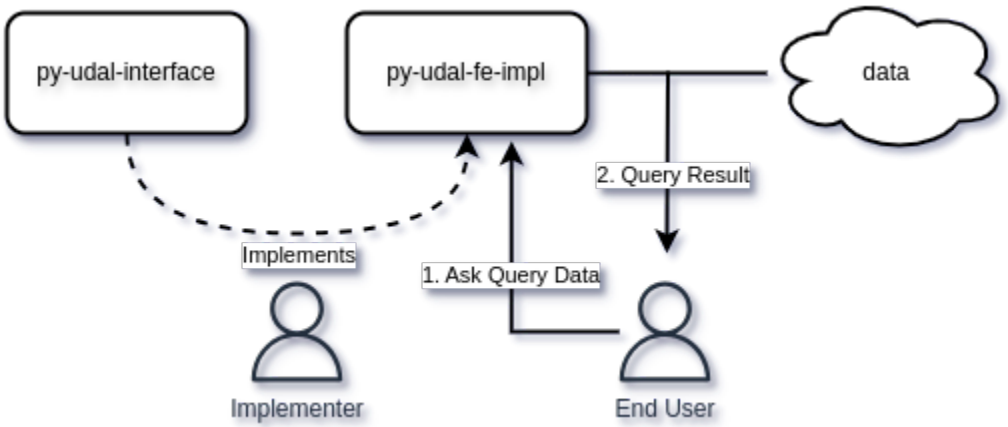


UDAL achieves the above ambition through a number of concrete building blocks:

(1) It introduces the idea of making a catalogue of data-demand. These are similar to existing catalogues of data-offer, but rather describe the "virtual" datasets (content, shape, quality, …)

(2) An abstract programming interface (currently in python only) that captures the basic interaction of any client side data-consumption in its most essential parts: possibly connecting and authenticating to a brokerage service, identifying the kind of data you need, sliding in some filtering parameters, and obtaining the resulting dataframe in the format it wants to deal with.

(3) Various plugable implementations of this interface that each follow different alternative strategies to perform the actual data retrieval and format conversions.



UDAL as a conceptual strategy was introduced in the FAIREASE.eu project in 2024. Within that project the goal is to challenge, test, and fine-tune the UDAL conceptual approach via a Proof-Of-Concept implementation in Python in a Jupyter Notebook context.

Material

- UDAL in FAIR-EASE slide-deck → 2024-04-12-UDAL 1.pptx
- @github py-udal-interface | py-udal-fe-impl | dataset-demand-register | IDDAS

- POKaPOK use case development: pokapok-projects / PKP8-OGI-BGC / qcv_udal_testings · GitLab

Tasks

During the internship, the ambition is to do practical work (create/modify code and specifications, write documentation, develop presentations) to further the development of UDAL, with a focus on concrete use cases within the FAIR-EASE project. Getting some acquaintance with that project will be a natural part.

Given the early stages of this approach, the opportunities are quite open and varied:

- Prototype and evaluate various UDAL implementation strategies (python code):
  (1) Rebundling access to data from existing tight data-flow pipelines to the UDAL way.
  This first-order, “hacky” way of "pretend UDAL" will give valuable information on the validity, completeness, and useability of any eventual UDAL interface design. We can collaborate on this with our data-providing partners from the FAIR-EASE project.
  (2) Handling various popular existing data-services.
  A diverse set of existing data-fragmentation services (ERDDAP, OpenDAP, openEO, OGC WFS layers, Examind, …) are in use – each of these will need a one-on-one UDAL implementation to show how actual mapping of existing services could work. This will involve selecting one of the services (at a time), getting acquainted with them, and implementing a prototype that can showcase the workings of UDAL.
  (3) Using a central brokerage service.
  While the UDAL is a client-side library, it can negotiate work with already existing data-discovery services. Within the FAIR-EASE project there is ongoing work on such a system, called the “IDDAS”, that operates on a catalogue that gathers all the provided datasets within the project. The plan is to provide a prototype UDAL implementation that collaborates with that.
- Prototype and evaluate the "data demand registry" (python code + RDF modelling).
  The tasks to achieve this are spread over the team and involve:
  - creating test entries to evaluate and influence the design,
  - tuning the structure and used vocabularies to describe these "virtual datasets"
  - investigate a UDAL implementation that also factors in the descriptions in this registry as part of its brokerage magic.

The process will be to first investigate into the current state of the specification and existing implementations of UDAL to get a feel and understanding of its design and the most urgent fixes or completions that are needed. Which aspects to then tackle will be the result of a dialogue with the implementation team at VLIZ and the broader FAIR-EASE community.

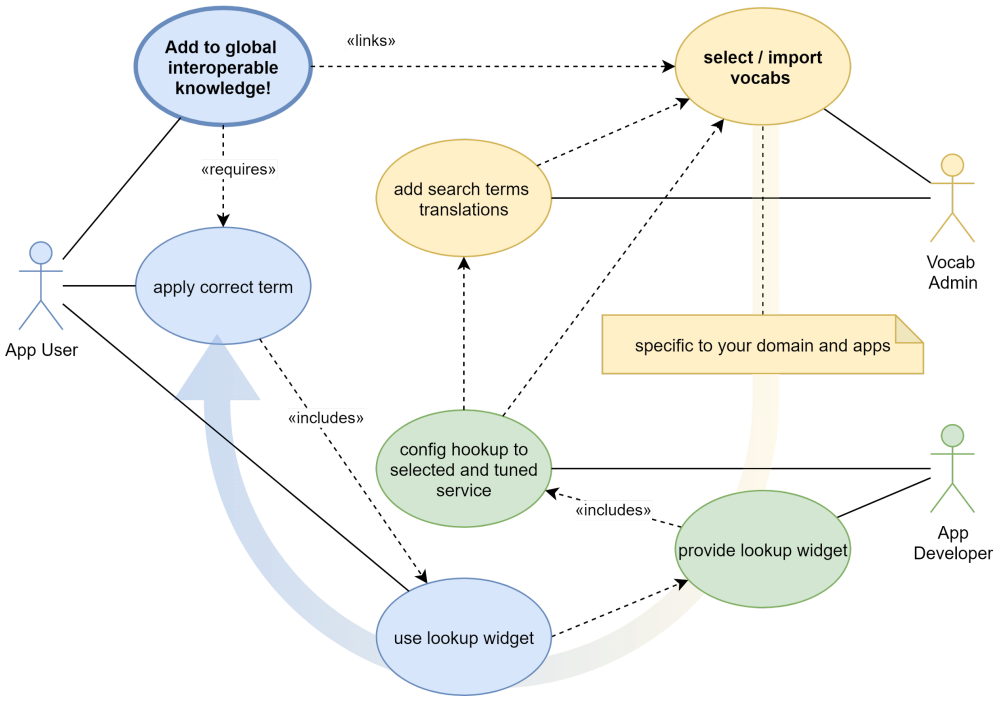
2 | Term Label Translation Management Support

Intro

Achieving, maintaining, enhancing the semantic interoperability of data in any domain is challenging and plays out on different levels. A big part involves the development of vocabularies and lists of reference terms. The practical benefits of these efforts depend on the subsequent correct and ubiquitous usage of these vocabularies in the various data management systems that are run by different communities and organisations. When done correctly, this makes data understandable to all, and connectable to and between all the data systems, allowing for all sorts of (serendipitous) correlations and links to be discovered.

To enable various data management systems to reuse vocabulary term lists across the board, we developed a so-called "vocab-terms-lookup-service" that delivers both an API and directly plugable widgets that provide consistent term selection in the UI of these systems. End-users simply select the correct option from a list that gets tuned-as-they-type, filtering on matching words in the label, identifier, and description of the term they know to look for.

Additionally, these term lists are defined in RDF which has built-in support for alternative strings in various human languages, and thus the lookup for matching terms can be supported in any of those specific languages. Provided, of course, the translated variants of those labels are being made available.

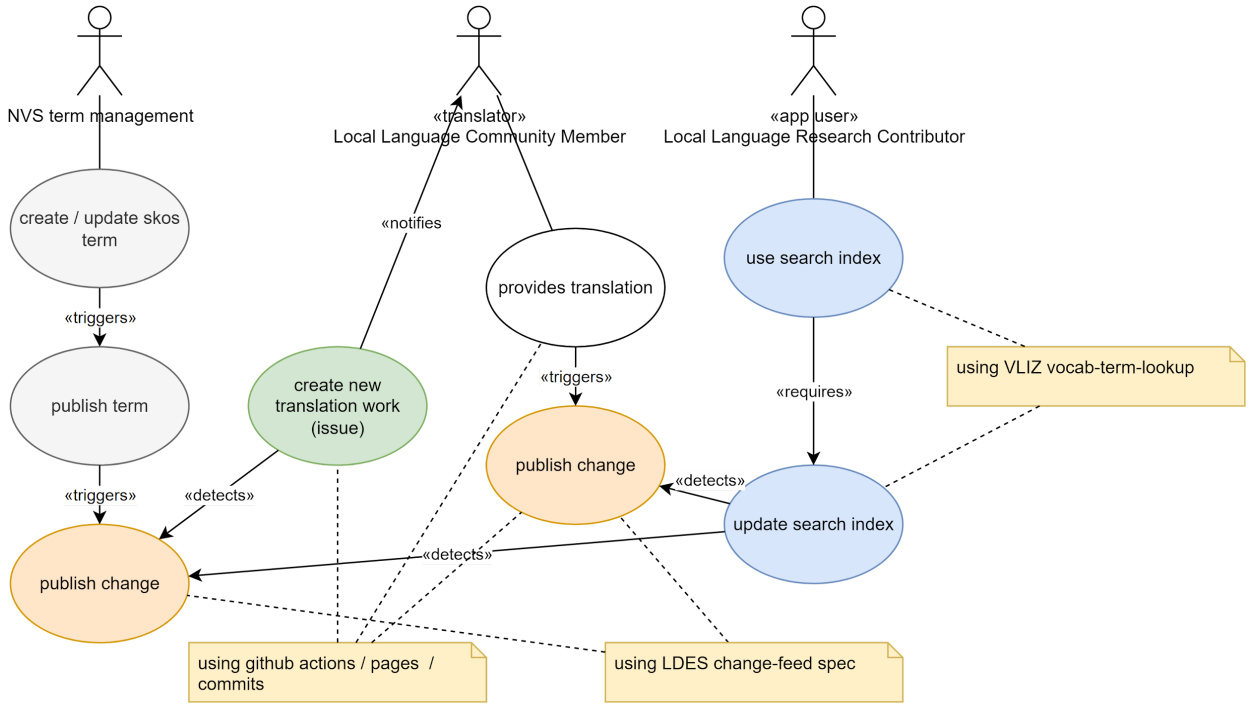


This is where LDES and some translation work are popping up.

LDES (Linked Data Event Streams) is a standard that allows for the changes to any dataset to be published as Linked Open Data. We use this stream of update events from our managed vocabularies to feed the vocab-term-lookup-server. This ensures it is kept up-to-date. Additionally we are now developing a git-based translation workflow that supports a certain language community through

- automatically reading the incoming stream of updates from the original (English) source;
- flagging updated terms as "to be checked" , converting them into simple yml files for translation into another language;
- grouping the work in batches for the community to fix;
- upon approval (merge and commit) the changes get published using LDES again.

Basically, we will help communities who want to translate particular vocabularies into their own language, by providing the technical assistance for their linguistic inputs. They provide the translations; we provide the method for the ingestion of those translations.



The described term-translation workflow is developed within the context of Phase V of the EMODnet Biology project.

Material

- Slides
  - intro to the vocab-term-lookup
  - vocab alignment task force in EMODnet Bio V
- github repo current development

- diagram current implementation flow
- Vocabularies & LDES
    - NVS by bodc
        - collections: https://vocab.nerc.ac.uk/collection/
        - feed - per collection: https://vocab.nerc.ac.uk/ldes/{collection_code}/
    - MarineRegions.org by VLIZ
        - web: https://marineregions.org/
        - feed: https://marineregions.org/feed/

Tasks

During the internship, the ambition is to do practical work (create/modify code and specifications, write documentation, develop presentations) to further the development of this github-based translation-management flow in the context of EMODnet Bio Phase V.

Given the early stages of this work, the field and opportunities are quite open and varied:

- The core setup of this approach is to leverage the version management features of github, so the intern will need to be able to work at that basic interface level. It is unlikely that the people providing the translations will also be fluent in handling technical details like git operations, github services or even yml files: we want to lower the threshold to allow them to focus on their core contributions (the translations) by designing and prototyping a user interface (browser based and/or packaged as a vs-code plugin)
- Additionally we want to investigate the possibility of adding in translations provided by a “robotic-community-member”, i.e. by integrating with automated translation services to handle some parts of the actual translation work.
- Our current work on the github action that supports the translation flow needs extra development and testing. While this core job is mainly covered by a colleague in the OpSci team, interaction and collaboration with that work is as logical as it is inevitable.

The process will be to first investigate into the current state of the translation workflow as well as evaluating the optional platforms for developing and deploying the targeted UI. The main task is to design and prototype this UI. The VMDC staff includes people speaking an interesting variety of native languages that should allow you to organise a local user-group for testing and bouncing off design ideas.

Practical

Programming languages to be used:

- Python
- Javascript/Typescript

3 | Connecting the Science Knowledge Graph

Intro

The Open Science movement is big in Europe and the rest of the world. Its aim is to make all science outputs available for anyone to find and use: not just scientists in the particular domain, but any scientist, but also any teacher, student, journalist, policy-maker, company, … anyone and everyone, including robots or AI systems that are there to help them out. Much work has been going on in the past decade or so, building a consensus on what this means and how to do it. The current focus is on solving a number of technical problems that are limiting the practicality of Open Science: the fact that it is not enough to make science open, it has to actually be findable, understandable, and usable. One focus here is on interoperability: that data can be opened and read by anyone; and that data can be found by anyone by the use of accompanying descriptive metadata that is understandable to everyone.

Semantic web technology is an important instrument to overcome these interoperability challenges of Open Science. Imagine that all science is described, annotated, formulated so that it can all be “knit together” in a "Science Knowledge Graph", just like how web-pages are connected. Inside the marine domain, this has been happening on various levels ranging from individual datasets, through to individual institute programmes within many EU / EOSC initiatives and projects, all the way up to the UN / Unesco-level work on the Ocean Info Hub.

The promised end result of a science knowledge graph will be to produce the experience of querying a single database, via the web, that contains all the science that is. It is safe to say we are not there yet, but the pathway to this goal is well-established and being worked on from all sorts of directions by many organisations. At VLIZ, we have been working on a set of python services and libraries that together allow a single data scientist to easily build up a local aggregate of what can be found "out there", which they can then use as a basis for data analysis. We apply this technique already in a number of concrete projects and dashboards under development.

While we named this platform K-GAP (Knowledge Graph Analysis Platform), we have found that it is very useful in detecting actual knowledge gaps, giving the name a double meaning.

Material

- Slides
  - 20240418 FRDN VLIZ LOD & Open Science
  - 20240418 K-GAP Demo & Exercise
- Repositories
  - https://github.com/vliz-be-opsci/k-gap
  - https://github.com/vliz-be-opsci/sema
- Linked data providers
  - OIH - https://oceaninfohub.org/

    - NVS - https://vocab.nerc.ac.uk/collection/
    - Maregraph - https://www.maregraph.eu/
    - openaire graph - https://graph.openaire.eu/
    - https://orcid.org/
- Identified gap:
    - https://ror.org/
- Specs
    - fair-signposting - https://signposting.org/FAIR/

Tasks

During the internship the ambition is to do practical work (create/modify code and specifications, write documentation, develop presentations) to further the development of our own K-GAP platform, specific uses of it, and to pragmatically bridge the gaps in the platform.

Given the early stages of this approach, the field and opportunities are quite open an varied:

- Institute identifiers are a useful way to link people, projects, and institutes in the knowledge graph. Unfortunately, The fact that ROR.org, one of the registries of institutes, is not yet published as linked open data is an issue for us, as K-GAP ingests LOD formats. A clever usage of one of our semantic-uplifting tools on top of the provided ROR.org APIs will allow us to produce a publicly-available LOD publication of the ROR registry. Additional application of w3id.org and some .htaccess files could make that conform to classic LOD expectations about de-referenceability and content-negotiation.
- Active participation on some of our various K-GAP based projects will give a further practical experience of how to use this tool, as well as potentially disclose some other gaps to close (i.e. we need user testing!).
- Finally, since K-GAP is still under development, there are a number of extra new features in the books that are waiting for a willing contributor
    - an LDES client
    - an aggregator of various LOD resources
    - a set of export-import routines to
    - support for alternative triple stores
    - …

The process will be to first investigate the problem space and gain some experience with our LOD supporting python libraries. The main task is to create the ror.org/LOD publication as described above. But depending on how well that goes, we see a vast opportunity for related and very useful follow up tasks.

General remarks and practicalities

Work Schedule

- 13 weeks July-September (June 24th earliest)
- Bi-weekly short stand-ups to mention where you are and what your next moves are.
- Every 3 weeks presentation & online meeting with advisory team; dates & timing need to be agreed, set and communicated (see below)

Practical

- VLIZ does not offer any remuneration for these internships.
- As a team we struggle with this too as we understand these 'costs' set a filter that might be keeping out potential and deserving talent. We are happy to help in administrative paths that could get you some supporting grant to cover some costs, but cannot recommend at this stage any specific ones.
- Students need to look for housing themselves
    - best in region Ostend, Bruges, or Ghent
      (local spelling: Oostende, Brugge, Gent)
      expect to pay 650-900 €/month
    - consider joining forces and find some co-housing solutions
    - no preference on your location choice, just some info:
        - Train
        - Ostend (which merges into nearby Bredene for more options):
          closest, bike-distance options, tourism summer vibe, beach
        - Bruges: at 25k, 3 trains per hour, they take 15', historic cultural centre, nicely quiet, might have student housing options (locally named 'kot') available over summer
        - Ghent: at 60k, 2 trains per hour, they take 40', large, vibrant, diverse, lively, similar (but more) student housing opportunities here
- Language requirements
    - working language at VLIZ in general is English
      (the bulk of people are Flemish, which means native Dutch speaking; most of the staff has some understanding of French)
- How to reach VLIZ
- Contact & Questions via marc.portier@vliz.be

Admin

Please provide us with the following:

- formal notification of your acceptance by email
  (asap, and at least one month prior to arrival)
    - this to assure office space and laptop

    - please also include your ok to share your CV and contact details with the direct colleague you will collaborate with
- some practical information for HR
  (asap, but at a least 3 working days prior to arrival)
    - actual starting day
    - your name, date of birth, preferred contact points (email and phone)
    - the address you are staying at

Advisory Team

Regular online presentations of progress to a panel of experts
https://meet.jit.si/2024-vliz-intern-panel

Candidates / Members

Name (institute)	Email	expertise / domain / …	internship topic interest		
			#1	#2	#3
Damian Smyth (Marine Institute)	damian.smyth@marine.ie	Data Services, Platform Specialist	☑	☐	☐
Erwan Bodere (ifremer)	erwan.bodere@ifremer.fr	Information Systems Architect (holidays: 15/07 →15/08)	☑	☑	☑
Jérôme Detoc (ifremer)	jerome.detoc@ifremer.fr		☑	☐	☐
Paul Weerheim (maris)	paul@maris.nl		☑	☐	☐
Peter Thijsse (maris)	peter@maris.nl		☑	☐	☐
Alexandra Kokkinaki (BODC)	alexk@noc.ac.uk		☐	☐	☐
Gwenaelle Moncoiffe (BODC)	gmon@noc.ac.uk		☐	☑	☑
Mohamed Adjou (ISEN)	mohamed.adjou@isen-ouest.yncrea.fr		☐	☐	☐
Jorge Mendes (VLIZ)	jorge.mendes@vliz.be		☑	☐	☐
Cedric Decruw (VLIZ)	cedric.decruw@vliz.be		☐	☑	☐
Laurian Van Maldeghem (VLIZ)	laurian.van.maldeghem@vliz.be		☐	☐	☑
Marc Portier (VLIZ)	marc.portier@vliz.be		☑	☑	☑

Schedule of calls

week num	date	timing			remarks
		#1	#2	#3	
27	~ 2024-07-04				understanding, problem scope & initial plan
30	~ 2024-07-25	07-25 11h-12h	07-25 10h-11h		1st iteration status
33	~ 2024-08-15	08-14 11h-12h	08-15 10h-11h		2nd iteration status
36	~ 2024-09-05	09-05 11h-12	09-05 10h-11h		3rd iteration status & final commitment
39	~ 2024-09-26	09-26 11h-12h			closing