Co-designed Citizen Observatories Services for the EOS-Cloud

H2020 programme: Research and Innovation action

Deliverable 4.6
Data Use Notification Services

October 21th, Version 1

Project funded by the European Commission within the Horizon 2020 Programme (2014-2020)

Grant Agreement No.	863463

Type		
R	Document, report excluding the periodic and final reports	
DEM	Demonstrator, pilot, prototype, plan designs	X
DEC	Websites, patents filing, press & media actions, videos, photos, etc.	
SOF	Software, technical diagram, etc.	
OTHER	Flyers, etc.	

Dissemination level		
PU	Public, fully open.	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified	

Revision history

R#	Date	Description/Reason of change	Deliverable contributors
R0.1	20220715	First skeleton	Marisol Parrilla López (BINEO), Santiago Martínez de la Riva (BINEO), Robin Giles Ribera (BINEO)
R0.2	20220914	Draft content	Marisol Parrilla López (BINEO)
R0.3	20221014	Draft shared with Consortium partners - contributions & review	Robin Giles Ribera (BINEO), Santiago Martínez de la Riva (BINEO), Marisol Parrilla López (BINEO)
R1.1	20221018	Internal review	Ana Álvarez (ICM-CSIC)
R1.2	20221023	Updates	Robin Giles Ribera (BINEO), Santiago Martínez de la Riva (BINEO), Marisol Parrilla López (BINEO)
R2.1	20221026	Internal review	Sasha Woods (EarthWatch)
R2.2	20221028	Updates	Santiago Martínez de la Riva

			(BINEO), Marisol Parrilla López (BINEO)

Authors

Marisol Parrilla López, Robin Giles Ribera, Santiago Martínez de la Riva

Citation

This document is public and can be cited as: Cos4Cloud consortium D4.2 (2021) Parrilla, M. , Giles Ribera, R., Martínez de la Riva, S. Expert Portal for biodiversity data validation (D4.2). Project funded by the European Commission within the Horizon 2020 Programme GA #863463. Available in: http://www.cos4cloud-eosc.eu

Licence and attribution

Executive Summary

Throughout Cos4Cloud, Bineo Consulting developed a number of integrative services. We focus on this deliverable on Data Usage Notification Service (also known as DUNS). For a proper understanding of this service, one must understand the challenges Citizen Observatories (COs) and the citizens who collect data face.

In the first instance, once an observer makes an observation, they cannot track how that observation is used. There is no way for observers to know where or what their photographs of animals or plants are being used for. The observers may know from a general standpoint that they are helping to conserve species, but they do not know what scientists and experts may be doing with that information. Additionally, they have no idea how many times their observations have been cited in sources outside of the CO, or on which pages or in which countries your observations are cited. Information like this can be relevant to those collecting observations and can make them visible in their communities.

COs also face a number of challenges. Retaining users and showing them the value of their work is becoming increasingly difficult. In many cases, they cannot demonstrate how important their contributions are. With this information, COss could even develop gamification systems to reward citizens who provide observations that are most consulted or most referenced.

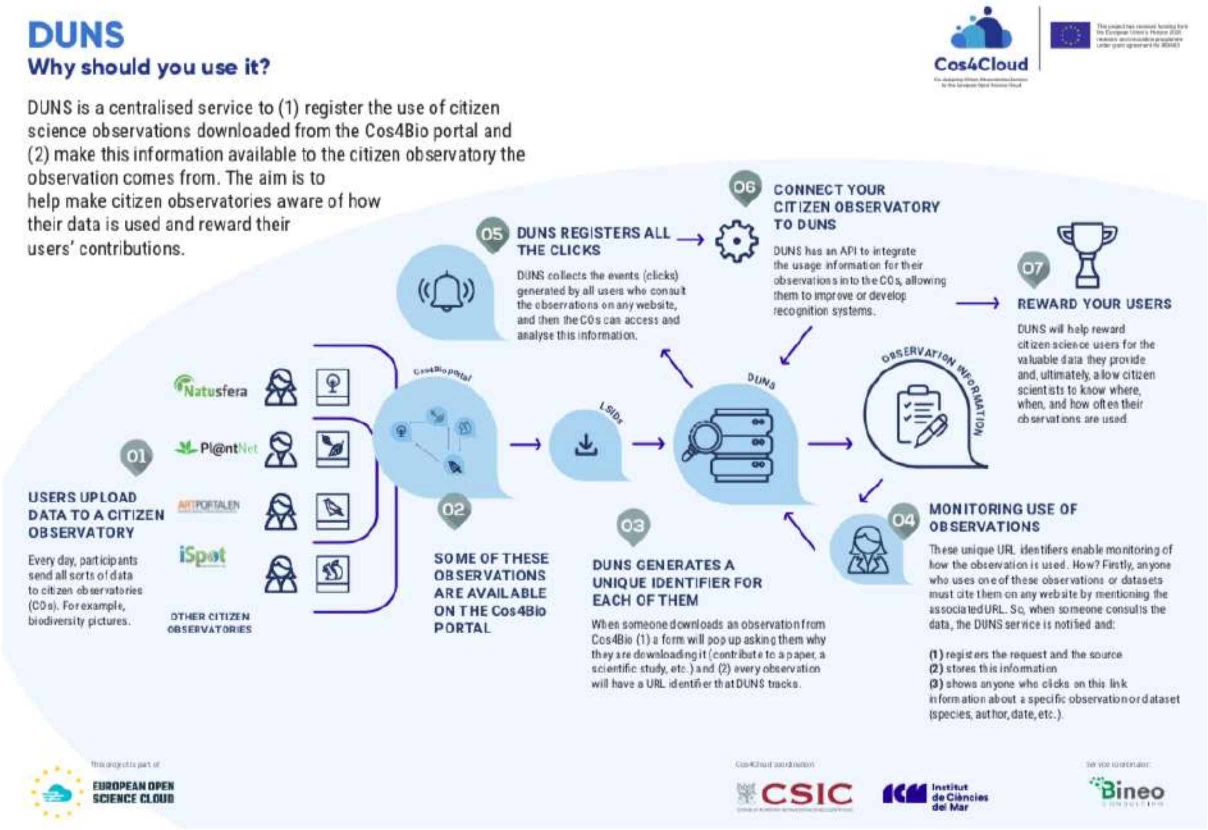
The main objective of this deliverable, aside from explaining the DUNS conceptualization and development process, is to demonstrate how this service can be integrated and useful in other Cos4Cloud services, such as Cos4Bio **(a service that integrates biodiversity observations from multiple COs in one place)**. We also want to show how DUNS can provide an additional data flow to a citizen science portal such as Natusfera, enriching the portal's information with the goal of improving the community and retaining the people who are part of it and who contribute information with their observations.

It is important to note that DUNS is a service that will be available on the EOSC Portal at https://marketplace.eosc-portal.eu/services/. The objective is that DUNS could be integrated into all Citizen Observatories that want to be part of Cos4Bio, and in this way they can receive notifications of use of each and every observation that experts download and use on other platforms through Cos4Bio.

In developing this new service, we have considered the following Minimum Viable Ecosystem (MVE) within the implementation of the prototype, which has been validated through technical meetings:

- Service that is interoperable
- The use of standards.
- DUNS integration in Cos4Bio.
- The following aspects have been recorded:
    - The global use of observations
        - By country.
        - By platform.
        - By language.
        - Use of observations per page where they have been used.
        - By browser.
    - Likewise, the user/portal has kept a record of usage.
        - By country.
        - By platform.
        - By language.
        - Use of observations per page where they have been used.
        - By browser.

To better understand DUNS and the value it provides, we must first understand how it works, so we will focus on the DUNS infographic (Figure 1) that shows how citizens publish observations to Citizen Observatory and experts download them from Cos4Bio. At that moment, both the citizen observatory and the user who published the observation are notified.



DUNS

Why should you use it?

DUNS is a centralised service to (1) register the use of citizen science observations downloaded from the Cos4Bio portal and (2) make this information available to the citizen observatory the observation comes from. The aim is to help make citizen observatories aware of how their data is used and reward their users' contributions.

01 USERS UPLOAD DATA TO A CITIZEN OBSERVATORY

Every day, participants send all sorts of data to citizen observatories (COs). For example, biodiversity pictures.

02 SOME OF THESE OBSERVATIONS ARE AVAILABLE ON THE Cos4Bio PORTAL

03 DUNS GENERATES A UNIQUE IDENTIFIER FOR EACH OF THEM

When someone downloads an observation from Cos4Bio (1) a form will pop up asking them why they are downloading it (contribute to a paper, a scientific study, etc.) and (2) every observation will have a URL identifier that DUNS tracks.

04 MONITORING USE OF OBSERVATIONS

These unique URL identifiers enable monitoring of how the observation is used. How? Firstly, anyone who uses one of these observations or datasets must cite them on any website by mentioning the associated URL. So, when someone consults the data, the DUNS service is notified and:

(1) registers the request and the source
(2) stores this information
(3) shows anyone who clicks on this link information about a specific observation or dataset (species, author, date, etc.).

05 DUNS REGISTERS ALL THE CLICKS

DUNS collects the events (clicks) generated by all users who consult the observations on any website, and then the COs can access and analyse this information.

06 CONNECT YOUR CITIZEN OBSERVATORY TO DUNS

DUNS has an API to integrate the usage information for their observations into the COs, allowing them to improve or develop recognition systems.

07 REWARD YOUR USERS

DUNS will help reward citizen science users for the valuable data they provide and, ultimately, allow citizen scientists to know where, when, and how often their observations are used.

Figure 1. DUNS infographic

As shown in Figure 1e, the process begins with a user uploading an observation to a citizen observatory. As a result of the interoperability layer developed for Darwin Core, these observations are added to the Cos4Bio portal. It is possible for the experts to consult the observations, to filter the information, to download it or to add information identifying the species. This point is very important, since most DOI-style tracking services track/use data sets rather than observations, however, with this system, an expert will be able to refer to an observation using its URIs, which will be unique within the observation. Every time that link is consulted, the DUNS service receives a usage report. It is possible for each citizen observatory integrated in Cos4Bio to consult this usage information on both a global level, i.e. the use of all observations of that Observatory, and at a specific level of its users. Therefore, citizen observatories integrated into Cos4Bio will be able to take advantage of this data to provide their community with more information about the use of observations, as well as create gamification systems that they consider appropriate, which will increase the retention percentages of their users.

Index

List of tables and figures

1. Methodological approaches

Before describing the methodological approaches, we should keep in mind that DUNS is a purely integration and API development service, so this technical part has been addressed at the technical meetings and in collaboration with the various Citizen Observatories - including Natusfera and Pl@ntNet - integrated into Cos4Bio.

We also want to emphasize that the integration of the service with a Citizen Observatory was not explicitly addressed in the Grant Agreement, but it seemed relevant to demonstrate how useful it would be for the observatories to display this initial information, expandable in the future, in order to attract more observatories once this prototype phase has ended.

Similarly to the rest of the services, we have followed Agile Methodology guidelines for the development process: generating a Decision Backlog table to analyze requirements before starting to implement the service, prioritising these requirements and finally including them in the planning process.

As a result of the Analysis phase, we were able to generate User Stories, which we associated with each development sprint, allowing us to develop each component or service in an Agile manner, and integrate them successfully with the rest.

DUNS is one of the most complex services we've implemented, since it involves notifying the uses of observations, and integrating several services at once: CO (Citizen Observatory) -> Cos4BIo -> DUNS -> DUNS -> CO. The process involves information feedback, where there can be no failure.

For this reason, once again, it was essential to rely on a continuous development system, **which** would allow us to carry out the implementation of DUNS. DUNS can be accessed from the following link: https://cos4duns.eu here, you can see the presentation page of the service, that explains what it is, its advantages , the data flow, and the set of operations/calls you can use, as well as how it can be integrated or how we can help you if you need it.

2. Analysis

2.1 Introduction

The analysis process carried out for the implementation of DUNS has been based on the same methodology used during the development of Cos4Bio (Expert Portal for Biodiversity Data Validation https://cos4bio.eu/ a service that integrates biodiversity observations from multiple citizen observatories in one place: save time in the species identification process and get access to an enormous number of observations):

"Design Thinking: Empathise, define, devise, prototype and test".

This methodology is divided into four phases that we have been developing throughout this block, adapted to the specific needs of the DUNS service:

- Empathise: research users' needs
- Define: users and their needs.
- Ideate: challenge assumptions and create ideas.
- Prototype: create solutions.
- Test: test solutions.

These phases are described in detail in section 2.3 below. Before each of the phases, we **hold** co-design and technical sessions to collect the suggestions of experts who were not involved in developing the initial architecture. Based on the list of suggestions, we were able to implement a Backlog decision table and on the whole, carry out the Design Thinking process now applied to the new DUNS service.

Below we refer to the meetings held during this second period and the suggested inputs to comment on each of the stages.

2.2 Co-design inputs

Co-design activities implemented for DUNS

- Sci&Tech meetings: bimonthly since April 2021.
- General meeting (June 2021): interoperability integration.
- Annual meeting (November 2021): key performance index for services and EOSC publication.

Thanks to the feedback received, we made a list of requirements as a Backlog table (Table 1. Backlog table) with the following columns:

- Id: Identifier of the suggestion.
- User 's feedback: suggested by an expert.
- Decision: Internal decision of whether or not to carry out the task, followed by the explanation column.
- Explanation: decision field justification.
- Priority: Low / Medium / High / Very High.
- Status: Done / Working on it / Discarded.

	Id	**User's feedback**	**Decision (yes/no)**	**Explanation (reason why we could do it or not)**	**Priority**	**Status**
🟢	1	Interoperability service	Yes	Interoperability is crucial for the service.	Very High	Done
🟢	2	Accessible service	Yes	Accessibility is essential to the service.	Very High	Done
🟢	3	It is essential to use standards	Yes	Standards must be used in accordance with the guidelines of the Cos4Cloud framework.	High	Done
🟢	4	Find some way to identify all the observations that come from the different Citizen Observatories.	Yes	Our system is based on URI.	Very high	Done
🟢	5	An integration with some of the Cos4Cloud services	Yes	In spite of the fact that it was not outlined in the agreement, we decided to demonstrate the	High	Done

				functionality of DUNS, and its degree of integration, in order to integrate it with Cos4Bio, thus further enhancing the service developed during the second period.		
●	6	Data on usage by user	Yes	We consider it important to have usage information at the level of each observation made by each observer.	High	Done
●	7	Usage statistics by the portal	Yes	Although the most relevant information and the main objective was to be able to show the use at the user level, we thought it interesting to be able to provide this information also at the CO level, which we believe will provide us with added value or an additional feature of Cos4Bio and in this way be able to attract more CO.	High	Done
●	8	Data usage by country	Yes	A basic statistic that will let experts/portals know where their observations are being used.	Medium	Done
●	9	Data usage by host	Yes	Specifying where, on what website, the URIs are being referenced was something we thought was very useful, so that users could investigate a little more	High	Done

				about the usefulness of their contributions on that specific website.		
🟢	10	Data usage by language	Yes	Learn more about usage by language.	Medium	Done
🟢	11	Data usage by Browser		Find out which browsers are most popular.	Low	Done
🟢	12	The number of times an observation is consulted	Yes	With this additional data, each Observatory can generate an indicator of relevance for each user.	High	Done
🟢	13	Data about temporary usage	Yes	When are each of the observations most frequently consulted?	High	In progress
🟢	14	Create a page where you can consult information about DUNS	Yes	Despite it being a highly technical service that will be housed in EOSC-Marketplaces, it is important to provide information about it, describing the processes an Observatory must follow to integrate it into its architecture.	High	Done
🔵	15	Create a tokenization system	No	While this aspect is not included in the prototype development, it could be a very interesting functionality when more CO are integrated after the prototype phase.	Low	Future work

🟢	16	**The integration with the citizen observatory Natusfera**	Yes	Although it was not specified in the Agreement, it was important to demonstrate the benefits of integrating this new service within the architectures of new COs.	High	Done

Table 1. Backlog table.

2.3 Stages

2.3.1 Empathise

One of the ideas behind the development of DUNS was to make it a service that generates transparent information for users based on queries made by experts and scientists. Thanks to DUNS and through the integration and interaction between citizen observatories and, for example, Cos4Bio, citizens can learn how their observations are being used. In spite of not knowing what occurs behind the scenes between when an observation is uploaded to when an expert or scientist makes a query, citizens will see that it's an improvement in both COs and Cos4Bio.

2.3.2 Define

Once we finished the analysis stage, we defined with the experts who could be the potential internal and external users for the DUNS service.,Compared to Cos4Bio and Cos4Env services, a larger set of roles were identified for DUNS, each participating in data generation; including, for example, services.

These are the user profiles that will be involved in DUNS:

Table of users	
Role	**What will they find on the platform?**
User/Citizen	Knowing how experts make use of their observations is important to the user.
Expert user	A scientist or person with extensive knowledge of the context in which they are working, for example biodiversity. It is necessary for an expert user to cite an observation in a simple and unique way, such as through a URL.
Cos4Bio (and other services?)	To add value to Cos4Bio, DUNS generates permanent, resolvable, and unique indicators for each observation downloaded by an expert, all without requiring much time to download. This would require a background download system.
Citizen Observatory (Natusfera)	Through a simple integration system, the citizen observatory can access information at both a global platform level and an individual level for each of its users and show usage statistics based on your observations.
External User that query the information	The user may be interested in information published by an expert or scientist. It is possible that the user has a significant level of knowledge and is curious about where the information cited by the expert comes from. It's important to get information in an easy-to-understand format.

Table 2. Table of users.

We further defined the typical user of our platform, which could be a person with the following profile:

- Approximate age between 25 and 65 years.
- Interested in citizen science, biodiversity, environmental variables and sustainability.

- And with advanced knowledge about the identification and validation of biodiversity observations and environmental measurements.

Following is a diagram that illustrates the role each plays in the complete DUNS service process, which will allow us to move onto the next step "Ideate".

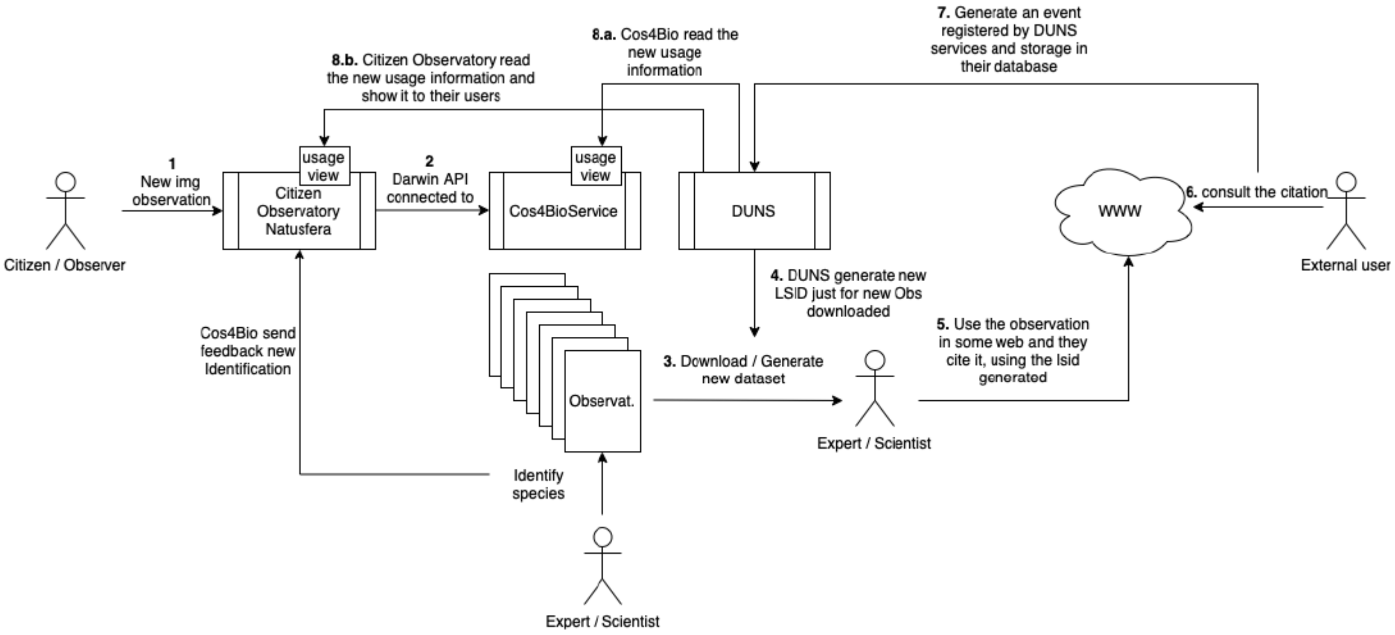

8.b. Citizen Observatory read the new usage information and show it to their users
8.a. Cos4Bio read the new usage information
7. Generate an event registered by DUNS services and storage in their database
1 New img observation
usage view
Citizen Observatory Natusfera
2 Darwin API connected to
usage view
Cos4BioService
DUNS
WWW
6. consult the citation
Citizen / Observer
External user
4. DUNS generate new LSID just for new Obs downloaded
5. Use the observation in some web and they cite it, using the lsid generated
Cos4Bio send feedback new Identification
Observat.
3. Download / Generate new dataset
Expert / Scientist
Identify species
Expert / Scientist

Figure 1. Data flow diagram.

2.3.3 Ideate

We have carried out an "Ideate" process that helps to clarify the aspects that we have to implement, taking into consideration that we can only define some visual aspects of the user interface after consulting Cos4Bio and Natusfera. Because this is not a web portal, we will not show a Site Map Diagram or wireframes, as we will not present a Look & Feel for the service, as was the case with Cos4Bio (see Deliverable X) or as we will in Cos4Env.

- Flow Diagram
- Task Flow Diagram
- Uses Case Diagram

All of these diagrams also helped us to identify improvements in information flows, detect useless functionalities and take all the relevant aspects of the service into account.

2.3.3.1 Flow Diagram

The objective of the flowchart is to represent the actions that users will be able to take to complete the different processes covered by the DUNS service.

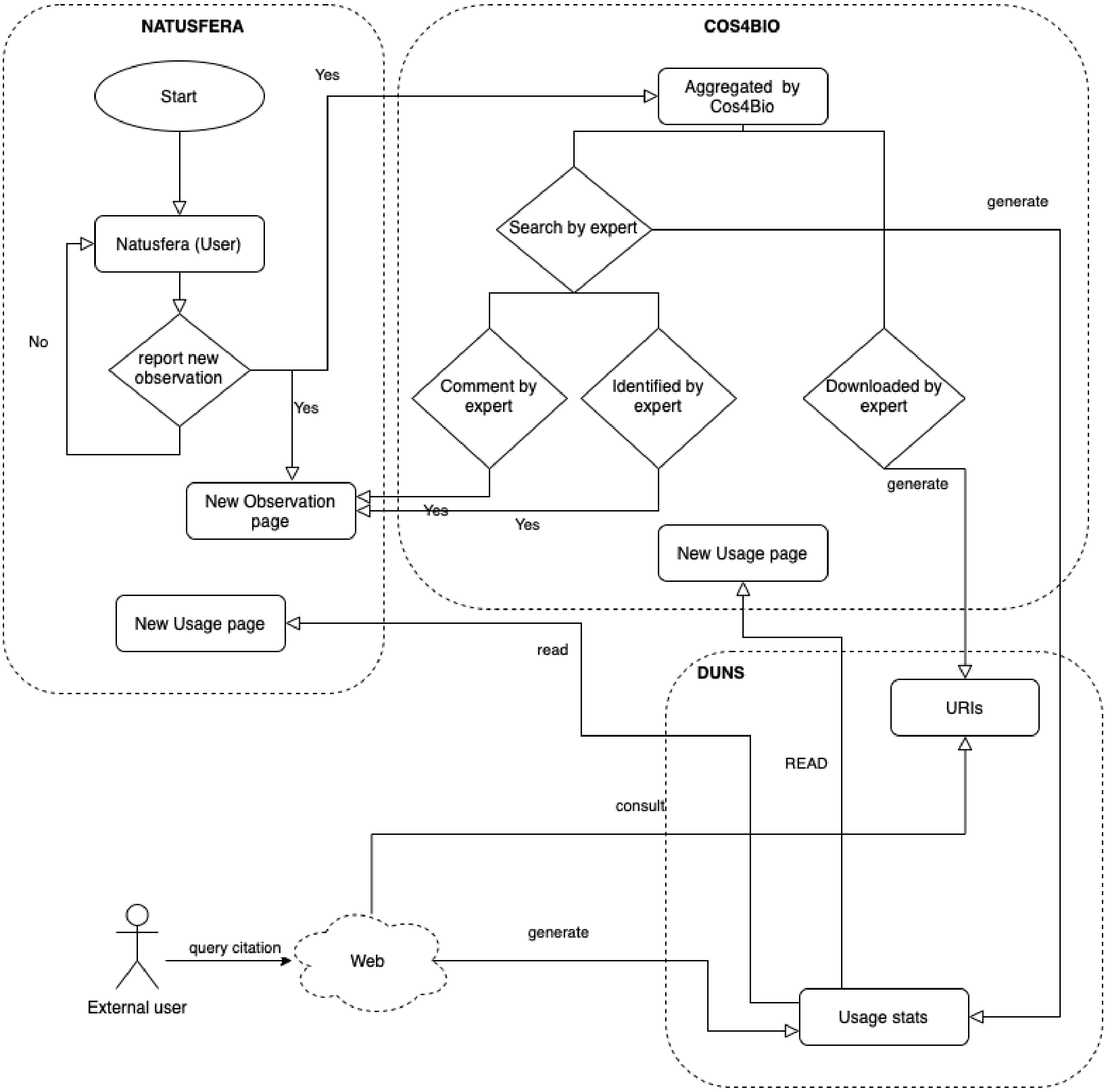


Figure 2. Flow Diagram.

The process begins with the contribution of a citizen or observer. When an observation is made in a CO that is integrated in the Cos4Bio framework, it is added by the system, and it becomes available to the experts. Therefore, the experts can take several actions from this point forward:

- Look for an observation (search by expert in the diagram).
    - There are three other flows that can arise from this action:
        - The expert simply consults the information, generating a new record of use.
        - The expert comments on the observation within Cos4Bio.
        - The expert identifies the observation.
    - Generate a new data set to download the observations.
        - For each observation downloaded, an individual URI is generated, which the expert can then use to cite the observation on a web page, generating new usage information for external users who are querying the source of the citation. By clicking on the URI, you will be redirected to the observation page, where the usage record is saved.
- It is possible to consult all the information generated as shown in the diagram from the CO as well as from Cos4Bio.

As a demonstration of interoperability and integration, these flows have been developed so that the CO can be enhanced. In the case of Natusfera and Cos4Bio, thanks to the continuous and interconnected development of services involved in the Cos4Cloud project..

2.3.3.2 Task Flow Diagram

The Task flows are focused on how users travel through the platform while performing a specific task. They generally show only one path and don't include multiple branches or pathways like a traditional user flow might. These are best used when the task being analysed is accomplished similarly by all users. When using task flows, it is assumed that all users will share a common starting point and have no variability in the way the task is carried out.

Each of the Task Flows is presented below, taking into account the changes made and the new functionalities implemented for the DUNS service.

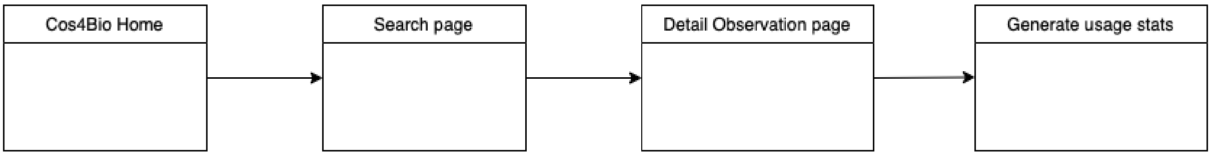


Figure 3. Generate usage stats by consulting the Detail Observation page. Task Flow.

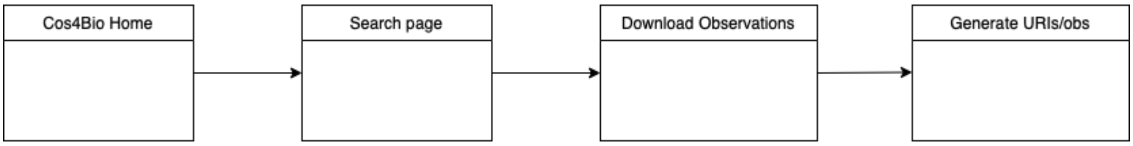


Figure 4. Generation of URIs during Download flow. Task Flow.

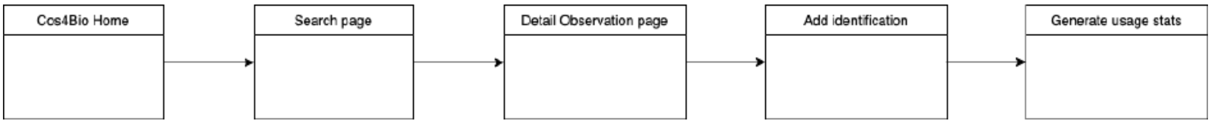


Figure 5. Usage stats generated during add identification flow. Task Flow.

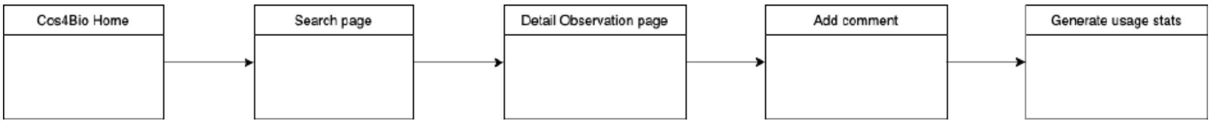


Figure 6. Usage stats generated during add comment flow. Task Flow.

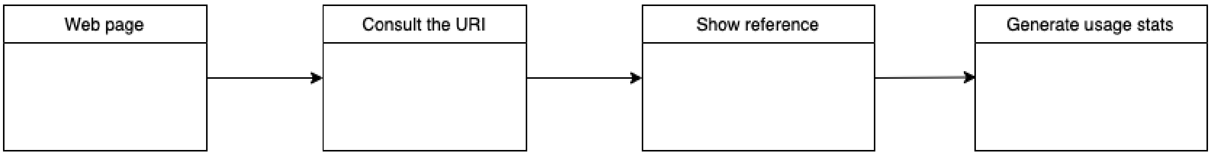


Figure 7. Generate usage stats per external consulting of URI citation. Task Flow.

2.3.3.3 Mockups

Mockups are a visual guide that represents the design of a website. They include the navigation structure and design elements in detail and help to visualise ideas and concepts. We have developed a new view to help users consult DUNS usage information, as we did with the Sitemap (https://cos4bio.eu/dashboard)

Figure 8: Now, in addition to consulting the experts' information, it is also possible to consult through the Dashboard the usage information generated by each user's observations, both globally and individually.

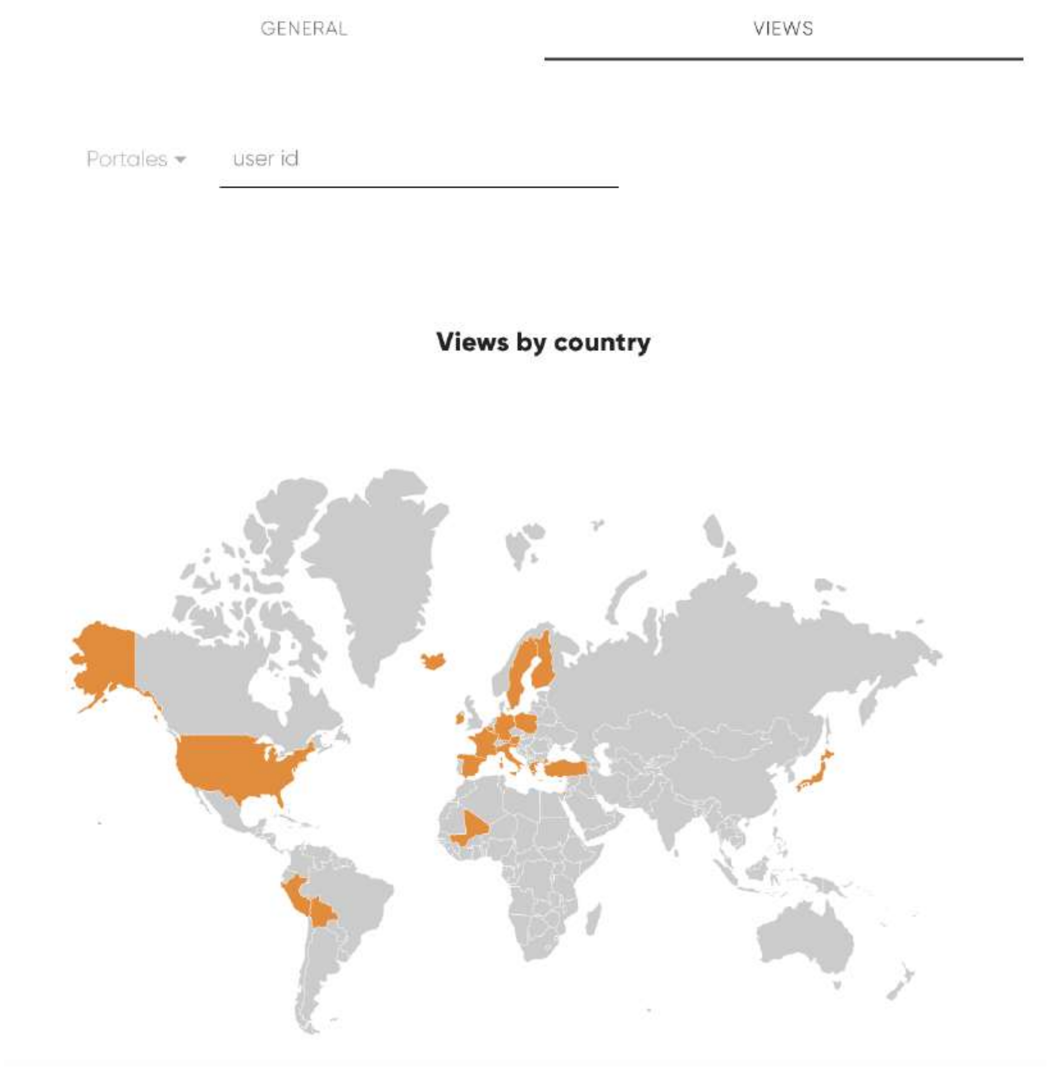


Figure 8. Mockup. Usage view with different usage stats; in this case, stat usage by country.

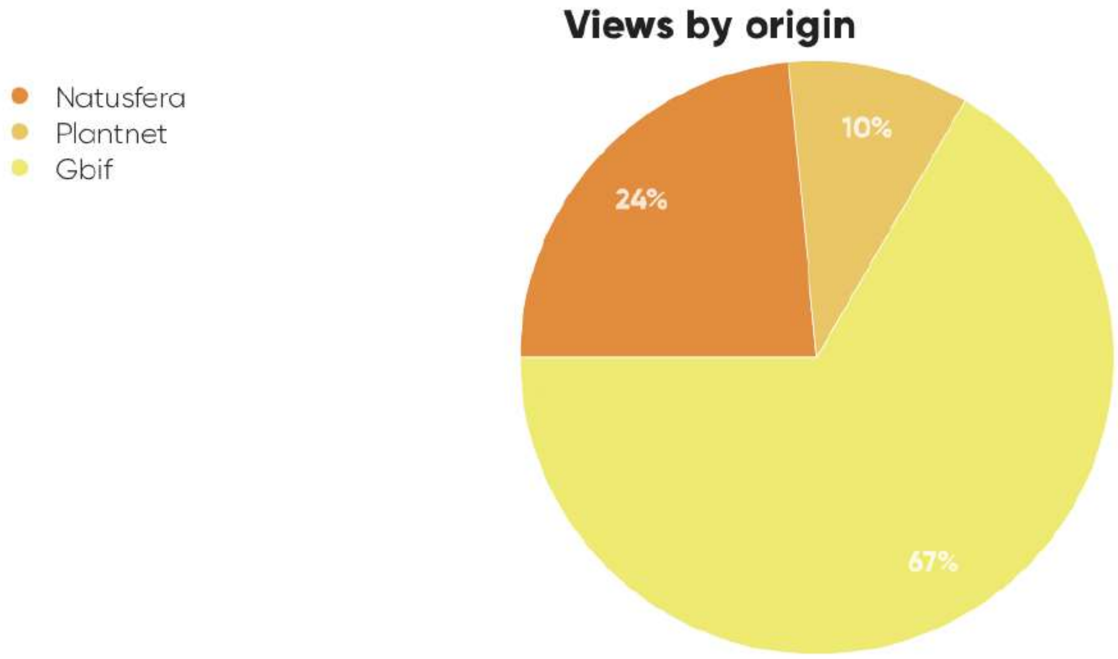


Figure 9. Mockup. Usage stats view Origin (Source Citizen Observatory).

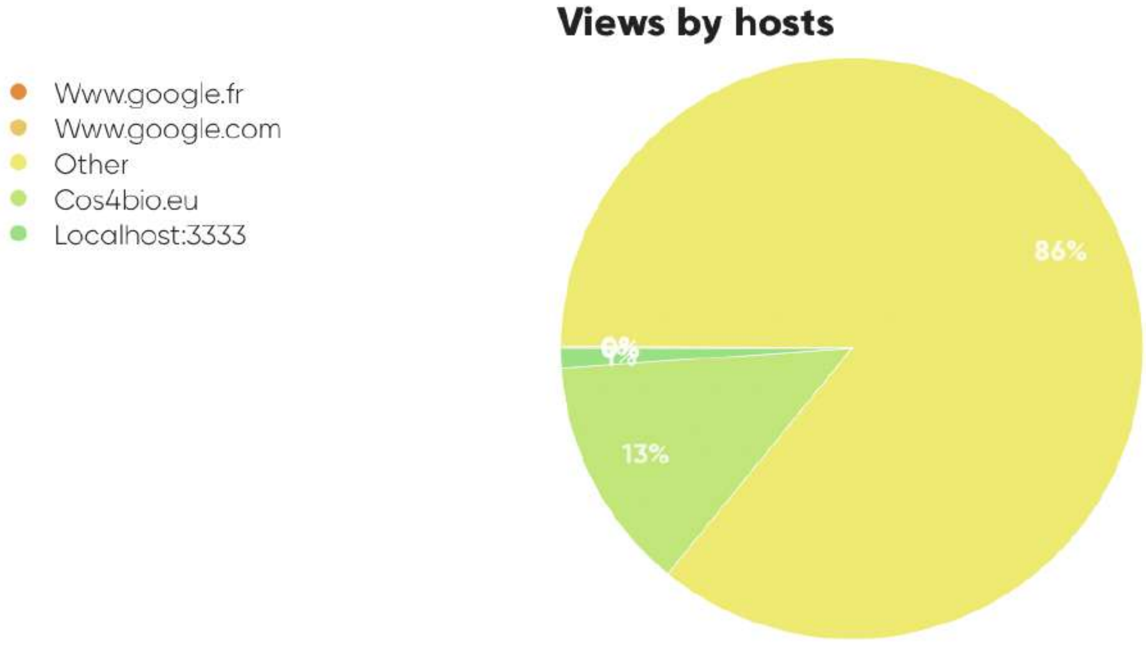


Figure 10. Mockup. Usage stat by host.

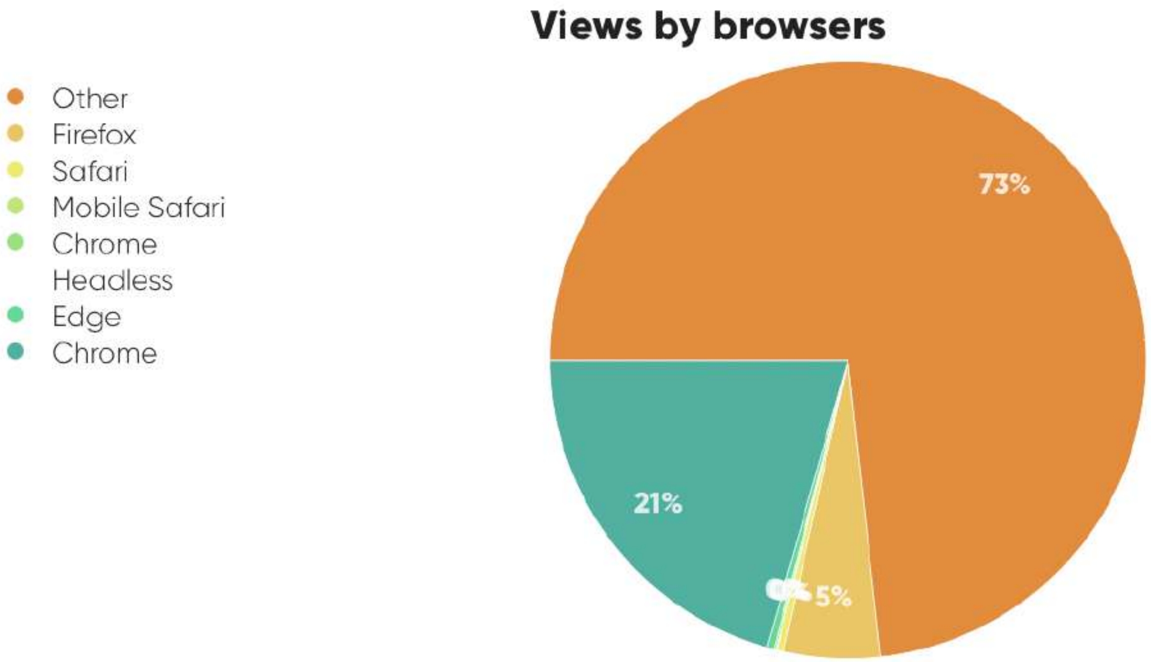


Figure 11. Mockup. Usage stat by browser.

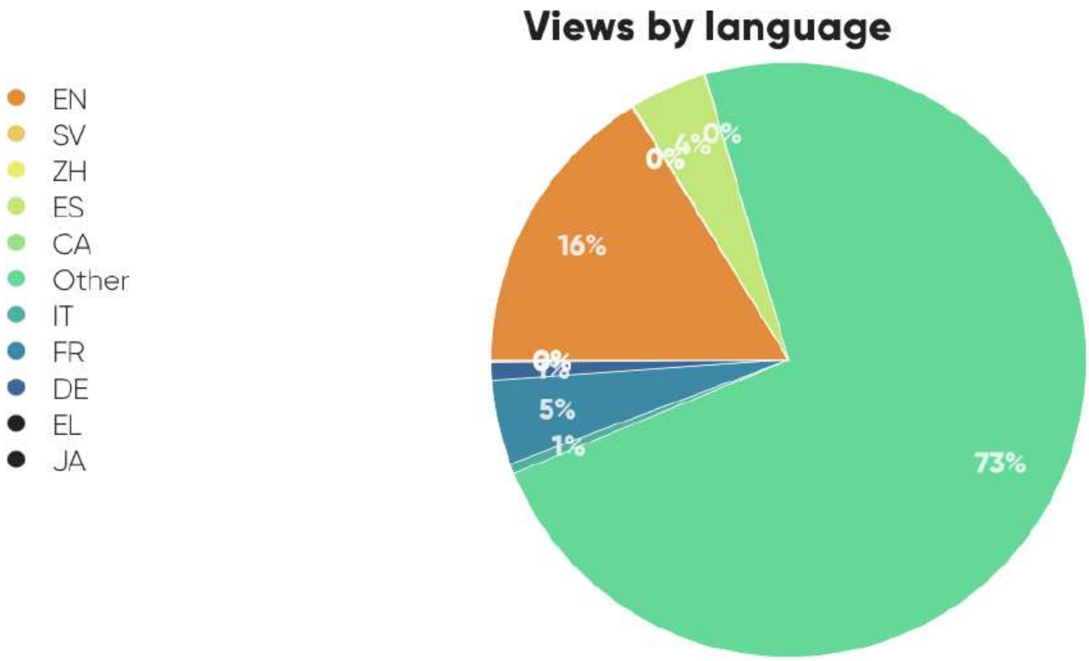
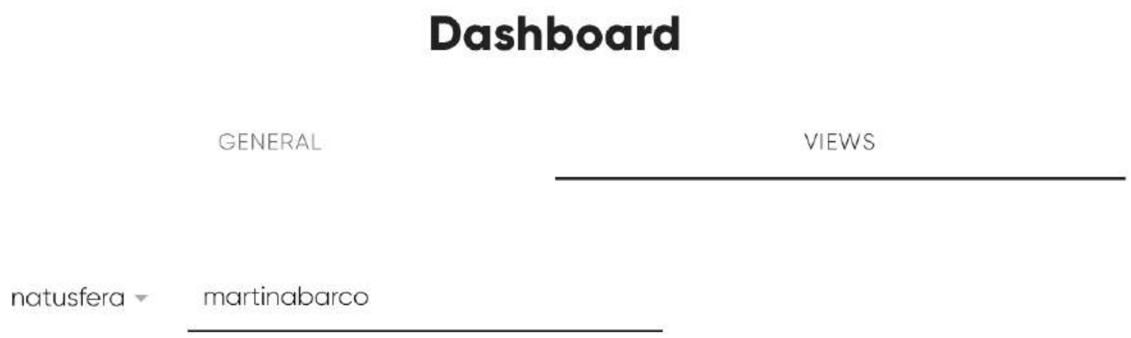


Figure 12. Mockup. Usage stat by language.

This information can also be consulted by a specific user of one of the platforms integrated into Cos4Bio, enabling them to l remain informed about the use of their observations. In order to do this, the user's identifier is included in the selector:



natusfera

martinabarco

Figure 13. Mockup. Usage selector by User and CO.

Following that, all usage statistics are filtered according to the selected user and platform:

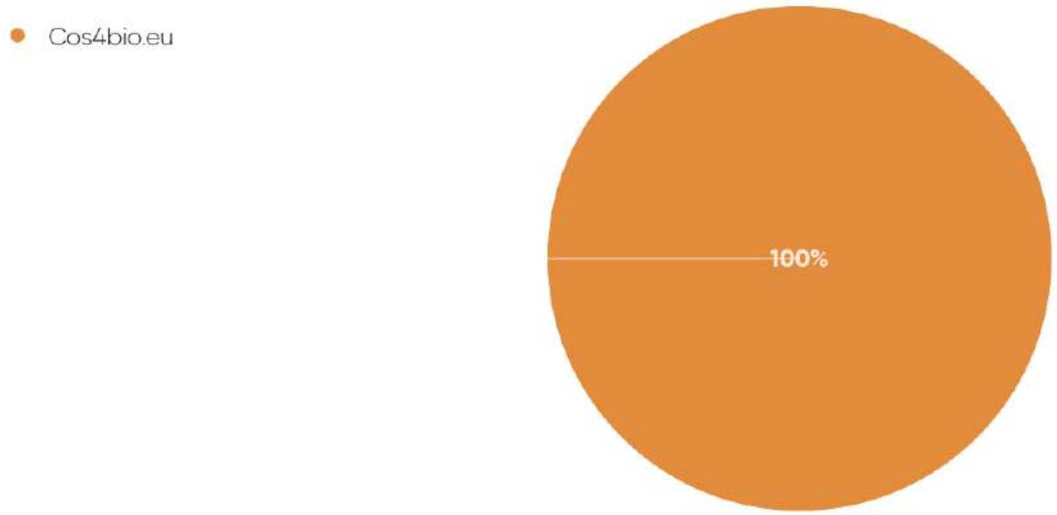


Figure 14. Usage stat by browser.

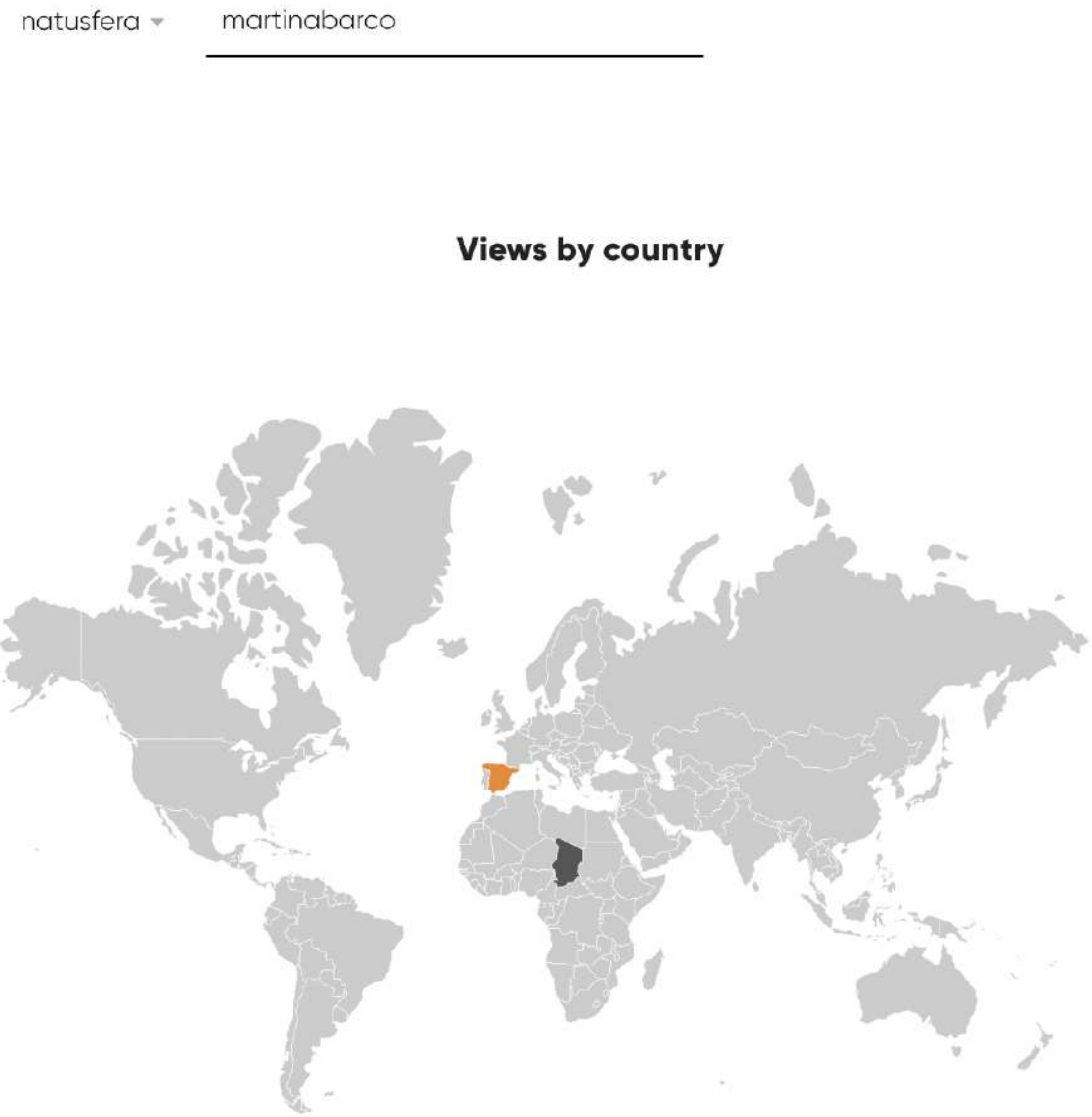


Figure 15. Mockup. Usage stat by country.

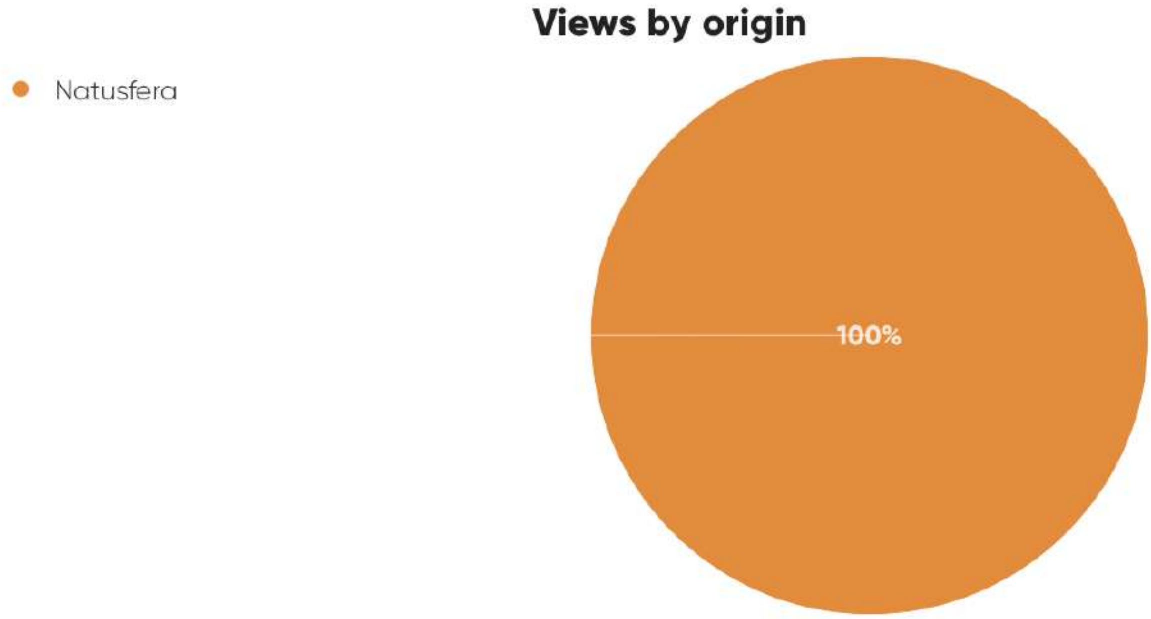


Figure 16. Mockup. Usage stat by origin.

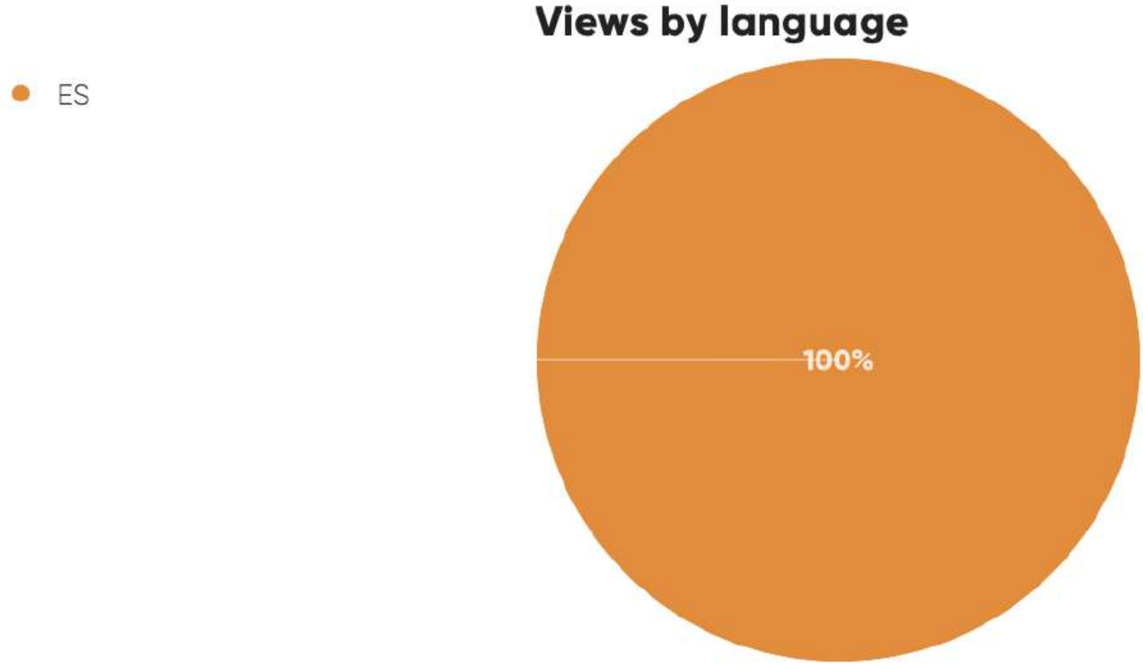


Figure 17. Mockup. Usage stat by language.

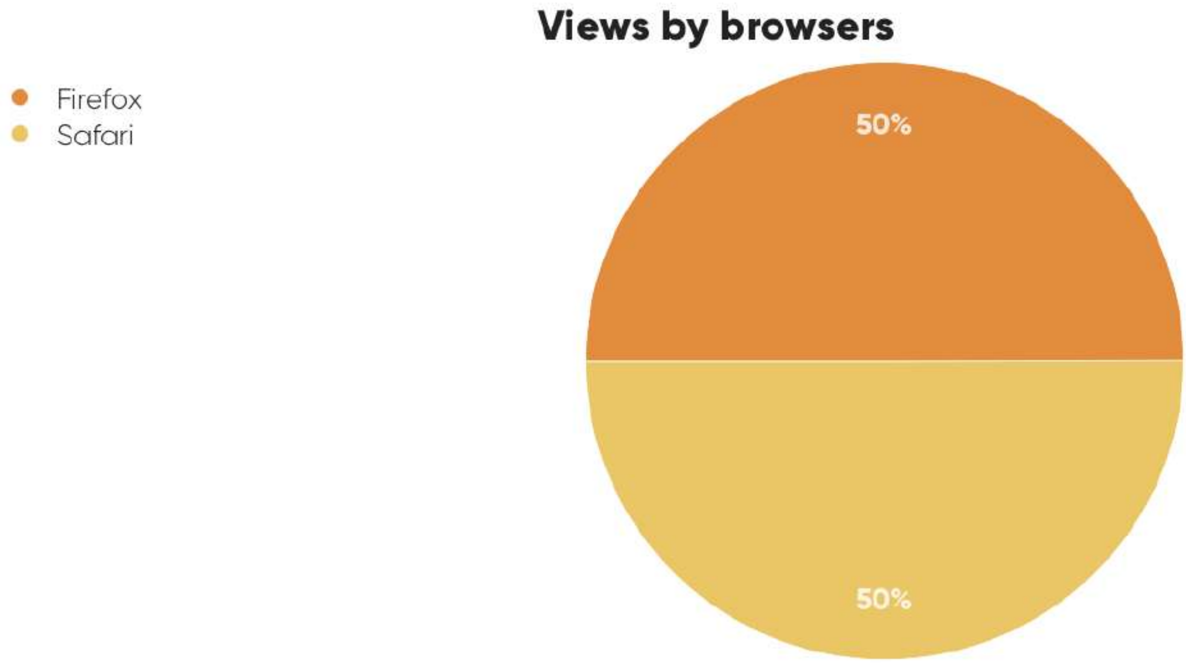


Figure 18. Mockup. Usage stat by browser.

2.3.4 Prototype

Having generated and agreed the task flows during the different co-design and technical meetings, we developed an initial prototype. This prototype was then tested by both internal experts and the experts of the Co-design group. Once reviewed, we moved to the next phase, the implementation phase, which we describe in section [3. Development](#). The prototype implemented during this phase is available at the following public link https://cos4duns.eu.

2.3.5 Test

In this section we detail which requirements were tested within the scope of Cos4Cloud, how testing has been implemented, and how it was applied throughout the project.

The following requirements were tested and can be viewed in [Table 3: Technical integration Experiment](#).

1. What are the **relevant user stories** developed for Cos4Cloud?
2. What **criteria have to be fulfilled** in order to consider the user story as “done”?
    a. What testable criteria can provide "proof" that the business requirement described by the story is implemented?
    b. Each user story shall have a list of acceptance criteria that have to be tested and re-tested respectively.
3. Determine the **maturity context**: Under what TRL is the user story considered important?
    a. What readiness level will the outputs have in terms of EU TRL (See Annexe: TRL Calculation DUNS).
    b. The story may span multiple levels but ideally, a story is tailored to match a particular one.
4. Under what **level of detail** have tests been performed?
    a. Gives an idea of the granularity under which testing took place.
5. What aspects have been tested (**functional and non-functional**)?
    a. *non-functional* covers all quality aspects and boundary conditions.
6. How testing has been **implemented and is applied** (for example; tools, workshops, continuous integration testing, or manually following a checklist)
    a. When: automated tests, manual tests, during each build/deployment, or before each release.
    b. How: see the list in functional and non-functional for examples
    c. Context: using Co-Design sessions, datathons, or hackathons, for example, to test certain aspects.
7. Is there any relation to other Cos4Cloud services? What are testable use cases when **integrating with other services**? For example:
    a. A cross-point in the Technical Integration Experiment (TIE) table (describe the particular use case).
    b. A service integration of any other kind.

2.3.5.1Test implementation

The next steps represented in Table 3, demonstrate what requirements are actually put under test in the scope of DUNS service, how testing has been implemented and how it was applied throughout the project.

User Story	Relevance to Cos4Cloud	Relevant TRL (s)	Acceptance Criteria	Testing Granularity (see the test pyramid)	Access to be Tested	How to Verify/Test?	Integration Test (e.g. TIE)
Citizen observatory or Cos4Bio use the DUNS API to display usage data.	-Usage stats. -API. -Services implementation. - Use of standards .	TRL-1 TRL-2 TRL-3 TRL-4 TRL-5	- Through an API, external actors can easily view usage information. - Data visualisation does not generate delay times. - The DUNS system can show the usage information for each Citizen Observatory by a specific user. - Scalable service is a must. - The system is easy to maintain.	- Service. - API test. - HTTP request information.	**Functional** - Statistics on global usage by platform. - Usage statistics by user. - Regional statistics. - Statistics by host about usage. - Statistics on usage by language. - Statistics on usage by reason. - Statistics on usage by browser **Non-Functional** - Provide optimal response times. - Easy to use through simple calls. - It is easy to integrate. - Observations can be tracked in terms of their use.	- Unit test. - API test. - Natusfera Integration. - Cos4Bio integration	References implementation: API implementation - https://github.com/Bineo-Consulting/DUNS DUNS integration on Cos4Bio https://cos4bio.eu/dashboard Note: New View tab that includes all new usage stats.

Experts consult one observation through Cos4Bio and DUNS registers the event.	- API test. - Integration. -Functionality.	TRL-6	- When a user consults one of the platforms integrated in Cos4Bio, DUNS must be able to register its use. - Current Cos4Bio operation shouldn't be affected.	- API test. - New Component Integration. - Interconnection between different services of Cos4Cloud.	**Functional** - Correct registration in DUNS of the user's platform event. **Non-Functional** Cos4Bio services will not be affected by the new DUNS integration.	- Unit test. - API Test. - Co-Design sessions. - Pl@ntNet integration. - Natusfera integration. - Cos4Bio Integration	References implementation: https://github.com/Bineo-Consulting/cos4bio-front https://cos4bio.eu/dashboard
Experts include new comment in one observation through Cos4Bio and DUNS register the event.	- API test. - Integration. - Functionality.	TRL-6	- As soon as a user adds a new comment to any of the platforms integrated into Cos4Bio, DUNS must be able to register the request.. - Current Cos4Bio operation shouldn't be affected.	- API test. - New Component Integration. - Interconnection between different services of Cos4Cloud.	**Functional** - Correct registration in DUNS of the user's platform event. **Non-Functional** Cos4Bio services will not be affected by the new DUNS integration.	- Unit test. - API Test. - Co-Design sessions. - Pl@ntNet integration. - Natusfera integration. - Cos4Bio Integration	References implementation: https://github.com/Bineo-Consulting/cos4bio-front https://cos4bio.eu/dashboard
Experts include new identification in one observation through Cos4Bio and DUNS register the event.	- API test. - Integration. - Functionality.	TRL-6	- When a user identifies an observation of one of Cos4Bio's platforms, DUNS must be able to register it. - Current Cos4Bio operation shouldn't be affected.	- API test. - New Component Integration. - Interconnection between different services of Cos4Cloud.	**Functional** - Correct registration in DUNS of the user's platform event. **Non-Functional** Cos4Bio services will not be affected by the new DUNS integration.	- Unit test. - API Test. - Co-Design sessions. - Pl@ntNet integration. - Natusfera integration. - Cos4Bio Integration.	References implementation: https://github.com/Bineo-Consulting/cos4bio-front https://cos4bio.eu/dashboard

Experts download new observations in Cos4Bio and DUNS generate new URL identifiers.	- The download time should not be affected. - A URL must be generated that can be used by an expert when citing the observation.	TRL-6	- When a user downloads an observation from one of the platforms integrated into Cos4Bio, DUNS must be able to register its use. - Current Cos4Bio operation shouldn't be affected. - A URI must be generated by DUNS for each downloaded observation.	- API test. - New Component. Integration. - Interconnection between different services of Cos4Cloud.	**Functional** - Cos4Bio Download process. **Non-Functional** - Cos4Bio services will not be affected by the new DUNS integration.	- Unit test. - API Test. - Co-Design sessions. - Pl@ntNet integration. - Natusfera .integration. - Cos4Bio Integration.	References implementation: https://github.com/Bineo-Consulting/cos4bio-front https://cos4bio.eu/dashboard
The user of a platform wants to check the usage information of one of his observations .	- Interoperability - platform collaboration - Engagement - Integration service.	TRL-7 TRL-8	- Include the MVE fields needed to be able to check the usage information. - Do not affect too much the existing model in Natusfera.	- Service. - Integration test. - API test.	**Functional** - Easy integration. - Easy to maintain. - Only accessible for the observation owners. **Non-Functional** - Easy to consult. - Fast response times.	- Unit test. - Co-Design session. - Cos4Cloud project in Natusfera. - API Test.	https://natusfera.gbif.es/observations/72437
External user consults citation done by Expert.	- Usability - Accessibility. - Standardisation -Observation query record. - New API service.	TRL-7 TRL-8	- Monitoring the use of observations from outside the citizen observatory is possible thanks to the users' queries.	- Service. - Integration test.	Non-Functional - Visibility. - Integration with others. services on EOSC.	- Unit test. - API Test.	

	- User engagement.						
EOSC publication	- EOSC compliant - Production environment	TRL-9	- In the prototype phase, the service will be accessible only through the platforms integrated with Cos4Bio. - The documentation. - Maintenance is ongoing. - Integrate with other EOSC Marketplace services. Cos4Bio, Natusfera, Pl@ntNet...	https://cos4duns.eu API documentation	Non-Functional - Visibility. - Engagement. - Communication. - Sustainability.		Add the repository to the EOSC Marketplace https://marketplace.eosc-portal.eu/services

Table 3: Technical integration Experiment.

3. Engineering software design

3.1. Introduction

In this section we present the DUNS Software Engineering analysis that will allow us to understand each and every one of the use cases present in the system, the interaction diagrams and the resulting domain model.

3.2. Vision

In DUNS, the main purpose is to record events related to the usage of observations aggregated through Cos4Bio. Users of the portals of origin can be notified when one of the URLs generated by DUNS is consulted by an Internet user. Citizen observatories could improve their retention indicators by providing more information to citizens who shared observations.

What are the differentiating elements and advantages?

The principal advantages that we find in DUNS are:

- Provides additional information on the use of observations.
- It is easy to integrate.
- Easy to maintain.
- Scalable.
- Provides usage information at the Cos4Bio, CO, user, and observation level, with various levels of detail.
- Observatories now have new opportunities for gamification.
- Enhances the potential of Cos4Cloud services like Cos4Bio.
- It is available on the ESOC-Marketplace.
- Standard protocols are used.
- Easily understandable API.
- In the prototype phase, the service is free.

Are there products or services that offer something similar?

DOI is arguably the only service similar to DUNS in terms of usage reporting today. This service provides usage information at the dataset level (if we speak in a Biodiversity context) and also at the scientific article level, but the DOIs do not provide tracking at the Observation level that is contained in each dataset.

Many times, an expert wishes to refer to a specific observation, but has no way of ensuring that the citizen who made the observation receives a report on their Internet use.

The two technologies/services could be complementary, which makes them powerful information mechanisms for platforms such as GBIF (Global Biodiversity Information Facility). As one of our future goals, we would like to include the connection of our services with the scientific community through platforms such as the EOSC centre and GBIF. Currently, DOIs are integrated into GBIF, but there is no service that allows monitoring at the observational level. Our goal with this integration is to increase the engagement and sustainability of DUNS in the future.

Is there an equivalent product or service?

Based on our previous comments, DOI may be the only service conceptually similar to DUNS, however it cannot be used at the observation level. So we can say that DUNS is truly innovative, despite its use of widely known and widely used technologies, and will prove useful to many platforms that wish to track their data usage.

Is there a workaround that people are using that is good enough but not perfect?

Up until now we do not know a service that performs the work we are carrying out, so we can say that DUNS is an innovative system at a Global level.

What are the strengths and weaknesses of the competition?

The DOI Foundation could be a possible competitor, since it has an adequate structure and service that, although it does not do the same as DUNS, could offer you:

- The DOI Foundation has already developed a broad and consolidated structure.

- The DOI technology performs a similar function, but at the level of papers and data sets.
- The scientific community supports it strongly.
- Their communication team is very important.
- On the EOSC Marketplace, they already have a presence: https://marketplace.eosc-portal.eu/services/datacite-doi-registration-service?q=DataCite+DOI+registration+service

All these positive aspects will allow us in the future to define together with the CoNNect Group (consisting of WPs 5, 6, 7 and 8) a sustainability and leadership strategy for DUNS.

However, DOI has a number of shortcomings such as:

- At the observation level, they do not track usage.
- Citizen science is a niche they are not well established in, which may make DUNS a good fit.
- Citizens may have difficulty understanding the service.
- Integration can be difficult for Citizen Observatories.

3.3. Use Case Diagram

Use cases (Figure 19) are functional requirements that indicate what the system will do. They are the main mechanism for discovering these requirements and defining them, so they define how the system will behave.

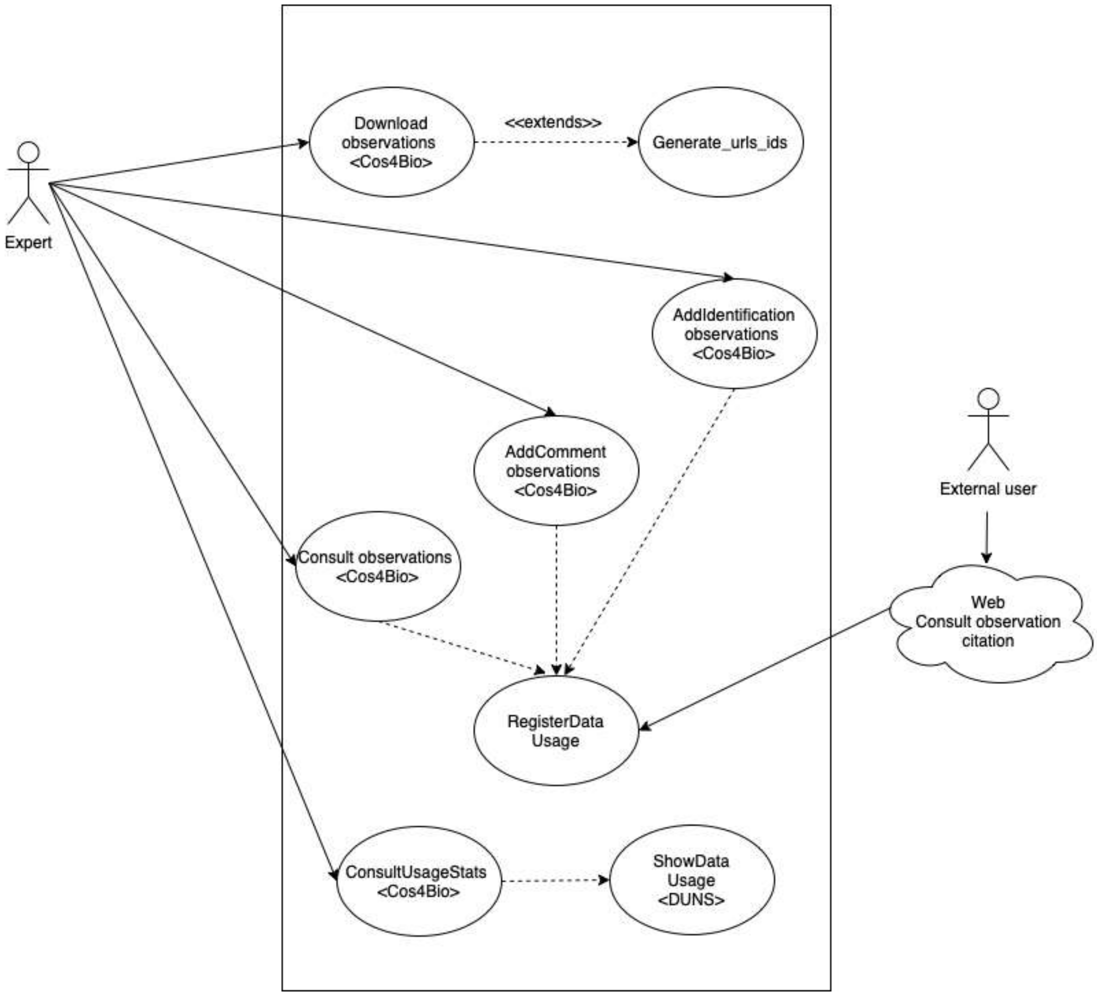
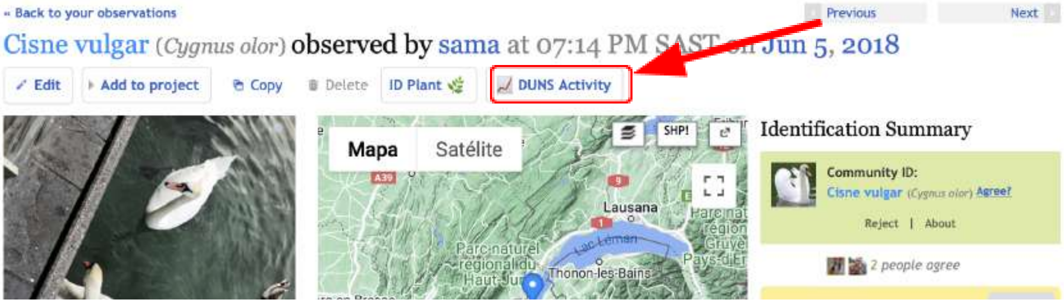

Expert
Download observations <Cos4Bio>
<<extends>>
Generate_urls_ids
AddIdentification observations <Cos4Bio>
AddComment observations <Cos4Bio>
Consult observations <Cos4Bio>
External user
Web Consult observation citation
RegisterData Usage
ConsultUsageStats <Cos4Bio>
ShowData Usage <DUNS>

Figure 19. Use Case Diagram.

In the example below (Figure 20), we can see that a new button has been added to the observations within the Cos4Cloud project in Natusfera:



Cisne vulgar (*Cygnus olor*) observed by sama at 07:14 PM SAST on Jun 5, 2018

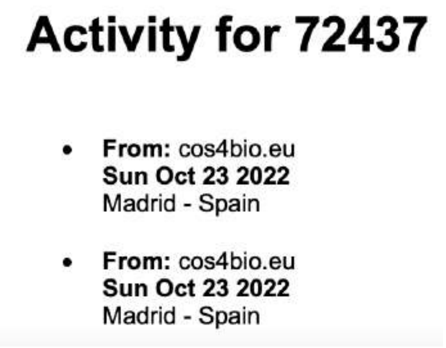
Identification Summary

Community ID:
Cisne vulgar (*Cygnus olor*) Agree?

2 people agree

Figure 20. Observation page with DUNS Activity boton.

The usage information is only accessible to the owners of the observations, and it is a very clean way to integrate Natusfera and DUNS. As a result, Natusfera users will have more information about where, how many and when their observations are used.



Activity for 72437

- **From:** cos4bio.eu
  Sun Oct 23 2022
  Madrid - Spain
- **From:** cos4bio.eu
  Sun Oct 23 2022
  Madrid - Spain

Figure 21. Usage activity information reported in Natusfera using DUNS service.

4. Development

4.1 Introduction

In the development section, we will explain the technology that we have used based on the Agile Methodology and Co-design. We will also provide an explanation and an analysis from the point of view of the Frontend and Backend, to finally show an image of the system architecture.

4.2 Methodology

Additionally to Agile and Co-design methodology, we focus on reaching achievable milestones in short sprints to close the stages towards each objective. To reach a milestone, we follow three basic principles: Simplicity, Communication and Feedback.

- **Simplicity:** The fewer elements that disperse the target's attention, the easier it will be to achieve it.

- **Communication:** This principle is indispensable. Each difficulty, doubt, comment or change is put in common language to assess whether the final goal is achievable or what elements are prioritised over others.

- **Feedback:** The team seeks constant feedback from those responsible for the project, both when it comes to resolving doubts and commenting on positive points, strong points of the project, elements for improvement or possible changes. Thus, the team advances with confidence and with a clear objective.

4.3 Technology

In the technology section, we explain all technologies and services for the decision making regarding the methodologies used in the development and the choice of each one of them.

4.3.1 Life Science Identifiers

Life Science Identifiers are a way to name and locate pieces of information on the web. Essentially, an LSID is a unique identifier for some data, and the LSID protocol specifies a standard way to locate the data (as well as a standard way of describing that data). They are a little like DOIs used by many publishers.

An LSID is represented as a uniform resource name (URN) with the following format:

urn:lsid:<Authority>:<Namespace>:<ObjectID>[:<Version>]
The lsid: namespace, however, is not registered with the Internet Assigned Numbers Authority (IANA), and so these are not strictly URNs or URIs.

What is the reason for not using LSIDs?

There has been a lot of interest in LSIDs in both the bioinformatics and the biodiversity communities, with the latter continuing to use them as a way of identifying species in global catalogues. However, more recently, as understanding has increased of how HTTP URIs can perform a similar naming task, the use of LSIDs as identifiers has been criticised as violating the Web Architecture good practice of reusing existing URI schemes. Nevertheless, the explicit separation of data from metadata; specification of a method for discovering multiple locations for data-retrieval; and the ability to discover multiple independent sources of metadata for anything identified were crucial parts of the LSID and its resolution specification that have not successfully been mimicked by an HTTP-only approach.

The use of HTTP URIs is very similar to that of LSIDs when used as identifiers. Web architecture simulates what LSID intends. LSIDs offer explicit separation of data from metadata, multi location discovery, and multiple independent source discovery; but neither of these are necessary for DUNS. DUNS only requires a way to normalise observation URLs, which HTTP does perfectly.

4.3.2 DOI

A DOI is an identifier or persistent identifier that can be used to uniquely identify multiple objects, and has been standardised by the International Organization for Standardization ISO. It is commonly used to identify academic, professional, and government information, including journal articles, research reports, data sets, official publications, and commercial videos. In order to achieve this, the DOI is linked to metadata about the object, such as its URL. Documents' DOIs remain constant while they exist, but their location and other metadata can change. Referencing a document by its DOI should provide a more stable link than using its URL directly. However, if your URL changes, the publisher must update the DOI metadata. Publishers are responsible for updating the DOI database. If they don't, the DOI resolves to a dead link, rendering it useless.

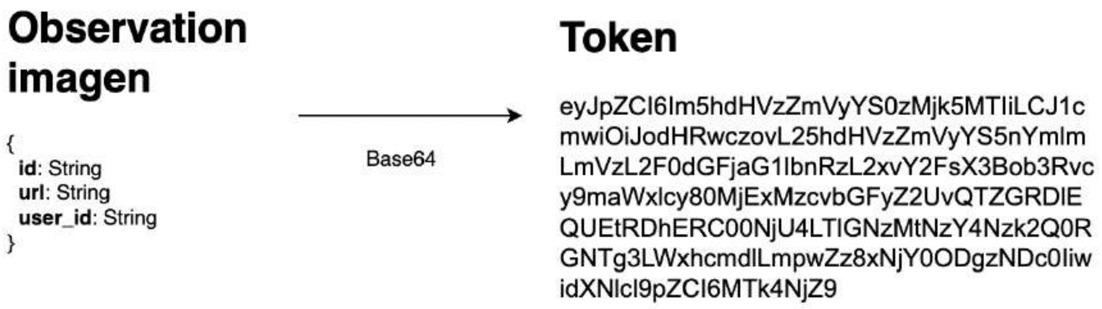
What is the reason for not using DOIs?

Due to the magnitude of this project, DOI would seem to be an unnecessary extra effort. By maintaining metadata and synchronizing, it may instead introduce many more problems than it solves.

4.3.3 HTTP + JSON

Among the variety of protocols and technologies available, we have selected the most simple and elegant solution, which does not require complex implementation solutions. By using HTTP properties to send metadata along with URIs and JSON to represent metadata, we were able to standardize an identifier for the observations URIs with the HTTP properties.

In an HTTP GET request, the JSON metadata travels in the URI parameters, but we need to convert it to a string format suitable for URIs. By converting JSON to base64, we form a URI with a token like this: https://**{endpoint}**/proxy?id=**{token}**



```
{
 id: String
 url: String
 user_id: String
}
```

eyJpZCI6Im5hdHVzZmVyYS0zMjk5MTIiLCJ1c
mwiOiJodHRwczovL25hdHVzZmVyYS5nYmlm
LmVzL2F0dGFjaG1lbnRzL2xvY2FsX3Bob3Rvc
y9maWxlcy80MjExMzcvbGFyZ2UvQTZGRDlE
QUEtRDhERC00NjU4LTlGNzMtNzY4Nzk2Q0R
GNTg3LWxhcmdlLmpwZz8xNjY0ODgzNDc0Iiw
idXNlcl9pZCI6MTk4NjZ9

Figure 22. Image in token base64 format.

By doing so, we only need to interpret the token on the server and extract the metadata to determine the observation's origin and owner.

What is the processing location of this URI?

URIs are generated on the fly when observation information is consulted in the various portals, as well as when observations are downloaded.

4.3.4 Reverse proxy

A **reverse proxy** is the application that sits in front of back-end applications and forwards client (e.g. browser) requests to those applications. Reverse proxies help increase scalability, performance, resilience and security. Resources returned to the client appear as if they originated from the web server itself

With the help of a proxy server that interprets these tokens, we can provide the client with a specific resource coming from a portal as if the resource really originated from the same server. Through the same DUNS service, we can respond with an image as well as analyse the request and save it in a database for future notifications.

4.3.5 Reverse proxy Architecture

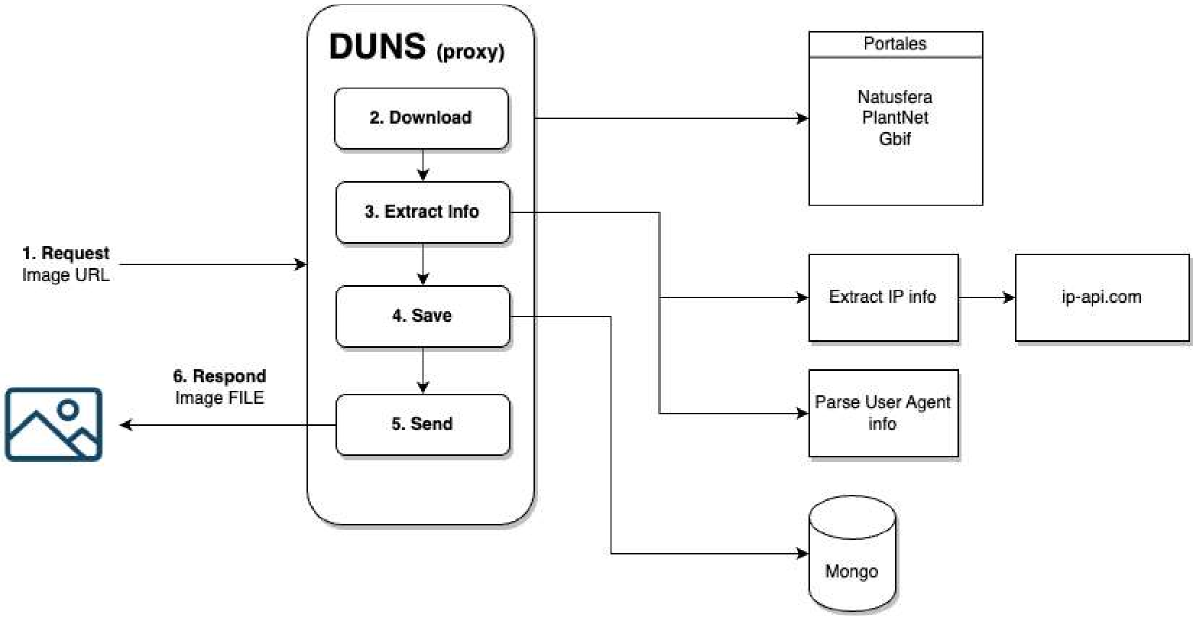


Figure 23. Reverse proxy DUNS architecture.

This figure illustrates the entire architecture of the DUNS proxy:

1. **Request:**
   In this step, a "token" with the observation's metadata is used to request the observation. By downloading Cos4Bio itself or by using the Cos4Bio service, users can obtain this token.

2. **Download:**
   Here, the URL is understood by the server, and the resource is downloaded to the specific portal.

3. **Extract info:**
   Obtain information about the HTTP request:
   a. Extract IP info: The IP address of the client is included in the HTTP call. IP-API processes this information (https://ip-api.com/) it provides information on the geolocation of the same request.
   b. Parse User agent:

The User-Agent request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting user agent.
Using this information, we are able to identify which device, operating system, language, or software version is being used to make the call, which is crucial to analysing the generated data later.

4. **Save**:
   We save the geolocation data and the details of the user agent in a database once we have them. The data at Cos4Bio is stored in MongoDB. It is this information that is used to generate the DUNS API.

5. **Respond**
   Lastly, we send the requested resource to the client. The server downloads the resource to memory, then sends it as part of the request. Servers are responsible for sending files in their proper format (content-type).

It is possible to determine the origin of HTTP requests by following these sequential steps. Because it is a JSON object, this system can be extended to provide more functionality or improve security.

4.3.6 Backend

The core of the backend that we use for DUNS, Cos4Bio and Cos4Env is NodeJs runtime. This Backend can be deployed on a server with Docker or on a Serverless infrastructure like Cloud functions. If the backend is deployed on a Serverless infrastructure, Swagger and Express are not required, but are required when deployed with Docker or a dedicated Server.

This backend is used for different purposes:

- Download observations
- Mapping observations
- Mapping URIs for DUNS
- Query differents portals
- Verify credentials of the users using Authenix
- Expose an API for stats and data

For this case we have used a very popular stack (Node - Swagger - Express - Docker):

- **Node:** We chose this framework because it is lightweight and is the same language as the frontend.
- **Swagger:** We need an API that maps the different portals (Natusfera, iSpot, PlantNet, ArtPortalen, ...), for this we have documented the API with the Swagger specification.
- **Express:** Very popular node microframework, high performance and stable since 2012.
- **Docker:** Users can download the docker virtual container/image to install locally and make changes or customise the software when needed.

Stats and Mongo Database

We have chosen MongoDb for the following reasons:

- It is NoSQL and document-based and stores data in JSON-like structures.
- Since we are using NodeJS as a backend, the most used architectures with Node and Mongo are MERN (Mongo Express React Node.js) or MEAN (Mongo Express Angular Node.js), which suits us like a glove.
- It has a powerful syntax for querying and grouping.
- Lastly, and most importantly, it is open source.

Mongo helps us build this list of features:

- We store information about downloads, so we can display historical download information for each user..
- We store user information that comes from Authenix service, this helps build the profile for each user. Also, with this information we show stats like charts and counters of the number of queries in each user profile.
- Every comment made from the portal is stored in Mongo. This helps us build stats for comments and to display comments on portals that don't have comment features like PlantNet and Gbif.
- The information stored in Mongo helps us build a global Dashboard. This dashboard consists of a list of counters (comments, downloads and users) and a series of charts (Pie and Bars) showing data of the Portal.
- We also store information about User Agent and Geolocation HTTP requests for DUNS, in order to build a dashboard and expose an API for different portals.

The architecture of the API that consume the Mongo database consists in three concept endpoints:

- We have these models: Users, Comments, Downloads and Logs
- Each model has a generic endpoints:
  - /${model}/:id
  - /${model}/search
  - /${model}/agg
- /:id return an specific document by the id
- "Search" lists all the rows of the models stored
- "Agg" lists stats grouped by Months and by type
- And we can also specify the format of the request (only available for logs). For example, the svg format responds with an image that contains an aggregated representation of the log data, and the html format returns interactivity with the image:

  - /logs/agg **(default JSON)**
  - /logs/agg**.html**
  - /logs/agg**.svg**

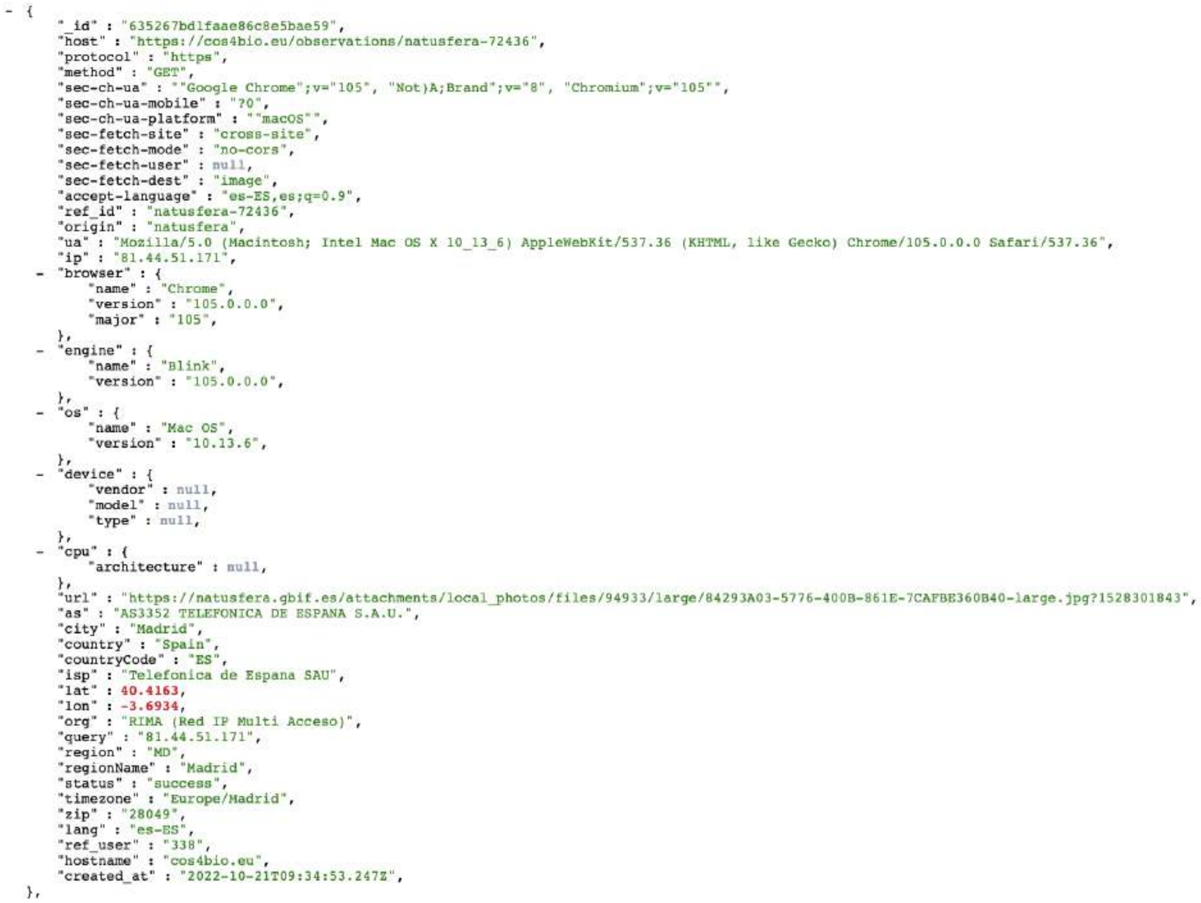
API DUNS

DUNS API uses the previous endpoint architecture and extends some formats to build an image or add interactivity.

Here is a list of the calls that can be made using the DUNS API:

- https://cos4bio.eu/api/logs/all

  Returns the entire list of document logs, in descending order by “created_at”.

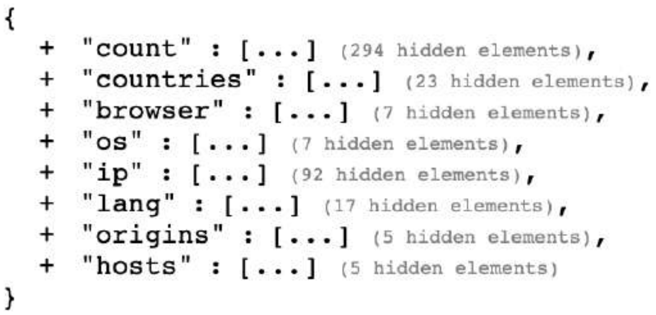


```
- {
      "_id" : "635267bd1faae86c8e5bae59",
      "host" : "https://cos4bio.eu/observations/natusfera-72436",
      "protocol" : "https",
      "method" : "GET",
      "sec-ch-ua" : ""Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"",
      "sec-ch-ua-mobile" : "?0",
      "sec-ch-ua-platform" : ""macOS"",
      "sec-fetch-site" : "cross-site",
      "sec-fetch-mode" : "no-cors",
      "sec-fetch-user" : null,
      "sec-fetch-dest" : "image",
      "accept-language" : "es-ES,es;q=0.9",
      "ref_id" : "natusfera-72436",
      "origin" : "natusfera",
      "ua" : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36",
      "ip" : "81.44.51.171",
    - "browser" : {
          "name" : "Chrome",
          "version" : "105.0.0.0",
          "major" : "105",
      },
    - "engine" : {
          "name" : "Blink",
          "version" : "105.0.0.0",
      },
    - "os" : {
          "name" : "Mac OS",
          "version" : "10.13.6",
      },
    - "device" : {
          "vendor" : null,
          "model" : null,
          "type" : null,
      },
    - "cpu" : {
          "architecture" : null,
      },
      "url" : "https://natusfera.gbif.es/attachments/local_photos/files/94933/large/84293A03-5776-400B-861E-7CAFBE360B40-large.jpg?1528301843",
      "as" : "AS3352 TELEFONICA DE ESPANA S.A.U.",
      "city" : "Madrid",
      "country" : "Spain",
      "countryCode" : "ES",
      "isp" : "Telefonica de Espana SAU",
      "lat" : 40.4163,
      "lon" : -3.6934,
      "org" : "RIMA (Red IP Multi Acceso)",
      "query" : "81.44.51.171",
      "region" : "MD",
      "regionName" : "Madrid",
      "status" : "success",
      "timezone" : "Europe/Madrid",
      "zip" : "28049",
      "lang" : "es-ES",
      "ref_user" : "338",
      "hostname" : "cos4bio.eu",
      "created_at" : "2022-10-21T09:34:53.247Z",
  },
```

Figure 24. A log example. Represent a display of an observation.

- https://cos4bio.eu/api/logs/agg

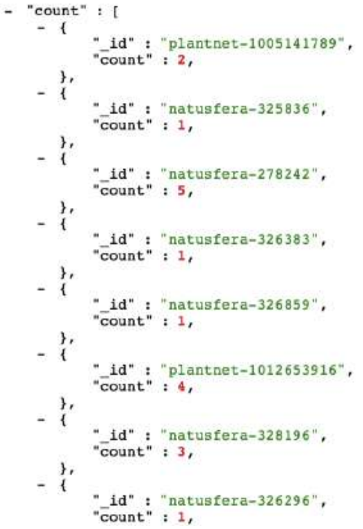
  This response contains an aggregated list of all the records in the logs document. Following is a list of aggregations:



```
{
   + "count" : [...] (294 hidden elements),
   + "countries" : [...] (23 hidden elements),
   + "browser" : [...] (7 hidden elements),
   + "os" : [...] (7 hidden elements),
   + "ip" : [...] (92 hidden elements),
   + "lang" : [...] (17 hidden elements),
   + "origins" : [...] (5 hidden elements),
   + "hosts" : [...] (5 hidden elements)
}
```

Figure 25. Reverse proxy DUNS architecture.

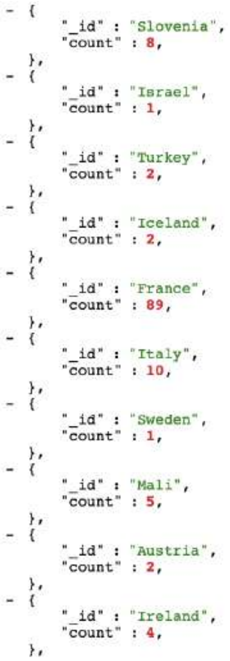
- **Count**, we can see how many times an observation has been consulted:



```
- "count" : [
   - {
         "_id" : "plantnet-1005141789",
         "count" : 2,
     },
   - {
         "_id" : "natusfera-325836",
         "count" : 1,
     },
   - {
         "_id" : "natusfera-278242",
         "count" : 5,
     },
   - {
         "_id" : "natusfera-326383",
         "count" : 1,
     },
   - {
         "_id" : "natusfera-326859",
         "count" : 1,
     },
   - {
         "_id" : "plantnet-1012653916",
         "count" : 4,
     },
   - {
         "_id" : "natusfera-328196",
         "count" : 3,
     },
   - {
         "_id" : "natusfera-326296",
         "count" : 1,
```

Figure 26. Aggregate stats number of queries by Observation.

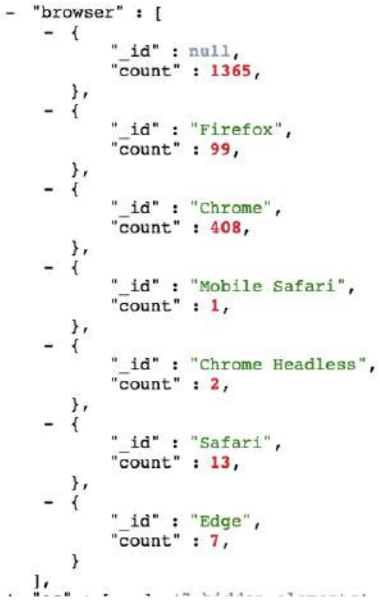
- **Countries**, it shows us how many queries are made by country:



```
- {
      "_id" : "Slovenia",
      "count" : 8,
  },
- {
      "_id" : "Israel",
      "count" : 1,
  },
- {
      "_id" : "Turkey",
      "count" : 2,
  },
- {
      "_id" : "Iceland",
      "count" : 2,
  },
- {
      "_id" : "France",
      "count" : 89,
  },
- {
      "_id" : "Italy",
      "count" : 10,
  },
- {
      "_id" : "Sweden",
      "count" : 1,
  },
- {
      "_id" : "Mali",
      "count" : 5,
  },
- {
      "_id" : "Austria",
      "count" : 2,
  },
- {
      "_id" : "Ireland",
      "count" : 4,
  },
```

Figure 27. Aggregate stats by Country.

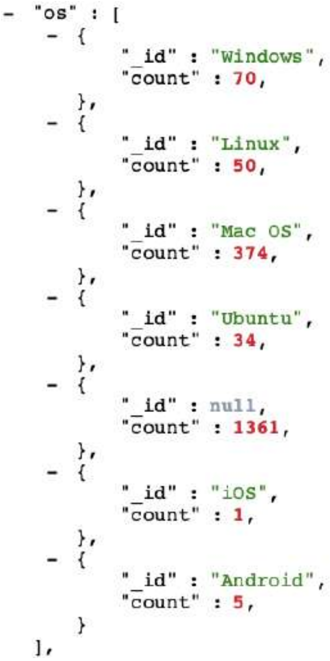
Browser, it shows us how many queries are made by browser:



```
- "browser" : [
   - {
          "_id" : null,
          "count" : 1365,
     },
   - {
          "_id" : "Firefox",
          "count" : 99,
     },
   - {
          "_id" : "Chrome",
          "count" : 408,
     },
   - {
          "_id" : "Mobile Safari",
          "count" : 1,
     },
   - {
          "_id" : "Chrome Headless",
          "count" : 2,
     },
   - {
          "_id" : "Safari",
          "count" : 13,
     },
   - {
          "_id" : "Edge",
          "count" : 7,
     }
  ],
```

Figure 28. Aggregate stats by browser.

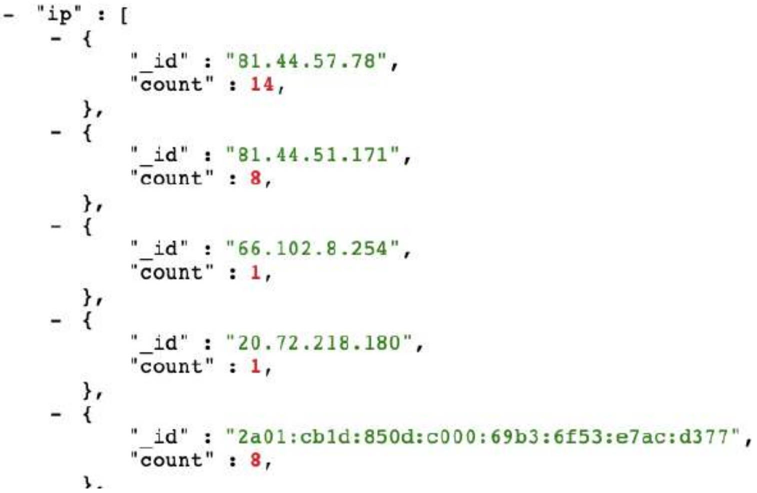
- **Os**, it shows us how many queries are made by operating system:



```
- "os" : [
   - {
          "_id" : "Windows",
          "count" : 70,
     },
   - {
          "_id" : "Linux",
          "count" : 50,
     },
   - {
          "_id" : "Mac OS",
          "count" : 374,
     },
   - {
          "_id" : "Ubuntu",
          "count" : 34,
     },
   - {
          "_id" : null,
          "count" : 1361,
     },
   - {
          "_id" : "iOS",
          "count" : 1,
     },
   - {
          "_id" : "Android",
          "count" : 5,
     }
  ],
```

Figure 29 Aggregate stats by OS.

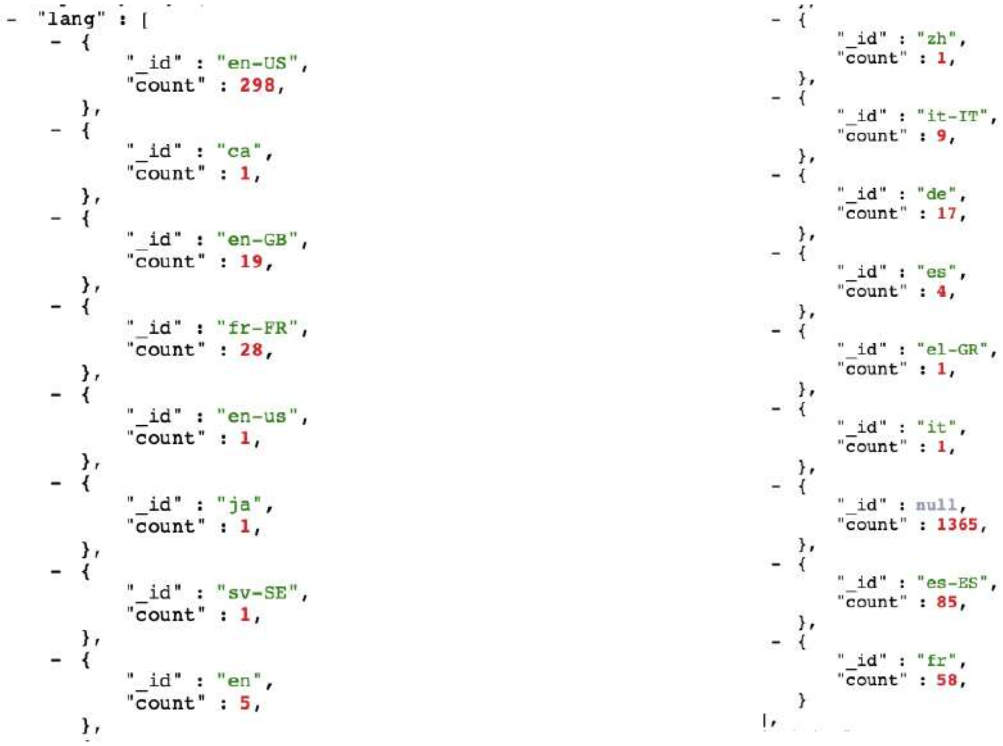
- **Ip**, it shows us how many queries are made by IP address:



```
- "ip" : [
   - {
          "_id" : "81.44.57.78",
          "count" : 14,
     },
   - {
          "_id" : "81.44.51.171",
          "count" : 8,
     },
   - {
          "_id" : "66.102.8.254",
          "count" : 1,
     },
   - {
          "_id" : "20.72.218.180",
          "count" : 1,
     },
   - {
          "_id" : "2a01:cb1d:850d:c000:69b3:6f53:e7ac:d377",
          "count" : 8,
     },
```

Figure 30. Aggregate stats by ip.

- **Lang**, it shows us how many queries are made by language:

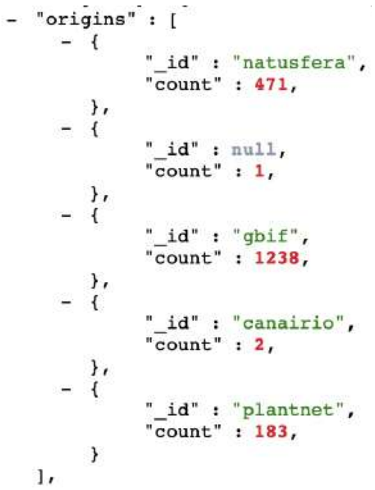


```
- "lang" : [
   - {
          "_id" : "en-US",
          "count" : 298,
     },
   - {
          "_id" : "ca",
          "count" : 1,
     },
   - {
          "_id" : "en-GB",
          "count" : 19,
     },
   - {
          "_id" : "fr-FR",
          "count" : 28,
     },
   - {
          "_id" : "en-us",
          "count" : 1,
     },
   - {
          "_id" : "ja",
          "count" : 1,
     },
   - {
          "_id" : "sv-SE",
          "count" : 1,
     },
   - {
          "_id" : "en",
          "count" : 5,
     },
```

```
   - {
          "_id" : "zh",
          "count" : 1,
     },
   - {
          "_id" : "it-IT",
          "count" : 9,
     },
   - {
          "_id" : "de",
          "count" : 17,
     },
   - {
          "_id" : "es",
          "count" : 4,
     },
   - {
          "_id" : "el-GR",
          "count" : 1,
     },
   - {
          "_id" : "it",
          "count" : 1,
     },
   - {
          "_id" : null,
          "count" : 1365,
     },
   - {
          "_id" : "es-ES",
          "count" : 85,
     },
   - {
          "_id" : "fr",
          "count" : 58,
     }
],
```

Figure 31. Aggregate stats by language.

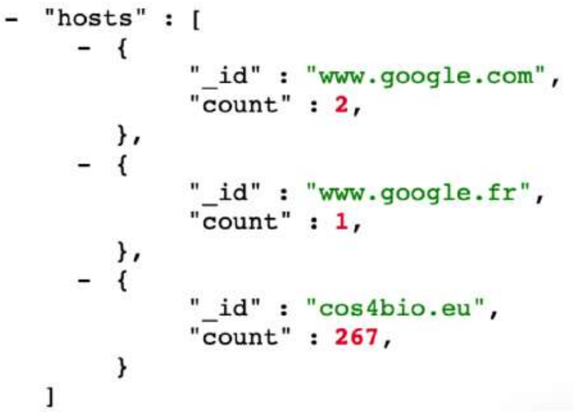
- **Origin**, how many times Observations from each Citizen Observatory integrated into Cos4Bio have been consulted:



```
- "origins" : [
    - {
              "_id" : "natusfera",
              "count" : 471,
      },
    - {
              "_id" : null,
              "count" : 1,
      },
    - {
              "_id" : "gbif",
              "count" : 1238,
      },
    - {
              "_id" : "canairio",
              "count" : 2,
      },
    - {
              "_id" : "plantnet",
              "count" : 183,
      }
  ],
```

Figure 32. Aggregate stats by origin.

- **Host**, the number of queries made by each host, grouped by Observatory.:



```
- "hosts" : [
    - {
              "_id" : "www.google.com",
              "count" : 2,
      },
    - {
              "_id" : "www.google.fr",
              "count" : 1,
      },
    - {
              "_id" : "cos4bio.eu",
              "count" : 267,
      }
  ]
```

Figure 33. Aggregate usage stats by host.

- https://cos4bio.eu/api/logs/agg.svg

The aggregated data of all the records of the logs document is returned as an image in SVG format. Here's an example:

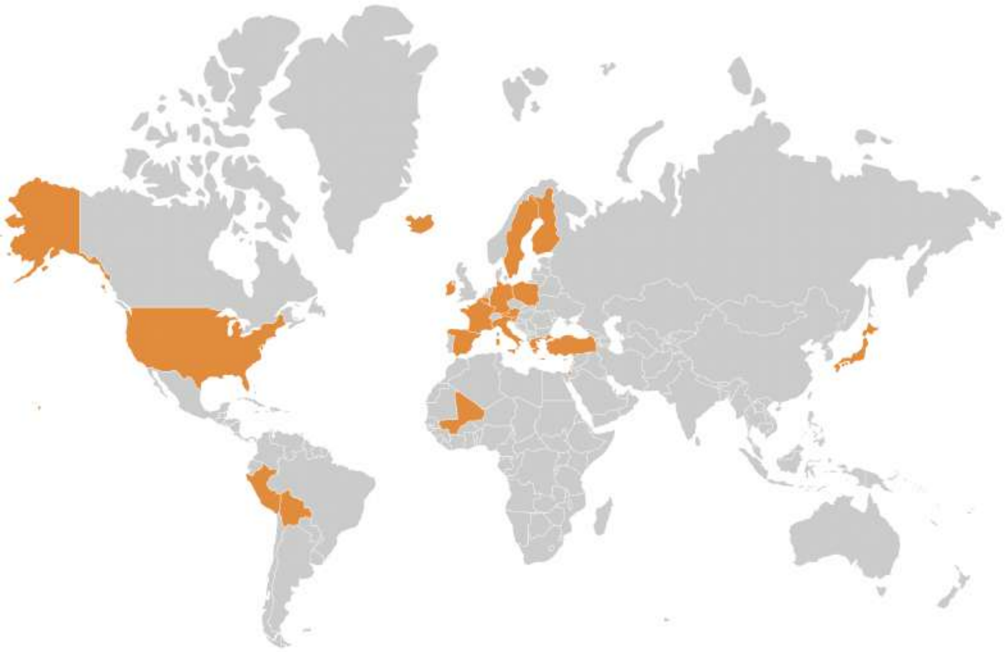


Figure 34. Usage stat by country localization.

- https://cos4bio.eu/api/logs/agg.html

A response in HTML format with an image in SVG format, interacting with mouse events through javascript. Here's an example:

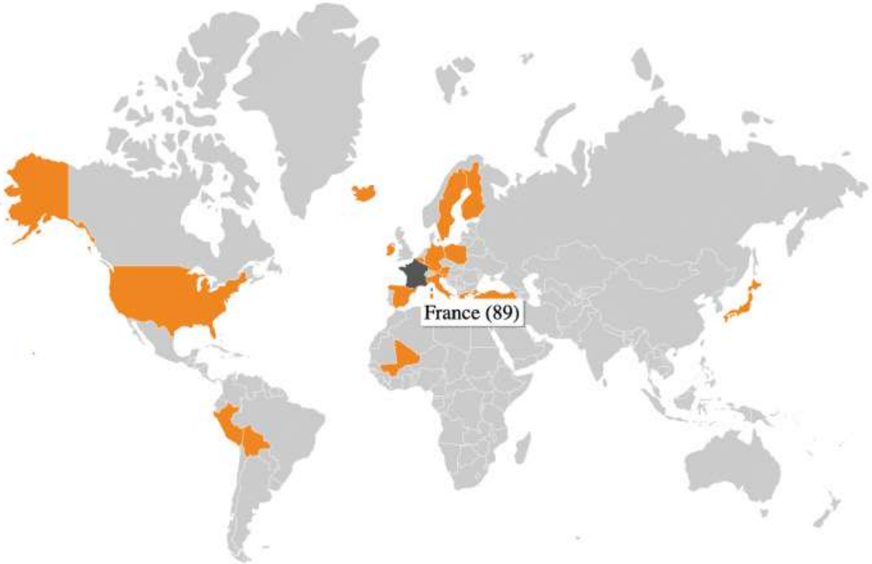
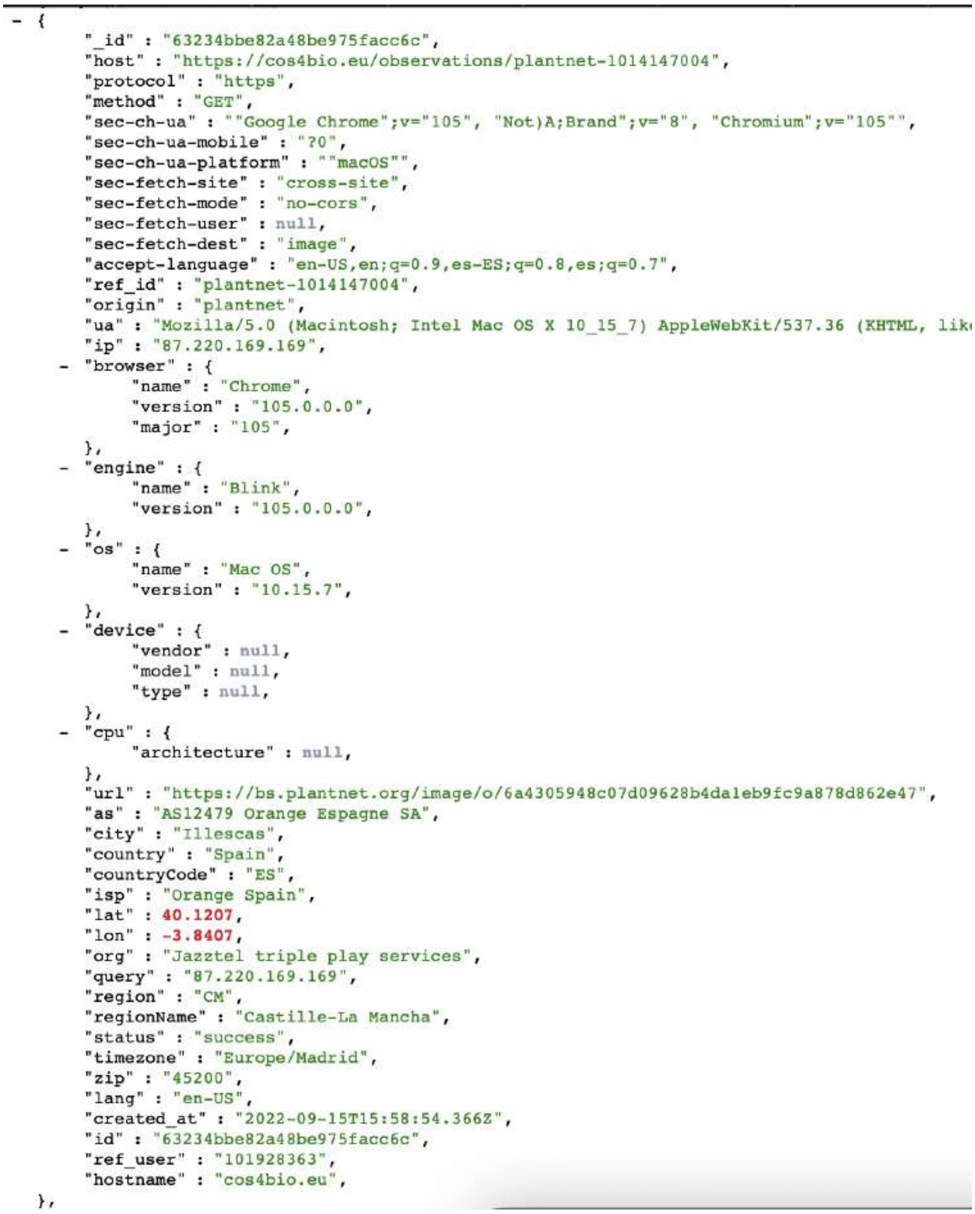


Figure 35. Usage stat by country localization.

- https://cos4duns.eu/api/logs/all?origin=plantnet

The record shows all the observations from a given "origin", in this case from a Citizen Observatory integrated with Cos4Bio. Using the next API call, Pl@ntNet could also inform its users about the use of their observations by using this call.

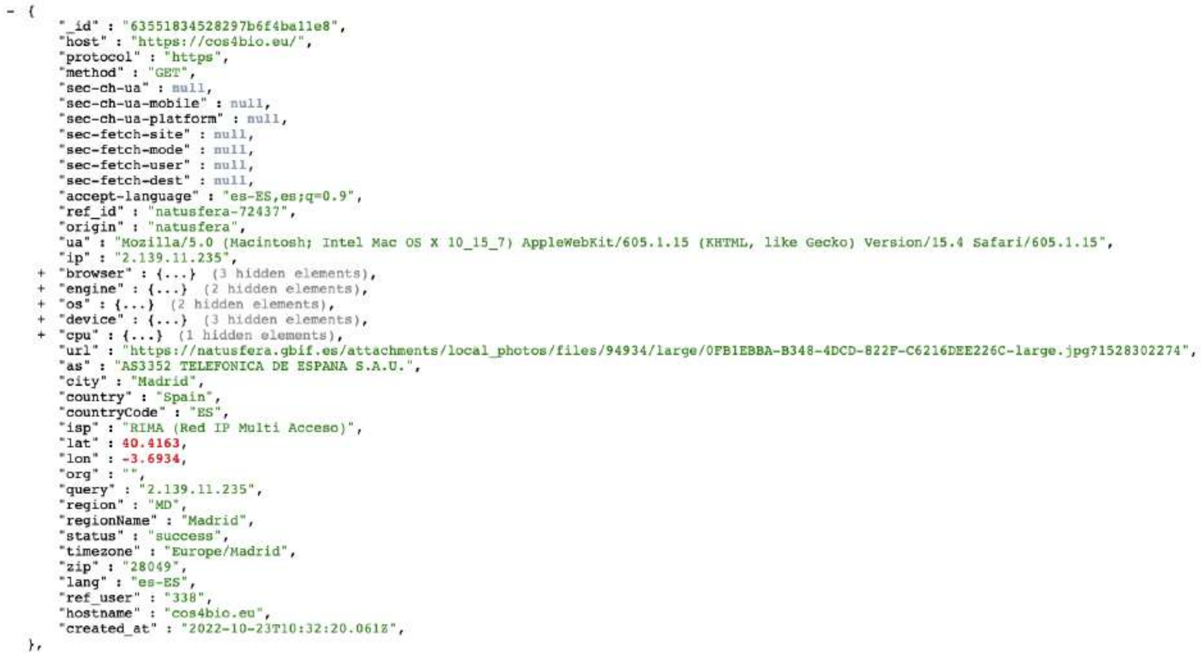


```
- {
        "_id" : "63234bbe82a48be975facc6c",
        "host" : "https://cos4bio.eu/observations/plantnet-1014147004",
        "protocol" : "https",
        "method" : "GET",
        "sec-ch-ua" : ""Google Chrome";v="105", "Not)A;Brand";v="8", "Chromium";v="105"",
        "sec-ch-ua-mobile" : "?0",
        "sec-ch-ua-platform" : ""macOS"",
        "sec-fetch-site" : "cross-site",
        "sec-fetch-mode" : "no-cors",
        "sec-fetch-user" : null,
        "sec-fetch-dest" : "image",
        "accept-language" : "en-US,en;q=0.9,es-ES;q=0.8,es;q=0.7",
        "ref_id" : "plantnet-1014147004",
        "origin" : "plantnet",
        "ua" : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, lik
        "ip" : "87.220.169.169",
    - "browser" : {
            "name" : "Chrome",
            "version" : "105.0.0.0",
            "major" : "105",
        },
    - "engine" : {
            "name" : "Blink",
            "version" : "105.0.0.0",
        },
    - "os" : {
            "name" : "Mac OS",
            "version" : "10.15.7",
        },
    - "device" : {
            "vendor" : null,
            "model" : null,
            "type" : null,
        },
    - "cpu" : {
            "architecture" : null,
        },
        "url" : "https://bs.plantnet.org/image/o/6a4305948c07d09628b4da1eb9fc9a878d862e47",
        "as" : "AS12479 Orange Espagne SA",
        "city" : "Illescas",
        "country" : "Spain",
        "countryCode" : "ES",
        "isp" : "Orange Spain",
        "lat" : 40.1207,
        "lon" : -3.8407,
        "org" : "Jazztel triple play services",
        "query" : "87.220.169.169",
        "region" : "CM",
        "regionName" : "Castille-La Mancha",
        "status" : "success",
        "timezone" : "Europe/Madrid",
        "zip" : "45200",
        "lang" : "en-US",
        "created_at" : "2022-09-15T15:58:54.366Z",
        "id" : "63234bbe82a48be975facc6c",
        "ref_user" : "101928363",
        "hostname" : "cos4bio.eu",
  },
```

Figure 36. Usage stats from one Observation Pl@ntNet.

- https://cos4duns.eu/api/logs/all?ref_user=101928363

A reference_user parameter contains an identifier that shows how the observations were used by a particular user.



```
- {
      "_id" : "63551834528297b6f4ba11e8",
      "host" : "https://cos4bio.eu/",
      "protocol" : "https",
      "method" : "GET",
      "sec-ch-ua" : null,
      "sec-ch-ua-mobile" : null,
      "sec-ch-ua-platform" : null,
      "sec-fetch-site" : null,
      "sec-fetch-mode" : null,
      "sec-fetch-user" : null,
      "sec-fetch-dest" : null,
      "accept-language" : "es-ES,es;q=0.9",
      "ref_id" : "natusfera-72437",
      "origin" : "natusfera",
      "ua" : "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/15.4 Safari/605.1.15",
      "ip" : "2.139.11.235",
    + "browser" : {...}  (3 hidden elements),
    + "engine" : {...}  (2 hidden elements),
    + "os" : {...}  (2 hidden elements),
    + "device" : {...}  (3 hidden elements),
    + "cpu" : {...}  (1 hidden elements),
      "url" : "https://natusfera.gbif.es/attachments/local_photos/files/94934/large/0FB1EBBA-B348-4DCD-822F-C6216DEE226C-large.jpg?1528302274",
      "as" : "AS3352 TELEFONICA DE ESPANA S.A.U.",
      "city" : "Madrid",
      "country" : "Spain",
      "countryCode" : "ES",
      "isp" : "RIMA (Red IP Multi Acceso)",
      "lat" : 40.4163,
      "lon" : -3.6934,
      "org" : "",
      "query" : "2.139.11.235",
      "region" : "MD",
      "regionName" : "Madrid",
      "status" : "success",
      "timezone" : "Europe/Madrid",
      "zip" : "28049",
      "lang" : "es-ES",
      "ref_user" : "338",
      "hostname" : "cos4bio.eu",
      "created_at" : "2022-10-23T10:32:20.061Z",
  },
```

Figure 37. Usage stats from one specific user.

The model for Logs consist of data that comes from HTTP User-Agent, Geolocation information about IP and others mapped attributes:

- **Term**: It is a term that is part of the answer.
- **Value**: The possible values that the terms could have.
- **Source**: The term in more detail.
- **Description**: An explanation of each term.

Term	Value	Source	Description
host	https://cos4bio.eu/	Host	The Host request header specifies the host and port number of the server to which the request is being sent. If no port is included, the default port for the service requested is implied (e.g., 443 for an HTTPS URL, and 80 for an HTTP URL).
sec-ch-ua	"Google Chrome";v="105", "Not)A;Brand";v= "8", "Chromium";v="105"	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-CH-UA	The Sec-CH-UA header provides the brand and significant version for each brand associated with the browser in a comma-separated list.
sec-ch-ua-mobile	null	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-CH-UA-Mobile	The Sec-CH-UA-Mobile user agent client hint request header indicates whether the browser is on a mobile device. It can also be used by a desktop browser to indicate a preference for a "mobile" user experience.
sec-ch-ua-platform	macOS	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-CH-UA-Platform	The Sec-CH-UA-Platform user agent client hint request header provides the platform or operating system on which the user agent is running. For example: "Windows", "Android" or “macOs”.
sec-fetch-site	cross-site	https://developer.mozilla.org/e	The Sec-Fetch-Site fetch metadata request header indicates the

		n-US/docs/Web/HTTP/Headers/Sec-Fetch-Site	relationship between a request initiator's origin and the origin of the requested resource.
sec-fetch-mode	no-cors	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-Fetch-Mode	The Sec-Fetch-Mode fetch metadata request header indicates the mode of the request.
sec-fetch-user	null	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-Fetch-User	The Sec-Fetch-User fetch metadata request header is only sent for requests initiated by user activation, and its value will always be ?1.
sec-fetch-dest	image	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Sec-Fetch-Dest	The Sec-Fetch-Dest fetch metadata request header indicates the request's destination. That is the initiator of the original fetch request, which is where (and how) the fetched data will be used. This allows servers determine whether to service a request based on whether it is appropriate for how it is expected to be used. For example, a request with an audio destination should request audio data, not some other type of resource (for example, a document that includes sensitive user information).
accept-language	es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Accept-Language	The Accept-Language request HTTP header indicates the natural language and locale that the client prefers. The server uses content negotiation to select one of the proposals and informs the client of the choice with the Content-Language response header. Browsers set required values for this header according to their active user interface language. Users rarely change it, and such changes

			are not recommended because they may lead to fingerprinting.
ref_id	natusfera-328175		
origin	natusfera		
ua	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:105.0) Gecko/20100101 Firefox/105.0	https://www.w3.org/WAI/UA/work/wiki/Definition_of_User_Agent	The User-Agent request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting
ip	81.44.51.171		IP information comes from the HTTP request
browser	{"name":"Firefox","version":"105.0","major":"105"}		Information about browsers is extracted from the User-Agent request header.
engine	{"name":"Gecko","version":"105.0"}		Information about browser engines is extracted from the User-Agent request header.
os	{"name":"Mac OS","version":"10.13"}		Information about Operating Systems is extracted from the User-Agent request header.
device	{"vendor":null,"model":null,"type":null}		When it is available, this is information about the model of the device, and is extracted from the User-Agent request header.
cpu	{"architecture":null}		When it is available, this is information about the architecture of

			the device, and is extracted from the User-Agent request header.
url	https://natusfera.gbif.es/attachments/local_photos/files/419517/large/DSCN2706.JPG?1651699320		Uri that contains the observation.
as	AS3352 TELEFONICA DE ESPANA S.A.U.		Network information.
city	Madrid		City name.
country	Spain		Country name.
countryCode	ES		Country code.
isp	Telefonica de Espana SAU		Internet Service Provider.
lat	40.4163		Latitude.
long	-3.6934		Longitude.
org	RIMA (Red IP Multi Acceso)		
query	81.44.51.171		IP address that does the query.
regionName	Madrid		The name of the region.
status	success		Status of the query.
timezone	Europe/Madrid		Time zone.
zip	28049		Postal code.

lang	es-ES		Language.
ref_user	15372		Represents the user for the different portals.
hostname	cos4bio.eu		Represents the host name from where the query was made.
created_at	2022-10-21T09:43:24.861Z		Creation date.

Table 4: Mapping terms table.

4.4. Code

At this moment all the code is available in a public GitHub repository: https://github.com/Bineo-Consulting/DUNS

5. Future work

Once the implementation has been tested and validated by experts and the DUNS service has been published in EOSC, future work will focus on several aspects that we detail below:

- Continuous maintenance of the system.
- Continuous monitoring of the usage stats generated by DUNS.
- Integration of new Citizen Observatories through Cos4Bio.
- To be in contact with the Scientific Community.
- Include more graphics.
- Carry out communication tasks to publicize the service hand in hand with the Connect Group.
- Resolve possible incidents that may occur.
- Keep collecting experts' feedback.
- Integrate DUNS with Cos4Env and other environmental observatories.
- Implement a tokenization system that allows integrate DUNS directly.
- Think about adding new features.
- Plan and think about ways of sustainability for the future.

6. Conclusions

As long as we find the right support, and with the rest of the services we have developed, DUNS can become a great benchmark within the EOSC Marketplace. We believe that DUNS has all the potential necessary to value citizen observations of biodiversity and the environment. In addition, DUNS opens new paths for Citizen Observatories that want to integrate this service, since they will have more insight into the use of user observations, which will enable them to provide more visibility to their users, as well as implement gamification mechanisms to keep their platforms challenging. It will now be possible to see the observers' efforts rewarded in some way.

As well as these aspects, DUNS offers a number of other advantages:

- Collaboration with other citizen observatories.
- Integration of other services of Cos4Cloud, such as Cos4Bio.
- Application of the defined Agile Methodology.
- Implementation of new API.
- Continuous learning from many experts thanks to Co-Design meetings.
- Improvements in each cycle through technical meetings every 15 days.
- Join the EOSC community.

These factors have made DUNS one of the most transversal services we have implemented, because it has been integrated with other services implemented during the Cos4Cloud project, such as Cos4Bio and Natusfera improvements.

Through DUNS, the Cos4Cloud services are enhanced, as well as Citizen Observatories, but most importantly, the citizens themselves are empowered by the knowledge of how their observations are used.

Glossary

Term	Description
DUNS	Data Usage Notification Service.
Cos4Bio	It's the name that we have defined for the service of Expert Portal for Biodiversity data validation.
DOI	A DOI (Digital Object Identifier) is a unique and never-changing string assigned to online(journal) articles, books, and other works. DOIs make it easier to retrieve works, which is why citation styles, like APA and MLA Style, recommend including them in citations.
LSID	Life Science Identifiers are a way to name and locate pieces of information on the web. Essentially, an LSID is a unique identifier for some data, and the LSID protocol specifies a standard way to locate the data (as well as a standard way of describing that data). They are a little like DOIs used by many publishers.
URI	A Uniform Resource Identifier (URI) is a character sequence that identifies a logical (abstract) or physical resource -- usually, but not always, connected to the internet. A URI distinguishes one resource from another. URIs enable internet protocols to facilitate interactions between and among these resources. The strings of characters incorporated in a URI serve as identifiers, such as a scheme name and a file path.
Expert Platform	Prototype that integrates data from various citizen observatories, allowing the experts who access it to perform various operations such as: searches, downloads, identifications, comments, with the benefit of reducing the investment of time when finding data on which to investigate, and that allows increasing the degree of participation of experts in the processes of identification of the observations made by citizens.
Citizen Observatory (CO)	Web Portal that allows the interaction between Citizens and Scientists in obtaining data, be it data on flora, fauna, odors, temperatures,

	precipitation measurements, emissions ..., and that involves an identification process , validation, quality and study.
Observation	It is the element observed by a person or device, be it Flora, Fauna or an Environmental Variable.
Identification	When the observations are not known by the citizens, they are presented in a random state, not identified, since they lack the scientific knowledge to be able to define the observation with precision. For example: In the case of a magpie, it would be identified when a user identified it as Pica Pica, and defined it through its correct scientific name.
Casual status	An observation is accidental when it has received an identification, but this has not yet been validated by any expert in its corresponding Citizen Observatory.
Research grade status	An observation is scientific grade, when it has been identified and in turn has been validated by an expert.
Citizen	Person who participates in Citizen Observatories sharing their observations related to biodiversity or environment context that can be used by the scientific community.
Expert	Person with sufficient knowledge to be able to carry out identifications or scientists dedicated to the world of Biodiversity or the Environment.
Standard	It is a common framework validated by the community, in this case it applies to the use of communication protocols, the representation of information, the visualization of images and procedures.
Interoperability	We can define it, within the project, as the layer that allows the data from the different Citizen Observatories to coexist in the same environment. This layer allows the Expert Portal to understand the information in each portal, performing a mapping task and showing it to the end user in a unified and standardized way.
FAIR	FAIR are data which meet principles of findability, accessibility, interoperability, and reusability. A March 2016 publication by a consortium of scientists and organizations specified the "FAIR Guiding Principles for scientific data management and stewardship" in

	Scientific Data, using FAIR as an acronym and making the concept easier to discuss.
Actor	it is something with behaviour, such as a person, identified by a role, computer system or organization.
Scenario	Specific sequence of actions and interactions between the actors and the system under study.
NoSQL	NoSQL databases (aka "not only SQL") are non-tabular databases and store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.

Annexe

Cos4Cloud TRL Calculation

Service:
DUNS: Data Usage Notification Service

TRL 1: Basic principles observed

Question	Criterion fulfilled (yes/no)
Is there a research hypothesis defined guiding the development of the service?	yes
Are the basic algorithms needed for the implementation defined?	yes
Is there a document (specification, publication) describing the ideas of the service?	yes

TRL 2: Technology concept formulated

Question	Criterion fulfilled (yes/no)
Is there an initial design description of the service?	yes
Did you define the users stories to be implemented?	yes
Did you identify the research and development activities necessary for developing your service?	yes
Did you perform first tests with example data?	yes

TRL 3: Experimental proof of concept

Question	Criterion fulfilled (yes/no)
Did you define the main components of your service?	yes
Is there an overview design of your service available (e.g. component diagram)?	yes
Did you define performance criteria (e.g., number of users, amount of data) for your service?	yes
Did you establish a test environment for component tests (e.g., tools for conducting unit tests)?	yes

Are the individual components of your service implemented in a preliminary version?	yes
Did you test the core functions of the service?	yes
Did you document your first tests?	yes

TRL 4

Question	Criterion fulfilled (yes/no)
Is there a detailed design of the components of the service available?	yes
Did you complete the first running on an internal system (development machine)?	yes
Did you define the environment in which the service will be operated?	yes
Did you successfully complete the testing of the integrated components?	yes
Are the test results of the integrated components documented?	yes

TRL 5: Technology validated in relevant environment

Question	Criterion fulfilled (yes/no)
Did you determine the requirements for scaling your service (e.g. from test users to a public audience)?	yes
Did you establish a testing environment similar to the later operational environment?	yes
Did you successfully demonstrate your implementation in a testing environment?	yes
Did you conduct tests with consortium partners?	yes
Are the test results from the test environment documented?	yes

TRL 6: Technology demonstrated in relevant environment

Question	Criterion fulfilled (yes/no)
Are the system requirements finalised?	yes
Is the operating environment fully defined?	yes
Are the necessary implementations of a prototype that takes into account the requirements of the operational environment?	yes
Did you conduct tests with selected external users (e.g. as part of hackathons)?	yes

TRL 7: System prototype demonstration in operational environment

Question	Criterion fulfilled (yes/no)
Is there a fully integrated prototype available that was demonstrated in an operational environment?	yes
Did you verify the performance of the system?	
Did you conduct tests within a broader community with previously unknown users?	yes
Is all software testing completed?	yes
Does your service provide value on its own?	yes
Is your documentation available in the English language?	yes
Did you specify privacy statements, terms of use and Service Level Agreements?	yes
Is there a help desk available?	yes
Is there a plan established for ensuring regular updates of your service?	yes

TRL 8: System complete and qualified

Question	Criterion fulfilled (yes/no)
Did you define a life cycle for your software? Is there a clear plan on how to handle the end of life?	yes
Did you make your service available via the EOSC portal?	yes
Did you fix all known open issues/bugs?	yes
Is there a full documentation of your service available?	yes
Has the qualification test on EOSC been passed?	yes

TRL 9: Actual system proven in operational environment

Question	Criterion fulfilled (yes/no)
Is your service operational in the EOSC?	yes
Is there continuous support for your service available?	yes
Did you finalise all related documentation?	yes
Does your software meet all specified requirements?	yes
Are all user stories successfully implemented?	yes