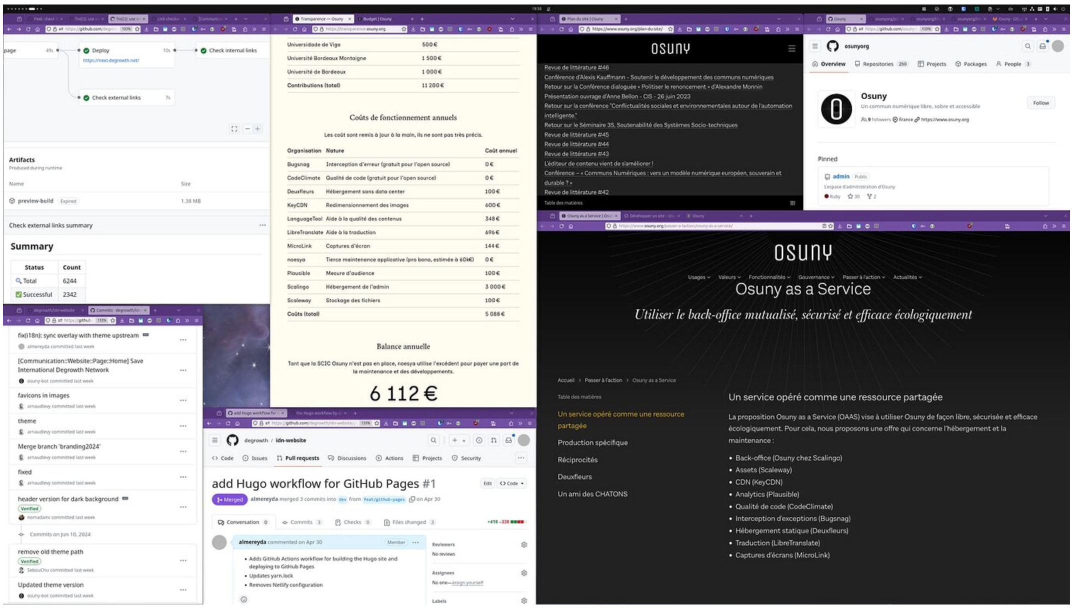
Working with Osuny out in the open: a replication study to increase the resiliency of the computational Common (2/5)

yala 1 Juin 22, 2024, 1:06

This is the second part in a series of posts around the Ergonomy of development experience and common reproducibility questions around maintaining degrowth.net, which, after investigating the modes of production of the Osuny application, shifts the view to practical experiences gained when using it. This happens by mobilising its means of production to achieve a desired outcome.

The intent often differs from who you ask. For a community of practice, it may be to publish their peer reviewed website content, for an editor it is to apply a small change to create a new revision of the content, for a designer it is to bring in changes to the visual appeal of the generated interfaces, for an application developer it is to maintain consistency between model and side-effects and for a distributed platforms engineer it is to isolate side-effects to maintain eventual consistency between multiple loosely-coupled systems. They all will give you different reasons for why their intent is a legitimate change within the frame of the overall, say holistic system, which we conventionally call Osuny.



Summary

Status	Count
🔍 Total	6244
✅ Successful	2342

Universidade de Vigo	500 €
Université Bordeaux Montaigne	1 500 €
Université de Bordeaux	1 000 €
Contributions (total)	11 200 €

Coûts de fonctionnement annuels

Les coût sont remis à jour à la main, ils ne sont pas très précis.

Organisation	Nature	Coût annuel
Bugsnag	Interception d'erreur (gratuit pour l'open source)	0 €
CodeClimate	Qualité de code (gratuit pour l'open source)	0 €
Deuxfleurs	Hébergement sans data center	100 €
KeyCDN	Redimensionnement des images	600 €
LanguageTool	Aide à la qualité des contenus	348 €
LibreTranslate	Aide à la traduction	696 €
MicroLink	Captures d'écran	144 €
noesya	Tierce maintenance applicative (pro bono, estimée à 60k€)	0 €
Plausible	Mesure d'audience	100 €
Scalingo	Hébergement de l'admin	3 000 €
Scaleway	Stockage des fichiers	100 €
Coûts (total)		5 088 €

Balance annuelle

Tant que la SCIC Osuny n'est pas en place, noesya utilise l'excédent pour payer une part de la maintenance et des développements.

6 112 €

Revue de littérature #46
Conférence d'Alexis Kauffmann - Soutenir le développement des communs numériques
Retour sur la Conférence dialoguée « Politiser le renoncement » d'Alexandre Monnin
Présentation ouvrage d'Anne Bellon - CIS - 26 juin 2023
Retour sur la conférence "Conflictualités sociales et environnementales autour de l'automation intelligente."
Retour sur le Séminaire 3S, Soutenabilité des Systèmes Socio-techniques
Revue de littérature #45
Revue de littérature #44
Revue de littérature #43
L'éditeur de contenu vient de s'améliorer !
Conférence – « Communs Numériques : vers un modèle numérique européen, souverain et durable ? »
Revue de littérature #42

Un commun numérique libre, sobre et accessible

9 followers France https://www.osuny.org

Pinned

admin Public

L'espace d'administration d'Osuny

Ruby 30 2

Osuny as a Service

Utiliser le back-office mutualisé, sécurisé et efficace écologiquement

Accueil > Passer à l'action > Osuny as a Service

Table des matières

Un service opéré comme une ressource partagée
Production spécifique
Réciprocités
Deuxfleurs
Un ami des CHATONS

Un service opéré comme une ressource partagée

La proposition Osuny as a Service (OAAS) vise à utiliser Osuny de façon libre, sécurisée et efficace écologiquement. Pour cela, nous proposons une offre qui concerne l'hébergement et la maintenance :

- Back-office (Osuny chez Scalingo)
- Assets (Scaleway)
- CDN (KeyCDN)
- Analytics (Plausible)
- Qualité de code (CodeClimate)
- Interception d'exceptions (Bugsnag)
- Hébergement statique (Deuxfleurs)
- Traduction (LibreTranslate)
- Captures d'écrans (MicroLink)

fix(i18n): sync overlay with theme upstream
almereyda committed last week

[Communication::Website::Page::Home] Save International Degrowth Network
osuny-bot committed last week

favicons in images
arnaudlevy committed last week

theme
arnaudlevy committed last week

Merge branch 'branding2024'
arnaudlevy committed last week

fixed
arnaudlevy committed last week

header version for dark background
Verified
nomadami committed last week

Commits on Jun 10, 2024

remove old theme path
Verified
SebouChu committed last week

Updated theme version
osuny-bot committed last week

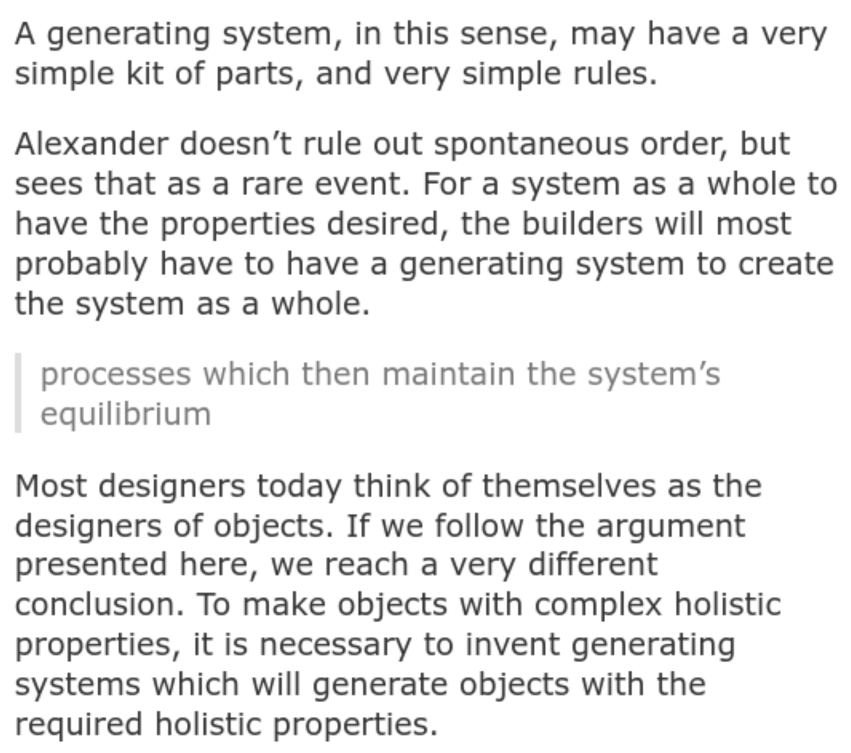
add Hugo workflow for GitHub Pages #1

Merged almereyda merged 3 commits into dev from feat/github-pages on Apr 30

almereyda commented on Apr 30

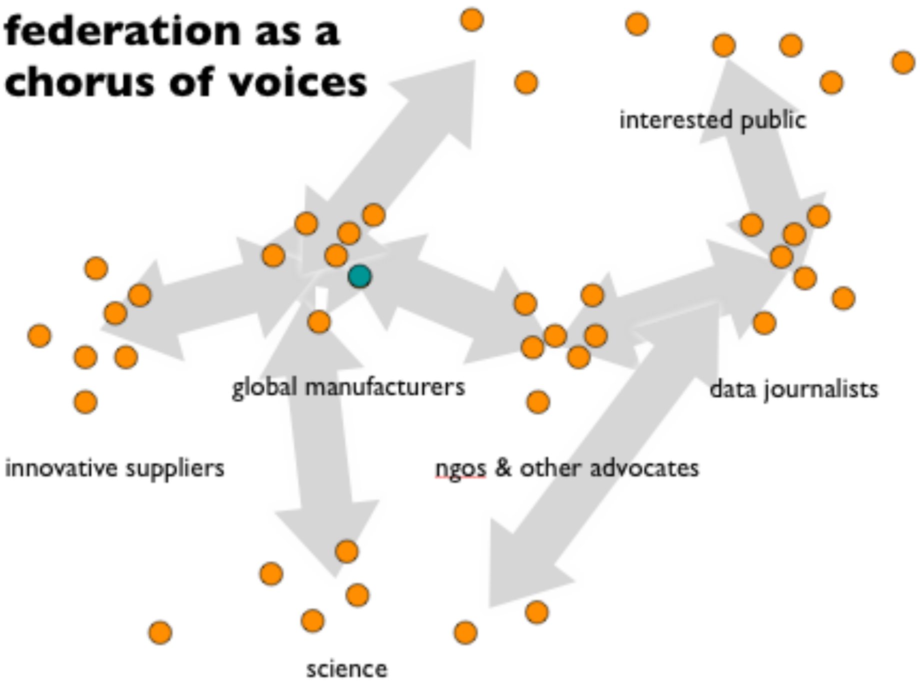
- Adds GitHub Actions workflow for building the Hugo site and deploying to GitHub Pages
- Updates yarn.lock
- Removes Netlify configuration

When pursuing the multi-modal multi-vector discussion in the multi-dimensional discursive space that makes up the whole of conversations about Osuny, we give acknowledgement to the fact, that Software is a continuing conversation that perpetually prefigures our day-to-day interaction. In applying holistic frameworks, such as the Numérique d'Intérêt Général reference frame or an Architectural design theory, we are making use of conceptual frameworks, against which we can compare the empirically gathered evidence and test the implementation in our hypothesis-driven experiment.



> A generating system, in this sense, may have a very simple kit of parts, and very simple rules.
>
> Alexander doesn't rule out spontaneous order, but sees that as a rare event. For a system as a whole to have the properties desired, the builders will most probably have to have a generating system to create the system as a whole.
>
> > processes which then maintain the system's equilibrium
>
> Most designers today think of themselves as the designers of objects. If we follow the argument presented here, we reach a very different conclusion. To make objects with complex holistic properties, it is necessary to invent generating systems which will generate objects with the required holistic properties.
>
> Excerpt from the bottom of **Architectural design theory excerpts**

Software is a conversation of a chorus of voices[1]. Allow me to add to the conversation a few perspectives from the view of a platform engineer, who likes to run all components of the platform with Free Software, in order to keep all necessary parts in the computational common. First we have to know which parts there are.

> 
>
>
> Chorus of voices. An analogy reused from [Chorus of Voices]

▶ [1] Some even have a data model for it and call it Conversation for Action.

We must leave the historical discourse analysis to anthropologists who make the effort to retrace the arguments passed along between participants in the Osuny community, unless you come join and participate yourself. Yet the scheme of an empricic recollection of computational artifacts used in running a live Osuny system will look very similar. We look into events and patterns observed during two factual use cases of Osuny within the International Degrowth Network to extrapolate preferrable conditions for running replicas of the Osuny application out in the open.

This investigation will allow to answer to the original hypothesis, now reformulated as a question:

> Is it possible for a computational commoning collective or Librehoster to run a replica of the Osuny application for the International Degrowth Network and possibly other communities?

Osuny

I have started working more closely with Osuny, when eco:bytes joined the IDN. It already seemed an interesting option, when we were evaluating the design of the new association website. A decoupled content management system with editing interface for editorial chores it must be, while providing the data via a content API, we collectively concurred.

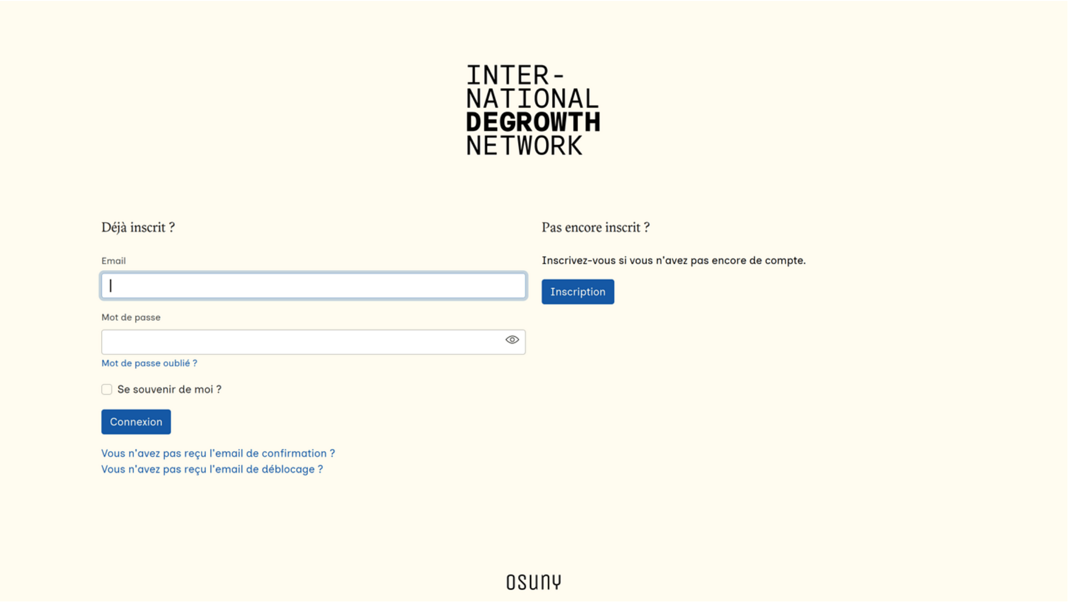
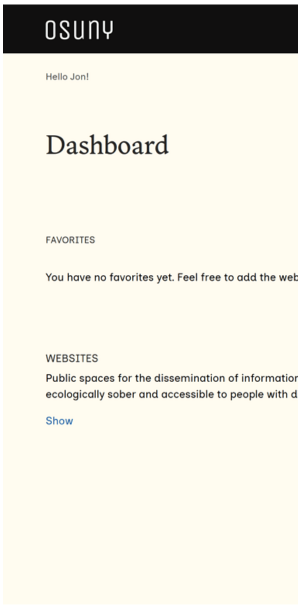
I was interested. And hooked. How does the data leave the editing interface? How does the rendering pipeline produce deployable artifacts? Which auxiliary services are there in use to pull this off?

This artifice is a permeating construct, depending on how you look at it.

> Definition:
>
> > Osuny is a website management system, that produces static webpages, which can be easily edited.

One of its strengths is its strict design system, which can give guarantees about accessibility of content published with it. Browsing its public documentation is very appealing, due to web typography done right.

Basically all supplemental Noesya and Osuny pages are also built with Osuny. One will be able to recognise its visuals after having seen some sites. But there are other creative interpretations of the design system in the portfolio, which showcase its flexibility.

Login	Dashboard
 INTER- NATIONAL DEGROWTH NETWORK Déjà inscrit ? Email Mot de passe Mot de passe oublié ? □ Se souvenir de moi ? Connexion Vous n'avez pas reçu l'email de confirmation ? Vous n'avez pas reçu l'email de déblocage ? Pas encore inscrit ? Inscrivez-vous si vous n'avez pas encore de compte. Inscription osuny	 osuny Hello Jon! Dashboard FAVORITES You have no favorites yet. Feel free to add the web WEBSITES Public spaces for the dissemination of informatior ecologically sober and accessible to people with d Show

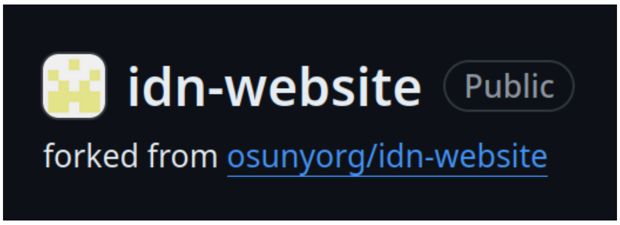
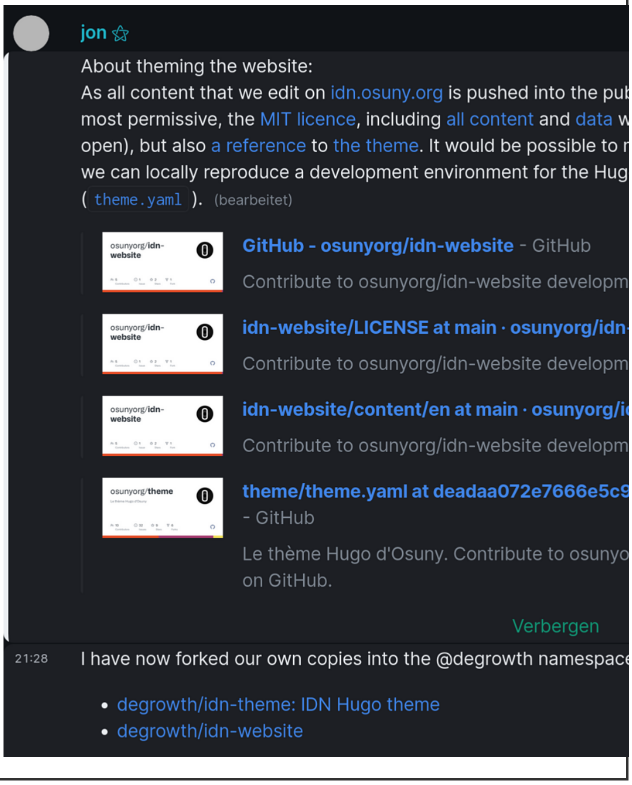
You start *seeing* it for the first time, when you login to your admin backend and start editing the content. A few moments after saving a page, the website will be updated and have an updated look and feel. This is your Osuny workflow: Go to this one website, change this other thing, click Save, be happy.

Before someone will have exchanged details with Noesya and Deux Fleurs to « activate » the website. Yet it happened, and off you go.

Because you don't know what's working behind the scenes for you.

Development

Until: during regular sweeps of lists of repositories and open source software, you discover, that the content of the websites is placed alongside the computational source code artifacts.

> > | Finding | Fork announcement |
> > |---|---|
> > | 
idn-website Public
forked from osunyorg/idn-website | 
jon
About theming the website:
As all content that we edit on idn.osuny.org is pushed into the pub
most permissive, the MIT licence, including all content and data w
open), but also a reference to the theme. It would be possible to r
we can locally reproduce a development environment for the Hug
(`theme.yaml`). (bearbeitet)
GitHub - osunyorg/idn-website - GitHub
Contribute to osunyorg/idn-website developm
idn-website/LICENSE at main · osunyorg/idn-
Contribute to osunyorg/idn-website developm
idn-website/content/en at main · osunyorg/i
Contribute to osunyorg/idn-website developm
theme/theme.yaml at deadaa072e7666e5c9
- GitHub
Le thème Hugo d'Osuny. Contribute to osunyo
on GitHub.
Verbergen
21:28 I have now forked our own copies into the @degrowth namespace
• degrowth/idn-theme: IDN Hugo theme
• degrowth/idn-website |
>
> The repository title of the **degrowth/idn-website** repository, forked from **osunyorg/idn-website** and a message in the internal IDN OC Tech Circle channel upon the discovery. Please note the use of the word *reproduce*.

Suddenly the whole environment unravels in front of you, based on a single code artifact, and new possibilities open up. It wouldn't have been communicated, until you assured yourself that the content of your website is contained there, publically stored without a license, out in the open.

> And you start investigating: How did my content end up here (input)? How is it transformed (process)? Where does it land (output)?

This recollection of artifacts, traces and journeys is basically the story of reverse engineering Osuny based on a few computational artifacts and publicly available information. The intend in doing so is not to seek security issues, but to document the build system that makes up a complete instance of Osuny.

We still want to know how to virtualise and run the system, and keep this as our focus for the rest of the expedition.

Data path

What we can infer about the system after inspecting the source code and retracing the data path:

It comprises of

- web site git source codes (input)
- web site git bot (input)
- web site git build pipeline (process)
- web site artifacts (output)
- web s3 deployment (fancy HTTP copy with a standard object storage protocol)
- web s3 server (HTTP GET by the browser engine)
- web sites (in the eye of the reader of the browser engine), where all comes together:

> Domain Name Systen, DNS Zone, DNS Resource Record (ALIAS), DNS Query, DNS Response (A or AAAA) Record, HTTP GET, HTML, et voilà!

- and a *back office*, which is also **available on the web** and most certainly feeds into the bot.

This is important, because we already find a live running system, which is needed to interact with the hosted build system at the git platform provider, which performs the website build from code to artifact that is deployed and ultimately served to the website visitor, by yet another system. We are collecting.

The git platform is also used for collaboration on the application and web site's source codes, which GitHub coined with the term « social coding ». The web server is something from another galaxy and must not concern us for a longer while. It is FLOSS.

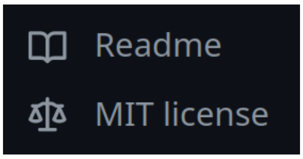
Garage - An open-source distributed object storage service

An open-source distributed storage service you can self-host to fullfill many needs.

Content

To a certain surprise, all website content is also pushed to the repository and has a bot running that directly commits to `main` on a regular basis, whenever an event in the admin interface occurs.

To a more converning surprise, the content was placed alongside the liberally licensed source code. It had no further licensing information applied., remains under copyright

> 
>
> Readme
> MIT license
>
> Readme and MIT license as shown to the right of a GitHub repository, when present.

This was alleviated by including a meaningful README with the repository. For now the content remains under copyright, until further declarations will be placed.

While no sign of them is present in the repository, images are « just there »   (on a first look).

Let's remember our findings for later and proceed diligently.

Movable parts, state and support infrastructure

Running a complete development and live broadcasting system for a contemporary architecture involves many movable parts, which through the levels of indirection they introduce bring many degrees of freedom. Not many people will be able to access a lot of these, as they never reach down the stack.

For a typical website deployment with Osuny, this is an unexhaustive list:

- domain name
  - at the domain registrar
- dns entry
  - at the DNS name server provider
- content management system
  - osunyorg/admin, in the example using the instance known as idn.osuny.org
- thematic design
  - osunyorg/theme and forks thereof

The system is clear, it is simple. It is a website, after all. You have seen this. Nothing to wonder about. You move on.

For a website developer, who might also want to adapt the theme, the situation looks quite different already. The focus shifts from the accessible interfaces provided by the runtime systems to the artifacts that make up those:

- source code repository
- build environments
- runtime environments
- content API
- API integrations (push/pull)
- CLI interfaces
- CI pipelines
- package repositories

A developer will spend time in a code editor of their choice to change some values in a YAML file, or might achieve greater Satori through forking their own theme. There is a lot of documentation about this layer of the application on **developers.osuny.org**.

A systems administrator will be interested to learn more about

- deployment manifests as Infrastructure as Code
- installation tutorial in the readme
- example configuration

When these pieces are documented and put together, they make up a sufficient description of a reproducible runtime environment. This could be expressed in Terraform, Docker Compose or Nomad manifests, Kubernetes Helm charts or other declarative environments, such as Nix.

Dependencies on and integrations with FLOSS platforms

Looking further for pieces that make up the running system, experience shows that Osuny makes use of three standard Internet protocols, which have many implementations in FLOSS land. They are:

- DNS
- Email
- S3

It is reasonably simple to get up and running with any of those on ones own.

The auxiliary supply infrastructure for the development process is FLOSS:

- Automated browser testing: microlink
- Automated machine translation: LibreTranslate

As is for user tracking:

- Analytics: Plausible

Dependencies on and integrations with proprietary SaaS platforms

The situatiom becomes more challenging, when the proprietary elements in the stack are also put into sight. Proprietary here means, it cannot be easily replicated, because it is not in the Commons.

By collecting from different sources, we are able to give a provisional inventory of proprietary components in Osuny. There might be others I am not aware of.

Environment	Feature behaviour	Implementation	ease of FLOSS replacement	alternatives
website + theme development				
	Chat notifications	Slack		Matrix, Mattermost
	PR Previews	Netlify		GitLab CI artifacts
	Code repositories	GitHub		GitLab, Gitea, Forgejo
			partially already available	on *GitLab*
application deployment				
	admin	PaaS: Scalingo		Dokku, Piku, Porter, Kubero
	Assets	S3: Scaleway		
	Status page	updown.io		statping-ngx, Uptime Kuma, Cachet
	CDN	KeyCDN		
	Email	Google Mail		
application development				
	Exceptions	BugSnag		Sentry

	Code scanning	CodeClimate		GitLab CI SAST, Bearer, Betterscan

There is a lot to say about any of these, but most concerning seem the dependency on the Assets and the CDN provider. Unfortunately these are not services that exist as not-for-profit, due to the involved scale to make it work. A useful addition to speed up image delivery nonetheless.

It could be interesting to see all of these presumably actually existing use cases documented.

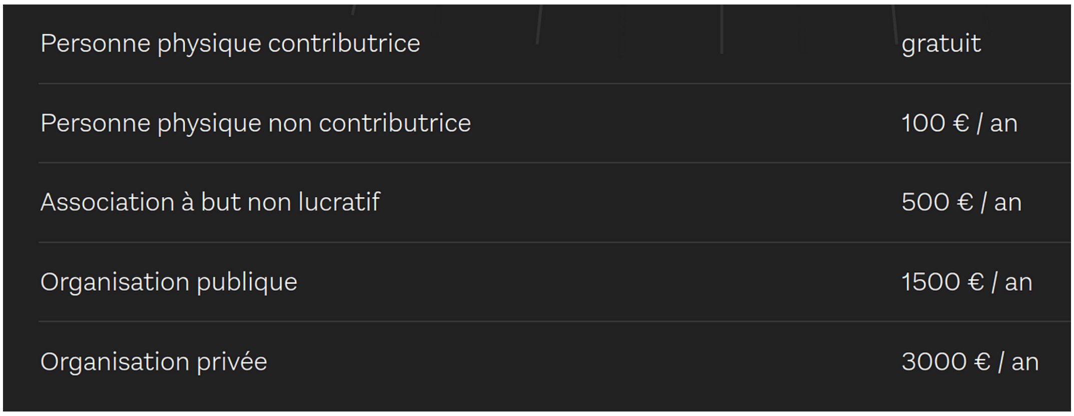
> Deliberation could be given to further liberating of the dependencies in the application stack.

Distribution model

These information were oddly not derived from a technical documentation, but from a description of the commercial service offer on the Osuny website.

- Osuny as a Service | Osuny

The site gives further insights into the economic functioning around the Osuny Common.



Personne physique contributrice	gratuit
Personne physique non contributrice	100 € / an
Association à but non lucratif	500 € / an
Organisation publique	1500 € / an
Organisation privée	3000 € / an

> List of Osuny prices per year

When looking for other Osuny deployments in the wild, there do not seem to be other providers than noesya, yet. Maybe I am missing someone in particular?

We can use the Osuny status page to find more instances of websites built with Osuny.

This ends the section about the modes and means of production that make up a technical perspective on the Osuny Common. After all this weary listing and cataloging items, let’s allow ourselves a break in perspective, and put our light some place else.

Another way of looking at the system is through the eyes of an actual user. What can we learn and pass on from experienced examples?

Let’s have a look at some of those in the next section, and see how it can help us to evaluate, whether we will be able to replicate the Osuny Common with simple-enough-means.

Osuny case studies: degrowth.net/ + explore.degrowth.net/ (3/5)

Observations and projections from half a year of using Osuny : "Co-construire un commun numérique" pour "Devenir un commun" (4/5)

Ergonomy of development experience and common reproducibility questions around maintaining degrowth.net