

# Observations and projections II (5/5)

---





























---

yala 

juin 2024

Bonjour Arnaud.

This is developing into an interesting direction.

arnaudlevy:

I think the whole is a Common, made of smaller commons

While I agree with the latter, I must politely disagree with the first.

My let's say investigation led me to discovering very blurred boundaries between noesya and Osuny as its product, and Osuny the Common (of noesya and its user base) which together manages the overall direction of the project.

That is, as proposed, I would suggest to introduce interfaces (in both directions) between the two entities, to be able to draw a clearer barrier.

noesyia coop + Osuny SaaS product > develops modifications to the shared code base derived from a large user base + voluntarily publishes > Osuny Common artifacts  
Osuny Common community requirements > collect, refine + build up knowledge common > additional empirical experimentation for noesyia to work with

As Commoners, we create these little membranes around us, to filter out the world of business interest and hegemonic logic from our praxis. From the presentation on the many websites, it is just really hard to see where the coupling points are. Maybe I am a radical Communist, in so far, that to me cooperative money-relations are happening outside of the Common and that for a good reason.

As a suggestion to look at it: while it is noesyia who receive public funding for a very specific project, it is the presence of the independently maintained Common which accommodates its outcomes and ensures its long-term sustainability. What I am getting at is: which are the practices in the Osuny community Common, which do not engage in reciprocal exchange?

I'm really interested in learning, how many non-SaaS clients of Osuny there are, how many non-paying instances that serve the General Interest?

arnaudlevy:

That's why we do monthly public assemblies (next one on the 4th of July [Assemblée Osuny #8 | Osuny](#))

A good reminder.

I see you are a friend of [selfdogfooding](#) already. It's interesting to witness how the presence of an admin interface that invites for easy contributions, you can use your static website environment as a semi-dynamic web application and have it very up to date.

Often with static websites it's just *fire and forget*.

arnaudlevy:

we started using a proper semantic versioning and release notes. But it failed, as we expected

Yes, the modern Software Delivery Lifecycles encourages append-only working styles with regular dated releases.

arnaudlevy:

because taking care of an instance takes something like a full time job for one person. So we don't give any energy to make it easier, as no-one seem concerned except us.

Doesn't have to be. Depending on the automations you provide and guarantees you can give about your system, e.g. by using the other suitable abstractions, you receive other leverage points to instrument and maintain your decoupled system.

Caring for DX in terms of platform engineering and Infrastructure as Code, like the whole GitOps train, has changed the way we build software. In a good way! Deterministic builds and declarative interfaces with clear separation of the twelve factors allows to implement for more resilient application engineering patterns.

Take Scalingo, for example. Their Heroku-like git push deployment model is not as proprietary, as it may seem. With Dokku, it costs close to zero effort to replicate an Osuny instance, if good instruction is given. They probably also just build a container for you, like Dokku and the many other PaaSs, so why not do it yourselves already? Then it's there and you can reason about it. Before that, all discussed here is a little theoretical.

I also don't see, why nobody else would be concerned. The world is so large and there are so many groups and people struggling with the exact use case Osuny provides. Why is there [sully.coop.ar/](https://sully.coop.ar/) for Jekyll? Why are people trying to use DecapCMS or Sveltia CMS, despite their horrible UX? Because there is a global demand for such nice editors for static websites.

arnaudlevy:

We try to make everything open and documented here [Contribuer à l'admin – Osuny](#) . There are, for example, quite detailed steps about connections and references here [Connexions et dépendances – Osuny](#) .

We are engaging in quite some intercultural dialogue here. My brain is not yet wired for connecting my technical French reading with my technical knowledge in the other two languages. Now that I take a second look at it, I am seeing some command examples.

I think the website - and Osuny in its backend - could make it easier to distinguish article drafts, those in a collaborative review process and those published. Here it's a very deep hierarchy (four levels) with many empty pages and only little visual distinction, content spread over many pages. Many parts of pages only contain bullet points. It's hard for me to follow a narrative under these conditions.

What I'm really asking for is just a little Docker Compose with a simple `up` command that gets you started right after cloning the repository, maybe even without modifying an `.env` example.

There is also a lot clarity to gain from this OpenAPI schema or that Rails ERD.

arnaudlevy:

The whole project is not only technical, but also a political and economical perspective on how we do websites.

I'd like to understand more, how you combine these three aspects. But this may not be the space.

I'm here for the websites and the Commoning, which also involves the three kinds of perspectives, but maybe in another light? I'm looking at this without the cooperative glasses, implying my earlier conjecture about concrete interfaces inbetween noesya and Osuny. As long as they are not established, Osuny is a product.

From what I understand, noesya, or its people in their spare time, I don't know, contributes a lot more to Osuny than Software artifacts. There is already writing documentation, engaging the community, reaching out to and involving with other partners and communities (IDN), running *pro bono*, in a reciprocal view *unpaid*, websites and probably a lot more.

Could be nice to see how others could be involved, too. For example, I'm *really* interested in having an English reference documentation of Osuny in my hands. I'm just clicking around in the French one and keep on getting lost in the many items and levels in the navigation tree. Is this worth a series of coordinated PRs and content changes in the admin of <https://developers.osuny.org/>? Which editorial workflows are recommended for Osuny?

## Gérer le multilingue

Ce qu'il faut savoir pour les sites utilisant plusieurs langues

While I could theoretically fork the repository and recreate the data structure locally, it seems limited by how many separate writing environments there are for the development. website, and how content changes progress between them. Would you happen to have a means to reimport content changes and new content from pull requests at your disposal? Does Osuny have a feature that allows me to check out « a branch » of the database for an individual website, edit ahead and then reintegrate that?

arnaudlevy:

It would make perfect sense to have another Osuny in south-america, for example.

Let's make it happen! I'm already looking forward to rip Osuny admin apart and to put it back together.

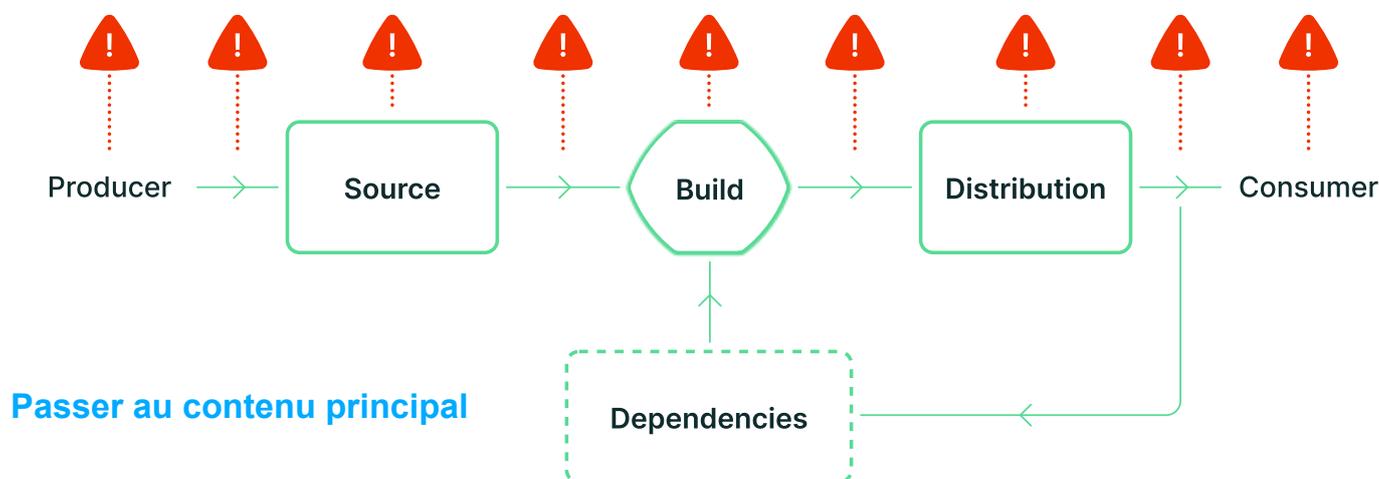
This should eventually be a prepared, documented and funded endeavour. But nobody stops us from requirements collection and seeking cooperation partners, for the Common and out in the open.

There is also a conversation to be had about publishing ActivityPub. You already have a long-running process (daemon) that can handle medium amounts of changes per time interval (limited by CI velocity & concurrency).

À la prochaine

Edit:

PS: [slsa.dev/](https://slsa.dev/) presents us with **Supply-chain Levels for Software Artifacts (SLSA)**, which can be attested and verified.



PPS: This is a nice example of a similar system, too. [Website Factory: NGO — Commit Global](#) « The only Website Builder designed specifically for NGOs. » 😊

PPPS: The Codespaces example for using a development container to build a web page is now here:



**feat: development container with Codespaces compatibility**  
osunyorg:main ← degrowth:next  
ouvert 🌐 28 juin 2024 almereyda +171 -43

- adds pagefind.yml to automate pagefind runs with less command line parameters  
...

yala 🛡️

juil. 2024

*Update:*

This article has a nice compilation of arguments for (1) using the AGPL and (2) decoupling the whole stack of an application from Commerce and open sourcing it into the Commons.



**On Open Source and the Sustainability of the Commons**  
On Open Source and the Sustainability of the Commons par Ploum - Lionel Dricot.

 Alimenté par Discourse