Summer of Nix 2025

Program report

Fedi Jamoussi Ivan Minčík Daniel Ramírez

November 28, 2025

Summer of Nix is a coordinated effort to support free and open source software (FOSS) projects that are funded by the European Commission's Next Generation Internet (NGI) initiative through the NGI0 consortium coordinated by the NLnet Foundation.

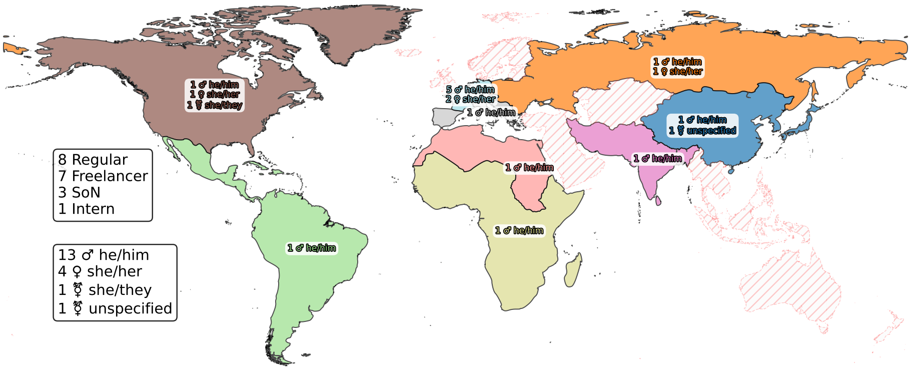
The program's main objective is to make more software available as Nix packages or NixOS service modules. This is in order to make the projects developed under NGI0 easier to obtain and run – something that is not a given due to the complexities involved in software development – and thus help end users and developers to reap the benefits of the public funding campaign. Nix itself is a proven technology for highly repeatable software builds and deployments, and supports an ecosystem of tools for building a fully transparent software supply chain.

Packaging artefacts for supported open source projects are collected in NGIpkgs Git Repository and presented via NGIpkgs Overview Web Interface.

This year, the team was newly formed to sustain packaging and maintenance effort with a kickoff meeting on Jan 13th. Long term goals were set using OKR framework. In addition to packaging work itself, the team has focused on redefining the NGI package content and design (package deliverables), packaging quality, tasks and project management and automation in order to provide the best possible contributor and user experience.

1 Summary

- Activities from January 2025 to November 2025
- Team's role and gender distribution (after UN M49 standard)



1.1 Expenses

Cost Category	Amount [€]
Paid (in balance)	82 200
Pending invoices	17 600
Allocated budget	55 800
Travel expenses	3 100
Total	**158 700**

- Time worked: 3 900 h
- Average compensation: 26.50 EUR/h

1.2 Packaging work

The following software projects where packaged:

Project	Funds
0WM	Core: 1
Arcan	Core: 2, Entrust: 1
Blink	Entrust: 1, Review: 2
Briar	Review: 2
Corteza	Review: 3
CryptoLyzer	Core: 1, Review: 1
Draupnir	Core: 1
ERIS	Review: 1
Galene	Core: 1
Gnucap	Commons: 1, Entrust: 2
Heads	Review: 2
Inventaire	Entrust: 1, Review: 2
Irdest	Core: 1, Entrust: 2, Review: 1
Kaidan	Commons: 1, Entrust: 1, Review: 4
Kazarma	Entrust: 1, Review: 1
Keyoxide	Review: 4
Liberaforms	Commons: 1, Review: 2

Project	Funds
LibreSOC	Entrust: 2, Review: 6
Librecast	Commons: 1, Core: 1, Review: 2
MarginaliaSearch	Core: 1, Entrust: 1
Mastodon	Commons: 1, Entrust: 1, Review: 1
Mox	Core: 1, Entrust: 1, Review: 1
Namecoin	Review: 5
Nitrokey	Commons: 2, Entrust: 1, Review: 1
NodeBB	Commons: 1, Core: 1
Nominatim	Entrust: 1, Review: 1
Openfire	Core: 2
PagedJS	Commons: 1
Re-Isearch	Commons: 1, Review: 1
ReOxide	Entrust: 1
Repath-Studio	Commons: 1
SCION	Core: 4, Entrust: 1, Review: 3
Seppo	Entrust: 1
Tau	Core: 1
Teamtype	Core: 1
ThresholdOPRF	Entrust: 1, Review: 3
Wax	Core: 1
Wireguard	Entrust: 2, Review: 5
holo	Core: 1
jaq	Commons: 1, Entrust: 1
kbin	Entrust: 2
nyxt	Entrust: 1, Review: 2
oku	Entrust: 1
openXC7	Entrust: 1
owasp	Core: 1, Review: 1
owi	Commons: 1, Core: 1
proximity-matcher	
slipshow	Commons: 1
stalwart	Core: 1, Entrust: 1
verso	Core: 2, Review: 2
xrsh	Entrust: 1
y-crdt	Commons: 1, Entrust: 1

1.3 Additional metrics

In the NGIpkgs repository, the team is maintaining

- 47 services and 55 programs
- 30 projects with demo available
- 97 NixOS tests, associated with 97 examples
- 80 derivations, 59 of which have an explicit update script

Funded by the following subgrants

- 16 Commons
- 38 Core

- 64 Entrust
- 102 Review

In the upstream Nixpkgs repository, the NGI team is maintaining 131 derivations, 107 of which have explicit update scripts.

1.4 Non-packaging work

In addition to packaging work, the following additional work was done:

- Designed and implemented a new NGI project schema for metadata and deliverables with type checking support using the NixOS module system
- Improved project and deliverable tracking with issue templates
- Implemented demos for programs and services as a practical way to showcase projects
- Implemented update script support and bulk updates support
- Implemented multiple improvements to the UI/UX of the project overview page such as:
  - implementation of website generation using NixOS modules
  - implementation of metadata, deliverables, demo and demo instructions sections
  - implementation of code snippets, with copy-paste and download buttons
  - implementation of highlighting of missing deliverables, such as programs, services, or demos, with links to implement them
  - implementation of breadcrumbs and dark mode
  - addition of usage instructions for multiple Linux distributions
- Implemented significant improvements of tasks and project management tools and processes (planning, backlog grooming and review sessions)
- Provided support for integration of Nix based development environments to upstream source code bases of NLnet funded software
- Provided regular weekly NGI team office hours for public audience
- Joined Outreachy program (1 mentor from the core team and 1 Outreachy mentee)
- Participated in NixCon conference and gave multiple talks

2 Evaluation

After the team was newly established early this year, it took us considerable time to properly scope our work and settle on suitable project and task management methods and suitable assignment of tasks to full-time and part-time team members. Also, in the chase for maximum efficiency, we suffered from premature attempts to automate our processes which didn't give us the expected results.

Nevertheless, this year's results are very positive. We have packaged significantly more software than last year (57 projects compared to 35 in 2024, representing a 63% increase) with higher quality. Each package now includes better documentation, tests, and demos.

The packaging effort was more consistent throughout the year, and thanks to improved task management, we were able to focus on high-priority software projects.

3 Future plans

In addition to continued software packaging and maintenance effort, we want to significantly improve our user feedback loop, which in turn will allow us to improve our product.

We also want to strengthen collaboration with upstream projects by encouraging adoption of Nix-based development environments and packaging best practices. This will help ensure that NGI-funded projects remain easily installable and deployable long after initial funding ends, maximizing the return on public investment in open source innovation.