State of the OCaml Platform 2023

Anil Madhavapeddy
University of Cambridge

David Allsopp
Tarides

Thibaut Mattio
Tarides

Thomas Gazagnaire
Tarides

***Abstract*—This paper reflects on a decade of progress and developments within the OCaml Platform, from its inception in 2013 with the release of opam 1.0, to today where it stands as a robust toolchain for OCaml developers. We review the last three years in detail, emphasizing the advancements and innovations that have shaped the OCaml development landscape and highlighting key milestones such as the migration to Dune as the primary build system, and the development of a Language Server Protocol (LSP) server for OCaml.**

We also outline our plan for the coming years. The roadmap is informed by community feedback, discussions with Platform tool maintainers, and insights from industrial users of OCaml. The final version of this evolving roadmap, designed to shape the future of the OCaml developer experience, will be presented at the International Conference on Functional Programming (ICFP).

I. Introduction

The OCaml Platform serves as the backbone for building software in OCaml, a functional programming language known for its expressiveness and safety. It is the recommended toolchain to work and be productive with OCaml, comprising reliable tools like the opam package manager, the Dune build system, the Merlin editor helper, and more. Alongside these work-horse tools, it also incubates new development tools aimed at improving the OCaml developer experience, such as dune-release for publishing Dune projects on opam repository, and Mdx to execute code blocks in documentation.

The OCaml Platform first started with the release of opam 1.0 in 2013, and this year, it celebrates its 10th anniversary. A lot has happened in 10 years and the Platform now features a dozen of tools in the Active and Incubate stages!

This paper reflects on the progress we've made in the past three years and delve into the future plans for the OCaml Platform, highlighting its vision, goals, and future developments in OCaml tooling. The goal is to communicate the exciting developments on the horizon for the OCaml Platform, and how they will transform the OCaml developer experience in the coming years.

II. Reflecting on the past three years

Three years ago, Anil Madhavapeddy and the OCaml Platform team unveiled an ambitious roadmap for the OCaml Platform. Our 2020 vision was to create a seamless editor integration, enabling users to open their editors and instantly be productive with OCaml. Though we are still on the journey towards fully realizing this vision, the progress we have made so far is significant. The landscape of tooling and the overall developer experience with OCaml has been transformed over the past 3 years with considerable advancements made on all tools of the Platform.

- Dune is now the primary build system used by OCaml developers, with 65% of packages on the opam repository using Dune as their build system.
- Ppxlib has become the best way to write PPX and most PPX have been migrated to Ppxlib. To get a sense of the effort that went into this, you can read the updates from 2019, 2020, and 2021.
- We've built a Language Server Protocol (LSP) server for OCaml, that powers the OCaml VSCode extension which has now been downloaded 100K times.
- We've built an RPC protocol in Dune that was released in Dune 3.0 and another one for OCamlFormat. These protocols are integrated with OCaml LSP to support some of its features.
- Odoc 2.0.0 was released with a complete re-write of the language model to make the generated documentation more accurate and also comes with a new rendering layer to allow output in different formats including HTML, LaTex and manpage. Odoc is also now used to generate the documentation of every OCaml package on ocaml.org.
- Opam 2.1.0 was released with significant improvements like the integrated depext system and the generation of lock files.
- We re-implemented the way we locate OCaml terms in Merlin which greatly improved the accuracy of the locate query and paved the way for the work on project-wide references. To achieve this, we added a new representation of the module structure in the OCaml compiler called "Shapes". Shapes are now used in Merlin, but also enable features in other Platform tools, like linking to source code in Odoc.

In the meantime, things have also changed for the broader OCaml ecosystem. OCaml 5 was released with support for shared memory parallelism and effect handlers; and we saw a new version of OCaml.org going live with a centralised package documentation, a job board, an interactive playground, and more.

Following the three priorities we adopted in 2022, we now want to intensify our efforts on prototyping new workflows for OCaml development. It is time to lay down a roadmap for the next three years.

III. A ROADMAP FOR THE NEXT THREE YEARS

We have been working on a future roadmap for the OCaml Platform, taking into account a wide range of factors including community feedback, industrial needs, and our vision for the ideal developer experience with OCaml. As of now, the OCaml Platform roadmap is a dynamic, evolving document. We have shared the initial sections with the OCaml community and begun collecting feedback. The final roadmap will be shaped by this feedback, and we aim to present a final version at the International Conference on Functional Programming (ICFP).

Below are some preliminary axes which, based on the past months of user research and feedback review, we anticipate to be in the final roadmap in one form or another.

A. A seamless editor integration

Our focus for the past few years has been on creating the ideal editor experience for OCaml, with the goal of being able to do everything from the editor. With Dune RPC providing access to the information and features from the Build System, OCaml LSP becoming more and more complete, and we are steadily moving towards this goal.

Support for package management in Dune will be a big milestone for the editor integration, as it will place Dune as the only tool needed to build OCaml projects. Editors will be able to interact solely with Dune RPC under the hood to power this seamless editor integration.

We'll also look into adding features to Merlin, such as the most anticipated project-wide references, and next project-wide renaming.

B. A unique tool for the OCaml Platform

The OCaml Platform is set to remain a collection of independent tools, and its governance is modelled around the fact that tools have a lifecycle: the package manager and build system we use today might not be the same as we will use in 10 years.

However, it is increasingly evident that in 2023, developers expect to have a single tool to interact with for all development workflows.

Dune has organically taken the role of the frontend of the Platform. It already integrates with OCamlFormat, Odoc, Mdx, and Merlin to name a few. With package management support, it will continue integrating with more Platform tools to provide package management and publication workflows.

Moving forward, we want to recognise the importance of a unified experience and consolidate the integrations of all the Platform tools in Dune.

C. Filling gaps in OCaml development workflows

Following user feedback on the development workflows supported by the Platform, we will focus our efforts on closing gaps and improving the developer experience of the Platform tools.

Notable new development workflows that we will explore include linting, creating installers, auditing code for security advisories, benchmarking, and more.

In the coming years, prototype new tools for these development workflows, with the goal of incubating them in the Platform and integrating them both in Dune and in the OCaml Editors.

IV. ACKNOWLEDGMENTS

We would like to express our deepest appreciation to every individual who contributed their expertise, ideas, and time to the formation of this roadmap. Your active engagement in brainstorming sessions, diligent contributions to the development of Platform tools, and enthusiastic participation in discussions, in conjunction with your vigilant review of work-in-progress and invaluable feedback, were integral to this endeavour. Crafting a user-centric roadmap necessitates a great degree of coordination, and we profoundly value your shared commitment.

Specifically, our heartfelt thanks go to Ulugbek Abdullaev, Jules Aguillon, Cuihtlauac Alvarado, Paul-Elliot Anglès d'Auriac, Raja Boujbel, Frédéric Bour, Fabrice Buoro, Javier Chávarri, Enguerrand Decorne, Kate Deplaix, Daniel Dickstein, Boning Dong, Carl Eastlund, Patrick Ferris, Ulysse Gérard, Gemma Gordon, Rudi Grinberg, Rafal Gwoździński, Sonja Heinze, Sébastien Hinderer, Hans Hoguld, Shakthi Kannan, Varun Kohli, Marek Kubica, Jon Ludlam, Etienne Millon, Yaron Minsky, Andrey Mokhov, Dmitri Nikulov, Nicolás Ojeda Bär, Ty Overby, Guillaume Petiot, Lucas Pluvinage, Thomas Refis, Louis Roché, David Sancho, Riku Silvola, Sabine Schmaltz, Stephen Sherratt, Mark Shinwell, Magnus Skjegstad, KC Sivaramakrishnan, Emile Trotignon, Arthur Wendling, and Leo White.

Additionally, we extend our sincere gratitude to the OCaml community for their proactive and constructive engagement. Their insightful responses to OCaml Discuss posts and community surveys have been crucial in shaping this roadmap.

Lastly, we acknowledge with gratitude the generous funding provided by the industrial OCaml users. Special mention goes to Jane Street and Tarides, whose financial support has significantly propelled the development of the OCaml Platform. This project would not have been possible without their unwavering support.