A New Chapter for Haskell:
The Haskell Foundation

Prelude

Haskell's slogan of "avoid success at all costs" was a clever and cheeky way of saying that innovation and research in programming languages, especially in functional programming, needed some insulation to succeed. Ideas that were not perfectly understood needed iteration to fully develop in the minds of language innovators and users. By avoiding the "success at all costs" mentality of other language communities, the Haskell community bought time and space to try ideas that were not perfectly understood at first. Since then, the Haskell language has sparked so many lasting innovations in language design that its impact is now beyond doubt.

What is the reason for this outsized impact? Haskell and related languages re-opened the connection between mathematical thinking on the one hand and compilers and programming languages on the other. It showed that these two fields should never have drifted so far apart. By removing the ceiling on the ideas that are easier to express in Haskell, it attracted the brightest minds and still does. It became a lingua franca for a large swath of CS research. In education, Haskell helps CS students learn to think better. The quality of ideas represented in the Haskell ecosystem has attracted both small and large companies. In many ways, the story of Haskell is one of success. Perhaps it was unavoidable after all.

As good as the core concepts of Haskell are, users adopting it are often exposed to some very rough edges. We think it is time for Haskell to have a much smoother on-ramp and more real-world use cases where it is the clear first choice. There are now countless stories of Haskell adoption in the industry, but not all are successful. A few companies have even switched away from Haskell, for reasons that include difficulties with the compiler, tooling and hiring. Once users can get past these obstacles, they can achieve amazing outcomes, but too often the path to get there is not clear.

Haskell is an open-source community. Almost all its tooling and community is created and run by a strong community of bright volunteers. But these contributors all have other day jobs, so the Haskell ecosystem still lacks crucial social, technical and organizational "glue".

We believe that every single obstacle standing in the way of Haskell adoption can be overcome.

All technical issues are tractable and require a modest investment of energy, commitment and financial resources to fix. Likewise the social/community issues: the Haskell community is packed with friendly, able and motivated contributors who share common values. We think that by focusing our attention on the critical story of adoption, Haskell can encourage yet more innovation in the future, while bringing the benefits of its core ideas to a much larger user base. We need to encourage adoption in new ways and renew focus on improving the Haskell ecosystem.

The Haskell Foundation

That's why we are forming a non-profit organization, provisionally called "The Haskell Foundation" (HF), dedicated to advancing the Haskell programming language, related tools, education and research.

Goals

HF has these goals:

- To foster a much broader understanding of the benefits of using Haskell among developers who do not currently use Haskell.
- To increase adoption of Haskell substantially, by erasing barriers that inhibit adoption.
- To identify and fill missing "technical gaps": the useful stuff that makes for a smooth user experience (pain-free installers, documentation, error messages, and much more).
- To help with "community glue", by nurturing respectful, inclusive communication across the community.

Together these goals form the mission of HF.

Principles and ethos

How we pursue the goals of HF is just as important as *what* the goals are. HF's actions will be guided by these core principles:

- **Open source**. Haskell is an open source community and HF will embrace the open-source ethos wholeheartedly. HF may develop, or sponsor the development of tools and infrastructure, but it will all be open source.
- **Empowering the community**. A major goal of HF is to augment, celebrate, and coordinate the contributions and leadership of volunteers, not to supplant or replace them.
- **Open, friendly, and diverse**. For many of us Haskell is more a way of life than a programming language. All are welcome, all can contribute.
- **Transparent.** All communication related to code and decision making will be publicly accessible, to enable asynchronous communication and collaboration. Only certain categories of sensitive information (e.g. financial, and matters concerning particular individuals) will be kept confidential.
- **True to Haskell's principles.** Haskell's design puts principle ahead of expediency, notably by cleaving closely to the principles of purely functional programming. Success, yes, but not at all costs!

We have learned from other open source communities. The Rust community has a code of conduct that has benefited the community. They do a remarkably good job of this, in large part because the Rust community is actively led and nurtured. The Apache Software Foundation has developed clear standards of transparency and consistent governance across all of its 300+ projects, run by volunteers. As we set up HF we have adopted successful approaches like these into our operating principles.

Organization and Funding

The organization will seek funding to ensure the longevity and continuous strengthening of the Haskell ecosystem.

Structure

The Haskell community comprises an amazing group of technical talent and functions today as an almost entirely volunteer effort. Our goal is make every member of the Haskell community, and every HF-affiliated Haskell committee feel more support and more productive. We want to enlarge and diversify our community.

We have received and incorporated lots of feedback about the best way to structure HF.

- HF will have a **Governing Board** ("Board" hereafter) that reflects the Haskell community and its stakeholders, including academics, commercial users, and individuals.
- HF will have a staff. The size of the staff will depend on funding, but we intend to hire an Executive **Director** (ED), who can organize Haskell outreach, support its funding activities and oversee the rest of the staff. The staff will mostly focus on funding, marketing, and key infrastructure.
- HF will work with existing and new open source teams to channel energy into various efforts like packaging, tools, libraries, compilers, languages, documentation, user experience, and infrastructure. To reiterate, we expect that most technical contributions will be volunteer, just as it is today, but we want to position HF to fill gaps that can help adoption.
- HF will establish a code of conduct and transparent decision-making that will apply to itself and any associated teams.
- To the extent HF funds and pursues technical goals itself, it will pursue those goals with the same transparency as we expect from any of the teams associated with it. In this way HF will augment the community in a transparent way.

We think that HF represents one of the final puzzle pieces for Haskell. A new organization will provide a way to fund and coordinate Haskell development going forward. We hope that all key committees that currently support Haskell will align HF's values and mission and we are working with those committees that wish to voluntarily affiliate with HF. Discussions with those teams are currently underway. We don't want to simply add another Haskell committee.

Who are the HF organizers?

The idea of the Haskell Foundation has been developed by an informal working group including

- Representatives from the haskell.org committee, the Core Library Committee (CLC), the Hackage Trustees and the GHC Devops Committee, and other Haskellers.
- Haskell companies, who are generously providing financial, advisory, and in-kind support.
- Numerous long-standing members of the Haskell community.

Funding

There have been other initiatives of this kind in the past, but they have proved hard to sustain. A big part of this has been a simple lack of bandwidth in a highly decentralised community run entirely by volunteers.

We expect to launch with a small group of founding sponsors. The Board and staff will take over that function after launch. Our goal is to raise around $1m/year in cash and in-kind contributions.

Improving the Haskell Adoption Story

Promoting Haskell adoption has three major components:

- Eliminating unreasonable and perceived barriers to adoption.
- Educating the tech community about the benefits of adoption, including decision makers.
- Enhancing the tooling, so that the risk of adoption is dramatically reduced.

Eliminating unreasonable barriers to adoption

We expect that with HF adding a little structure, some things can be improved immediately. For example, we need an entry point for Haskell that speaks to the needs of a range of users: from engineers looking for an easy on-ramp with our best advice on how to learn and use Haskell, to team leaders who want to assess Haskell adoption as a business decision. The Haskell community today does not cover the full range of content needed to promote adoption. This is one example where an organization with the right focus and some resources can easily have a positive impact.

Educating the tech community

Engineers are typically the ones who drive Haskell adoption. But they often need permission or sponsorship from managers 1 or 2 levels higher. These decision makers are people who are more concerned about speed of development, reliability, maintenance, and people. We need to explicitly address a broader audience and position Haskell as the best solution to many problems, while maintaining integrity and avoiding too much “marketing speak".

Conditions for telling this story are favorable. The days of a senior sysadmin dictating which version of Java or Python “shall be used” are dying. Containers and cloud technology have inadvertently conspired to permit engineering teams to make these decisions more often at a team level, even in larger organizations. Still, these teams need air cover and solid tools so that their decisions don’t look “rogue” to the rest of the organization.

If used correctly, Haskell can be unreasonably effective. Unfortunately, the “if” clause is not common knowledge. The HF recognizes that many of these advanced features are precisely the reason why Haskell can be successful in cases where other languages fall short. We trust that the end users will be able to make the right trade-offs that work for them. The HF will be there to help users that need to make these trade-offs by improving documentation and highlighting success stories. For example, Haskell has a great concurrency story and things like STM that are dramatically undersold in the marketplace. Some advanced features can be emphasized and taught.

The spectrum of developers is very broad: Haskell is used by both junior developers as well as long-time experts. This can cause friction, and some code may be unreadable even to other Haskell developers. This is unfortunate, because we believe that Haskell allows us to express ideas in a way that are clear and elegant. We need to delineate the common knowledge aspect of the language and tools to elevate more developers. HF should carry this banner.

Beyond that, the pesky problem of the marketplace for Haskellers remains. We often hear about fear of hiring bottlenecks and we know some engineering leaders that list this as the number one concern about Haskell. These leaders need to know one thing. You don’t have to hire a Haskell team: Haskell can make your team. Your smart Python developers can learn how to make reliable software and will love you for letting them. There is a role for HF to bridge the gap between the countless developers who want to learn and write Haskell, and the decision makers who currently believe it is hard to find these people.

Technical Agenda: Enhancing the tooling, filling the gaps

Haskell Foundation will identify a list of technical goals that will ease adoption and improve Haskell use in production. We have established an initial agenda and are seeking to refine it as we go forward. As HF evolves, we will engage technical discussion in a transparent way, with input from the community.