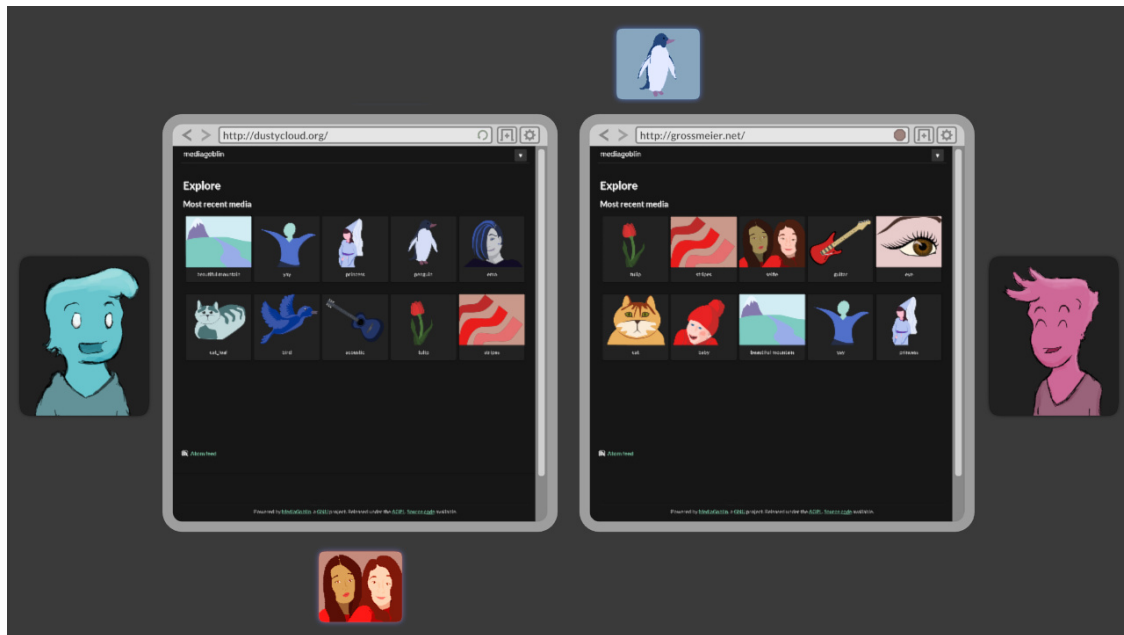


# MediaGoblin's application to Knight News Challenge

Christopher Allan Webber <cwebber@dustycloud.org>

April 28, 2014



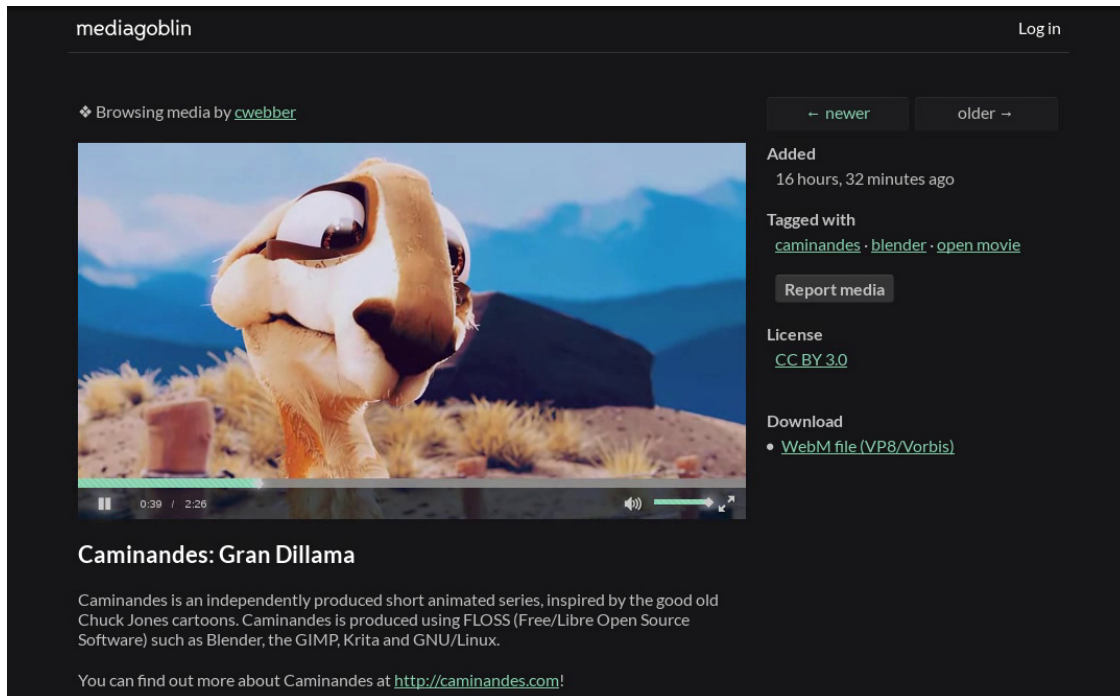
*MediaGoblin is a decentralized media publishing system. Users can share across nodes thanks to built-in federation tooling.*

## 1 Project summary

MediaGoblin is a decentralized and social media publishing system focused on user freedom. It supports many different media types; images, video, audio, 3D models and more. Instead of needing a separate service for each one (YouTube + Flickr + SoundCloud + Vimeo + Thingiverse), each controlled by a specific company with a walled-off network, when you use MediaGoblin you need just one account – and it's yours to control. For maximum independence you can host your own network node, or just create an account on a node you trust. It archives all your media and lets you share with as large or small a circle as you wish. MediaGoblin is free and open source software, making it the censorship- and surveillance-resistant solution for global media sharing.

## 1.1 Detailed description

MediaGoblin is a highly extensible free and open source media publishing system with a great plug-in architecture. This means that its evolution can be user-driven, allowing it to be adapted to a great variety of user purposes. Nodes focused on citizen journalism may have different requirements than music publishing nodes. If a person, project or business using MediaGoblin needs an extra feature, they will always have the right to extend the software or hire someone to extend it for them – as long as they contribute the code back to the community they build on.



*Thanks to MediaGoblin's built-in extensibility, many media types are supported. Here we see the film "Caminandes: Gran Dillama" playing on a MediaGoblin node.*

This extensibility is already evident in the areas of theming, media types, and plugins providing additional features like embedded OpenStreetMap support, and flexible authentication options including OpenID and Persona. MediaGoblin's codebase and legal tooling make it ideal for extension, and because it is written in Python, it is accessible and easy to modify.

The internet is becoming a fragmentary landscape of competing but incompatible services, with business models commonly designed to exploit and endanger users rather than aid them. But it doesn't have to be that way. MediaGoblin is designed for federation, the way email is. Federation helps people talk across different systems. Imagine if every email user of Blahoogle.com could only email other Blahoogle users, and needed another subscription at CatMail to talk to users there. That is what is happening across the media-rich social web. MediaGoblin doesn't want to see the internet become a hopelessly fractured and chaotic space where everyone has to keep up with a zillion services and networks, always chasing their feature and social needs. These services are not prioritizing the open standards and

user rights that protect the internet itself while keeping it an egalitarian and safe zone to live in. Rather than cultivate its own private commercial fiefdom that relies on preventing interoperability with everyone else, MediaGoblin builds tools and communities that benefit everyone. MediaGoblin is a powerful framework that people can build on, benefit from, and share the benefits back to the common good. We don't think people should have to trade their freedom for usability. MediaGoblin is designed to offer people the same kind of social media sharing features they find in their existing services, but better: simpler, more powerful, connecting a global user base, growing with every bit of user investment, and all while protecting user freedom.

The call for an independent media publishing system that resists censorship grows urgent as we see increasing cases of the centralized commercial services exerting censorship and restricting content – on behalf of their own interests, and in collusion with regional governments. The silos that are the main current points of media sharing are not only vulnerable to attacks on free speech, they also hamper important grassroots economic activity by privileging the interests of a tiny minority over those of most of the world. As more and more people conduct their lives online, the accumulation and commodification of granular personal data presents a tremendous encroachment on privacy, security and anonymity – not just the hallmarks of free society, but even the preconditions of our freedom of thought.

## 1.2 One-sentence summary

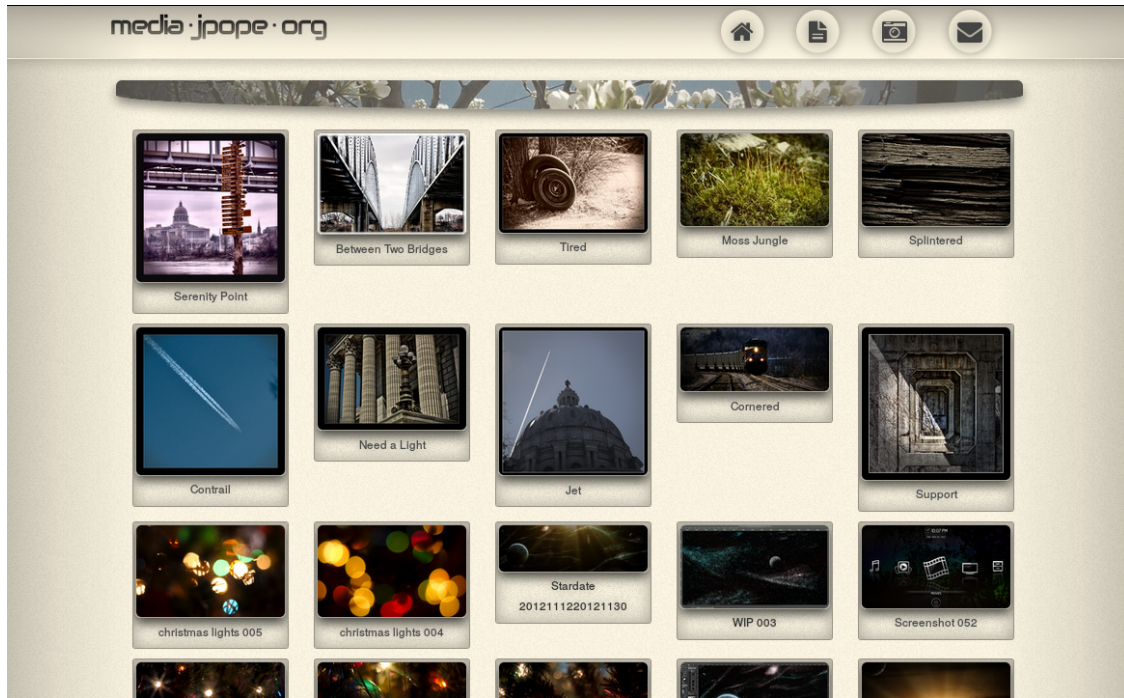
In ONE sentence, tell us about your project to strengthen the Internet for free expression and innovation.

MediaGoblin builds a decentralized, social media publishing system, and a set of tools any other project can use to be social *and* protect user freedom on the internet.

## 1.3 Who will benefit?

Who will benefit from what you propose? What have you observed that makes you think that?

Media authors – whether artists, journalists, individual users or organizations – gain an innovative publishing system without censorship, centralized surveillance, or corporate monopoly, on a social platform the user can customize and control. In place of being locked into centralized and vulnerable corporate media silos, users gain the enormous benefit of a world-wide independent media sharing network that can be extended to defy even national and language boundaries.



*MediaGoblin is flexible enough to fit the needs of whatever individual or organization is running it. If the project does not yet contain the features you need, it can be easily extended. Here a user has installed their own custom theme.*

## 1.4 What progress have you made so far?

After one year of crowdfunded development, MediaGoblin has issued 5 major releases and built the groundwork of an ambitious system that could reverse the trend of fragmentation into proprietary media silos, and empower the future of user-driven media communications on the web. Its active community of over 75 volunteers need to be augmented by full-time developers in order to implement the next key targets of MediaGoblin's roadmap such as federation and private media sharing. MediaGoblin would greatly benefit from more resources and is working hard to become a self-sustaining project, having launched successful community-funded crowdfunding campaigns while providing complete financial transparency related to the expenditure of those funds.

## 1.5 Successful outcomes

What would be a successful outcome for your idea or project?

MediaGoblin's success can be measured by the following criteria being met: complete and usable decentralized/federated media publishing software, the presence of well-functioning privacy features and private sharing, and growing public adoption with a multitude of MediaGoblin nodes.

## 1.6 Team and experience

Who is on your team, and what are their relevant experiences or skills?

Christopher Allan Webber is lead developer of the MediaGoblin project and provides the technical vision of the project, as well as the goblin drawings. He previously has worked as tech lead at Creative Commons and began development of Miro Community at the Participatory Culture Foundation. Deb Nicholson is community manager and additionally handles much of our communication. She has a background in social justice, currently works at the Open Invention Network, and previously worked for the Free Software Foundation. We also have a team of over 75 contributors from around the world that do everything from coding features to graphic design to documentation to translations.

## 1.7 Location

Madison, WI, USA

# 2 Questions and answers

## 2.1 Users and target customers

Who are the users or target customers of your project, and what have you learned from them so far? Please give specific examples.

MediaGoblin aims to be usable for anyone who is interested in publishing media on the web. However, we are particularly interested in targeting 1) artists who have a wide portfolio of media types to publish as social activists, 2) communities who are concerned about censorship and surveillance, and 3) those who want a media hosting platform that can be tailored to their individual needs. We are also interested in targeting galleries, libraries, archives, and museums.

MediaGoblin's early adopters are presently by fairly technically savvy communities or those who are more keenly aware of issues of technical freedom (including the Free Software Foundation hosting their conference footage from LibrePlanet on their own node<sup>1</sup>). However, we are working actively to expand beyond those areas. We have growing adoption in other spaces; MediaGoblin is presently working on a small grant to make the project more useful for archival and academic institutions, and we have seen at least one university adopting MediaGoblin, as the Rochester Institute of Technology is running their own node.<sup>2</sup>

## 2.2 Assumptions and testing

What assumptions are you making in what you propose, and how will you test them?

---

<sup>1</sup><http://media.libreplanet.org/>

<sup>2</sup><http://yacht.rit.edu/mediagoblin/>

We assume that artists, musicians, and other authors or publishing institutions of media are interested in a decentralized media publishing system with the features MediaGoblin provides. The large amount of discourse on the internet and in the press over the last year surrounding controversies of governmental and corporate censorship and surveillance suggest that many individuals would like to move to platforms that better support their freedoms, but feel as if they have no place to go. The best way to test these assumptions are to build MediaGoblin, make it easily accessible to use and run, while continuously seeking user feedback and talking to those in our target groups.

To get a better sense of what areas we are lacking in, we have also begun user experience testing sessions. Recently we ran an event called SpinachCon<sup>3</sup> to collect user experience stories from actual users. We have collected this data and are working towards acting on it. The event was successful enough<sup>4</sup> that the free software community diversity and outreach group OpenHatch<sup>5</sup> are looking to take it on as a recurring event.

Software design involves making many decisions based on assumptions, and MediaGoblin is no exception to this. Fortunately, MediaGoblin has the advantage of being transparent while encouraging user feedback, having an open bug tracker, and fostering community participation.

## 2.3 Getting the project out there

How will you get your project in front of the necessary people or organizations?

We intend to promote visibility of existing MediaGoblin deployments and also work on agreements with high profile and related communities and projects to assist them with MediaGoblin deployment. Thus, we can help build living examples of what can be done with MediaGoblin.

For those projects and individuals that can afford it and are interested in using the software but perhaps not maintaining a running node themselves, we are planning to offer paid hosting. We are also interested in this as a path toward sustainability of the project.

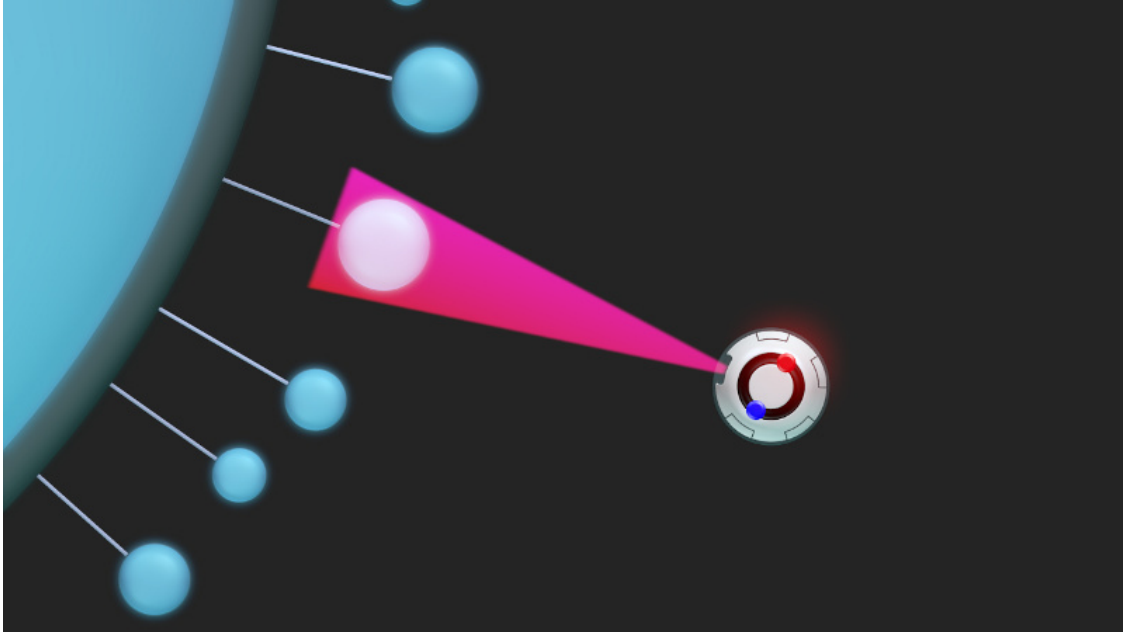
Additionally, MediaGoblin has put effort into building clear materials to make what MediaGoblin is easy to understand for newcomers, as well as the surrounding issues that MediaGoblin is attempting to address. We intend to continue to put resources into these efforts.

---

<sup>3</sup><http://mediagoblin.org/news/spinachcon-2014.html>

<sup>4</sup><http://openhatch.org/blog/2014/spinachcon-zero-a-huge-success/>

<sup>5</sup><http://openhatch.org/>



*MediaGoblin's messaging team works to build materials both to market the project and also to explain the underlying issues. This frame is from an animated video explaining the issues of automated censorship in a highly centralized environment.*

## 2.4 Addressing obstacles

What are the obstacles to implementing your idea, and how will you address them?

### 2.4.1 Technical obstacles

From a technical perspective, much of MediaGoblin already works, and while there are certainly many technical areas that need polish, refinement, and advancement, the core of MediaGoblin as a publishing tool functions well.

There is, however, one major piece missing: it is not enough just to "push media out there" in modern times with media publishing; users are accustomed to many social features such as favoriting media, adding media to collections, subscribing to channels and content, and so on. Adding these features to a decentralized environment is called "federation" and is a significant technical undertaking, one that we have begun to embark on but which will take significant effort to complete in a user-friendly way.

We have taken the "standing on the shoulders of giants" approach and are presently working towards implementing inter-site communication via a protocol called the Pump API.<sup>6</sup> We are adopting this to try to forge a future where many websites can be interoperable. It is possible that this assumption in protocol choice may not turn out to be correct, though we do believe that trying to collaborate with likeminded projects is the best way to move forward in advancing federation on the web. We also believe that if there are assumptions in

---

<sup>6</sup><https://github.com/e14n/pump.io/blob/master/API.md>

the protocol that are proven to not be ideal for a project such as ours which is media-heavy, we can collaborate with those authoring the specification.

### 2.4.2 Social and adoption obstacles

Much of the issue of social challenges has already been addressed in this proposal, but it is worth noting that for all truly decentralized web services, gaining adoption is difficult because of the challenge of network effects. While media publishing systems have less of a challenge than purely social networking sites,<sup>7</sup> users have come to expect the ability to interact with other users in relation to said media. If their family, friends and colleagues are not on the same network as them, this transition can be difficult. As such, we would like to make MediaGoblin as easy to migrate to as possible by providing easy transitional tools to migrate one's data from other larger sites. Making MediaGoblin easy to use, well featured, and well marketed will all of course be necessary as well.

### 2.4.3 Community obstacles



*MediaGoblin has both strong leadership and a sizable community which is tightly integrated into the project. Here a group of contributors are collaborating on improving the codebase at a hackathon at the GNU 30th event.*

<sup>7</sup>We say that MediaGoblin has less of a challenge than a more "pure" decentralized social networking system (such as one challenging Facebook) because the social network is not the only or even primary element of value being delivered, the media is. It is easy to share a video or an image with friends even if they do not have an account on that site or network.



One challenge that many free and open source software projects suffer from is that once the project grows large, the community suffers, or perhaps the primary organization funding the software chooses to make it proprietary. We have sought to avoid this issue from the very beginning of the project. For this reason, the project is under the stewardship of a nonprofit (the Free Software Foundation) which has a mission to preserve software freedom and maintain its integrity, and which has a strong track record of doing so. Additionally, we have chosen the GNU Affero General Public License because of its requirement to give back modifications to the community. Since there are many contributors to the software and no requirement for copyright assignment, it would be very difficult for someone to lock down the codebase by making it proprietary.

Despite these safeguards, community integrity remains on the forefront of our thoughts and actions. We have monthly contributor meetings, project decisions are made out in the open in a public chatroom, on mailing lists and on the bug tracker. While MediaGoblin is a project with strong leadership, it is also a project that emphasizes both transparency in decision making and community.

We also actively encourage and recognize contributions of all kinds... not just code, but also graphic design, documentation writing, translations, marketing, and so on. We work hard to make all contributors feel welcome, no matter what their skills are. We are also committed to diversity and are proud to have participated in projects such as the Outreach Program for Women, and will continue being proactive in building a diverse and welcoming community.

## 2.5 Project cost and expenses

How much do you think your project will cost, and what are the major expenses?

The large majority of MediaGoblin's costs will go into development of the software itself, which means paying developers and designers. (We have some advantage in that some of the administrative overhead can be reduced through our fiscal sponsorship through the Free Software Foundation.) We can make use of any amount of funding we can get, however the amount of funding brought in affects what kind of work we will be able to do. More details are given below in a range of three potential funding levels.

### 2.5.1 Basic plan budget: \$125,000

- MediaGoblin as a well functioning project. Full-time developer resource to focus on the programming of the project.
- Community management of the project, including reviewing and stewarding community contributions. This will be a full time management plus developer resource.

### 2.5.2 Polish MediaGoblin for everyone budget: \$250,000

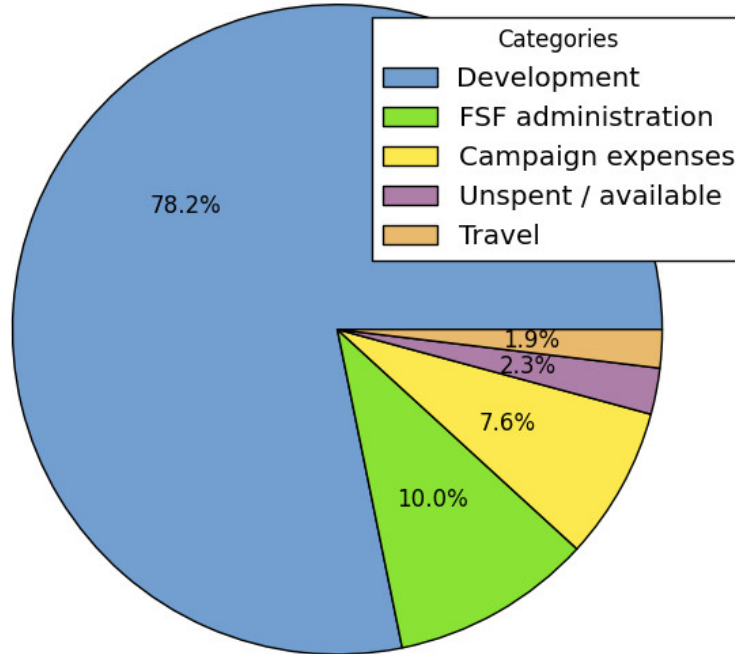
- MediaGoblin as a polished, well functioning, well featured project. Multiple full-time developers to focus on the programming of the project.

- Graphic design and user experience resource to make the project smooth, pleasant, and easy to use for everyone.
- Community management of the project, including reviewing and stewarding community contributions. This will be a full time management plus developer resource.
- Actively reach out to interested institutions and assist them with adopting MediaGoblin.
- Good marketing materials explaining MediaGoblin and why it matters
- Exploring MediaGoblin funding sustainability by partnering through interested resource to provide MediaGoblin as a service.

### **2.5.3 Strong sustainability budget: \$500,000**

- MediaGoblin as an amazingly polished, well functioning, well featured project. Multiple full-time developers to focus on the programming of the project.
- Graphic design and user experience resource to make the project smooth, pleasant, and easy to use for everyone.
- Community management of the project, including reviewing and stewarding community contributions. This will be a dedicated and full time resource.
- Actively reach out to interested institutions and assist them with adopting MediaGoblin.
- Actively working to bring the general public into MediaGoblin adoption.
- Strong marketing materials explaining MediaGoblin and why it matters.
- Multiple paths to exploring MediaGoblin sustainability and a resource focused on developing future paths to sustainability and self-sufficiency.
- Enough of a funding buffer to ensure that MediaGoblin has time to properly develop to the point of achieving sustainability.

### 2.5.4 A note on financial transparency



*Summary of MediaGoblin expenses in 2013.*

Note that it is already possible to get a sense of how MediaGoblin spends its funds. We have done a full financial transparency report<sup>8</sup> on past expenses, and we intend to continue in this direction.

## 2.6 Data collection and implementation

Who will do the data collection described in your proposal?

The majority of the project's data is actually user-submitted data, i.e. the media that users are publishing to their MediaGoblin nodes. Thus the vast majority of development is aimed at making the software work well so users can do this.

Additionally, we do data collection in the sense of bug tracking, our community wiki, and a development process which allows for public contribution. This helps us determine what we are currently being successful at and what we are lacking in.

We are also interested in both user experience testing (which we have begun doing through SpinachCon) and market research to better understand how users react to MediaGoblin and how to improve the software to those ends. If we receive funding, we will allocate some resources towards such projects.

## 2.7 Learning from projects

What have you learned from other projects in this space that informs your approach?

---

<sup>8</sup><http://mediagoblin.org/news/financial-transparency.html>

One lesson we have taken in observing other projects is in the construction of our community. When Diaspora first made its announcement during its initial crowdfunding campaign, it did a successful job of capturing the hearts and minds of many people. Unfortunately, the developers decided to do development privately and not until later did they open up to show their current progress. We feel this was a mistake: there was an opportunity lost to build a community from the excitement of this group of people. Instead when Diaspora's code was first opened, there was much frustration and confusion in that the codebase was not as well developed as many had anticipated.<sup>9</sup> Diaspora now has a much stronger community, but there could have been peer review by domain experts and an opportunity to build a community if all code had been built publicly from day one. For this reason MediaGoblin has always had a public and transparent development process.

We have also learned by paying close attention to what has happened between the decentralized social media systems StatusNet and Pump.io (which implement and are the main examples of the federation protocols OStatus and the Pump API, which we are using for federation, respectively). StatusNet achieved a significant amount of success as in terms of user popularity. The project was mostly fairly easy to use from an end user perspective. However, most users stayed on one very large server hosted at identi.ca, and there was not much success with true federation in this sense. Additionally, the federation protocol OStatus, while designed to be generic, was not well adopted by other software projects, partly because it was fairly difficult to understand, being cobbled together from multiple existing standards. When efforts transitioned to a new project named Pump.io, the new software instead switched from a user focus to a developer focus. The Pump API is much easier to understand for developers, and the goal of Pump.io is to become infrastructure to make developing federated software easier, but the software is harder for end users to make use of. Thus, we are taking the approach of making use of the developer-friendly Pump API and intelligently incorporating these technical efforts into our software to simplify efforts around decentralization technology decisions, but much of our focus will be around making these features easy to use for end users.

Lastly, MediaGoblin is itself founded by individuals who have worked extensively in many of these spaces. Chris Webber's experience as tech lead at Creative Commons and work on Miro Community and Deb Nicholson's employment history at places such as the Free Software Foundation and the Open Invention Network, coupled with years of community activism and nonprofit work, have greatly informed the direction and mission of the project. We work hard to incorporate the skills and experiences of our broad community, and stay in close contact with other projects at the forefront of the fight to keep the web decentralized.

Our community is strong, our project is growing, but it lacks significant resources. We believe with the support of the Knight Foundation we can bring MediaGoblin to the next level. Together, we believe we can bring freedom back to media publishing on the web.

---

<sup>9</sup>See "'Open Source' is not a Verb; thoughts on Diaspora" at <http://yergler.net/blog/2010/09/23/diaspora-community/> for a good summary of this situation.