# Why (and how) public institutions should release more of their hardware designs as Open-Source Hardware*

Carlos Serrano[1] and Javier Serrano[2]

[1]Lawrence Berkeley National Laboratory (LBNL), Berkeley, USA
[2]European Laboratory for Particle Physics (CERN), Geneva, Switzerland

February 2020

## 1 Prologue

Imagine a world in which public institutions create a common pool of knowledge and technology which *everybody* is free to access. A world in which commercial companies build upon that commons to add value and generate revenue, some of which is re-injected into the public sphere to fuel the continuous growth of the commons. Imagine we had a way to make this process very efficient. Just think about the amount of public institutions out there: universities, laboratories, government agencies... In this opinion piece, we argue that with coordination and appropriate support in terms of legal environment and business models, this vision can result in an enormous boost to innovation and welfare. If this sounds compelling to you, read on.

*Both LBNL and CERN have public statements on their mission, which includes the maximisation of positive impact of their developments on society. The *means* employed for reaching those goals differ, of course, as a function of the circumstances of each development and, to a lesser extent, are also subject to opinion. This article presents the personal opinions of the authors, hoping to foster productive discussion, and is in no way intended to represent an official communication from either LBNL or CERN.

# 2 Introduction

Software is omnipresent in today's modern societies. Wherever we look during a normal day, software is at work, helping us brew coffee in the morning, take our kids to school, work, meet friends and choose a good movie to watch in the evening. Software is everywhere, from schools to airports, from hospitals to stadiums. Everywhere.

The pervasiveness of software implies there must be many people developing it. Software development was democratised in the 80's and 90's of the last century. In the space of a few years, it became much easier to write code and produce a useful binary out of it. Computers also became cheaper, and these two effects allowed many people to become software developers, for work, fun or both.

There is a special quality to things which are both useful and easy to do. One example is crossing the border between France and Switzerland; it is easy to do, and it is very useful for many people who do it every day to go to work, visit or for any other purpose. Because it is both easy and useful, society as a whole benefits greatly if it can be done, i.e. if appropriate measures such as law and infrastructure are put in place so people can easily cross the border. An example of not-easy-and-useful is travel to Jupiter; although it would stand as arbitrary and somehow awkward, relatively few people would complain if it were forbidden.

Sharing software source code was not widely useful in the 50's. Few people had computers, and software was typically very specific to a given computer type, so the chances were slim for the recipient to be able to run the software (s)he would have received. It was also not very easy, typically involving costly copies and physical transport to the destination. With the democratisation of software development and the growth of the Web in the 90's, sharing software became very easy and very useful, so the case for putting appropriate measures in place to facilitate it became all the more compelling. The start of the GNU project, whose aim was to develop a free-as-in-freedom operating system, along with the publication of the GNU General Public Licence (GPL) are commonly considered the foundations of the Free and Open-Source Software (FOSS) paradigm. They established best practices for sharing source code and binaries effectively, and the adoption of these practices by projects such as the Linux kernel resulted in a revolution. Today, GNU/Linux and other FOSS projects power most of the infrastructure we use every day as we interact with friends, colleagues, vendors and other actors on the Internet.

It is our contention in this paper that hardware development is currently undergoing a similar democratisation process. Design tools exist, themselves FOSS, which allow non-experts to design hardware of a complexity which would have needed a small company a few decades ago. Manufacturing of products based on those designs has also become much easier and cheaper. While the production of a physical good will always be more costly than producing a binary file from a set of sources, the exchange of hardware design files themselves is just as easy as for the software case. Even if the easy-and-useful hypothesis is fulfilled to a lesser extent than in software, the case is still more than compelling enough to invest in appropriate ways to share hardware designs.

We also want to make a special point about public institutions. A private company represents the interests of a subset of society, namely those of its owners or stockholders. As such, it should not be surprising that its rational economic behaviour differs in some cases from that of a public institution, which represents the interests of a much bigger group of people: the complete population of a city, region, country or set of countries. In particular, in some circumstances, a private corporation might decide it is in the interest of its stockholders to not publish the source code of a program, or the design files for a hardware development, so as to exclude their potential competitors from this knowledge. In a similar situation, a public institution could judge that it is in the interest of society as a whole to publish these developments, so that free access to this knowledge can be used by companies and individuals alike to build upon as they see fit.

So, if the case is so compelling, why aren't most public institutions publishing the majority of their software and hardware developments as open source? There are, in our opinion, a number of factors, including inertia and negative external incentives. In the following sections, we detail them and we propose a road map to overcome these obstacles and unleash the huge potential contributions to society which can be brought about through a combination of preference for open source and a modicum of organisation among public institutions. If you are a practitioner of another discipline, e.g. software development or mechanical engineering, feel free to draw your own conclusions as to how much of the following applies to your case.

## 3 Current state of affairs

Our experience is in the community of developers of controls and data acquisition hardware for large experimental physics facilities, so this article is to be interpreted as applying broadly in that context. Discussions with members of other communities of publicly-funded hardware designers though, have convinced us that their issues are very similar.

A quick clarification is in order here regarding the type of hardware we are talking about: we do mostly Printed Circuit Board (PCB) design and Field Programmable Gate Arrays (FPGA) using Hardware Description Languages (HDL). In this realm, the easy-and-useful-to-share hypothesis applies fully. It applies less in the case of Application-Specific Integrated Circuit (ASIC) design, but it is very likely a matter of time that the democratisation we describe above reaches this domain as well.

## 3.1 Proliferation of similar developments

Controls and data acquisition are very generic tasks. For example, digitising an analogue waveform at a sampling rate of 100 million samples per second is something that many people need. Equipment designed to cater for that requirement does not need to know if the analogue signal comes from a telescope or a particle detector. Yet there is a bewildering variety of designs out there doing the same thing. The same happens for synchronisation systems, analogue waveform generators, high-speed communication links and many other basic building blocks of a control system. It is sometimes argued that independent development is necessary for scientific discoveries to be reproduced by at least two groups in a completely uncorrelated manner, discarding any possible systematic effects due to design errors in shared devices. This is true in principle, and would argue for a certain degree of variety, beyond which the extra cost for society is difficult to justify.

Unnecessary multiplication of efforts has other less visible costs beyond the obvious one, namely that society gets less and less for its money because the second Analogue-to-Digital Converter (ADC) board with almost identical specifications adds little value to the first one, and so the trend continues until the 100th one and beyond. Other costs include missed opportunities for sharing maintenance, economies of scale and improvements coming from communication among design groups. The software world is plagued by the same problem. On the one hand, developing software is accessible to more people, and that results in a tendency for more proliferation of redundant solutions. On the other hand, sharing developments is more established than in the hardware world, which compensates to a certain degree this unnecessary multiplication of efforts.

Why does this happen? These are the main reasons we have identified:

- Lack of information. Some people just don't know that a very similar development is ongoing in some other institutions or (for big places) even in the same one.
- Risk management and/or lack of trust. Some groups may decide to go solo instead of embarking on a collaborative effort where they don't

have a direct hierarchical line which would enable them to tell each designer what to do, easily change plans, etc.

- Low priority on Knowledge and Technology Transfer (KTT). In many cases, financing is tight and so are deadlines. Designers produce something, but there is a perception that preparing it for transfer to the outside world would incur extra costs, both in money and time. This includes things like proper documentation, design reviews and licensing, but would also typically involve a more-or-less sizeable rescoping of a project, because making something really useful for others often needs to tackle the solution to a more general problem than that at hand. If hierarchy does not give enough importance to this step, designers may naturally choose not to publish. KTT offices have as their main goal to maximise the impact of technological developments on society. Over the last 20 years, organisations around the world have agreed that this is a very good idea, and accordingly established KTT groups with this commendable goal. In many cases though, the budgets to support these activities do not match the ambitions set out in the mandates of these offices.

- Pernicious incentives. Research groups are made to compete. Some of them naturally develop a tendency to not share their developments, either until very late, or at all. They do this out of fear of helping their competitors catch up with them and outperform them. This is a very serious issue. Sometimes the very existence of a research group could be compromised by this sharing. Of course, the problem arises when one looks at the situation from the vantage point of a tax payer financing all these parallel efforts basically reproducing very similar designs.

- Another pernicious incentive does not directly apply to designers but rather to the KTT groups in their institutions. These groups are being increasingly asked to self-finance, or otherwise generate revenue for a number of purposes. Monetisation of proprietary developments is often perceived as more direct than its open source counterpart, so this can push these groups to prefer proprietary licensing arrangements whereby they can generate revenue out of royalties.

Whether the list above is complete or even correct, the fact is that there is an awful amount of publicly-funded development out there with very little (if any) impact outside the walls of the respective institutions. Developments often solve the problem they were intended to, and nothing else, foregoing opportunities to contribute more to economic activity and global welfare.

## 3.2 The funding agencies conundrum

One other reason why groups in public institutions do not share much publicly is the perception among certain managers that the results of their research should benefit exclusively the subset of people who have financed it. In the case of a city, it would be its citizens. For a regional institution it would be the individuals and companies from that region, and so on. For national and international laboratories, one often hears the objection in terms of the unsuitability of offering all our knowledge and technology for free to a very particular large country in Asia. We guess that the reasoning behind it is that this knowledge can then be used by these external actors to harm our economy, by offering very cheaply what companies in the financing states could have offered to the public at higher prices, using the higher margins to properly pay their employees and reinvest in innovation. We explain in section 4 why we think this reasoning is incomplete and can lead to decisions which forego important opportunities to benefit our societies.

## 3.3 Patents

The patent system was created to provide incentives for creators to disclose the details of their inventions. In exchange for a temporary monopoly on the economic exploitation of an invention, the patentee is expected to publish its details. Therefore, the patent system was initially designed to encourage sharing.

The current trend to patent inventions in public institutions arouse from a combination of two factors, which we have already mentioned: the expectation that KTT offices self-finance (totally or partially) and the wish to be able to favour a subset of companies (typically in the funding states/regions) by giving them exclusive or non-exclusive access to a technology. Regarding the former, its origin probably dates from the 80's of the last century, when a large subset of public opinion in the western world became increasingly convinced that the private sector was necessarily more efficient at assigning resources than its public counterpart. It therefore followed that the public sector should be made thinner, and adopt competitive practices wherever possible. Academic competition had already been a basic tenet of public research for ages, but it was felt that adding economic competition to the mix would make labs ever more productive.

Seen from that perspective, patents should be a basic component in the toolbox of any KTT office. Their use allows to very finely control the dissemination of a given technology, favouring only those actors we deem fit. It also allows the generation of revenue through royalties, which are then invested back to create more innovation. From a conceptual point of view, patents make public institutions resemble their private counterparts. These newly acquired assets are as exclusive as a good can get, and are therefore ideal for a public institution wishing to enter the markets and play the game of supply and demand.

The analysis above is an attempt to explain why so many laboratories and universities engage in the filing of patent applications. We will explain later why we think this chain of reasoning is fallacious, but we would like to be clear about one thing: it is our conviction that the people who push for such practices in public institutions do it because they think that is the best possible use for public money, and they may be right, i.e. it is possible that privatising knowledge generated with public money is a winning strategy for society in many cases. We respectfully disagree, but we may be wrong.

# 4 Why sharing is (almost always) better

Let's get something out of the way first: sharing *is* the natural state for an individual and for an organisation. When you discover something, your first instinct is to share the news, and to explain to others what you found. The current widespread practice of non-sharing is a constraint, artificially imposed, sometimes for good reason. One should therefore not have to justify sharing, but rather non-sharing. What's more, many of our public institutions have sharing enshrined in their mandates. In the case of scientific labs, reproducibility of discoveries, a basic tenet of science, is directly harmed by opaqueness and proprietary developments. Because people like to share in general, you will have an easier time hiring brilliant developers if you do not constrain their ability to exchange information with others, inside and outside your organisation. Besides, these developers have a better chance of being familiar with your technologies if they are open source, and they will continuously learn on the job as they interact with other brilliant people around the world. From that perspective, investing in open source is investing in the people you work with. This is not a theory. It is something we have seen repeatedly in our practice.

A quick disclaimer before we get into a systematic discussion: in the interest of conciseness, and considering that this document is already quite long, we do not fully develop some of the arguments that follow. We hope the explanations we give are enough and we remain available for further discussion if needed. We may complement this document in the future with a more academic one, heavier to read but with an effort to support all our claims with references from the extensive literature on the subject.

OK, so let's look at situations in which an institution might decide that not sharing is the best option. For example, the funding agencies conundrum we mentioned earlier. Let's suppose we really want to favour local industry and therefore decide to patent a given technology, or otherwise make it available only to a set of selected actors in the hope that it will become very popular and these companies will be able to generate great revenue and contribute to employment and welfare in the process. This is all great, except experience shows that it very rarely works. Most of the developments published as open source have openness as one of their key (often the main) selling points. Openness makes a product more appealing to the end user[1] because of the almost-zero risk of vendor lock-in in that case. This in turn makes the market bigger, sometimes beyond the critical mass at which a product can become profitable. Conversely, many private deals fail because the market for those products never gets off the ground. They are "one more signal acquisition system" or "one more signal analysis software suite". Nothing really sets them apart from many other competing developments. Openness is that key differentiator which can make publicly-funded developments attractive for the end users.

We all know of examples that succeeded using the proprietary paradigm, some spectacularly so. For example the MP3 coding format for digital audio, largely developed by the Fraunhofer Institute for Integrated Circuits, which brought it millions of euros in royalties. It is clear that looking at this from the perspective of the institute, and imagining for a second that the institute were private and had it as its main mission to generate revenue, this is a great success. One could adopt a different perspective and examine what the impact on society was and whether that impact would have been lower or higher in case Fraunhofer researchers had decided to open-source the development. Of course, the Fraunhofer partial self-financing model skews the choice towards the proprietary end, but that model could be challenged as well to begin with. Even this poster-child of proprietary dissemination is subject to controversial opinions. We should stop now and remind the reader that the case we just mentioned is extremely rare, even in institutions like Fraunhofer, which have as an explicit goal to generate this kind of technologies. Furthermore, note that Fraunhofer's non-discriminatory licensing policy was key to the success of MP3, so this is not a case which would il-

[1] This statement should be qualified. We are mostly talking about expert users here, such as ourselves in our respective institutions. It is true that other selling points, such as ease of use, are more important for the average user, but it is also true that we are seeing an increasing number of examples in less specialised settings where the lack of awareness of the lock-in problem ends up affecting the consumer in ways as diverse as cost or lack of privacy.

lustrate a means to favour this or that country or set of countries. Let us emphasise again that MP3 was a great success, both technically and in terms of impact. Its open-source alternatives at the time were much less successful on both accounts. Being open does not automatically make a project great.

The question is: if a project is great, what is the best way to optimise its impact on society? We are taking the case of MP3 as one of the most successful proprietary developments we can think of, and we still think that the impact would have been even higher had it been open-sourced. At the very least, it would have preempted many alternative open-source developments, such as the Ogg format, which also took a considerable amount of public and private resources and were ultimately less successful.

One example of the process of open-sourcing a technology which was great to begin with is BSD. The clean design and technical superiority of Unix was already acknowledged in technical circles from early on. In its BSD open-source incarnation, it arguably played a key role in the development of the Internet as we know it, and became the basis for Mac OS X and its successors. And then GNU/Linux took the impact to another level, some people argue that largely thanks to the adoption of the GPL licence.

Luckily, the vast majority of developments in public institutions which ultimately had a big impact on society were free to use by everyone. From the discovery of the structure of DNA to the invention of the Web, we even have a hard time imagining what would have happened if they had been licensed under a proprietary paradigm. Jonas Salk's answer to a journalist when questioned about who owned the patent for the polio vaccine vividly illustrates this state of mind: "Well, the people, I would say. There is no patent. Could you patent the sun?".

Let's now look at a recent example which we could more easily counterpose to the MP3 case: the development of the RISC-V instruction set architecture (ISA), started in 2010 at the University of California in Berkeley. RISC-V allows designers to come up with new processors, all sharing the same ISA, without any constraints. RISC-V was designed from the ground up with the explicit goal of having an ISA completely unencumbered by patents. The main ingredient to its success is openness. This is universally agreed. Had the original developers decided to release it under a proprietary licence, this would have been "one more ISA". Instead, RISC-V has gone on to become a revolution in computing and embedded systems, even playing a big role in the digital sovereignty debates in Europe and elsewhere. But wait a second, so does that mean that this development paid with American taxpayer money is actually helping Europe? Yes. Is that bad? Will Europe now use the technology to harm US economic interests? Absolutely no. The positive impact on the US economy arising from the RISC-V revolution is already evident. It would not be there if RISC-V had become a proprietary niche. Chances are that Europe would have come up with its own patent-free ISA if RISC-V had not appeared. So yes, RISC-V is helping Europe, and yes, it is great for the US economy. Go figure!

In practice, we see that when public institutions exclude our friends in the far east from the knowledge they generate, they also end up excluding lots of the taxpayers who paid for their work, including many small and medium enterprises, in a tragic outcome arising from using inappropriate tools to solve a problem which may not even exist. Here it is probably worth mentioning that big players have the means to reverse-engineer most developments, or even restart them from scratch once they are given an idea. Keeping design information secret is typically not an effective strategy against this type of actor. On the other hand, we have found that public institutions can still favour local companies after having open-sourced a design. Not everything is in the source: design intention, ideas that didn't work, plans for the future, are all examples of bits of information which can easily give a leading edge to one or a set of chosen partners. Again, we have seen this at work, and the fact that the biggest RISC-V company is in the Berkeley area is no accident.

Open-sourcing provides, as a side effect, a benefit to third parties. Keeping things secret from third parties means, as a side effect, to exclude many of our tax payers from this knowledge. Between these two side effects, the former is clearly to be preferred.

To summarise, it is our contention that many (most?) hardware developments are of a type where the goal of generating revenue for the public institution through proprietary deals will *compete* with the goal of having a positive impact on society. This is all the more unfortunate considering the fact that the number of patents or the amount of royalties received often feature prominently in official metrics used as indicators of impact, which is what most public institutions are ultimately mandated to prioritise.

OK, so we have already dealt with the material in section 3.2 related to the use of proprietary licensing to avoid helping people who did not pay for our developments. Let's now focus on the list of reasons not to share in section 3.1. This is basically a list of disincentives created by the fact that, somewhere along the way, the local interests of managers and developers have become misaligned with those of the taxpayers who pay for their work. Seen from the perspective of the latter, all this unnecessary duplication of work is a waste. It is also a pity that the work of so many bright people has so little impact beyond solving their particular problems at hand. Regarding the objection that open-sourcing requires extra effort, it is often the case that those things required for effective open-sourcing are good engineering practice in themselves. So the argument can be turned around, arguing that open source forces us to add quality to our developments, through careful design, documentation, testing plans, etc. This is especially important in institutions where developments have support lifetimes which often exceed the typical duration of an employment contract, leading to problems in keeping the knowledge needed to guarantee trouble-free operation and evolution of the designs. If the goal of this section is to convince the reader that it is better to open-source, there should be little else to add, provided we agree that "better" is to mean whatever maximises positive impact on society. We will see in the next section what can be done about this.

And finally, a few words on patents. Some open source advocates take a very maximalist position, arguing that the very notion of *owning* an idea is preposterous. Let's for a moment accept that the patent system grew off a societal need and plays a positive role in some cases, even if it is evident that it is severely broken and probably doing more harm than good these days. The main argument one hears from people defending the current system is that it would be "impossible to change", which reveals how much intrinsic value they assign to the basic principle of patents. If this is the case, i.e. if this is a bad system which is a necessary evil, then surely those who can should avoid using it, right? And this is the case of public institutions. There is in principle no need for them to use patents, and the only reason they would is if their funding agencies left that as the only option for them to have adequate funds, i.e. if governments decided that public institutions should increasingly resemble their private counterparts. In practice what we see is that private companies are great at what they do, and public labs are very bad at trying to behave as a private company, not to mention the potential issues with competition law one could bring up. So you have lots of efforts at privatising knowledge, including patenting, which ultimately result in lots of wasted time and money. More resources assigned to lawyering[2] and less to engineering and science. Some KTT offices have a "patents as a last resort" policy: they will do them if a development cannot be completed without external private investment and the investors require some kind of exclusive right. This is justified insofar as the alternative is that the development does not see the light at all, but should be exceptional and usually just means

[2] Although different organisations are able to cooperate on proprietary developments, these generally require a complex cooperation and knowledge transfer agreement, and this is even more likely to be the case where the cooperation may lead to patentable inventions, where significant controls on retention of confidentiality are required to ensure that any potentially patentable invention is not inadvertently made public (and lose its ability to be patented through lack of novelty). The complexity and time taken to put such an agreement in place is itself a disincentive to cooperation. The FOSS paradigm provides a much simpler low-friction way for entities to cooperate.

that the public institution in question should be better funded.

So hopefully we all agree by now that having public institutions release more of their work as open source is good. Does that mean that they will not create economic activity, that commercial companies will not be able to capitalise on these developments and generate revenue, contributing to global welfare? No, of course not. Please read on, we're almost there!

# 5 The way forward: a call to action

It is quite clear that there is room for improvement in how public institutions develop and disseminate hardware designs. What can we do to make things better? One specificity of hardware is that commercial companies are not optional, i.e. you need at the very least somebody who will manufacture the hardware and sell it to you and others. This is fortunate, because companies are natural actors when it comes to generating economic activity around a development. On the other hand, companies cannot monetise open source. This statement may sound a bit controversial, but it isn't: it is clear that one cannot sell what everybody can get for free. You cannot sell air, and you cannot sell open-source code. Even companies whose main object is FOSS, such as Red Hat, make money off something different than the source, namely the guarantee that "things will work". This, in turn, is supported by the experience and skills stored in the brains of their developers, which are as proprietary as it gets. There are many other business models in FOSS; all of them consist of selling something which is not the source. And they all rely on the availability of a solid FOSS foundation which adds value to whatever service you want to implement on top. The incentive to contribute to this commons is strong, especially if rebuilding it from zero is deemed risky and costly. Public institutions, on the other hand, can have this open-source foundation as their ultimate object, because they do not need to monetise developments as commercial companies do, at least in principle.

Many things can be done to improve on the current state of affairs. In the following sections, we describe some of them, going from what individuals can do without any external help, to more ambitious actions only reachable at the government level. In the middle, inspired by the case of FOSS, we identify what we call the foundation/consortium level, which would only need moderate effort and a bit of collaboration to put in place and can bring about great benefits.

## 5.1 Helping at the individual level

OK, so you are an individual developer, or a small team in a public institution and you are convinced by the arguments in this document. You want to open-source your designs. Great! Well, just go and ask your supervisor, discuss with your KTT office, trigger debate if needed, put the subject on the table. You may be surprised to see that their views are more aligned with yours than you thought, and if they aren't you may want to organise a discussion on the subject, and bring in colleagues from other institutions, including KTT officers, so they can help illustrate the process of positive impact on society with real examples.

Another way to bring about change is to publicise your open-sourcing efforts in conferences and other gatherings. We all have our reference meetings, where we share our latest findings and plans with people doing similar things in other places. These are ideal occasions to promote collaboration through open source. Projects which publish all their details, with appropriate licensing for those who want to see, modify and republish under an open-source paradigm, have a competitive advantage, in an almost-darwinistic sense, over projects which are more proprietary in nature. Sometimes it takes a relatively modest initial effort to get the ball rolling, and through the magic of open-source projects can develop into very ambitious endeavours, generating in the process lots of positive impact on society.

Finally, there is one more way in which you can bring about more openness, which is to favour open source when you purchase hardware (and software). We discuss this at a higher level in section 5.3 but there are many institutions which will happily delegate this decision to the individuals purchasing equipment.

## 5.2 Activity at the Foundation/Consortium Level

It is in the ethos of open source, and we can see it at work in many examples, to capitalise on past developments, to improve designs and add value to them incrementally. Many people have worked on many technologies in the past, and published them as open source. It is therefore very often the case that projects do not start from scratch. Similarly, when one looks for a solution which will allow us to marry the interests of society with those of private companies, and to find ways of generating positive incentives for both public and private actors, one can build on top of many existing efforts. In particular, the FOSS world has seen the emergence of foundations and other bodies which act as hubs to house projects, and provide a solid legal basis and guidelines for governance. They also provide a basic economic model, which each project is free to complement in order to come up with a sustainable environment. Two salient examples of such institutions are the Linux Foundation and the Apache Software Foundation. There are other, more grassroots-like, actors such as the Software Freedom Conservancy and Software in the Public Interest.

We need a similar kind of institution for hardware. We can create it from scratch or ask one of the existing bodies in the Open-Source Hardware (OSHW) world. Good candidates could be the Open-Source Hardware Association (OSHWA) and the community organising the Gathering for Open Science Hardware (GOSH). This is a non-exhaustive list of services this institution could offer to our community:

- Provide a template for the governance of projects, as well as their legal foundations (including licensing of designs) and basic business model.
- Provide a meeting ground for public institutions, private companies and individual developers, with compelling incentives for each of them to contribute to projects in a sustainable way.
- Fund research on unsolved problems, such as the tracking of impact on society without compromising openness.
- Provide information to decision-makers in funding agencies and public institutions so they can see the benefits of contributing to the financing of OSHW projects and they can in turn come up with appropriate incentives for the employees in public institutions, fighting the disincentives listed in section 3.
- Provide education, including best practices for sharing designs, both on the technical and legal fronts.
- Foster developments which will allow easier and more efficient sharing, such as common formats for design files.
- Give visibility to good open-source projects to reduce the chance of people inadvertently re-inventing the wheel.
- Provide a platform for gathering societal needs and matching them with existing OSHW technological developments, facilitating the organisation of collaborations to customise solutions or launch new projects as required.

There is also a more controversial subject that this institution could help with, namely that of the generation of revenue by public institutions to fund their KTT offices and reinvest in more innovation. One could discuss whether asking KTT groups to self-finance, totally or partially, is justified or productive. But if it is deemed appropriate, then research should be conducted to find the best ways for generating revenue without giving up on any of the basic tenets of open source. Consortia typically provide good examples for how to do this. Membership fees can be stored in a common account and distributed following a set of rules, and then the main problem to solve is how to make membership attractive. There are many things one could offer to members without touching openness, such as the ability to use a given trademark, benefiting from the marketing efforts of the consortium in general, early access to design documents, or the possibility of steering the direction of development to a bigger or lesser degree. Access to developers for training or consultancy could be another incentive for membership.

So let's start the discussion in OSHWA, GOSH and any other community out there which could be a good candidate to launch this effort.

## 5.3 The government level

Public institutions also have a very important different role. They often act not as developers/creators but as users/customers looking to purchase a solution. We have discussed extensively the OSHW supply side, but there is a demand side which could easily be excited by giving preference to OSHW in purchasing departments of public bodies. The benefits of open source when it comes to purchasing have been extensively documented in literature and all well-known: lack of vendor lock-in, much less problems with obsolescence (programmed or otherwise), less concerns about security (e.g. trapdoors) and the ability to easily collaborate with companies in design tasks to name a few. This information, which is common knowledge in designer circles, needs to make it to the management and purchasing spheres in organisations. Developers can help, and so can the foundations and consortia we described above. A complementary push from the funding agencies would also be very useful. It is becoming abundantly clear in these last years to governments in the EU and the US, to name two outstanding cases, that open source should play a preferential role in public procurement. Public institutions can also be beta-testers or initial users/customers for projects which need a certain critical mass to get off the ground and self-sustain. This role as a "tractor" institution is not new. We just mention it here for completeness.

From the supply side of things, funding agencies could request that publicly-funded projects release their designs as OSHW. In some cases, this could be tricky, either because secrecy is a design requirement (e.g. in military applications) or because a large part of the investment comes from private funding with an associated requirement for exclusive rights. But OSHW-by-default, with the requirement to justify any departure from that base policy, looks like a very realistic measure to put in place.

The combination of both supply and demand policies described above would have a huge positive impact on society, first and foremost as direct economic activity, but also as a strong message that public institutions (e.g. labs and universities) would then interpret as an incentive to carry on these policies to their local realm.

# 6 Conclusion

This brings us to the end of this document, whose main purpose is to put some ideas on the table and trigger debate. It is our conviction that hardware development in public institutions could be done more efficiently and have more impact on society. We try to go beyond a mere assessment of the situation and suggest a way forward, in the hope that at least some of you will find it interesting enough to help us improve our proposal and trigger concrete actions for real change.

# Acknowledgements

Thanks to all reviewers of the initial drafts, who supplied interesting complementary points of view and suggestions. They helped us clarify and improve the text. Any remaining mistakes, conceptual or otherwise, are exclusively our own.