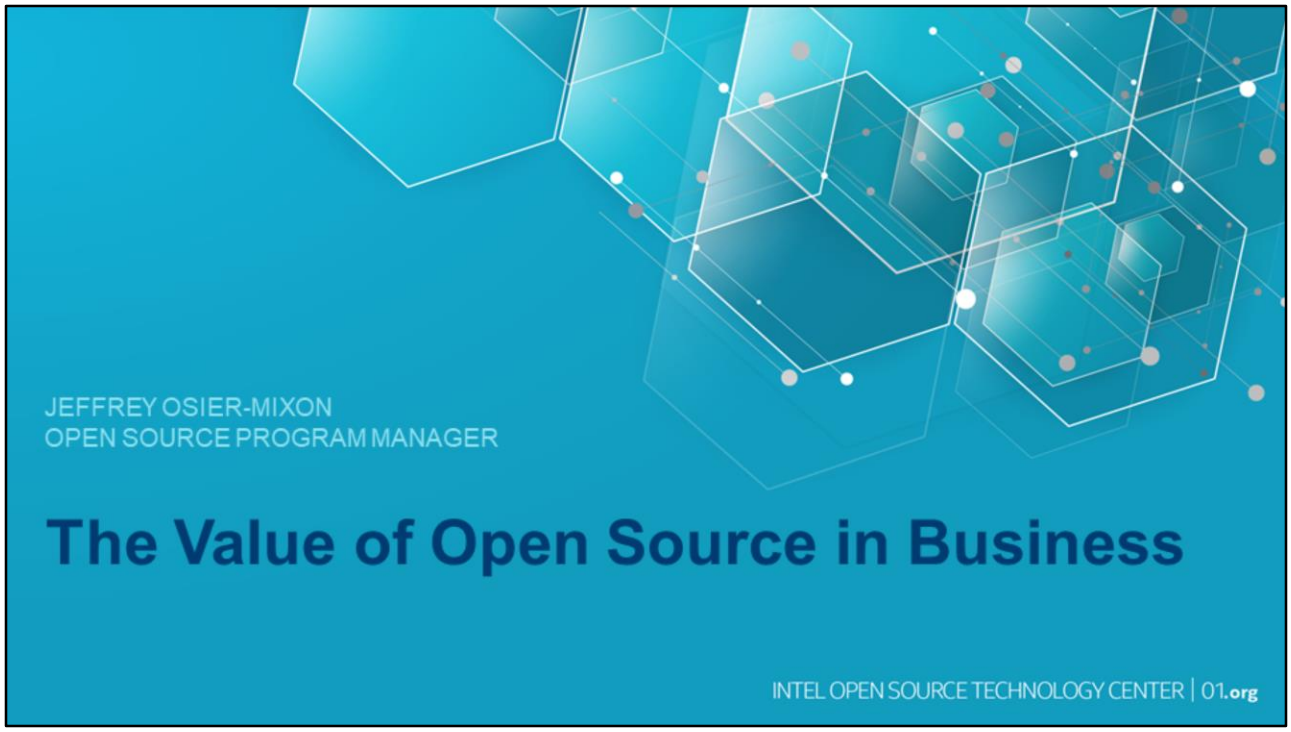


Good afternoon. My name is Jeffrey Osier-Mixon, but everyone calls me Jefro. That has been my nickname in the open source community for a very long time.

Today I want to talk about the business values of open source, but perhaps in a slightly different way from my colleagues.

We are hearing a lot today about the benefits of open source and the way it is created and managed. These are important messages from the large companies who are trusted advisors to the government of Canada.

As a longtime community manager, which is a role somewhere between program management and psychotherapist, and also as "that guy", I try to look this ecosystem from a slightly different perspective. I want to start with a story about values.

Story of the Wolf

https://pxhere.com/en/photo/1077389

One evening an old Cherokee told his grandson about a battle that goes on inside people.
He said, "My son, the battle is between two "wolves" inside us all.
One is Evil. It is anger, envy, jealousy, sorrow, regret, greed, arrogance, self-pity, guilt, resentment, inferiority, lies, false pride, superiority, and ego.
The other is good. It is joy, peace, love, hope, serenity, humility, kindness, benevolence, empathy, generosity, truth, compassion and faith."
The grandson thought about it for a minute and then asked his grandfather: "Which wolf wins?"
The old Cherokee simply replied, "The one you feed."

So what do we feed and how do we feed it? the way I interpret this story is that the wolves represent values, and the way we feed them is to distill fundamental principles from what we see as valuable, and then follow these principles, which creates more value. The better we understand what our values are, the stronger they get.

Story of the Wolf

Values

Actions ← Principles

https://pxhere.com/en/photo/1077389

Thus we have our virtuous circle, in which our values inform our principles, which, when we follow them, lead us back to values.

Another way to say it is that if we live by our values, if we follow our integrity as a strategy, our principle will distill from that and inform our tactics, and we will feed the wolf positively.

But how on earth does this relate to software development?

We can start with some history.

Open Source Beginnings

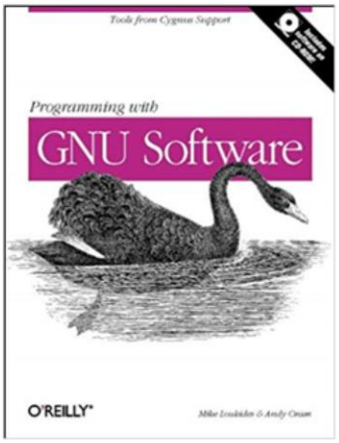
wikimedia

Open source had its beginnings in academia and in the corporate research labs, particularly Bell and Xerox, going back to the Mad Men era, due to the natural desire for researchers to share with each other. Sharing, collaborating, and building on prior art was a value, and competition tended to stifle creativity. This remains a value in academia and science but it has taken a while to be recognized in the pinnacles of capitalism. It was not recognized as a value.

Cygnus – commercial support for free software

I first started working with "open source" in 1992, at a company called Cygnus Support. It was a startup not far from Stanford, 28 people in an apartment building with coax strung through the trees between apartments, and now regarded as the first commercial enterprise based on open source. This picture is the day we moved into an actual office building in 1992.

GNU: GNU's Not Unix
Cygnus: Cygnus Your GNU Support



This startup company provided support for "free software", in this case a set of software development tools called the GNU project, which coincidentally is celebrating its 35th anniversary this week. GNU originated at MIT in the mid 1980s as an effort to provide an entire computer - operating system, text tools, compilers and development tools - under a permissive license and building on the principles of the UNIX operating system, which grew out of that research world.

The GNU tools were licensed to be available free of charge, but also to have their source code available to everyone, with the provision that the changes and improvements people would make needed also carry the same set of freedoms through the license. This license was the GNU Public License, or GPL, and it started a quiet revolution in the software industry.

I was a software developer and a technical writer at Cygnus, and eventually a manager for the tech pubs team. Our biggest challenge as a startup was, on the surface, to convince people to pay us for a product they could get for free. Makes no sense, right?

Perspective

https://www.flickr.com/photos/dret/33753688055

This was not the real challenge, though, as it turns out. Sometimes it takes reframing a problem to see it from a new perspective and understand what the real issue is underneath.

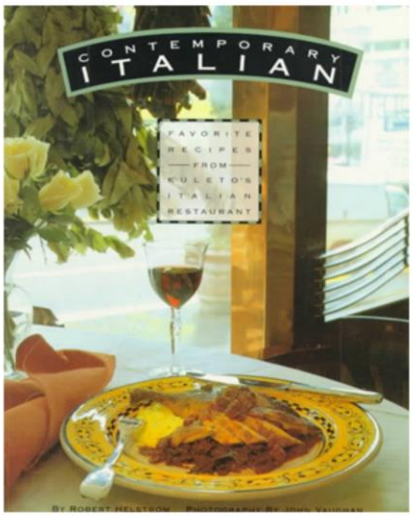
We addressed the problem by offering services on top of a free product - namely technical support, customizations and professional services, and training. That is not terribly uncommon for free or inexpensive products that would otherwise not have a viable market space. In fact, I think of it every time I go to a restaurant.

What's open source about a restaurant?

This isn't just a throwaway line - does anyone know what license is used for recipes? as a collection of ingredients, a recipe can not be copyrighted. It is by its nature the open source of food. This is a gross generalization of course, but it is a good example of the nature of open source.

Also, this is a restaurant I worked at in college.

Path to Success: Give Away Knowledge



If a restaurant publishes a book of recipes, does that affect the restaurant's success?
Do people stop going to that restaurant because they can cook at home, or share it with others, or even alter the recipe to create their own and then publish their own book?

In the foodie word, publishing a recipe book is a mark of success for a restaurant and they see the value in that transparency. It proves their actual values to the world - creativity, attention to detail, reverence for culture and history, ingenuity, comfort. We go to restaurants to be impressed and comforted, that is their value to us and it's why we spent a hundred bucks on something we could reasonably replicate at home for ten.

But again, it's a reasonable question whether this actually applies to software. Isn't software a commodity, where fine dining is more of a creative act?

What are we selling?

wikimedia

We realized that what we were selling wasn't software, or even the support features around software.
The most important commodity or item of value that we sold was the concept of open source that has caused it to proliferate with abandon over the past 26 years.

What was that value that sold so well that the company eventually grew into the largest open source provider in the world?

Open Source Values: Trust

It was trust.

Trust means different things to different people.

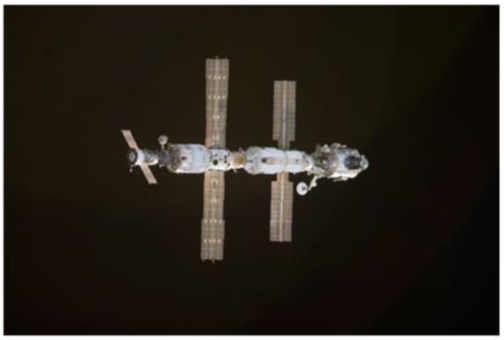
- self-reliance - confidence that you can accomplish what you need with the resources that you have.
- reliance on others - trusting that help and resources are available when you need it.
- friendship and camaraderie, the joy of accomplishing goals together.
- integrity - to see inside something, to reduce the need for blind trust. Transparency.
- sharing - contributing something of value to a wider system and taking the benefits of what others share.
- freedom - freedom to change, free as in free speech, free as in free love, free as in free beer.

So liberte, egalite, fraternite. Open source is essentially French.

Open source provides all of these things, and all over the world, these values generate and provide open source.

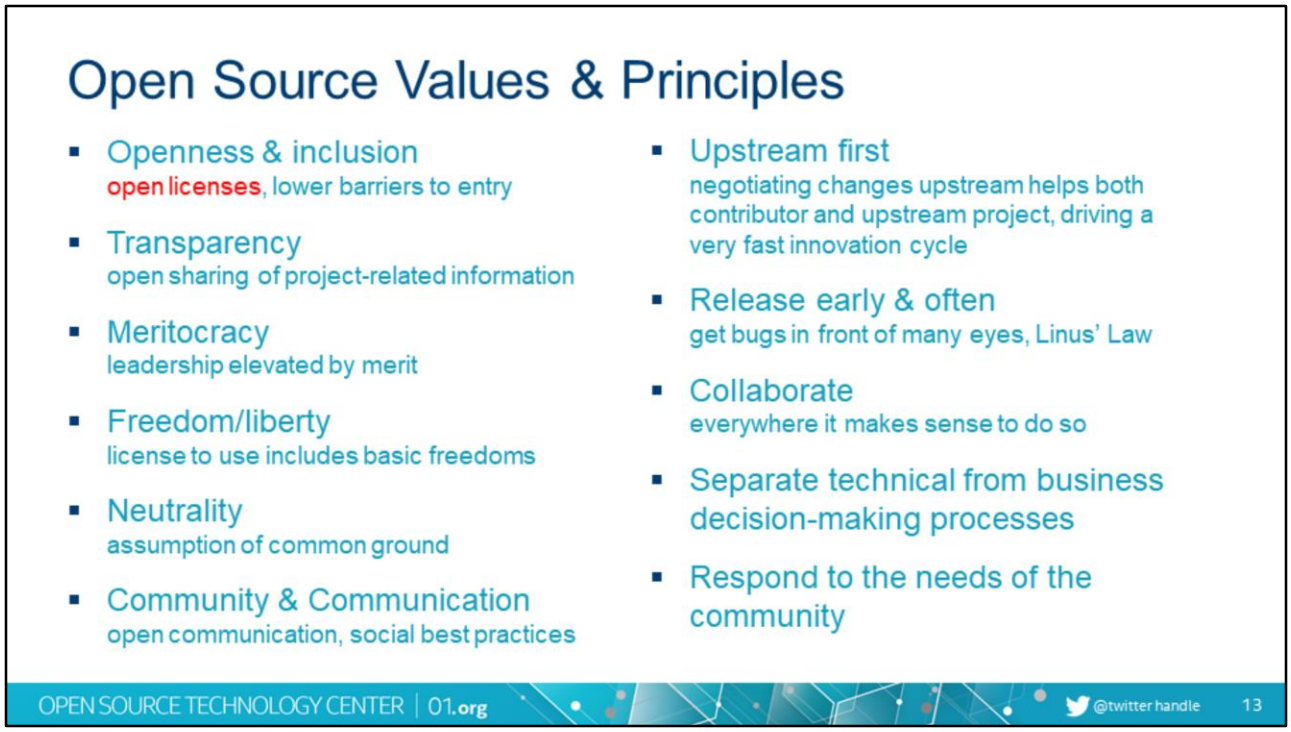
"We migrated key functions from Windows to Linux because we needed an operating system that was stable and reliable -- one that would give us in-house control.

So if we needed to patch, adjust, or adapt, we could."



Official statement of the United Space Alliance,
regarding the computer systems for the International Space Station (ISS)

Even the International Space Station has adopted open source and is able to recognize value from this process. Ironically, sharing with others gives you the highest level of control over your own software, and it is built on a core value of trust with the open source development community.



This focus on trust as a primary value, but it is not the only one.

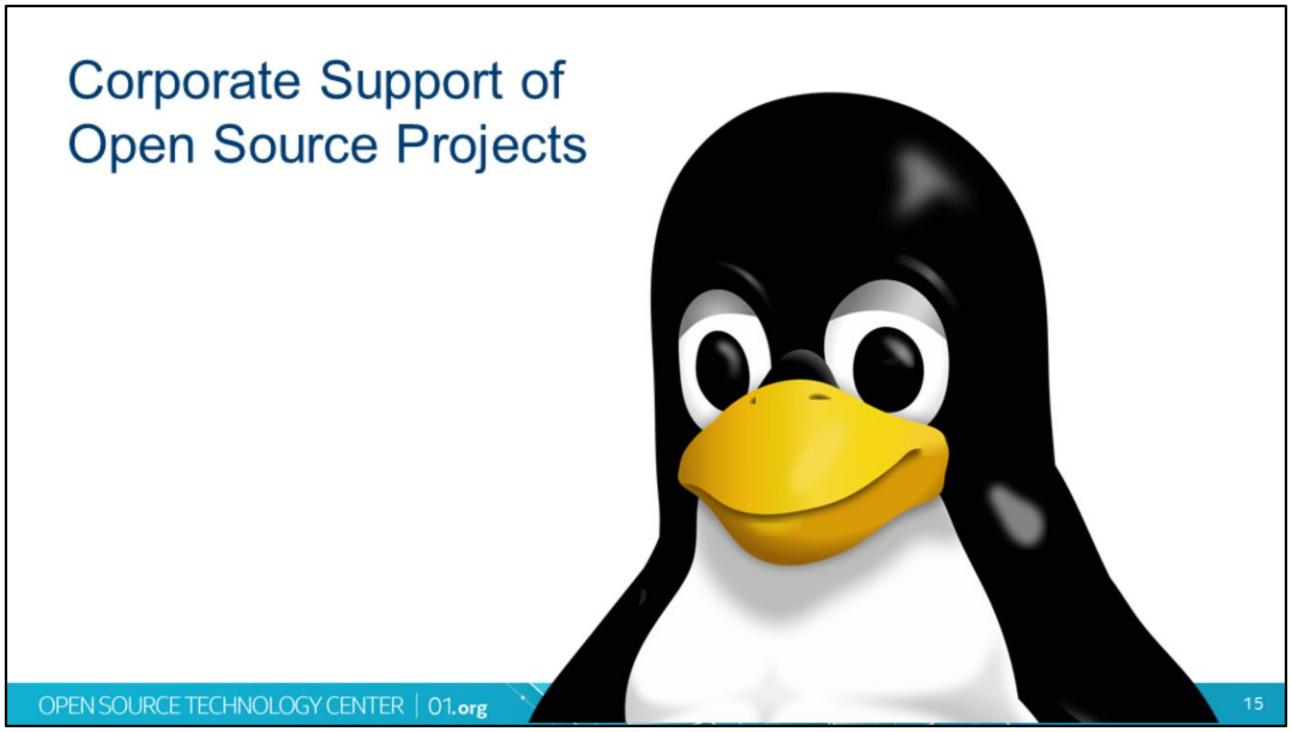
I think of values as what I find important, and principles as guides to help me get there. For example, the concept of upstream first is a value in theopen source world. This is the process of sharing improvements or bug fixes back to the original project, under a compatible license, so that others may also take that work and build on it or adapt it to their own needs.

This focus on trust as a primary value, but it is not the only one.

I think of values as what I find important, and principles as guides to help me get there. For example, the concept of upstream first is a value in theopen source world. This is the process of sharing improvements or bug fixes back to the original project, under a compatible license, so that others may also take that work and build on it or adapt it to their own needs.

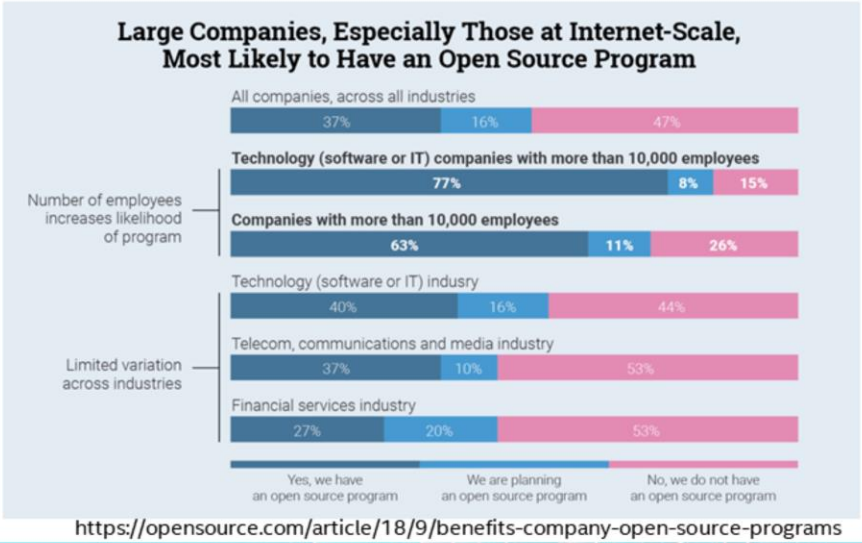
Why upstream? Isn't that giving away the farm? It depends on where the true value lies. If I fix a bug in the Linux kernel and submit it upstream, and it is accepted, that patch now belongs to the community, who can adapt it to other problems and keep it in sync with the rest of the system. If I keep that patch to myself, not only does it not benefit anyone else - attempting to raise my boat out of the water by itself - but now I must maintain that patch going forward in an environment that moves very, very quickly for innovation, bug fixes, security threats - and with every release it will be harder for me to backport this patch. It becomes a monkey on my back. Upstreaming is demonstrably better and less expensive in all respects, plus it makes you a good citizen, it's a win for everyone.

So maybe sharing or staying in sync is the value, but upstreaming is the best practice, the principle that nets the highest value.



Intel was an early corporate contributor to the Linux ecosystem, and has for the last 10 years been the largest single corporate contributor, because Linux is everywhere and Intel wants it to run very, very well on Intel processors, which it does. And today, Intel is cooperating with hundreds of companies in several markets, even working with direct competitors to raise the boats. Because market share of a larger system is more intrinsically valuable than the same market share of a smaller system. Intel formed the Open Source Technology Center in 2003, and still runs it as an internal center of excellence at Intel. As an open source program office, we advise software projects from all over the company.

Open Source Program Offices



https://opensource.com/article/18/9/benefits-company-open-source-programs

I don't know the original discussions or who had those initial discussions, but I am aware of some of the discussions that go on even today as new projects form and they need to convince management of their value. Intel, as well as Microsoft, Redhat, Facebook, Google, and many, many other companies, launch hundreds of projects every year, some small projects to fill ecosystem gaps, some large ones to address a market need or to capitalize on new hardware features or emerging technologies.

For organizations with even a little bit of open source experience under their belts, the discussion is usually not one of "should we work with open source", but "how can we work with open source most effectively and gain the advantages of community-based development and capture all that value"

What about competition? Reframing again - some people like the term co-opetition. I'm just going to give a quick example of how this works.

I worked for a small company about ten years ago that was an embedded Linux OSV - operating system vendor. there were about half a dozen primary companies in this space, and each of them had a staff creating development tools. It was crazy because none of them charged money for it. It was expected to be part of their offering, if you offer an OS basically as a kit, you also offer tools to build it into an image. Hardware support was actually a big part of the business also, because there were no standards, everyone did their own thing.

In 2010, Intel and Wind River, one of the other major OSVs, decided to address the gap by capitalizing on an open source build system called OpenEmbedded. They took this loosely managed open source project and put industry-level processes around it - QA, good documentation, regular releases, a mailing list where people responded to questions, and then presented it to two dozen companies to see what kind of traction it would get - would it resolve some of their issues. 22 of the 25 companies responded positively. All of the OSVs, several silicon manufacturers including Intel and TI, some consultants who immediately saw the value, and the OpenEmbedded project itself also joined. The project launched in the

fall of 2010 and then created an advisory board as its administrative governance in 2011. I joined Intel then as the community manager and chair of the advisory board.

The project succeeded wildly by addressing a clear need - it lifted all the boats. The OSVs all got a working build environment, the silicon manufacturers suddenly got a de facto standard for board support, the consultants got a world-class basis for their work - everybody won. The project now has 25 member organizations, provides over 60% of all commercial embedded Linux builds, and features as a basis for other highly successful open source projects. By every metric it is a successful corporate-sponsored community project.

I could tell this story about a number of open source projects. Even if they can be difficult to sell initially, the benefits simply outweigh the costs.

The choice to invest really breaks down into just a few decision points.

- can this software improve our relationship with our audience
- can this software improve our footprint, or open up new markets
- can this software improve people's lives

Or, can I meet my core business values by matching them up with the values of this open source project? And can I afford to be left behind when everyone else is working with open source?

Answering these questions is not always easy and often involves the burden of providing data to prove a point, but it is increasingly the choice of companies and governments around the world. It's not just good citizenship, it is a business decision that dramatically increases the speed and quality of development while also building communities and

It isn't just software. Building systems and ecosystems also requires the creation of standards and protocols. Best practices are also "open sourced". The best standards and practices evolve from the cooperation of stakeholders who understand the values inherent in what they work on, distill principles from those values, and then use the principles to create more value.

And it also isn't just companies. Educational institutions still drive a large number of software projects, and increasingly, governments are involved not only as users but as active participants and contributors in projects that serve their needs.

Takeaways

- reframe to see the real value of what you do
- build values into principles and actions
- understand the open source process

wikimedia

What I'd like you to walk away with is the understanding that it is incredibly valuable to reframe the question from "how do I keep my value safe" to "how do I express my real value as a set of principles, let that drive my actions, and let those actions create more value". And understand that that is the open source process at work, and it is recognized by companies and governments around the world. Maybe it'll work for you as well.

FURTHER READING

- *Art of Community 2nd ed.* – Jono Bacon
- *Seven Habits of Highly Effective People* – Stephen R. Covey
- *The Cathedral and the Bazaar* – Eric S. Raymond


INTEL OPEN SOURCE TECHNOLOGY CENTER | 01.org
@jefro_net

TYPES OF OPEN SOURCE PROJECTS

Type	Organization	Governance	Costs	Infrastructure	Example
Sole Provider	Only one	internal	Up to provider	Provided or free (Github etc)	hardware drivers
Workgroup	Loose assoc, no formal framework	ad-hoc	Low to none, contributions by providers only	Provided or free (Github etc)	Systemd, Linux print proj
Lightweight Collaborative	Group-created or foundation	TSC only	Low, funded by contributions only	Provided, free (Github etc), or paid service	OpenEmbedded, ACRN Project
Collaborative (Consortium style), Umbrella projects	Group-created or foundation	TSC Gov Board	Medium to high, funded by dues	Group-created or paid service	OpenStack, Yocto Project, Zephyr Project
Benevolent Dictator	Neutral, often a foundation	BD (+TSC?) Gov Board ?	Entire spectrum	Group-created or paid service	Linux Kernel, many smaller

TYPES OF OPEN SOURCE FOUNDATIONS

- **Single-project foundation**: provides legal/financial services only for one project
- **Collaborative foundation**: exists for many projects to collaborate on one subject (example: Eclipse Foundation, Open Compute Foundation)
- **Open Services foundation**: provides limited services to many projects regardless of type (example: Apache Foundation)
- **Single Service foundation**: provides a single service set, e.g. legal organization along with financial management (example: Software in the Public Interest, Software Freedom Conservancy)
- **Full Service foundation**: provides extensive services, including legal, marketing, infrastructure, project management, and governance, to open source projects organized around many business opportunities (example: Linux Foundation, Open Infrastructure/Openstack Foundation)

JEFFREY “JEFRO” OSIER-MIXON

Jefro works for Intel Corporation in the Open Source Technology Center as community advocacy manager for virtualization projects and as an advisor for open source activities around the company. He has been involved in open source since the dawn of recorded time, in roles as varied as receptionist, technical writer, software developer, program manager, and board chair. You can find him at open source conferences around the world.