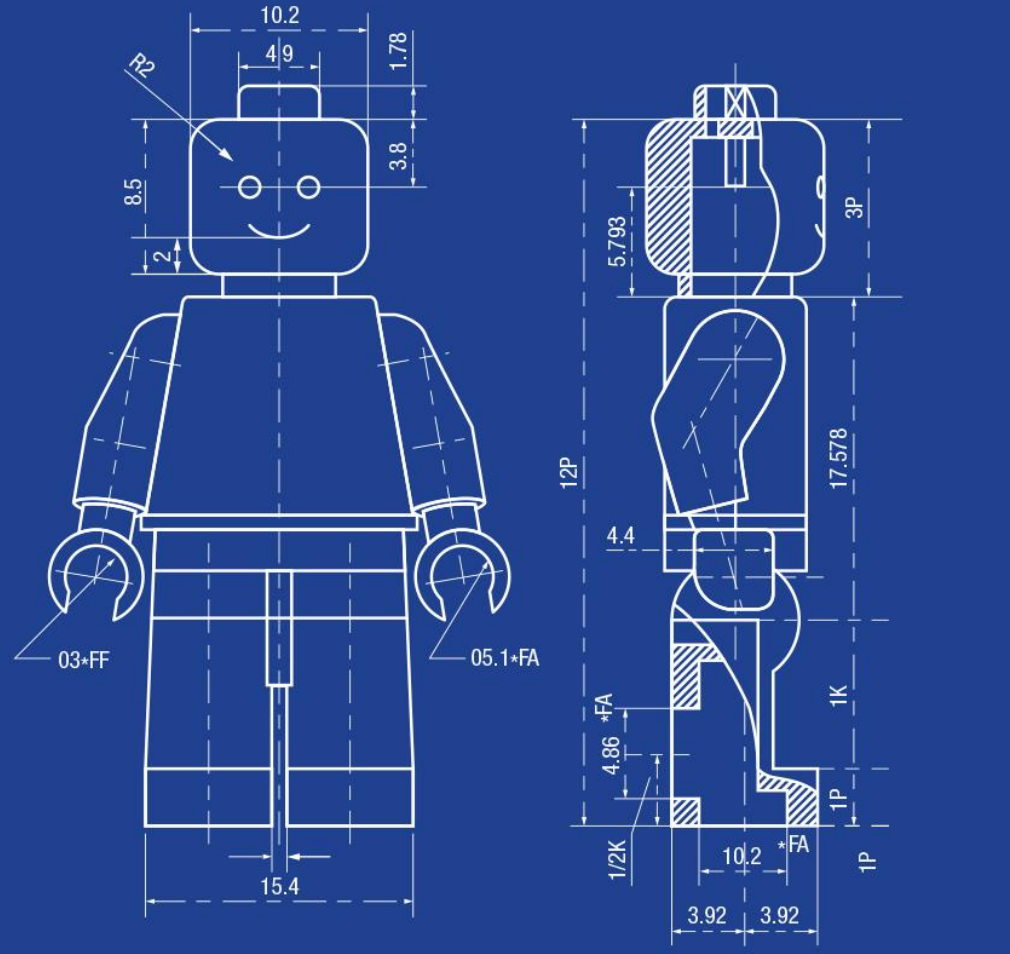
RI.
SE

JOHAN LINÅKER

Open Source Software And Open Science

About Johan



LEGO

TOY FIGURE

PUBLICATION NUMBER	USD253711 S	PUBLICATION DATE	DEC 18, 1979
PUBLICATION TYPE	GRANT	FILING DATE	FEB 14, 1978
APPLICATION NUMBER	US 05/877,900	PRIORITY DATE	AUG 29, 1977

What is Open Source Software?

Photo by Billy Huynh | https://unsplash.com/photos/v9bnfMCyKbg

Liberally Licensed, Collaboratively Developed Software

Photo by Billy Huynh | https://unsplash.com/photos/v9bnfMCyKbg

Open Source Software (OSS) today

- Approximately…
  - 90+ % of all software contains OSS
  - 75% (2020) of companies' code bases consists of OSS (up from 36% 2015)
  - 56 million developers collaborate on OSS projects on GitHub. Estimated to increase > 100 million 2025
  - Collaboration in and between verticals, including Energy, Automotive, Telco, Health

Photo by Pedro Lastra | https://unsplash.com/photos/Nyvq2juw4_o

In other words, it’s everywhere

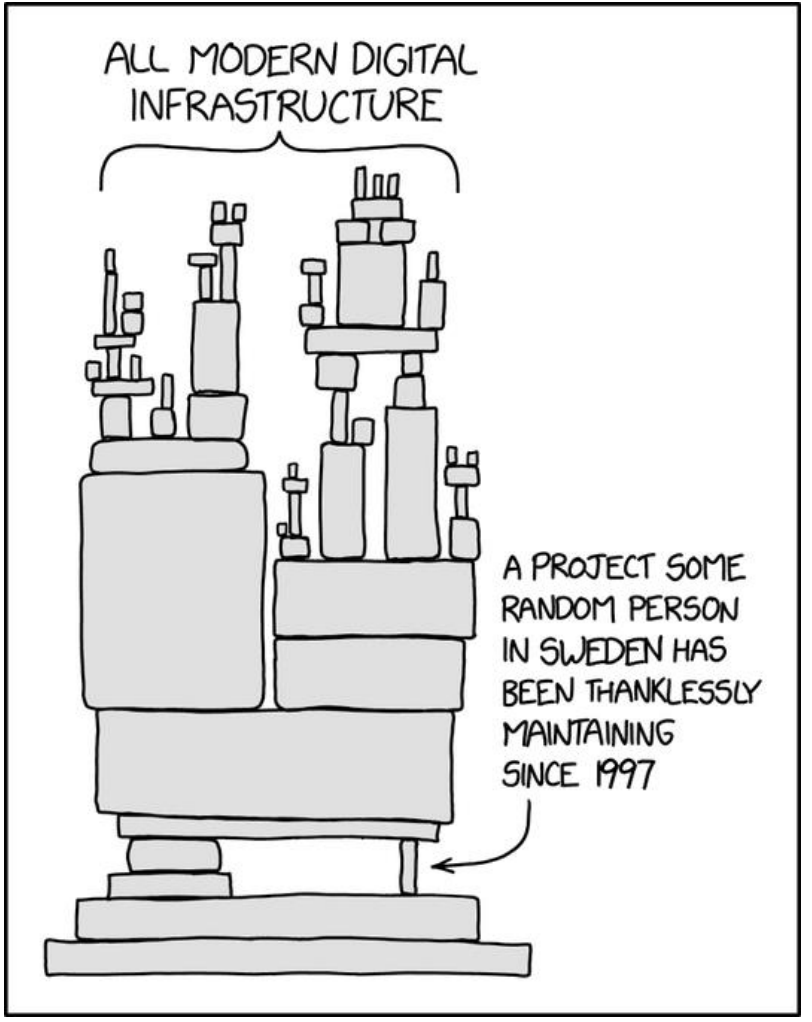
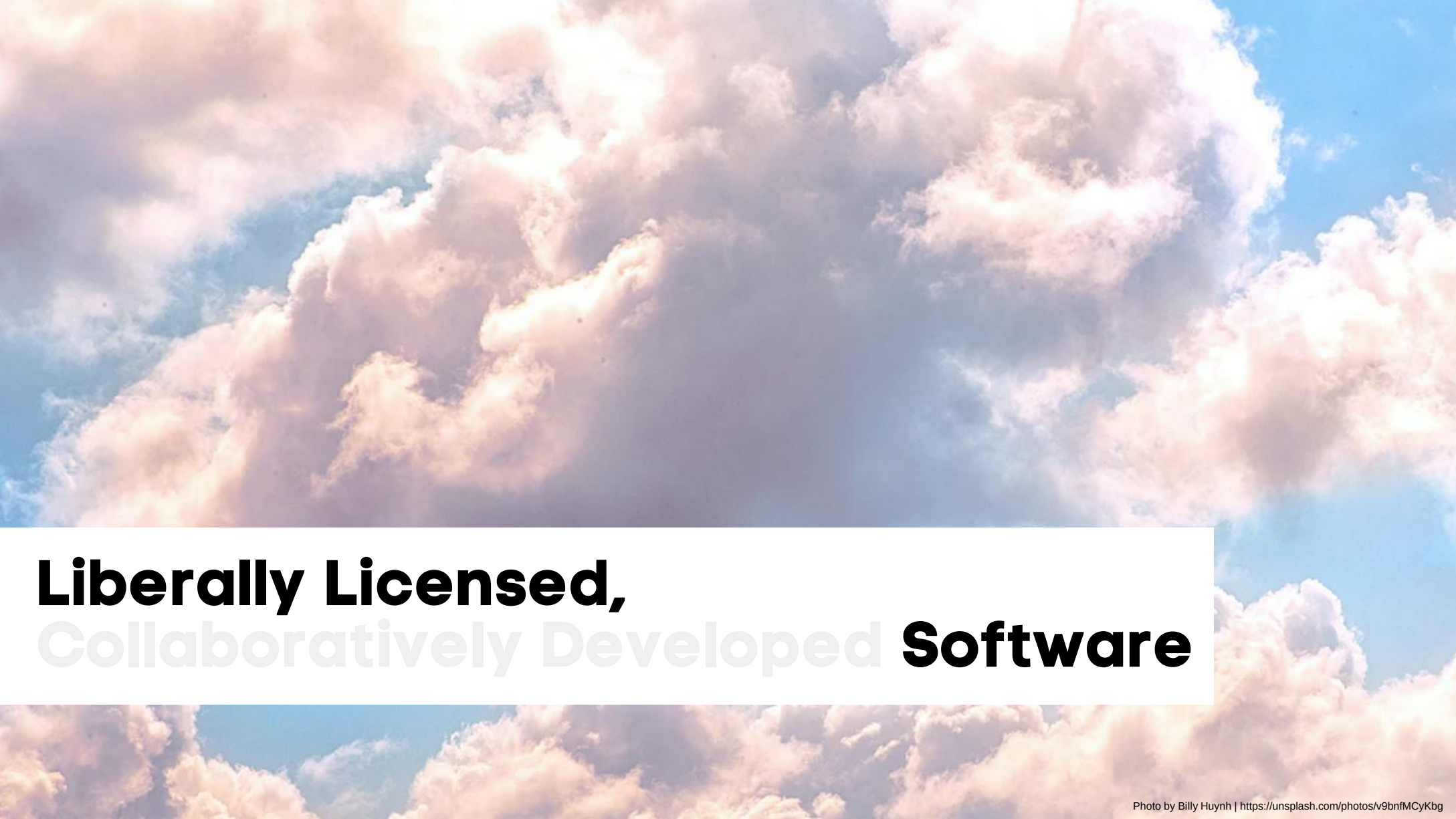


Photo by @tsjost and XKCD | https://twitter.com/tsjost/status/1295662118113619968



Liberally Licensed, Collaboratively Developed Software

Photo by Billy Huynh | https://unsplash.com/photos/v9bnfMCyKbg

Liberally licensed software

- Software available under an OSS license
- License follows the Open Source Definition and approved by Open Source Initiative (http://opensource.org)
- Who ever, for what ever reason may inspect, use, modify and redistribute the software
- Further conditions may vary between licenses

Photo by Fabian Fauth | https://unsplash.com/photos/1_EedIBc6jY

Permissive vs. Copyleft licenses

- Permissive licenses – do whatever you want, as long as you recognize the copyright holder
  - E.g., MIT, BSD, Apache
- Copyleft licenses – Above + share any modifications, additions and connecting code under same license.
  - GPL 2, GPL 3, AGPL
- Permissive common for standardiszations and collaboration on non-differentiating software
- Copyleft common when copyright holder wants to capture value back

Photo by Fabian Fauth | https://unsplash.com/photos/1_EedIBc6jY

Liberally Licensed,

Collaboratively Developed Software

Photo by Billy Huynh | https://unsplash.com/photos/v9bnfMCyKbg

Collaboratively Developed Software

- Software developed as projects by networks of individuals and organizations, aka. Open Source Communities
- ”Members” of the community commonly both users and developers
- Are united by a common vision and goal around the Open Source Software.

Photo by Annie Spratt | https://unsplash.com/photos/QckxruozjRg

Incentives for going open source

- Individuals:
  - Sense of belonging,
  - Recognition for contributions,
  - Solves paintpoint,
  - Build CV
- Organizations:
  - Lower costs,
  - Increased innovation,
  - Branding and PR,
  - Strategic tool

Photo by Annie Spratt | https://unsplash.com/photos/QckxruozjRg

Incentives for going open source

- Public policy:
  - Transparency
  - Competition
  - Economic growth
- Researchers:
  - Disseminate research outputs
  - Sustain OSS development between project
  - Collaborate with partners and scientific community
  - Enable reproducibility

Photo by Annie Spratt | https://unsplash.com/photos/QckxruozjRg

Risks, costs and complexities

- Companies:
  - Differentiating functionality, competitive edge and commoditization
  - Sensitive IPR and patents
- Public administrations
  - Compete with industry
  - Ethical aspects and responsibility
  - Integrity and confidentiality
- General:
  - Internal budget and resource constraints
  - Modularity and technical architecture

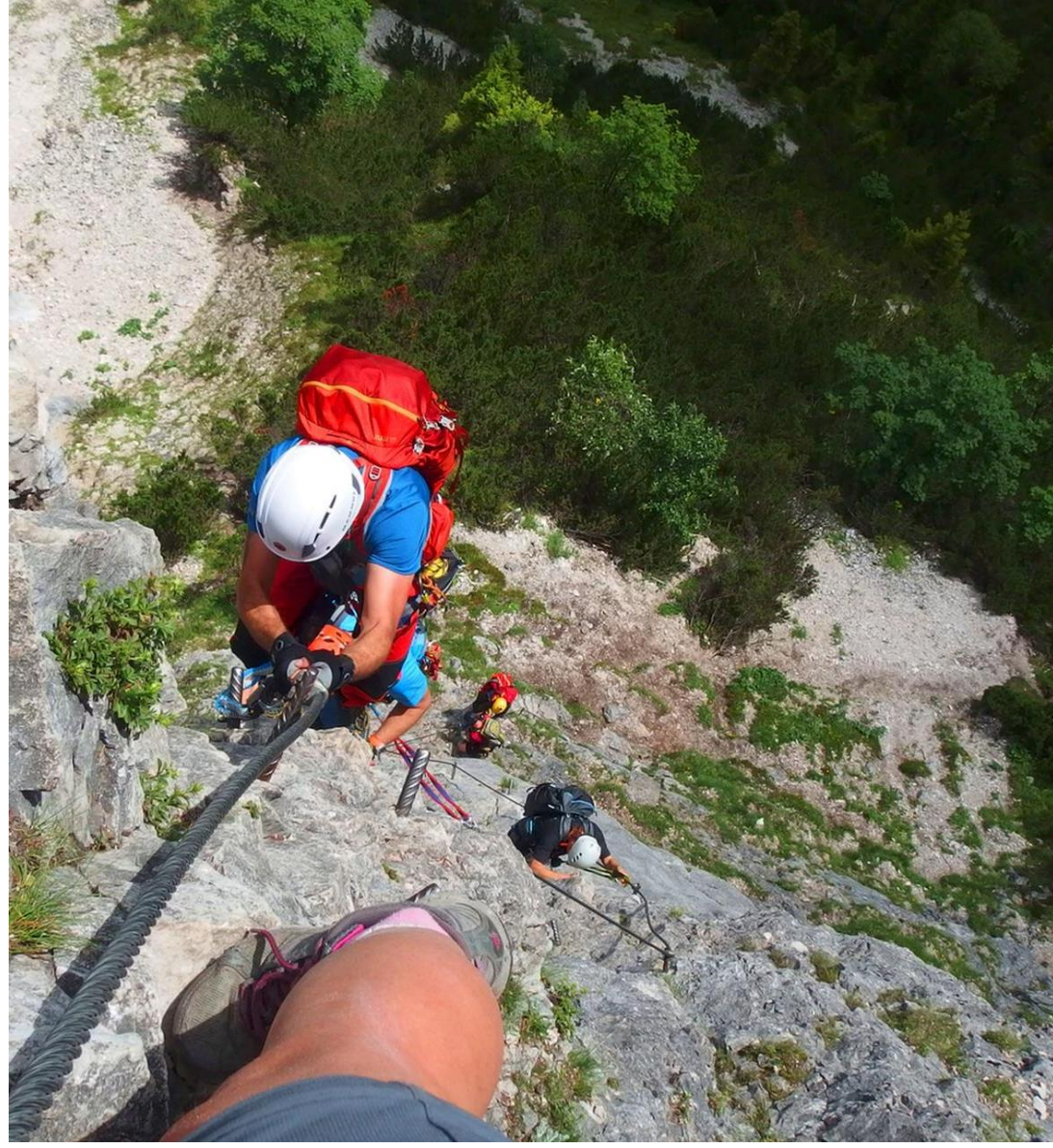
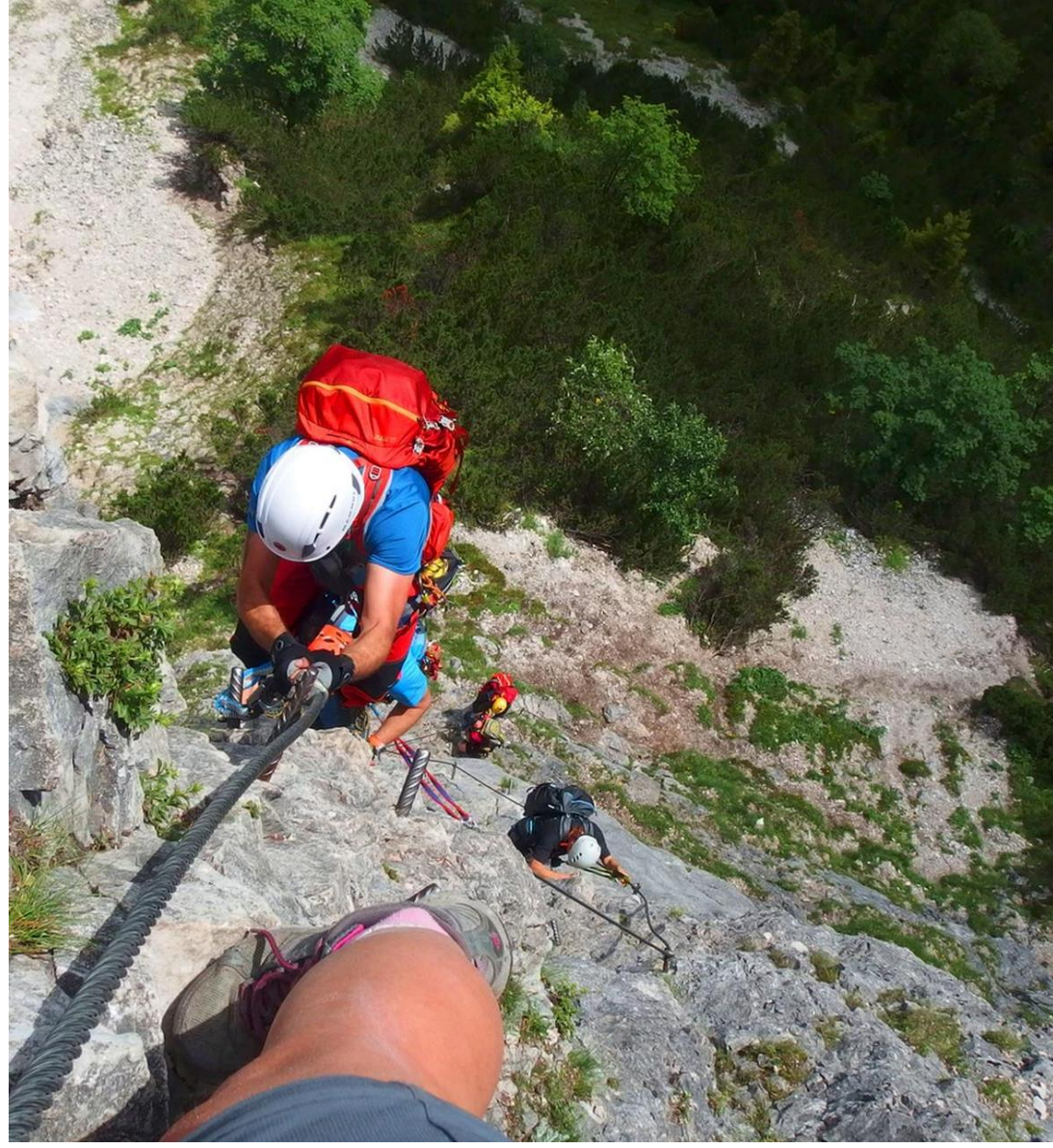


Photo by Maja Kochanowska | https://unsplash.com/photos/E-v6j5Y9xAs

Risks, costs and complexities

- Researchers:
  - Differentiating functionality, competitive edge and commoditization
  - Sensitive IPR and patents
  - Compete with industry
  - Ethical aspects and responsibility
  - Integrity and confidentiality
  - Internal budget and resource constraints
  - Modularity and technical architecture



Technical and non-technical contributions

- Development of new functionality and bug fixes
- Requirements identification, analysis, and prioritization
- Testning and quality assurance
- Documentation, marketing and community management
- Financial and infrastructural support
- ...

Photo by Amélie Mourichon | https://unsplash.com/photos/hcLSA1MiU3A

Open development process

- Informal structure, dependent the community
- Focus on openness
  - Whomever can contribute
  - Influence through merit
  - Self-appointment of tasks
- Traditional development
  - Structured in silos
  - Influence through hierarchy
  - Appointed tasks

Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Open development process

- Meaning, you cannot...
  - Expect quick and professional support
  - Expect to get your feature requests implemented
  - Order individuals to act according to your agenda

Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Open development process

- Transparent and open discussions on bug reports, features, and road map
- Conversations and information persisted in an open infrastructure
- Requirements fragmented and decentralized
- Community full of (un)known stakeholders, all with their own agendas

Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Governance for OSS projects

- Means and processes for
  - Deciding on the technical direction of the OSS project, and
  - How the collaboration should be coordinated to enable this direction.

Photo by krakenimages | https://unsplash.com/photos/Y5bvRlcCx8k

Community Structure and Governance

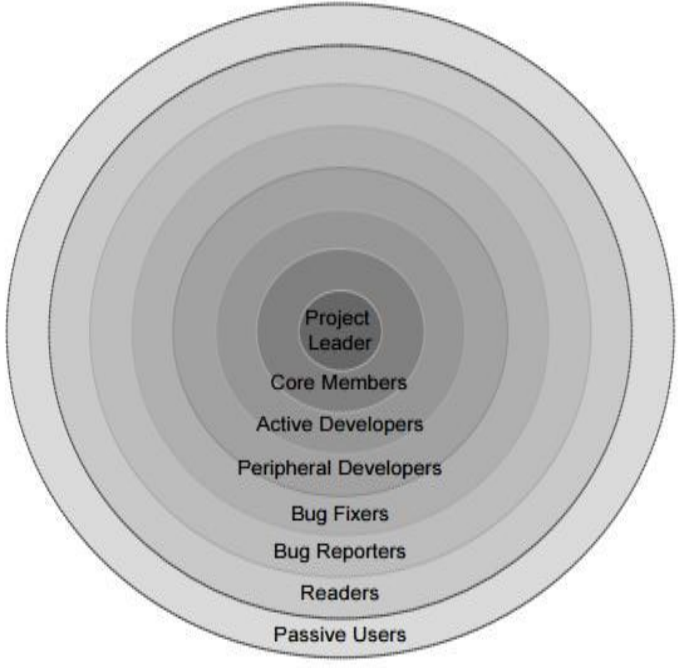


Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Community Structure and Governance

Leadership

Maintainership Maintainership Maintainership

Committers Committers Committers Committers
Committers Committers Committers Committers

Contributors Contributors Contributors Contributors Contributors
Contributors Contributors Contributors Contributors Contributors

Users Users Users Users Users Users Users Users Users Users
Users Users Users Users Users Users Users Users Users

@johanlinaker | https://linaker.se

Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Governance structures

- Autocratic governance
  - Centralized steering where roles assignment and influence over development is decided top-down
  - Usually the actor(s) that founded the project
- Democratic governance
  - Decentralized steering where roles assignment and influence over development is decided collectively, and gained through active engagement and contributions
- Transitions and combinations common

Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Governance structures

- Centralized governance
  - Formal steering and maintenance through a single or collective organization
  - Commonly pooled ownership of copyright
- Decentralized governance
  - Informal steering and maintenance through existing community
  - Distributed ownership of copyright
- Commonly transitions from decentralized to centralized structure

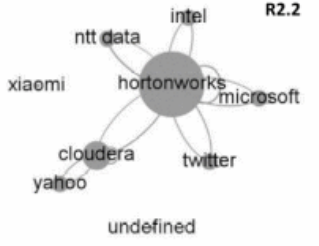
Photo by Leon | https://unsplash.com/photos/Oalh2MojUuk

Type of community

- Developer-driven community
  - Steering and maintenance typically performed by those who contribute to the development of the project
- User-driven community
  - Steering and maintenance typically performed by the end-users of the project.
  - Development performed through acquired resources

Photo by Vonecia Carswell | https://unsplash.com/photos/0aMMMUjiiEQ

Communities evolve constantly



RI.
SE

Relationship with community

- Symbiotic
  - Win-win for both firm and community
  - Contributing to influence projects according to internal agenda and improve health to mitigate security risks
- Commensalistic
  - Gain for firm, community indifferent
  - Use project and doing lighter contributions. Project in line with internal agenda and healthy with others already supporting it.
- Parasitic
  - Firm free-riding on community.
  - Using as is not giving anything back. Worst case expecting free work for nothing in return. Looked down on from communities.

Photo by Kelly Sikkema | https://unsplash.com/photos/4le7k9XVYjE

OSS Project health

- The OSS project’s capability to stay maintained to a high quality, long-term without interruptions
  - Productivity: There is an active development of the project
  - Robustness: The development is open and spread out on several (independent) individuals
  - Openness: Users of the project can influence and contribute to the development of the project

OSS and our Digital Infrastructure

- Open Source Software makes up a vitale building block in our digital infrastructure
- Needs maintenance as with physical infrastructure to stay secure and robust

Photo by Modestas Urbonas | https://unsplash.com/photos/vj_9l20fzj0

The Dualism of Quality

- Open Source Software is...
  - full of, or receptible to, vulnerabilties ready to be exploite
  - always more secure than proprietary alternatives

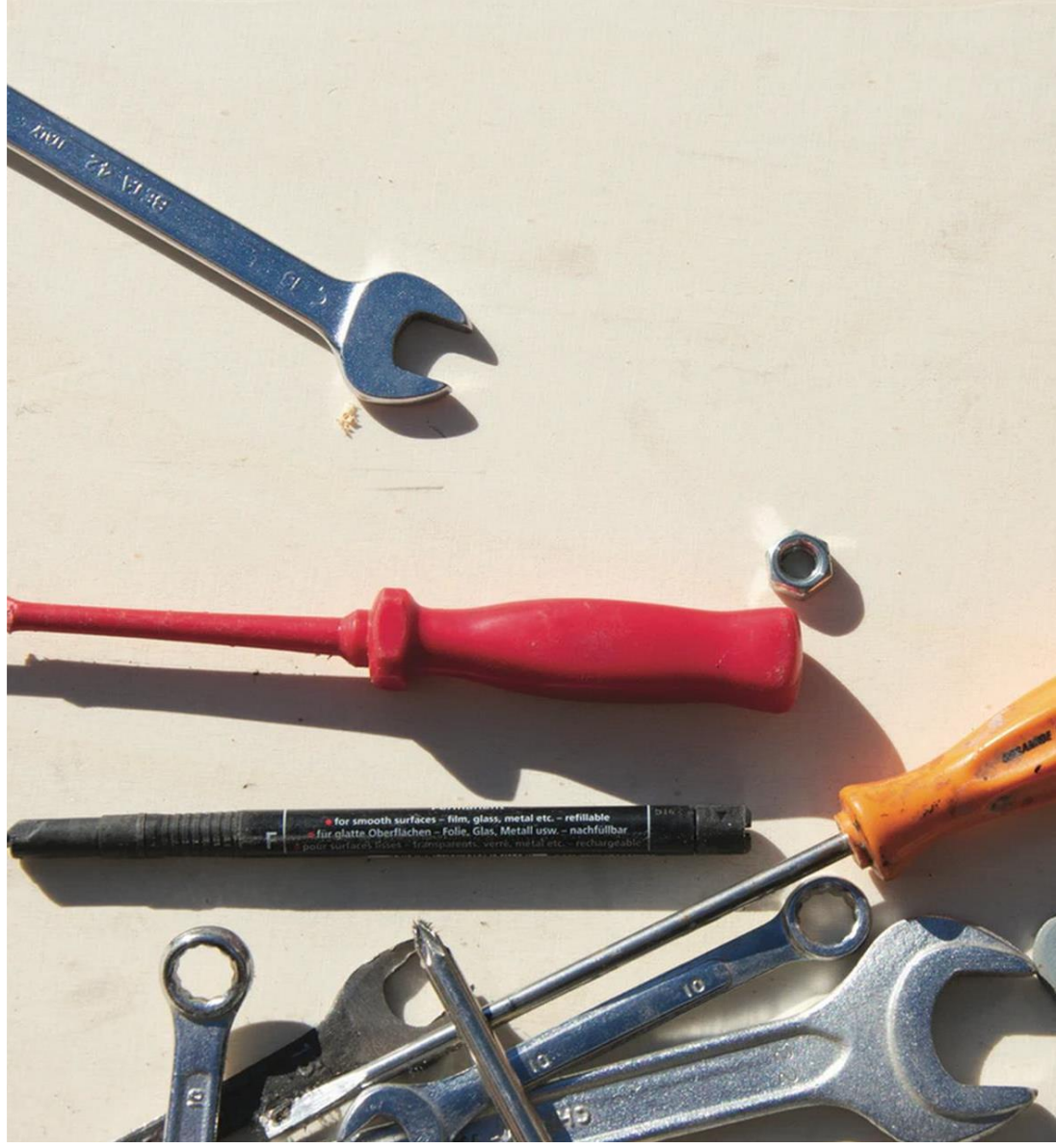


Photo by Elena Rouame | https://unsplash.com/photos/9JU2CKqtw0M

The ”Many-Eyes” effect

- Also known as Linus’ law →
  - ”Given enough eyeballs, all bugs are shallow”
  - Requires that enough eyeballs actually reaches the codebase

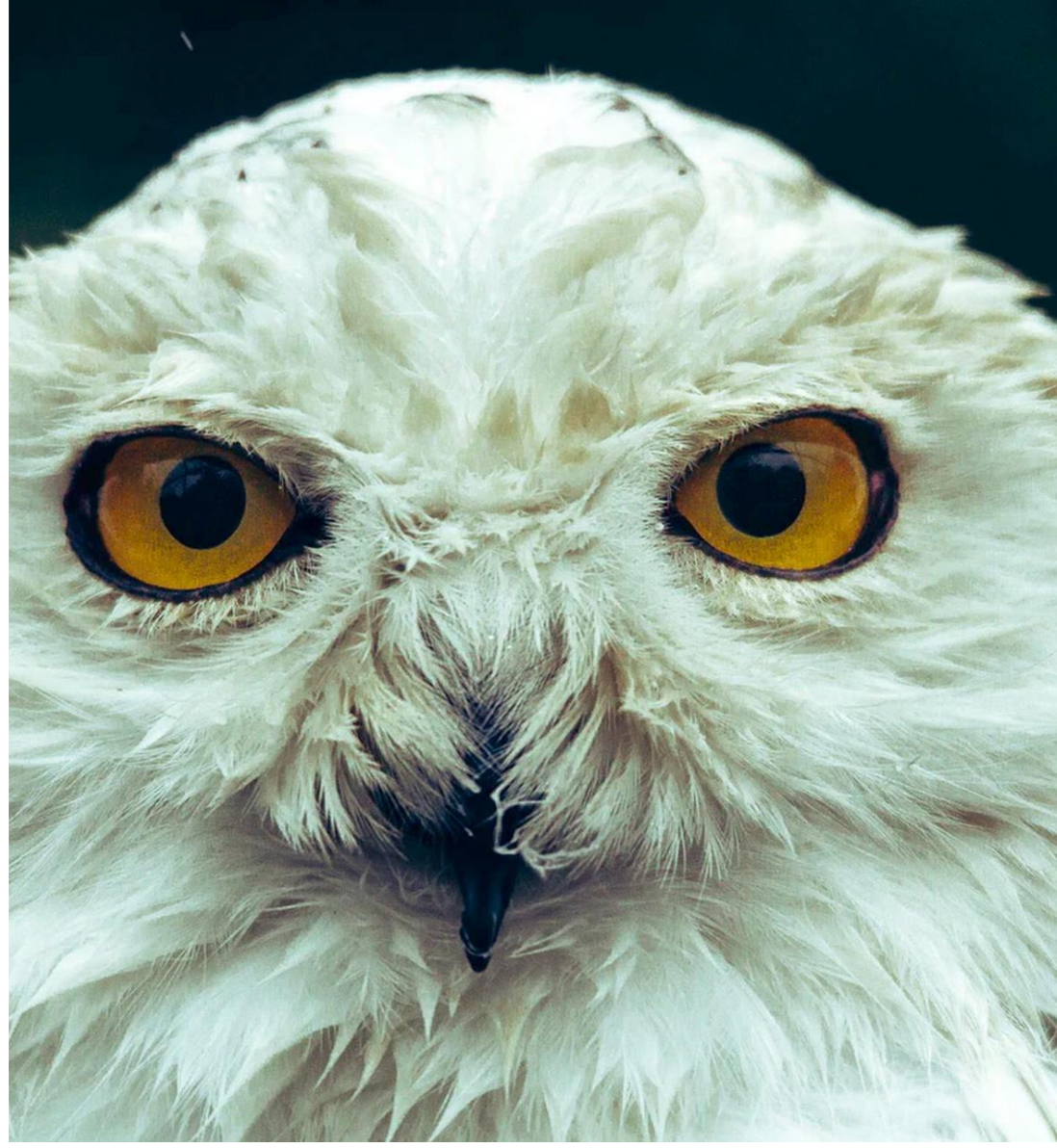


Photo by Frida Bredesen | https://unsplash.com/photos/c1fFv08N7qE

Development Resources are Depletable

- Maintainers are humans, not robots
  - Burnout, changed family or working conditions
- Companies must adapt to stay competitive
  - Refactorization, new products, changed business model

Photo by NeONBRAND | https://unsplash.com/photos/KYxXMTpTzek

Who’s responsible for the SW quality?

- Maintainer(s)?
- Developer community?
- User community?
- Individuals vs. Companies vs. Government?

Photo by Scott Blake | https://unsplash.com/photos/x-ghf9LjrVg

Importance of growing a healthy community

- Collectively grow and communicate a common vision for the project
- Be responsive and helpful in communication
- Grow an open, inclusive, and supportive culture
- Enable on-boarding and self-support through
  - detailed documentation,
  - standardized tooling
  - clearly defined development and governance processes

Photo by William White | https://unsplash.com/photos/people-raises-hands-TZCppMjaOHU

Common challenges for scientific OSS

- Culture, knowledge and organizational support for OSS lacking
  - E.g., license selection, business models, communiy growth…
- Growing sustainable funding for the OSS project’s development and maintennace
- Typically very technical and specific knowledge required to contribute
- Narrow groups of end-users
- Parallell academic hiearchy inhibiting open collaboration and governnace

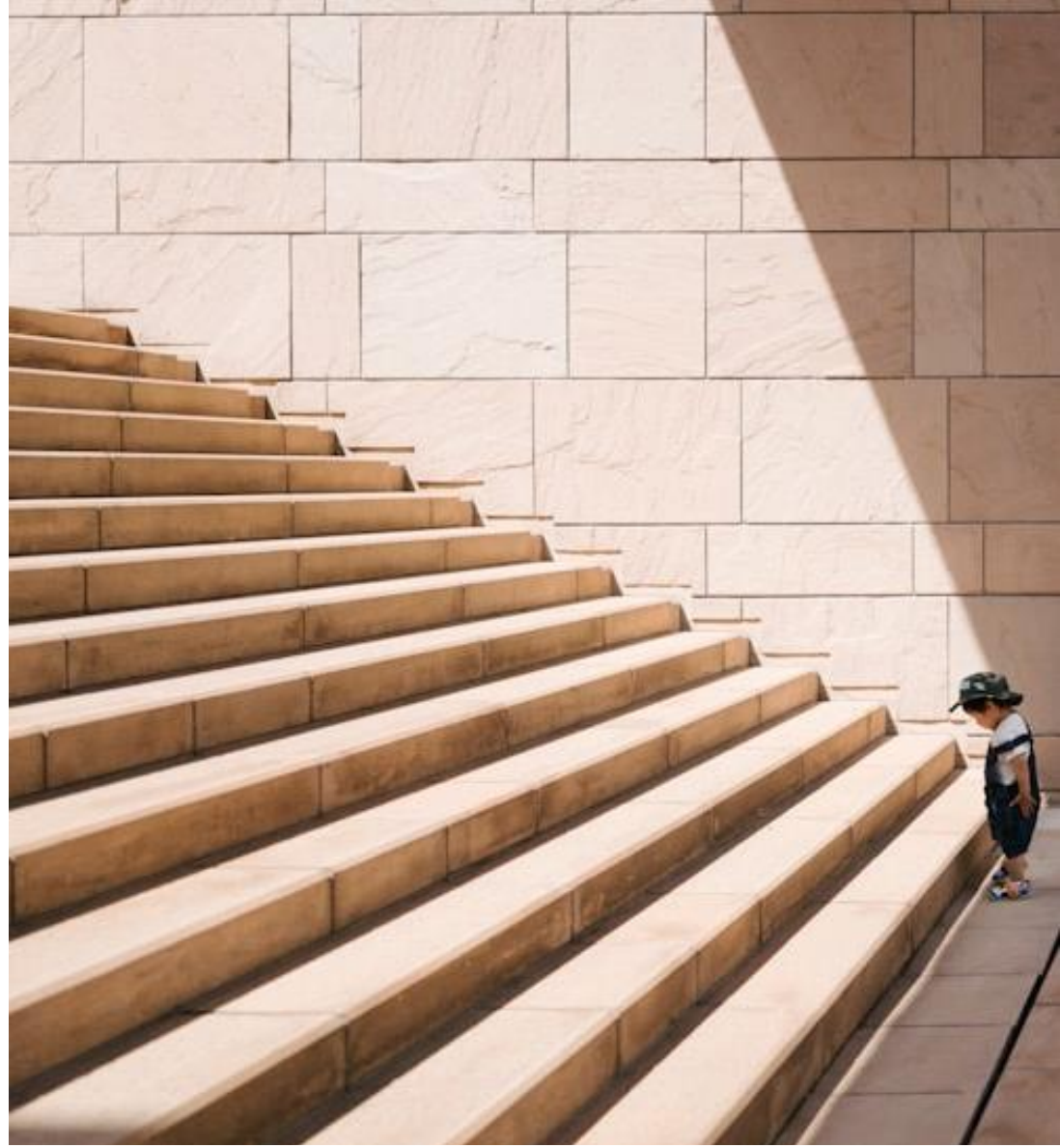
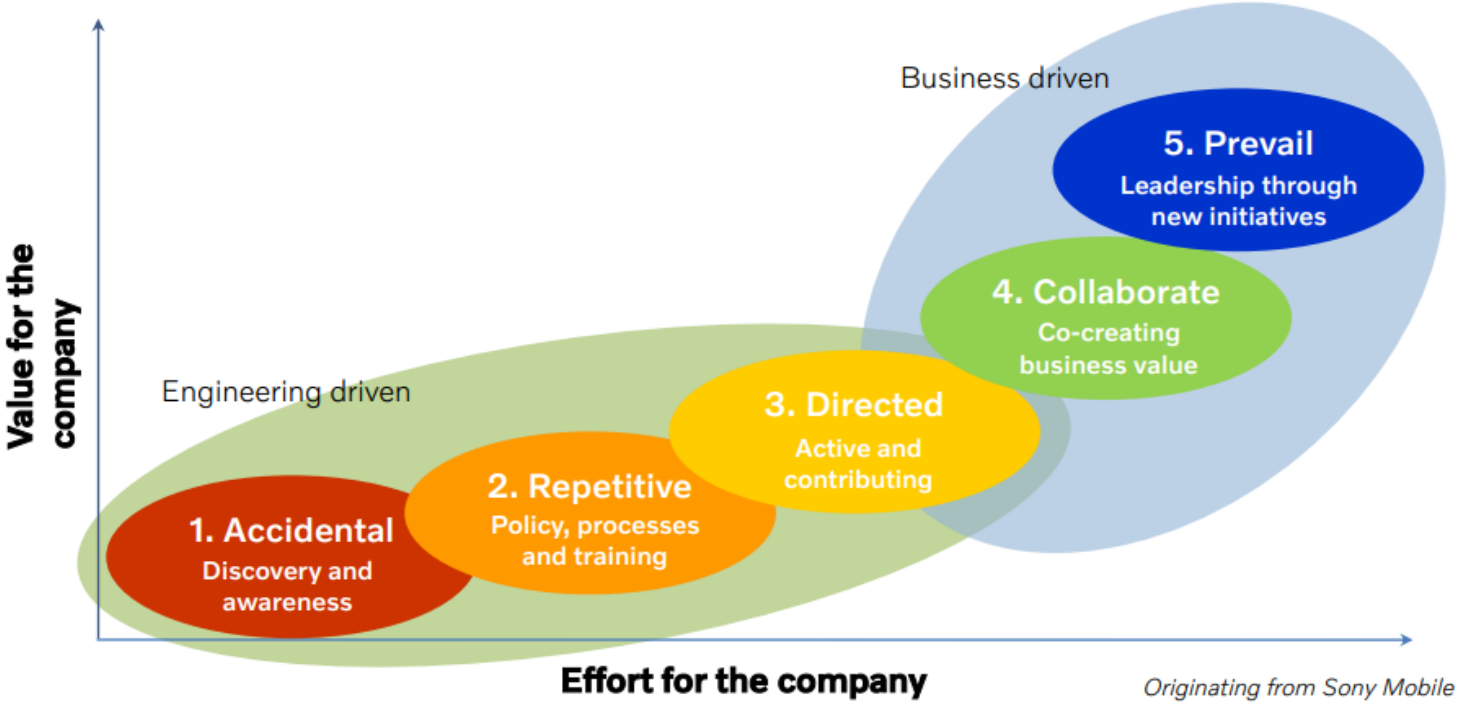


Photo by Jukan Tateisi | https://unsplash.com/photos/toddlers-standing-in-front-of-beige-concrete-stair-bJhT_8nbUA0

Maturing from consumption to leadership



Originating from Sony Mobile in 2011
/ Adapted by Carl-Eric Mols 2023

RI.SE

Open Source Program Offices (OSPOs)

- Center of competency and support
- Drives organizational readiness and maturity forward on open source
- Designs and executes an organization’s overarching open source strategy
- Provides voice of reason and objectivity on the benefits, risks, and costs of open source and how to balance between
- Supports use, development, and collaboration on open source

Photo by Quino Al | https://unsplash.com/photos/dhGFLj3rI0Q

Case: Trinity College Dublin

- Small team within the Technology Transfer Office with a business developer and legal expert
- Focused on supporting researchers in using OSS as part of a business model through the commercialization of research outputs
- Supports grant writing and IPR management in research consortiums
- Provides education and training to researchers and under grad students (to various degrees)

Photo by Jonathan Singer | https://unsplash.com/photos/library-interioe-NgU7IJ5XuyY

Case: LERO

- Constituted by an internal community of subject matter experts
- Supports and trains researchers in how to develop, collaborate and disseminate software-based research-outputs as open source
- Considers open source as an instrument for open science, with a broadening interest for other areas within
- Ambition of extending the OSPO and open source as an instrument to the Technology Transfer Office, similar as to Trinity College Dublin

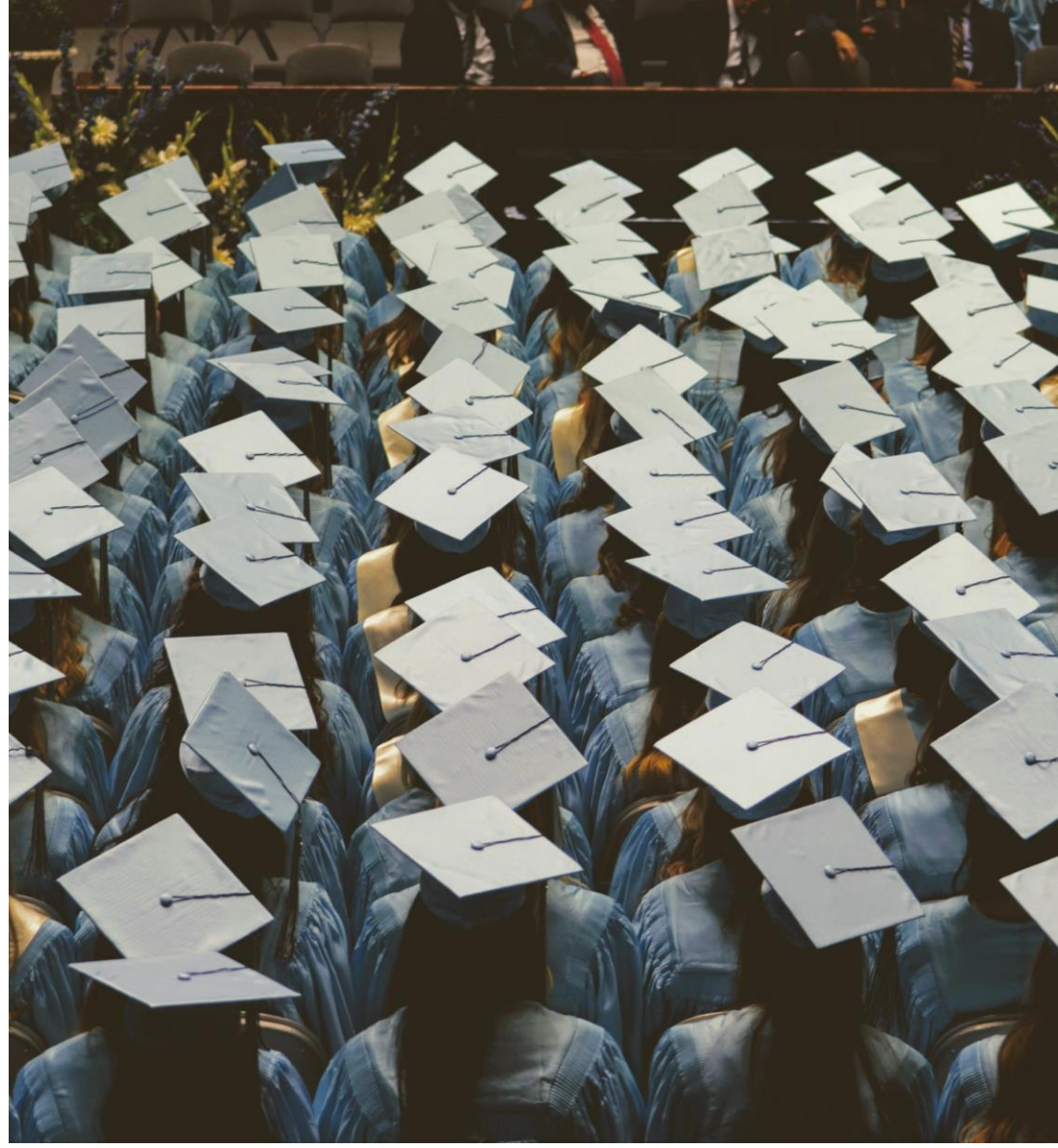
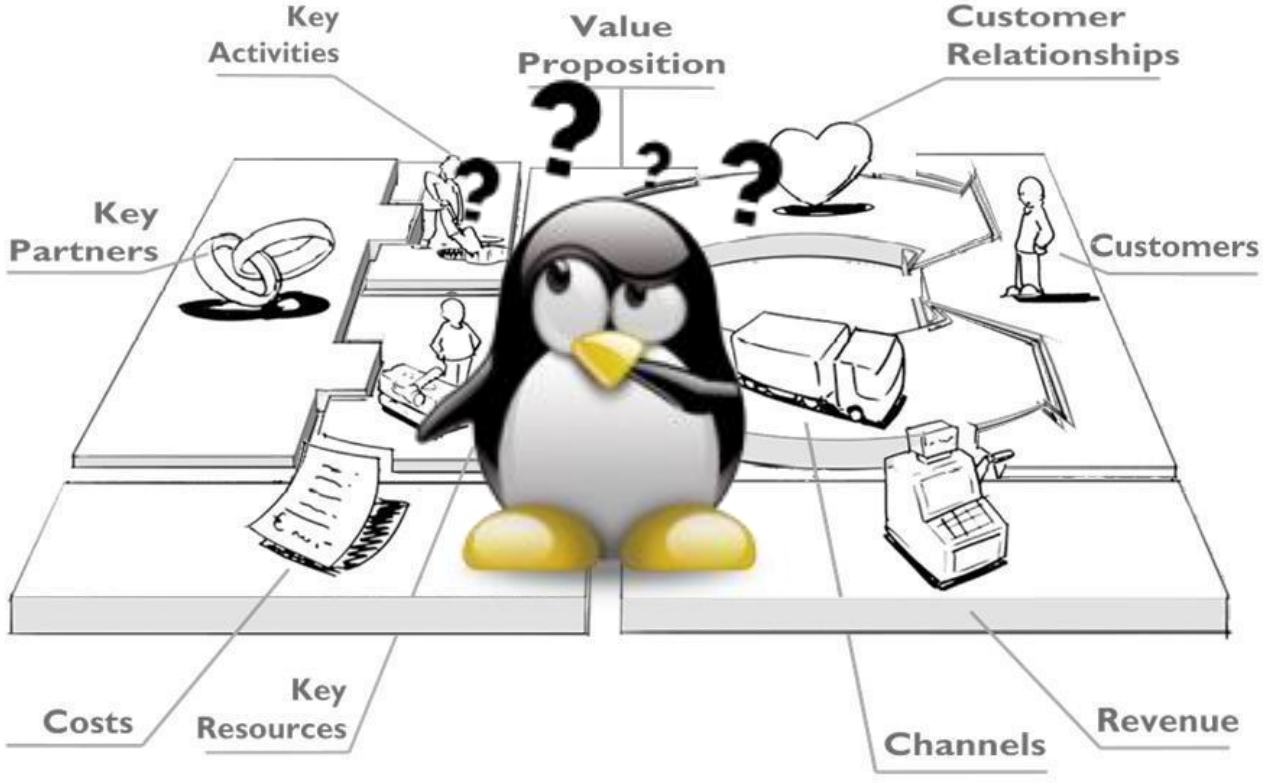
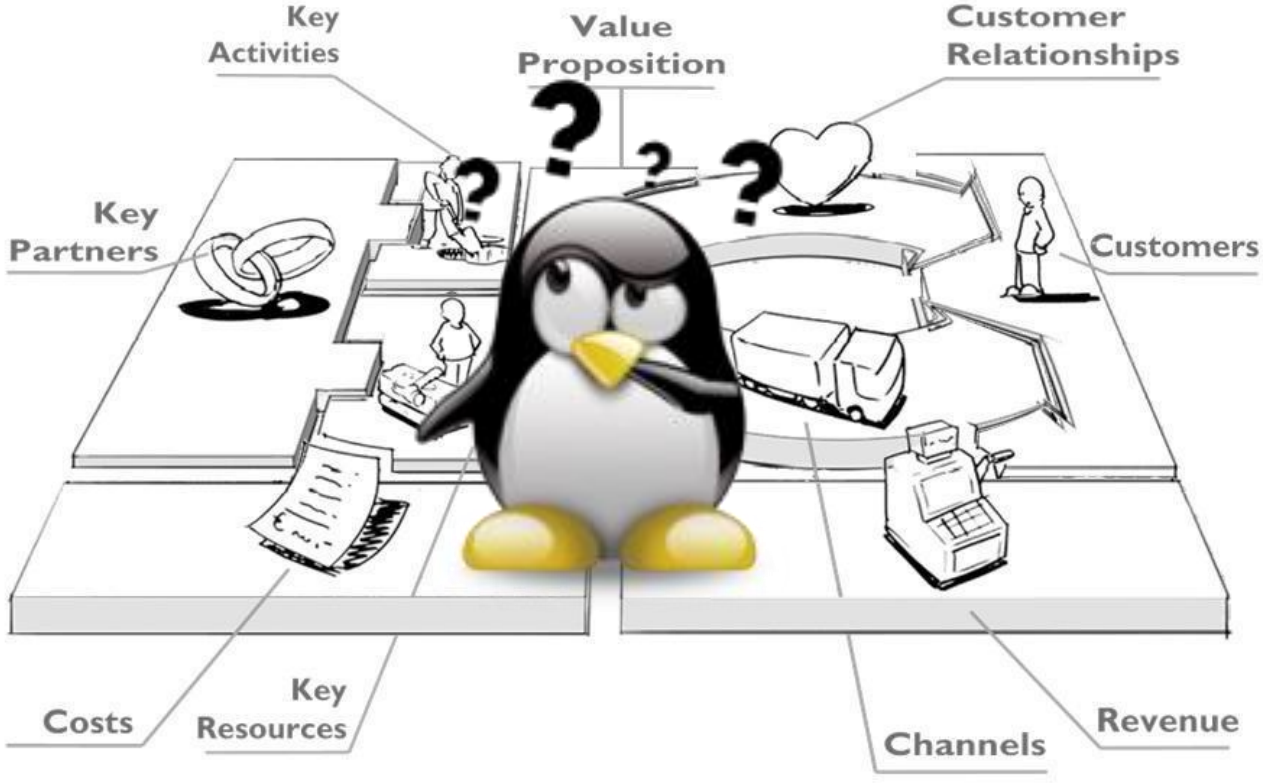


Photo by Joshua Hoehne | https://unsplash.com/fr/photos/cappello-accademico-blu-e-bianco-iggWDxHTAUQ

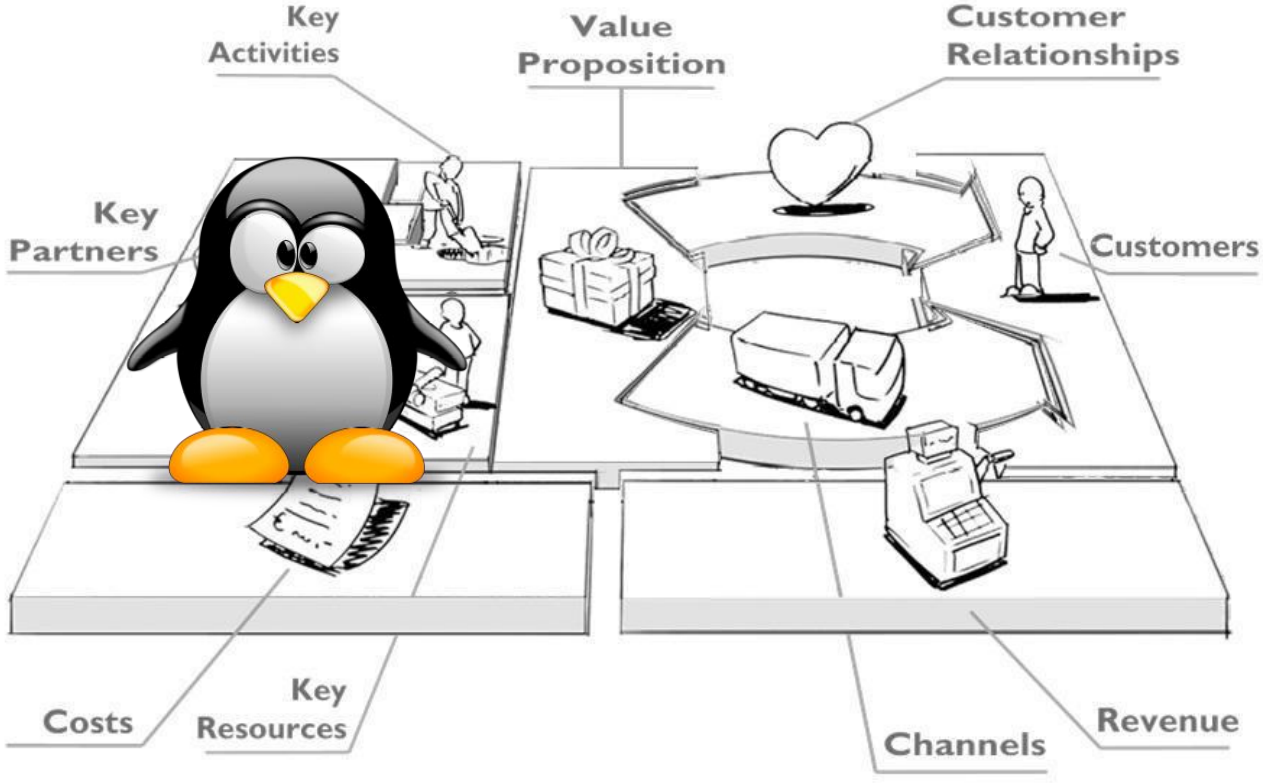
Open Source and Business models



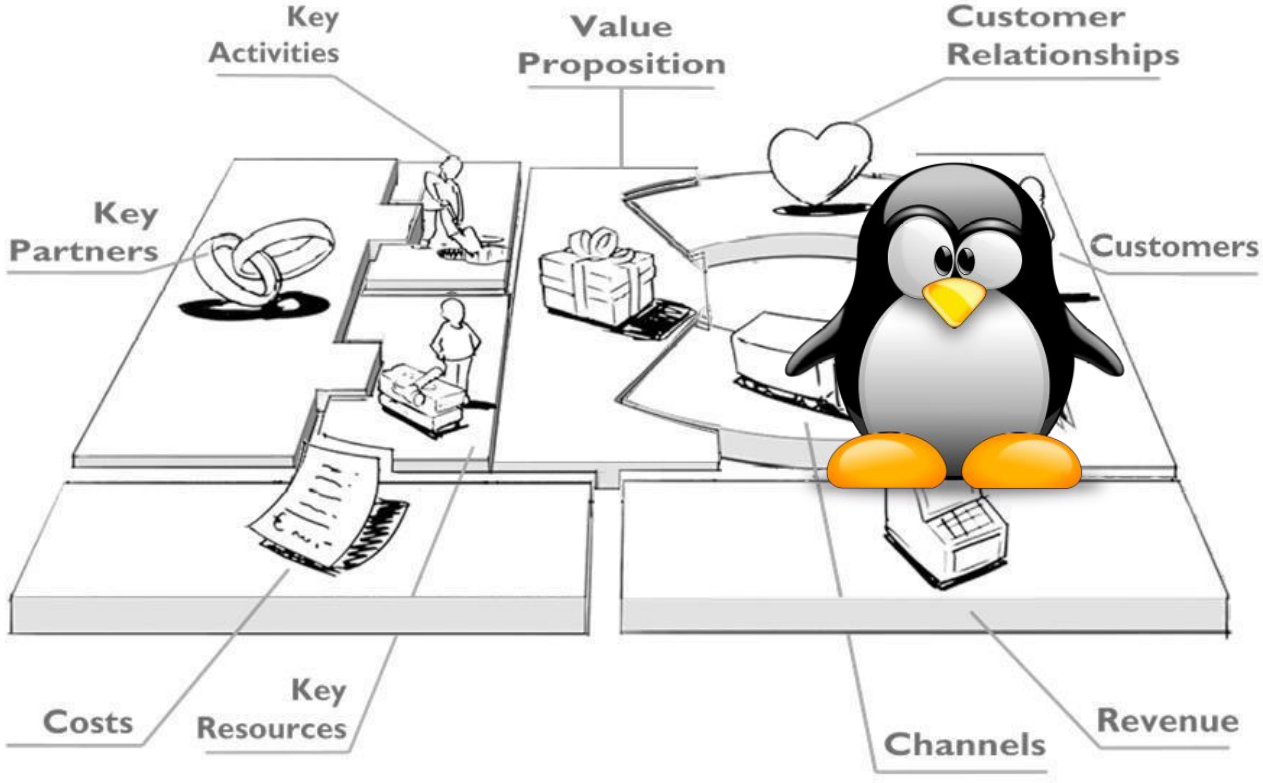
Open Source and Business models



Building block and complement



Distribution, CRM, Sales funnel



Case@RISE: Hopsworks

- An open source data platform for Machine Learning (ML) with a Python-centric Feature Store and MLOps capabilities
- Created in 2016
- 2500 users on the free serverless version on app.hopsworks.ai. 923 stars on GitHub.
- All code is developed by employees of Hopsworks AB
- AGPL 3.0

- Grew from an EU-funded research project aimed at integrating Apache Hadoop. A data science layer was added on top and labeled Hopsworks.
- The software outputs were released as OSS, and eventually, a startup was spun out with the involved researchers.
- Venture capital was initially challenging to attain, given the initial focus why the startup pivoted to focus on the data science layer today, still known as Hopsworks.

RI.
SE

Case@RISE: OpenModelica

- Framework for modeling, simulation, optimization and debugging of cyber-physical systems based on the Modelica open-standard.
- Created in 1997
- ~8000 downloads / month
- 72 unique committers
- https://github.com/OpenModelica
- Dual AGPL and OSMC-PL

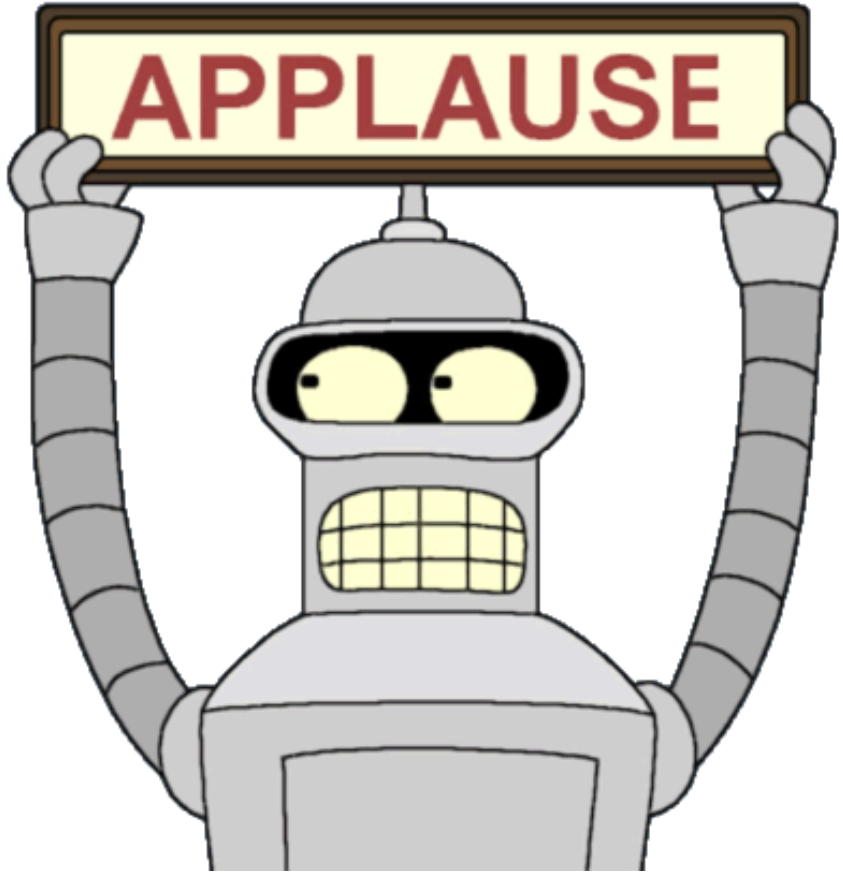
- Supported via funding by companies and universities via the Open Source Modelica Consortium since 2007.
- Creating a non-profit association may help to ensure the long term survival and maintenance of your open-source project via membership fees or direct funding development.

RI.
SE

Case@RISE: ContikiNG

- IoT operating system with support for IPv6-based mesh networking with support for many IoT application protocols (CoAP, LwM2M, HTTP, MQTT).
- Created in 2002 (Contiki) 2017 Contiki-NG
- Several commercial products use Contiki or Contiki-NG. Widely used in academia for low-power IoT research
- 229 contributors (GitHub)
- BSD License
- Project funding – VINNOVA, EU, ITEA, and some directly funded projects
- Several R&D projects have helped to sustain the project
- Forked from the Contiki project after internal disagreements in the community when founders left the project. Separation successful attributed detailed planning.

RI.SE



RI.
SE