



Choosing an open source license and business model for the success of your project

Gilles Gravier

Director, Senior Advisor – Open Source / Blockchain

20190212 – FOSSC, Muscat, Oman

Defining Open Source

Choosing an appropriate license



An open source license gives you permission in advance to innovate via using, studying, improving and sharing the code but also brings obligations to the developer making use of it in their project.

<http://www.opensource.org/licenses/>

<http://choosealicense.com/>

Defining Open Source

One definition to rule them all

For the sake of consistency

1. Free Redistribution of Source Code
2. Source Code Available
3. Allows Derived Works
4. Allows Integrity of Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License with Code
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral



<http://www.opensource.org/licenses/>

Defining Open Source license categories

Choose wisely

What do you want your users and downstream developers to do with your code?

Non-Reciprocal

(non-copyleft)

- “Attribution” licenses
- Fewest requirements, unrestricted scope of use.
- Unrestricted development of derived works
- Examples:
 - Apache: Android, Tomcat
 - BSD: Sudo, FreeBSD
 - ISC: Ripple

File-Level Reciprocal

(weak copyleft)

- Community fostering licenses
- Files derived from common files must be shared with the commons.
- Other derived works may be licensed in any manner.
- Examples:
 - MPL: Mozilla, LibreOffice 4.0 and above
 - LGPL: LibreOffice 3.6 and before

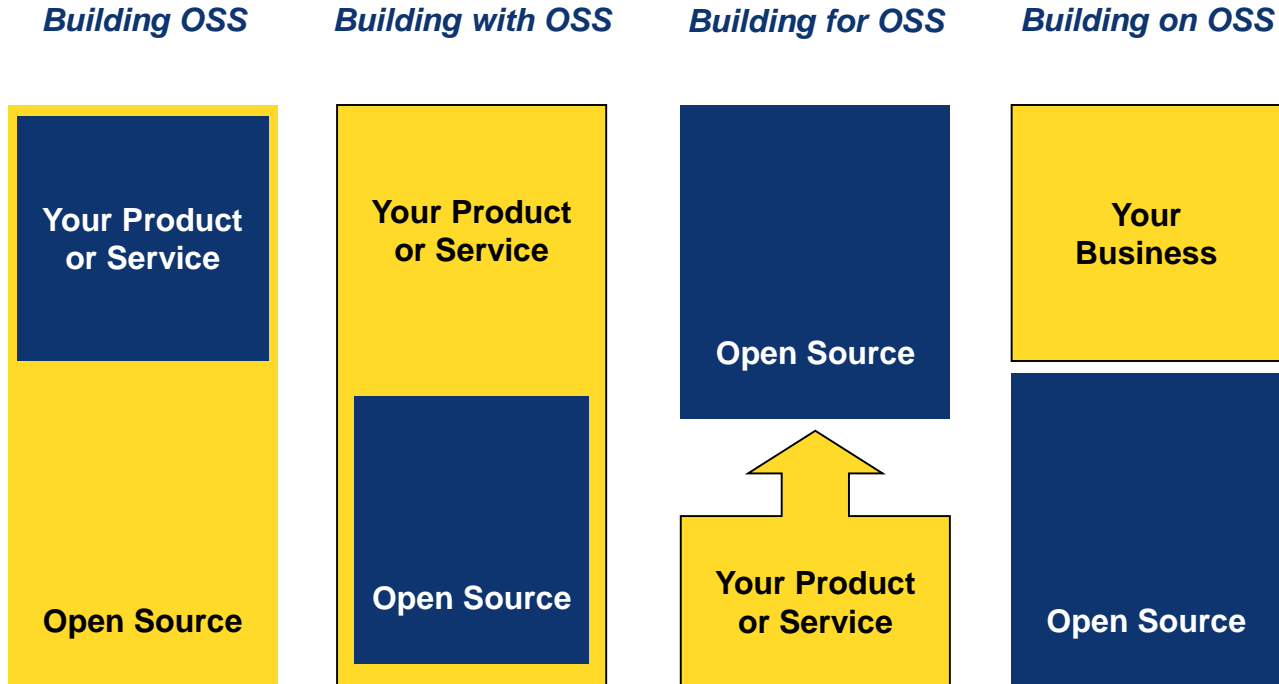
Project-Level Reciprocal

(strong copyleft)

- Project fostering licenses
- Most requirements: seeks to propagate open source.
- All work (including derived works) must be shared with the commons.
- Examples:
 - GPL: Linux
 - AGPL : Funambol
 - CERN OHL : Easy Φ

Business Strategies

Leveraging Open Source software (OSS)



Monetization tactics

Deliver a Service

Deliver a revenue-bearing service using OSS



- AWS is implemented primarily with OSS and OSS-as-a-Service
- Necessary to run Linux, Kubernetes, etc. workloads
- Fast growing component of Amazon revenue stream

The Netflix logo, consisting of the word "NETFLIX" in a white, bold, sans-serif font centered within a red rectangular background.

- High quality internet streaming is THE delivery vehicle
- Nearly all software published as OSS
 - Facilitates uptake, optimization by web carriers



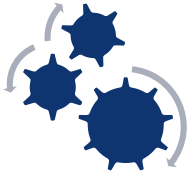
Monetization tactics

Sell services and support

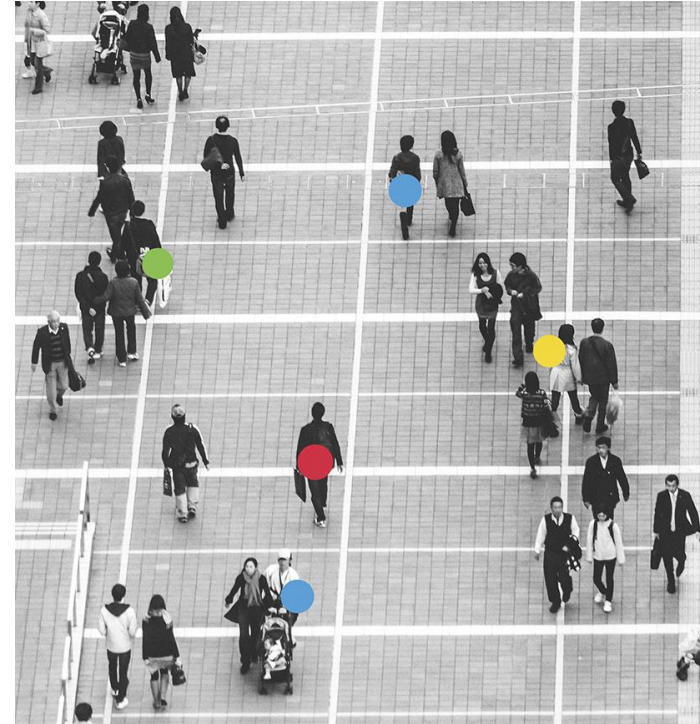
Garner services revenue for technical support / integration of OSS



- Offers technical support for top OSS packages
- Also offers other OSS and non-OSS related services



- Deliver custom solutions using OSS components
- Also integrate non-OSS related services



Monetization tactics

Sell hardware

Reduce cost of OSS components in system products



- Use for OS and network components of most products
- Linux integration essential for many products
- Drive OSS initiatives to lead product compatibility

SAMSUNG

- Use for OS in many products
- Build on Android for market compatibility
- Drive some OSS initiatives to build ecosystem



Monetization tactics

Dual licensing

Garner revenue for commercial re-distribution rights



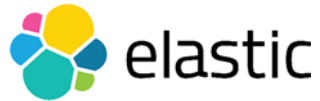
- Server and Tools licensed under AGPLv3.0 (now SSPL)
 - Requires sharing any enhancements (even SaaS)
 - Anti-tivoization conditions embedded use
 - Most commercial users will prefer to pay for a more commercially compatible license
- Also garner revenue from supported product packages



Monetization tactics

Open source with commercial upgrades

Garner revenue from enhanced commercial upgrades



- Elasticsearch and Log products as open source
- Sell a suite of add-on capabilities



- Primary sponsor for Hadoop OSS project
- Offer commercial products with OSS + enhancements



- Creators of Apache Kafka OSS project
- Offer Enterprise version with ops and admin tools



Monetization tactics

Packaged open source software

Sell product packages with support and maintenance



- Create and maintain a curated distribution of Linux
- Sell RHEL – branded, supported and maintained
- Apply similar models with OpenStack, OpenShift, Ansible, Cloud Forms, etc.



- Support and build around standard distributions of Hadoop, Apache Kafka, etc.



Monetization tactics

Strategic objectives

No direct monetization of OSS – Catalyse market change



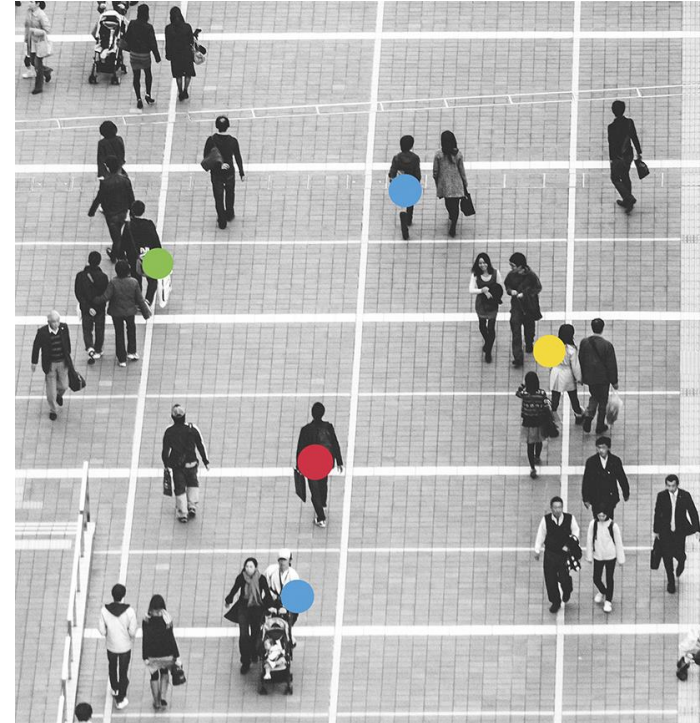
- Drive DevOps platform to share costs and increase developer retention and recruitment (Hygieia)



- Android to dominate mobile devices – sell services
- Kubernetes to drive container apps to their cloud



- Attract all important OSS workloads to Azure
- Make all of their products Azure and cloud compatible



More on indirect strategies

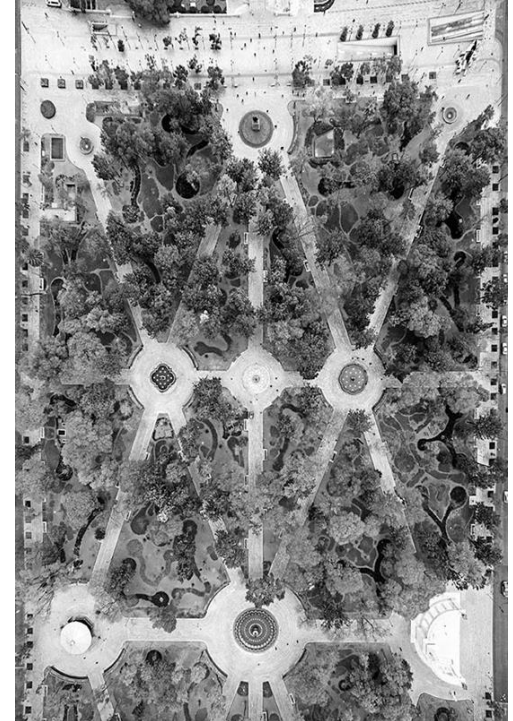
Catalyse market change or trade-offs with direct monetization

Strategies to catalyze market change

- Establish a new paradigm or product category
- Build an ecosystem or market
- Create or strengthen a standard
- Create new competition

Trade-offs with direct monetization

- Restrictive licenses (Dual Licensing) will slow uptake
- Commercial initiatives make potential ecosystem partners into competitors
- Filling gaps with commercial offerings can accelerate uptake



And last....

Strategy first!

Basis

- **Goals & Benefits** – corporate and stakeholder-specific
- **Challenges** – change, compliance, security, competition

Strategy Dimensions

- **Technology** – what components/functionality from where
- **Monetization** – what blend of tactics to realize revenue
- **License** – in-licensing analysis and out-licensing choices
- **Community** – engaging developers, users, coupled OSS projects
- **Ecosystem** – engaging customers, partners, competitors
- **Support** – how to adequately support community, ecosystem members and customers
- **Communication** – how to create awareness and motivate engagement
- **Resources** – financial and human
- **Implementation** – how to realize the goals and benefits



Successful Implementation depends upon *buy-in* by key stakeholders



Thank-you

Gilles Gravier

Director, Senior Advisor – Open Source / Blockchain



<https://www.linkedin.com/in/gillesgravier>



<https://twitter.com/gravax>