

Open Source Projects

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

COSS C02

Licensed under CC BY 4.0 International

Agenda

1. Process definition (open source)
2. Project communities
3. Developer-led foundations
4. User-led foundations
5. Labor economics
6. Example governance models
7. Project life-cycle

1. What is an Open Source Project?

The Other Definition of Open Source

“**Open source is a development method** for software that harnesses the power of **distributed peer review** and **transparency of process**. The promise of open source is **better quality, higher reliability, more flexibility, lower cost**, and **an end to predatory vendor lock-in.**” [O10]

The Cathedral and the Bazaar [R99]

- The Cathedral
 - Carefully crafted work
 - Performed by small group
 - Evolves slowly
 - Bugs and problems are tricky, require deep analysis
 - Project is led by command and control approach
 - No early releases nor beta
- The Bazaar
 - Incremental work
 - Performed by large group
 - Evolves rapidly
 - Linus' law: "Given enough eyeballs, all bugs are shallow"
 - Project is lead by convincing people, nudging them on
 - "Release early, release often"

The Linux Kernel and the Linux Foundation

- The Linux kernel
 - Is an operating system kernel
 - Was originally developed by Linus Torvalds
 - Has a hierarchical development model (BDFL)
- The Linux Foundation
 - Is a U.S.-based 501(c)(6) foundation
 - Was founded to protect and stabilize Linux
 - Is now a major home for new open source initiatives



The Apache Way

- Official [A10b]
 - Collaborative software development
 - Commercial-friendly standard license
 - Consistently high quality software
 - Respectful, honest, technical interaction
 - Faithful implementation of standards
 - Security as a mandatory feature
- Curcuru [C17]
 - Charity
 - Community
 - Consensus
 - Merit
 - Open
 - Pragmatic

A “Patchy” Web Server and the ASF

- The Apache Web Server
 - is the most widely used http demon (> 50% market share)
 - gave rise to a new development model, codified as the Apache way
- The Apache Software Foundation
 - Is a 501(c)(3) foundation “in the public interest”
 - Was founded to stabilize and protect the Apache open source software



Open Collaboration [R+09]

- **Open Collaboration**

- **Egalitarian**

- Open for contribution
 - Everyone can contribute

- **Meritocratic**

- Public discussion process
 - Decisions based on merit

- **Self-organizing**

- People find their process
 - People find their project

- **Traditional Work**

- **Hierarchical**

- Closed and hidden silos
 - Assigned to project

- **Status-based**

- Public + private discussions
 - Hierarchical status decides

- **Assigned tasks**

- Prescribed process
 - Prescribed jobs

Open Source Project (= People + Software)

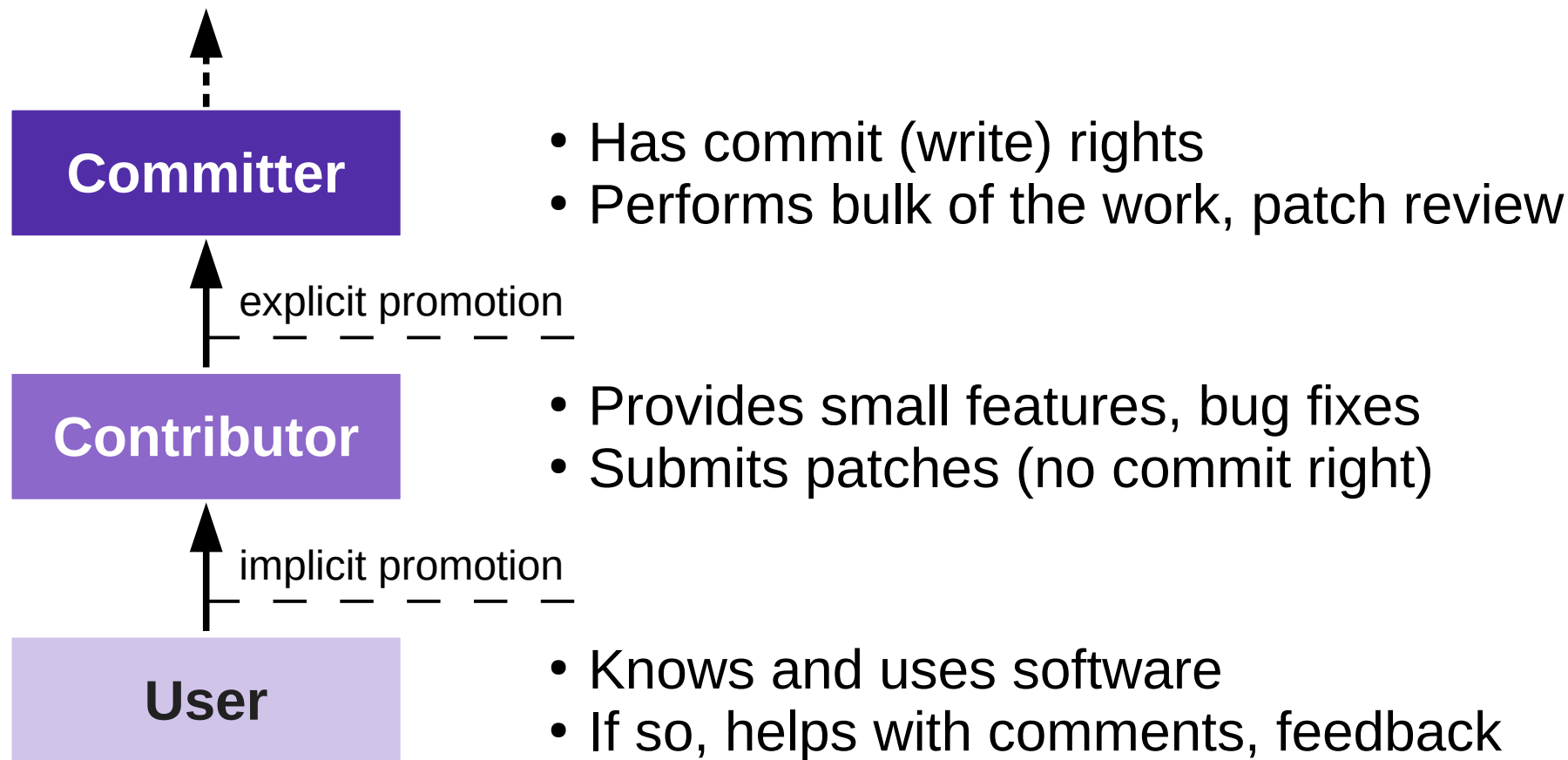
- An **open source project** is
 - Not a project but rather a piece of software
 - Provided under an open source license
 - **Developed by a community of people**
 - **Using the principles of open collaboration**
- A community of people here is a
 - Group of people independent of each other
 - That relate to and collaborate with each other
 - For a shared common goal of the software
- Apache's definition of independence
 - At least from three different legal entities
 - Legal entities = {juristic person, natural person}
 - Called committer diversity

2. Project Communities

Open Source Project Community

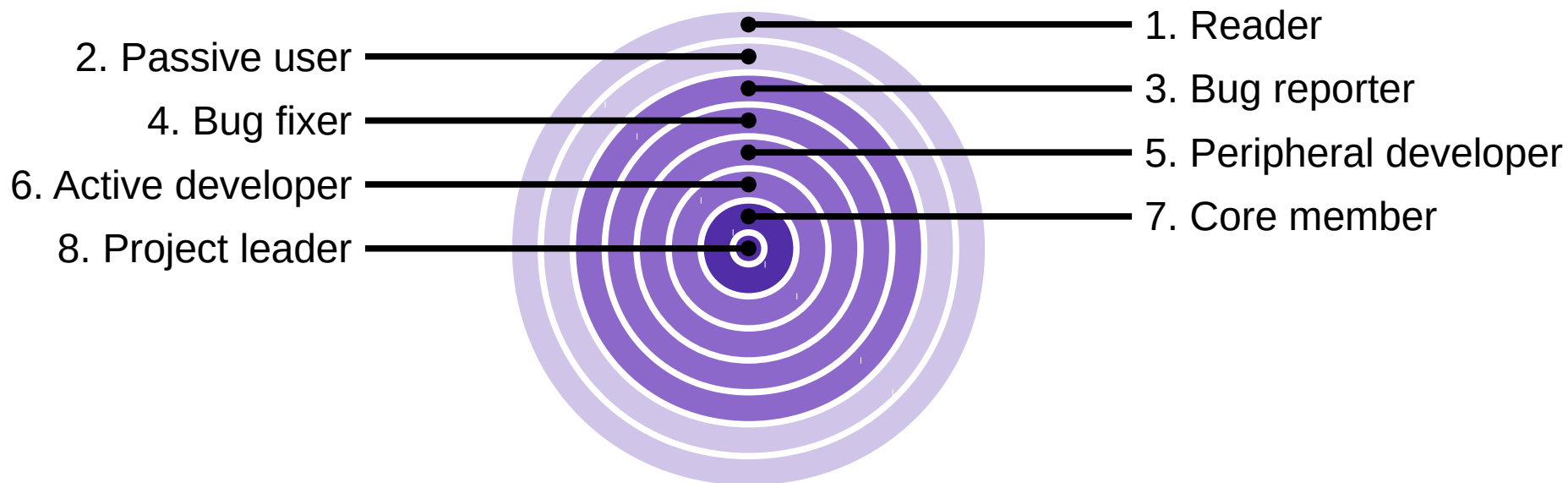
- An **open source project community** is
 - The group of people and companies engaged in an open source project
- The **developer community** is
 - The subset of the project community that is developing the software
- The **user community** is
 - The subset of the project community that is using the software

The Basic Career Path [R15a]



[1] Committers are called “maintainer” (mostly in the Linux context)

The Onion Model [CH11]

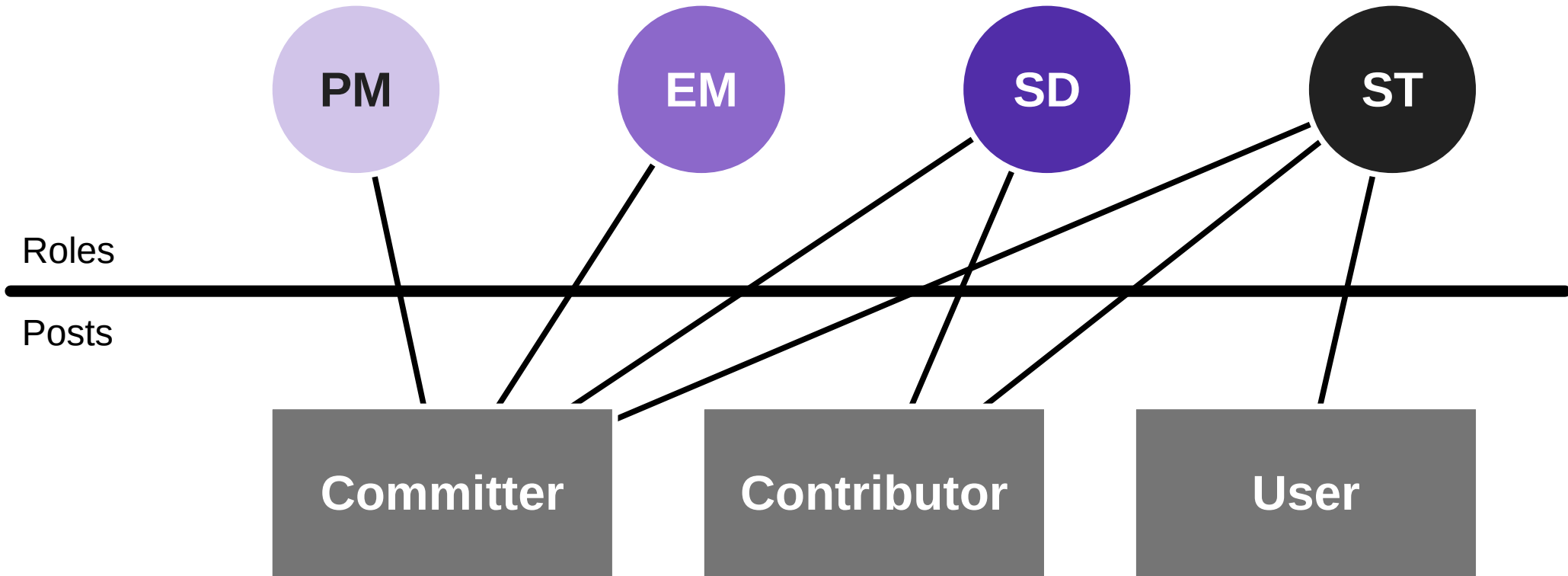


User

Contributor

Committer

Roles and Posts in Open Source Projects



Product Management in Open Source

- Strategic product management
 - Does not take place in open source
- Technical product management
 - Product roadmapping
 - Some is performed but often is ad-hoc
 - Product specifications
 - Barely exist as documents (wikis, to-do lists, other)
 - Progress tracking
 - Managed by time, it is done when it is done

Engineering Management in Open Source

- Release planning
 - See product management: Some is performed but often is ad-hoc
- Resource allocation
 - Committers can prod contributors
 - Usually contributors pick up what they like to work on
- Process improvement
 - Ad-hoc, if any

Software Development in Open Source

- Programming
 - Like in closed source, but in general with less visibility as to completion

Quality Assurance in Open Source

- Code review
 - Is the core ingrained best practice followed by open source projects
- Automated testing
 - Like in closed source, though perhaps a bit more ad-hoc in general
- Manual testing
 - A lot of user testing, significantly more than in closed source
- Release management
 - Like in closed source, committers play release manager

Governance

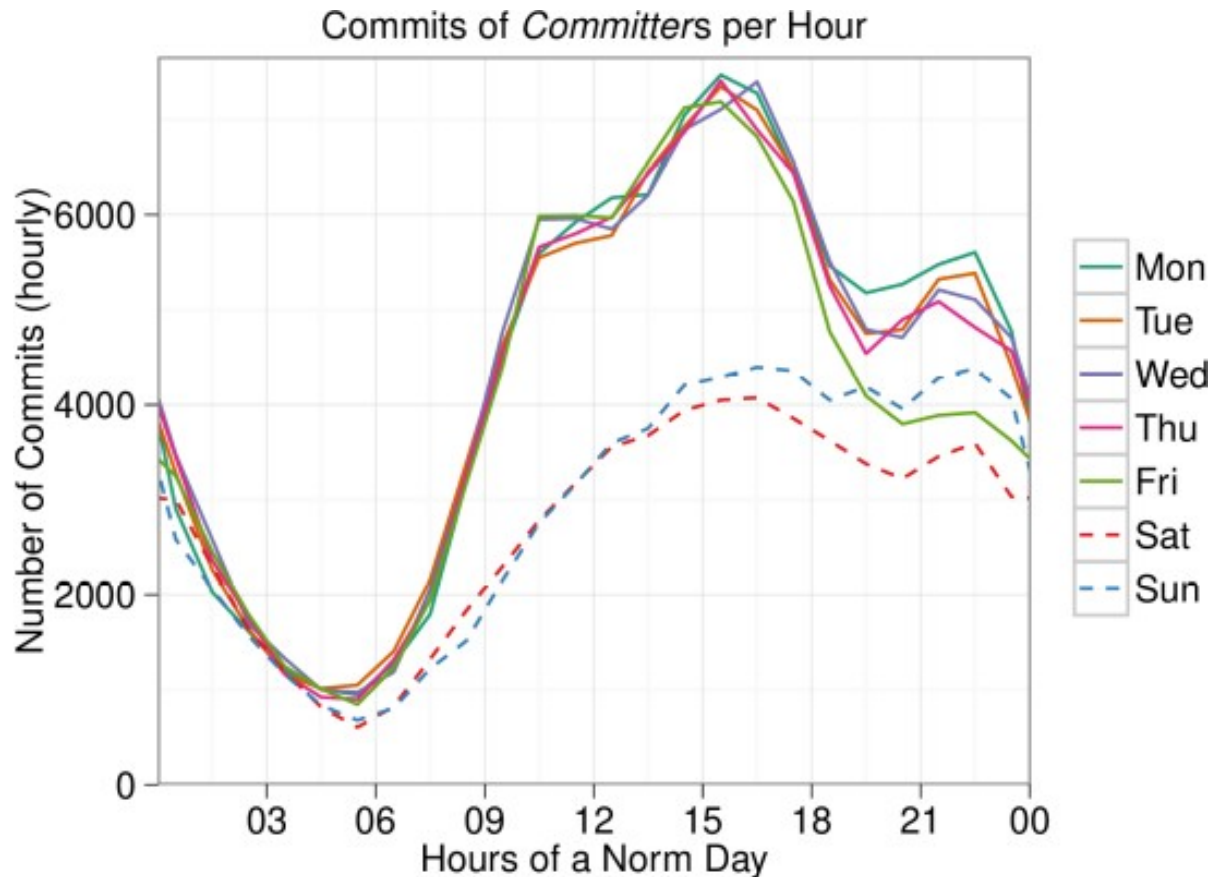
- Governance is
 - The set of processes, practices, institutions, and roles used to lead and manage a social system

Governance Models

- The BDFL (“benevolent dictator for life”)
- The peer group (cf. Apache)
- The do-ocracy

“At least 65% of the code which went into 2.6.20 was created by people working for companies.” [C07]

Daily and Weekly Work in Open Source Projects [R+14]



3. Developer-led Foundations

Developer-led Open Source Foundations (Recap)

- An **developer-led open source foundation** is
 - a **non-profit organization** (foundation, consortium)
 - with the purpose of **managing and performing the development** of
 - **non-differentiating open source software**
 - made available to foundation members and **the general public**
- Typical members of a developer-led foundation are
 - Individual developers
 - Software vendors
 - Service providers
 - Consulting firms

Motivation for Developer-led Foundations (Recap)

1. Support and protect people and their projects
 - Protect developers from lawsuits
 - Accept donations; manage books
 - Legally represent project interests
2. Create a stronger more competitive ecosystem
 - Increase share of customer wallet
 - Reach more price-sensitive customers
 - Grow addressable market
3. Create a fair and equal playing field for members
 - Regulate access to intellectual property
 - Regulate access to marketing channels
 - Regulate product roadmap and development

Solutions to Abuse of Relevant Property Ownership

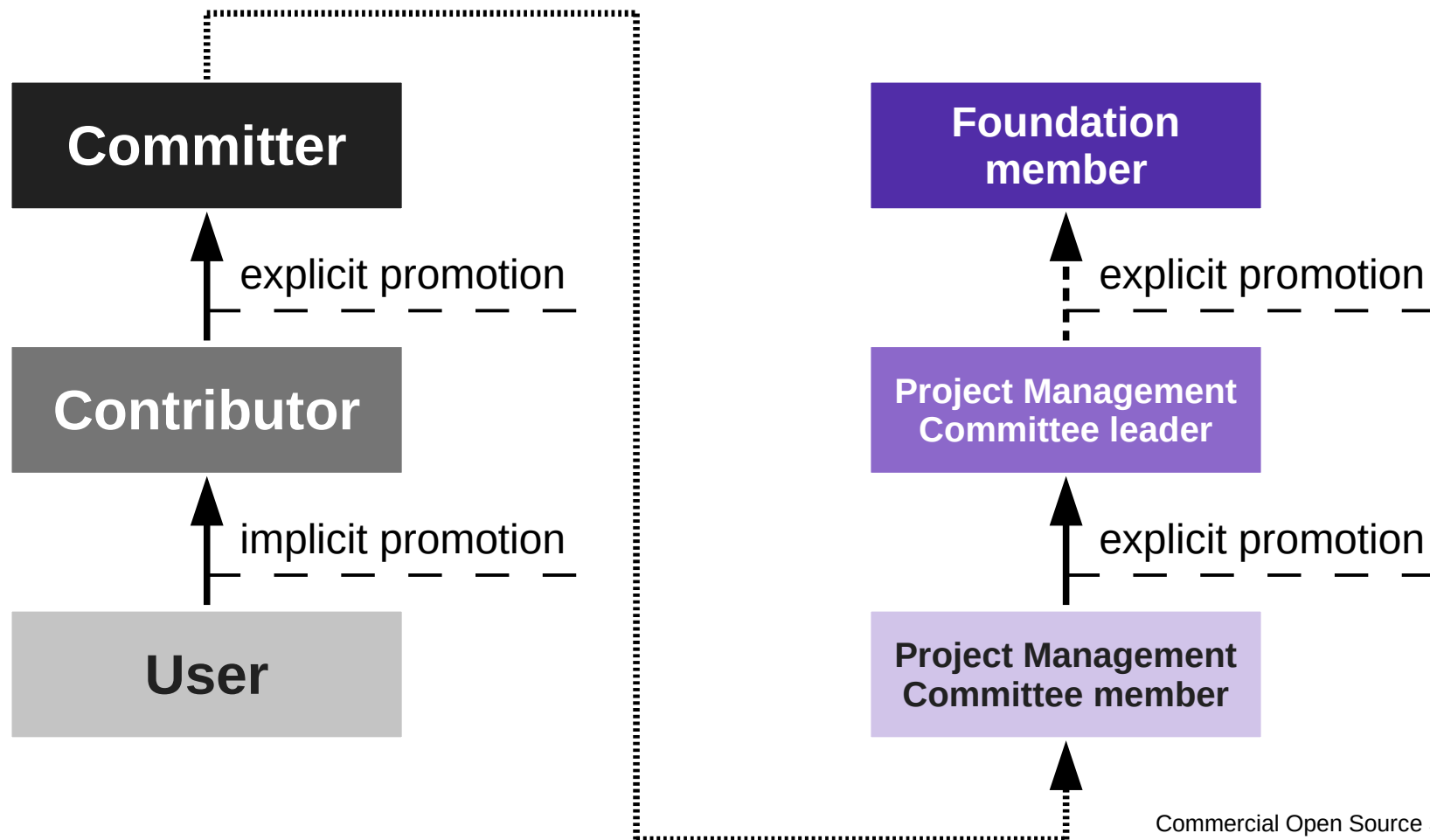
- The foundation becomes the owner of all relevant (“chokepoint”) properties
 - Re: **copyright**
 - The foundation acquires copyright via contributor license agreement
 - The foundation licenses out using an agreed-upon open source license
 - Re: **patent rights**
 - The foundation chooses an open source license with a patent usage rights grant
 - The foundation arranges for a FRAND [1] patent usage rights grant (uncommon)
 - Re: **trademark rights**
 - The foundation acquires all necessary trademark rights
 - The foundation clarifies access to and usage rights of trademarks
 - Re: **other properties** (e.g. domain names)
 - The foundation acquires them and handles them appropriately

[1] FRAND = “fair, reasonable, and non-discriminatory”

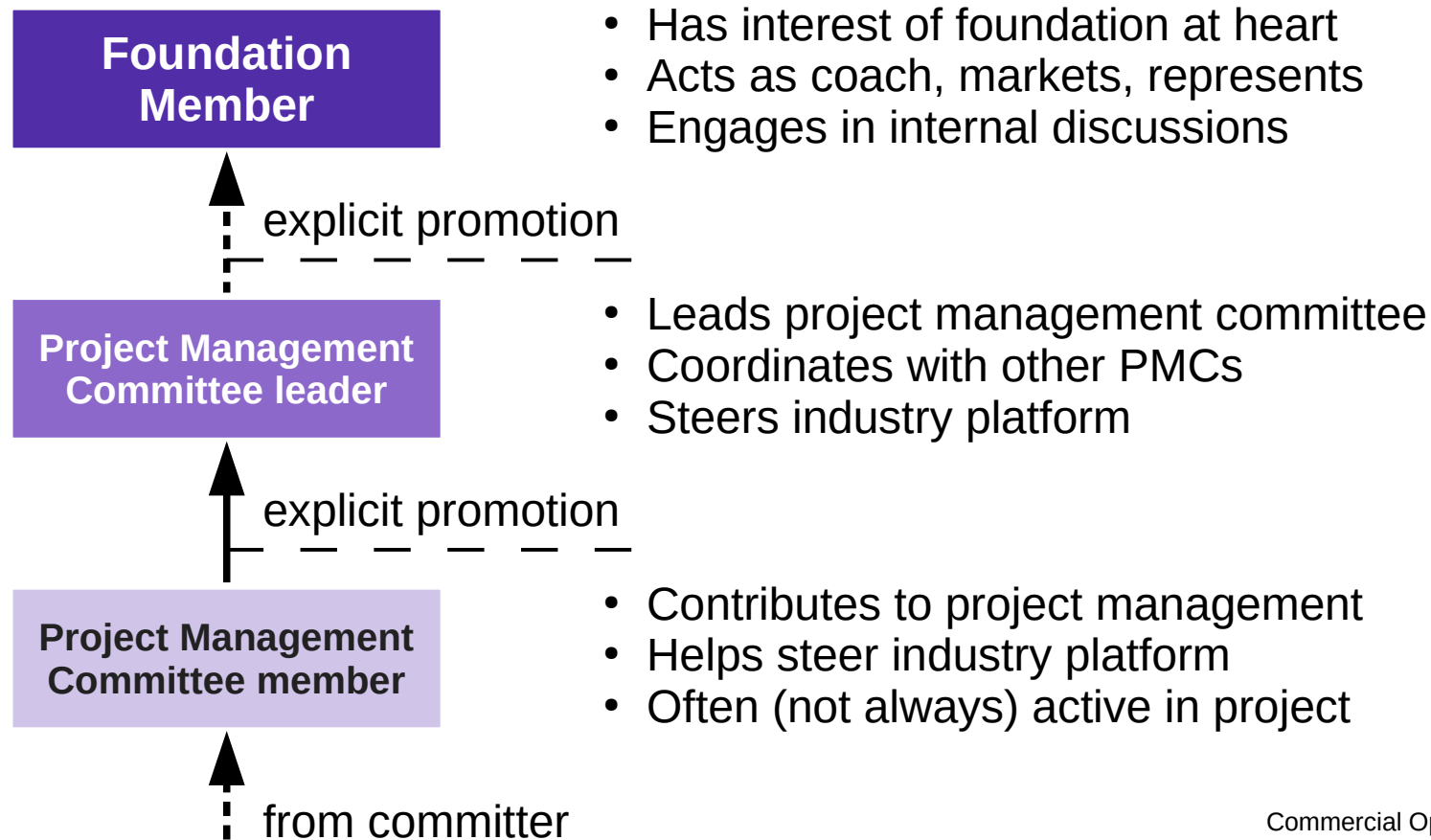
Solutions to Abuse of Control of Position of Power

- The foundation defines governance processes that resolve project strife
 - Re: **social leadership**
 - The foundation sets-up appropriate processes to avoid strife in the first place
 - If necessary, the foundation can still step in if project leaders misbehave
 - Re: **committer rights**
 - See abuse of social leadership

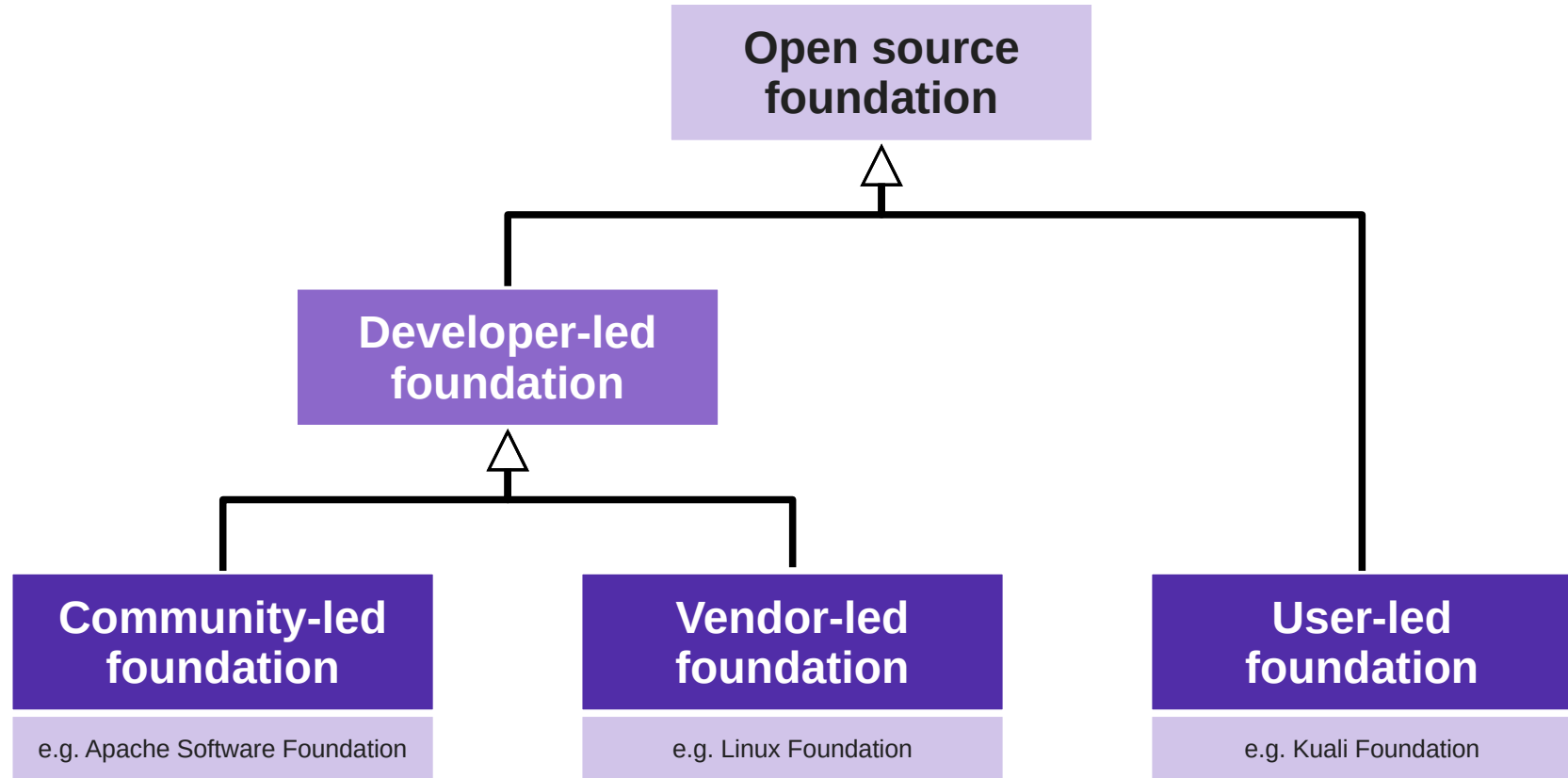
The Extended (Foundation) Open Source Career Path



Tasks and Purpose of New Positions



Types of Open Source Foundations (Recap)



Dysfunctions of Vendor-led Foundations

- Misuse of projects as marketing channels
- Special provisions in bylaws for ensuring influence

4. User-led Foundations

User-led Open Source Foundation (Consortium)

- A **user-led open source consortium** is
 - a **non-profit organization** (foundation, consortium)
 - with the purpose of **funding and managing the development** of
 - **non-differentiating open source software**
 - made available to foundation members and **the general public**
- Typical members of a user foundation are
 - Software user firms
 - Software vendors
 - Consulting firms
 - Service suppliers

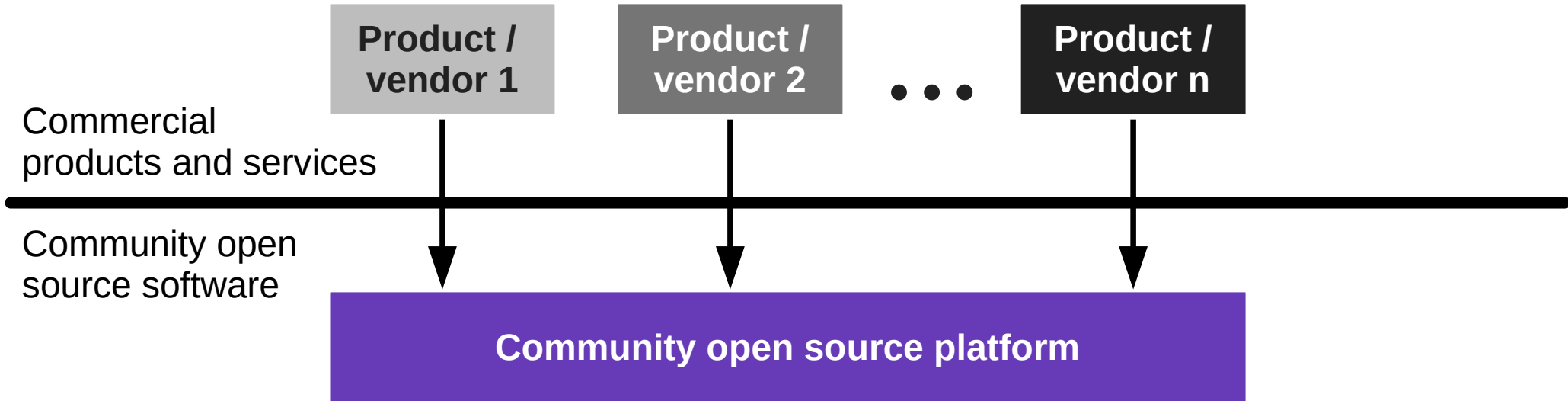
Example User-led Foundations



Purpose of a User-led Foundation

- To get rid of or reduce the problems of vendor lock-in
 - High total cost of ownership
 - Innovation blockage
 - Operational risk
- By establishing a supplier ecosystem
 - For software products and services that
 - Is based on community open source software and
 - Conforms to the economics of the sponsors

Software and Services Ecosystem of Suppliers

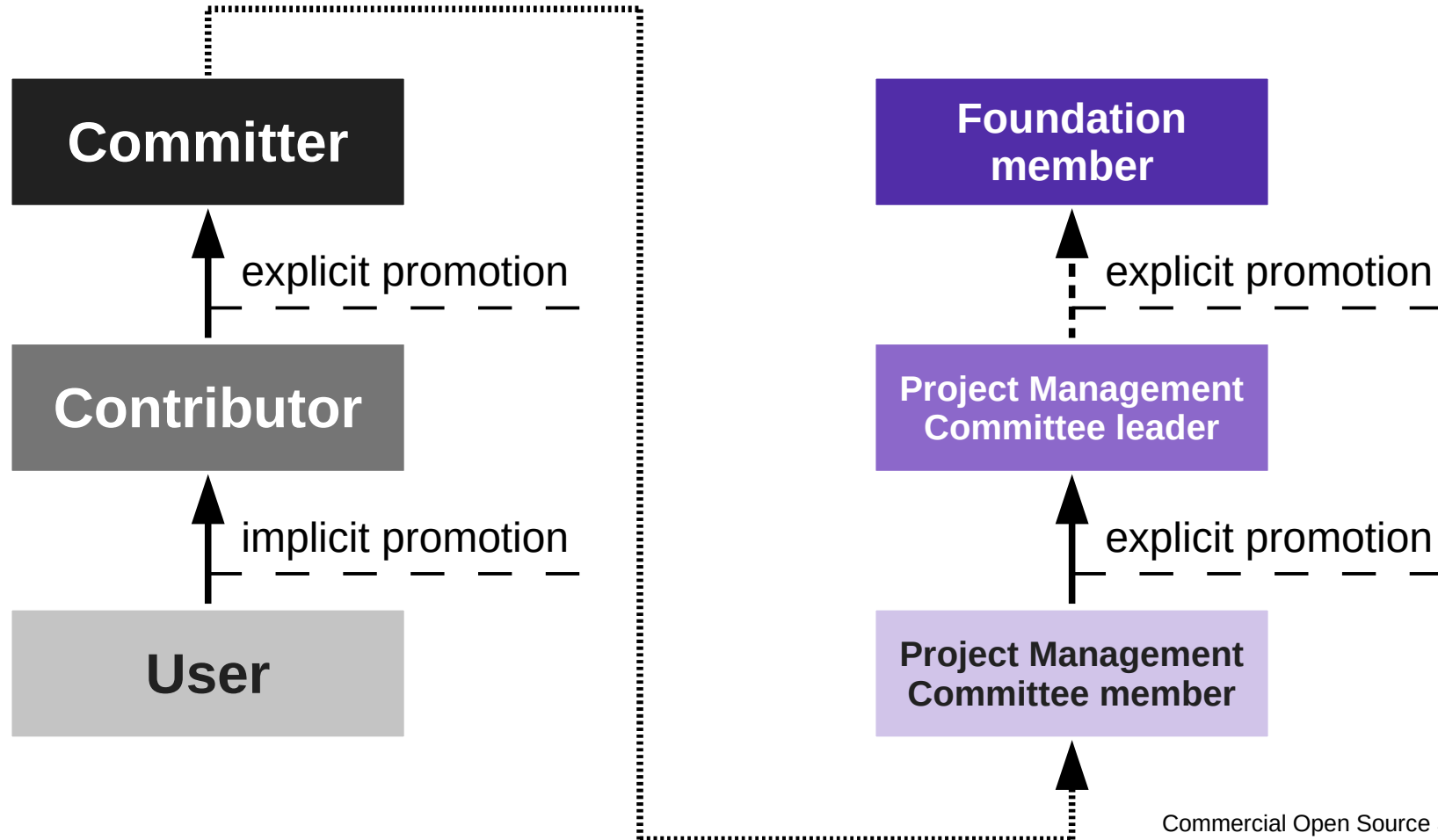


The Next Billion Dollar Opportunity

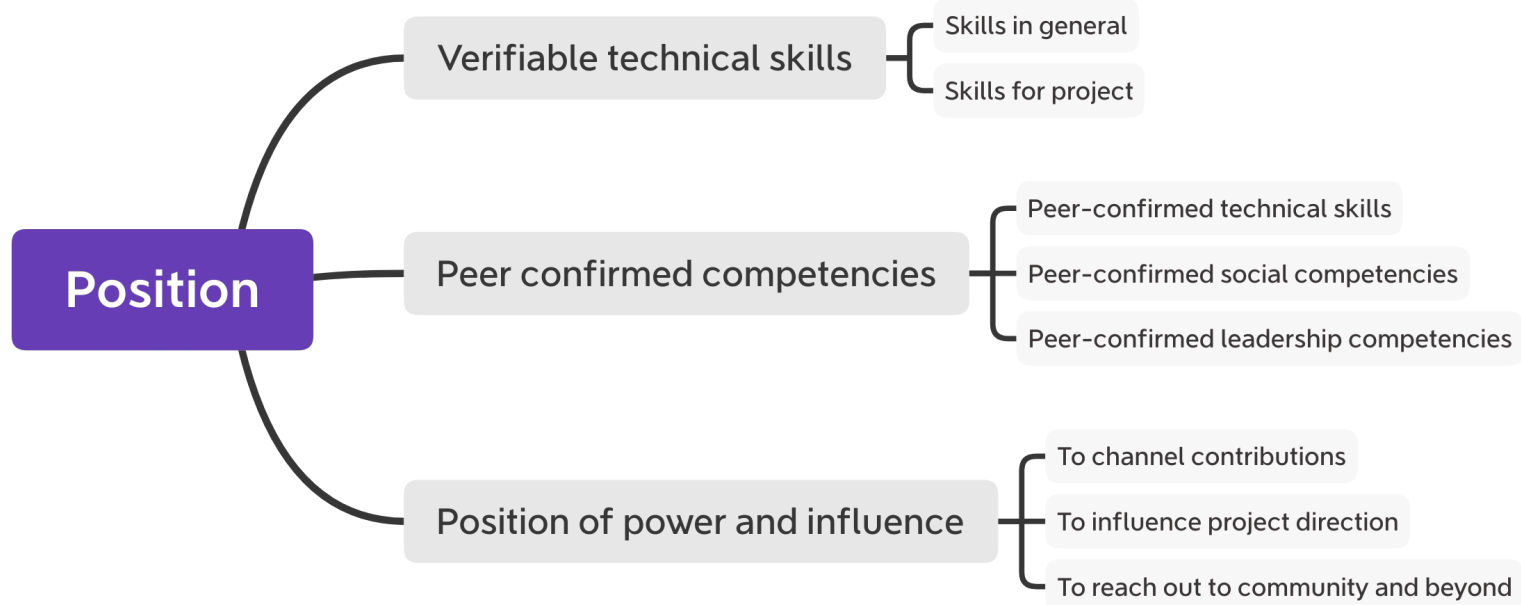


5. Labor Economics

Position in Project = Status (Recap)



Value of Status in Open Source Project to Employers



Resulting Value to Developer

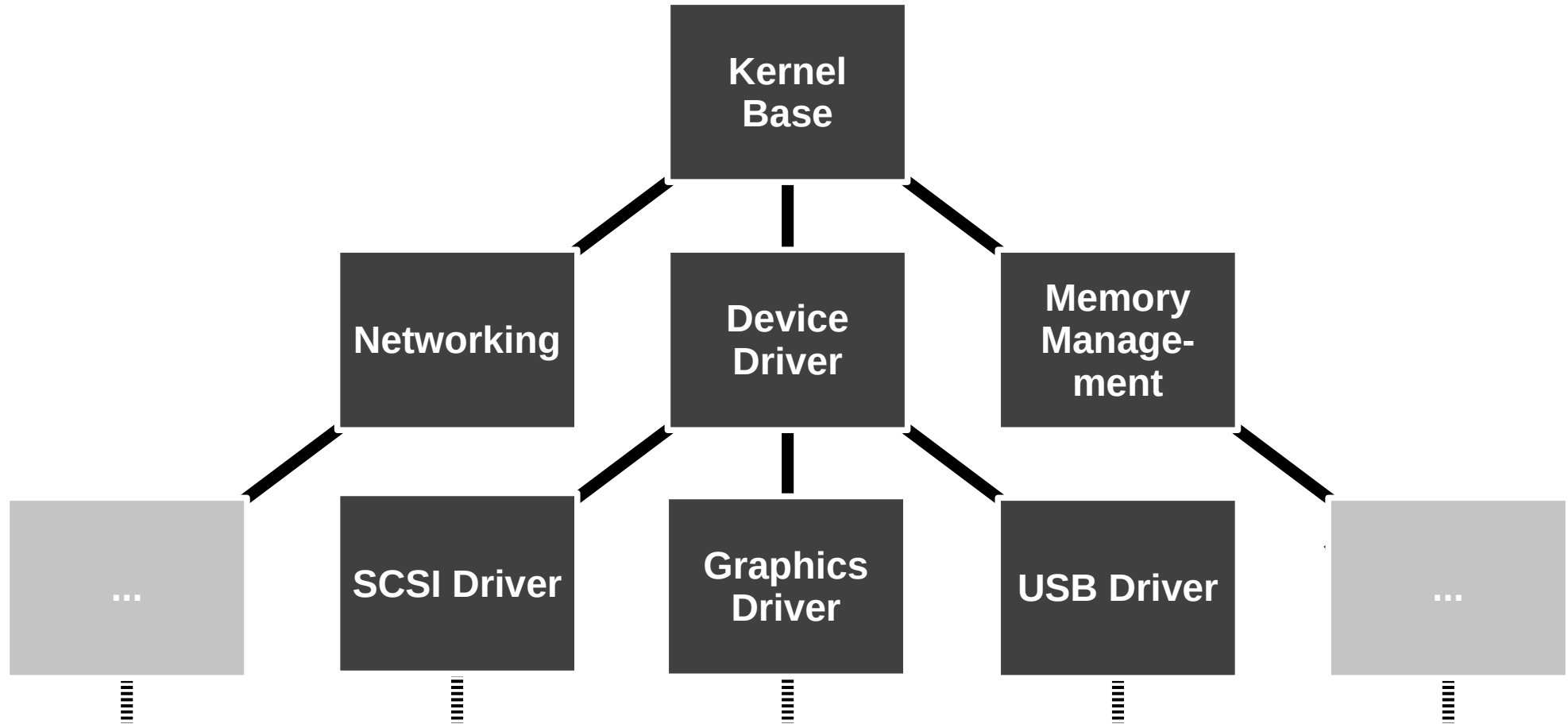
- Better negotiation position
- Higher job security
- Richer job experience

The Developer Labor Market

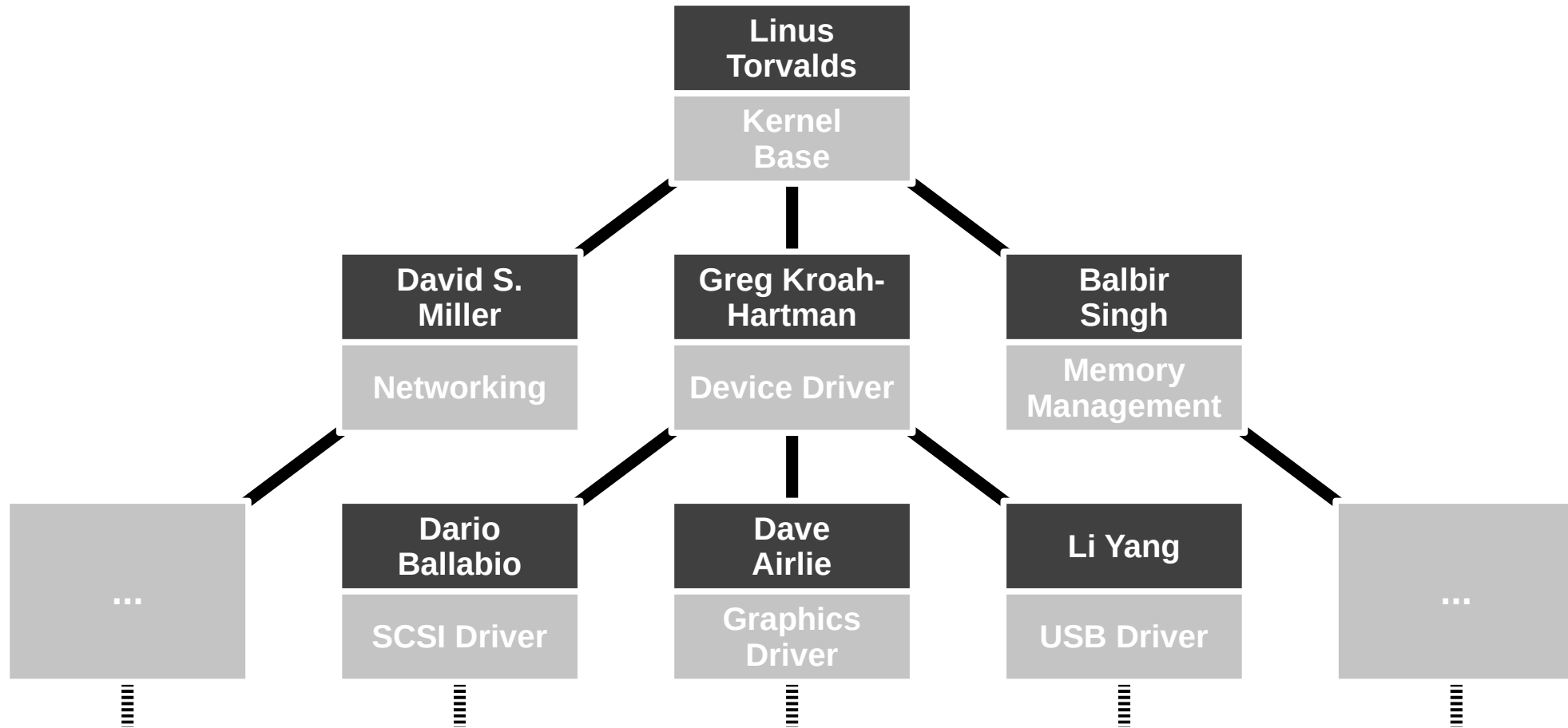
- A weaker position towards an employer
 - Because of more competition in general
- A stronger position towards an employer
 - If can get a coveted open source position

6. Example Governance Models

Illustration of the Code Architecture of the Linux Kernel

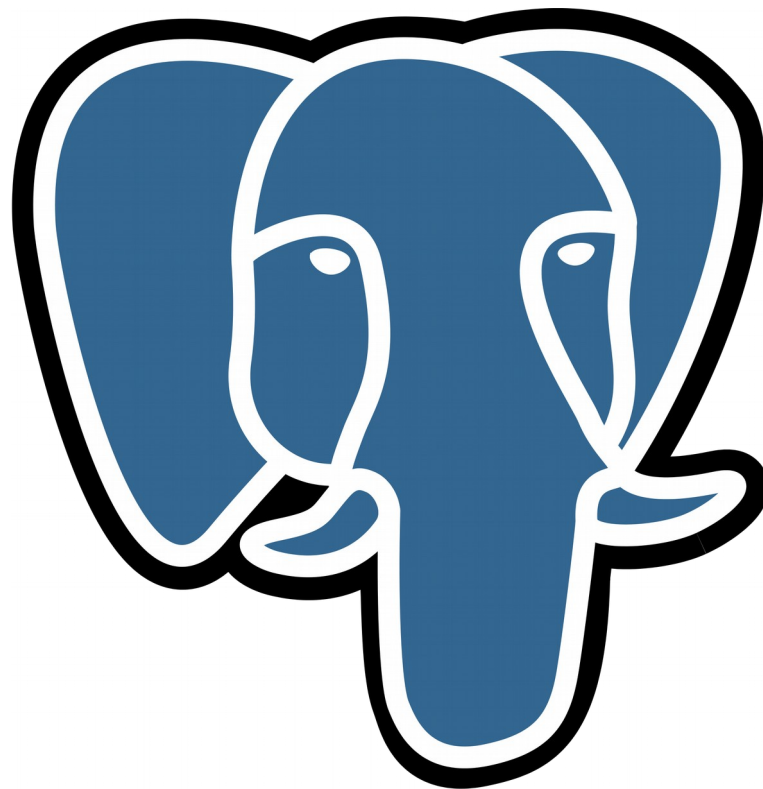


The BDFL Model or Conway's Law Applied



Engineering Management in PostgreSQL

- Core team (long-term stable six members)
 - Coordinate release management activities
 - Facilitate consensus building process
 - Decide if community consensus fails
 - Administer major assets (website, repository)
- 20+ major contributors (committers)
 - Have commit rights, contribute to code base
- (Regular) contributors
 - Contribute to code base through committers
- “Hacker Emeritus” and past contributors

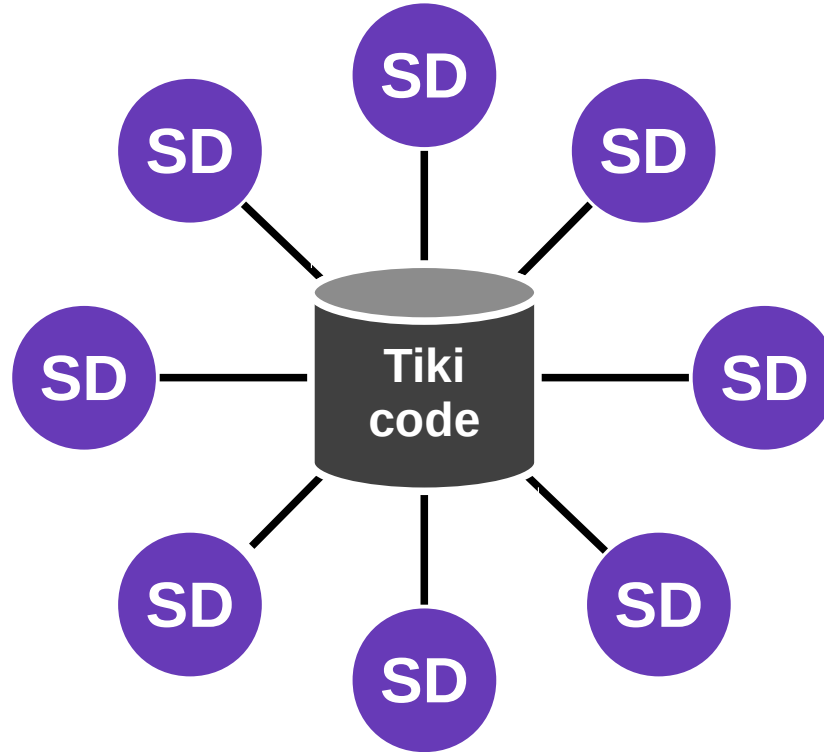


Product Management in PostgreSQL

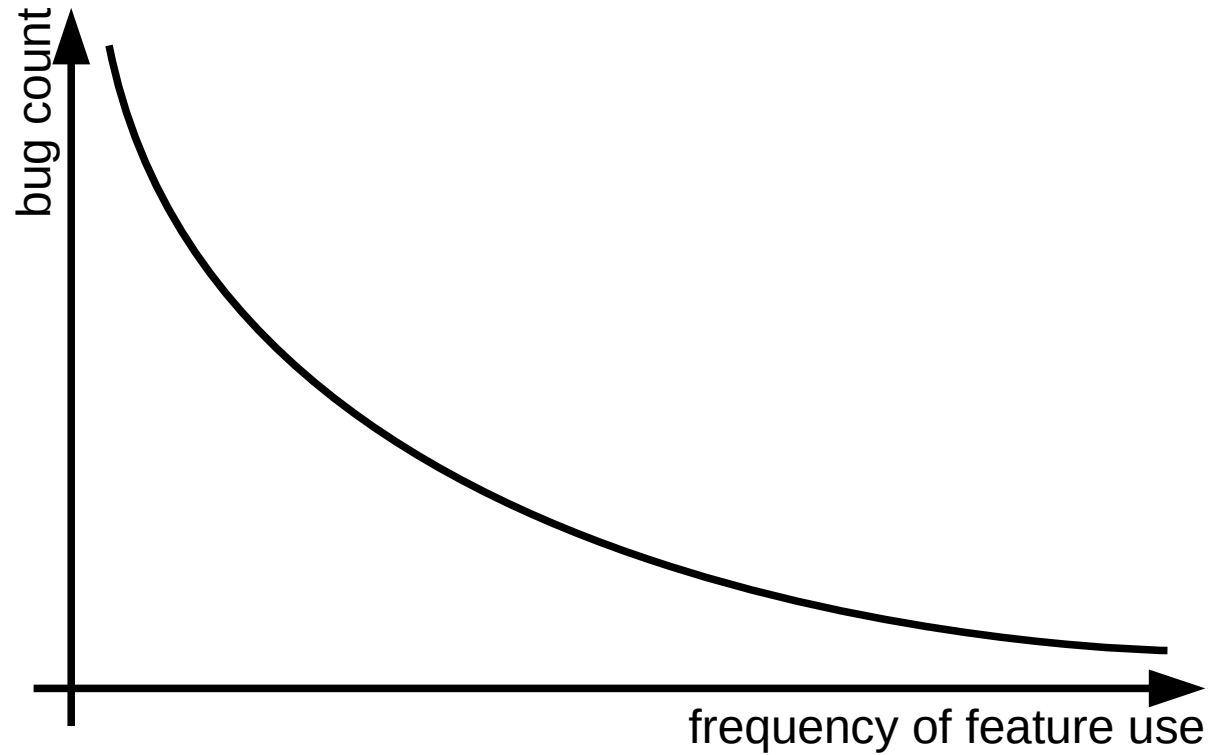
- Product roadmap is defined by core team [2]
- Product backlog is maintained as a to-do list
- There is no prioritization: “Pick your feature”
- Community uses user polls for market research

Tiki's Do-ocracy

- Tiki has a plug-in based software architecture
- Everyone gets access (upon request)
- Has strong code ownership



An Example of “Just Good Enough”

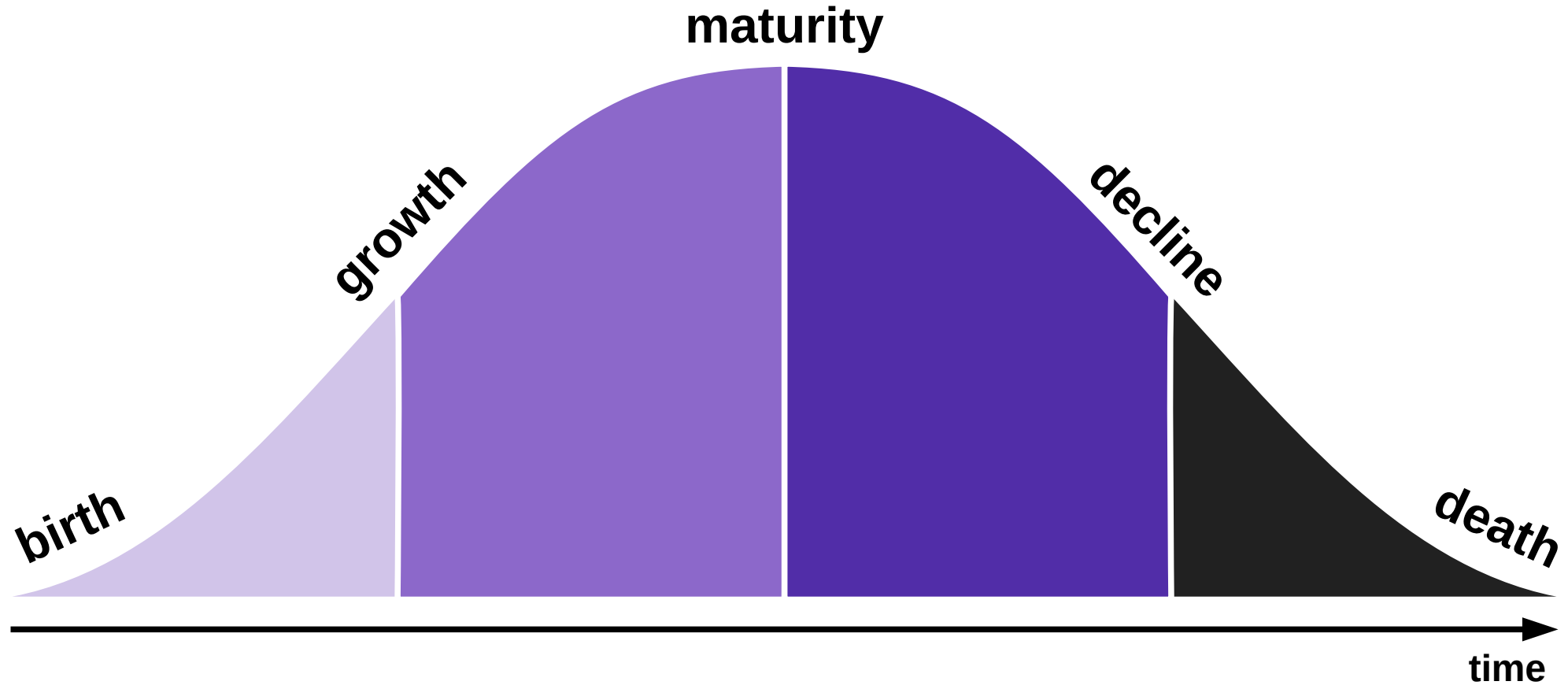


Three Different Governance Models

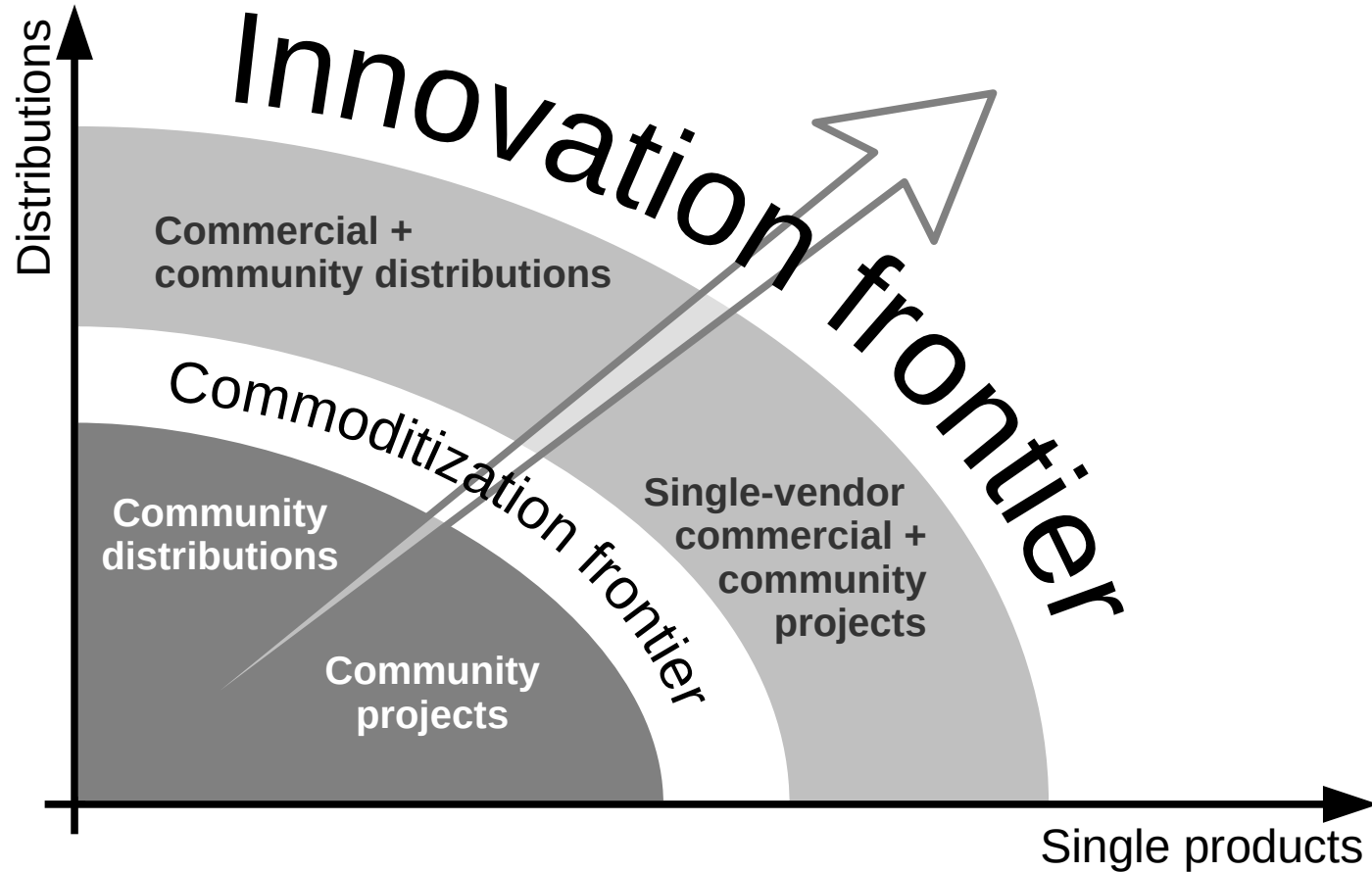
- The BDFL Model
 - Hierarchical with strong code ownership
 - Example: The Linux kernel
- The Peer Group model
 - Core group of committers and evangelists
 - Example: The PostgreSQL RDBMS
- The Do-ocracy
 - Those who do, rule
 - Example: Tiki Wiki CMS Groupware

7. Project Life-cycles

Life-cycle of Open Source Projects (Similar to Products)



Innovation and Commoditization



Summary

1. Process definition (open source)
2. Project communities
3. Developer-led foundations
4. User-led foundations
5. Labor economics
6. Example governance models
7. Project life-cycle

Thank you! Questions?

dirk.riehle@fau.de – <http://osr.cs.fau.de>

dirk@riehle.org – <http://dirkriehle.com> – [@dirkriehle](#)

Credits and License

- Original version
 - © 2020 Dirk Riehle, some rights reserved
 - Licensed under [Creative Commons Attribution 4.0 International License](#)
- Contributions
 - None yet