Open Source Developer Foundations

Prof. Dr. Dirk Riehle
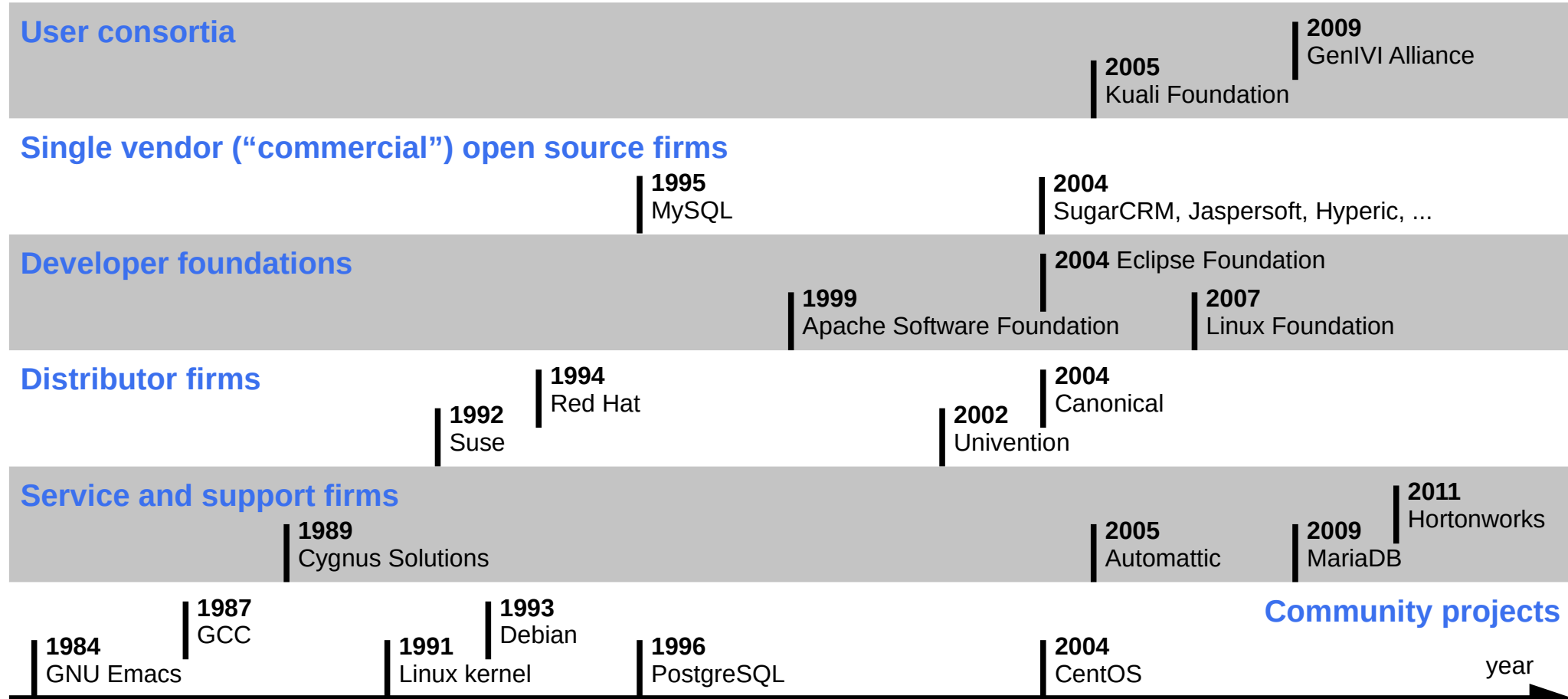
Friedrich-Alexander University Erlangen-Nürnberg

FOSS C02

1. **Legal innovation**
2. **Process innovation**
3. **Software tool innovation**
4. **Business model innovation**

Evolution of Open Source Projects (Recap)

User consortia

2009 GenIVI Alliance

2005 Kuali Foundation

Single vendor ("commercial") open source firms

1995 MySQL

2004 SugarCRM, Jaspersoft, Hyperic, ...

Developer foundations

2004 Eclipse Foundation

1999 Apache Software Foundation

2007 Linux Foundation

Distributor firms

1994 Red Hat

1992 Suse

2002 Univention

2004 Canonical

Service and support firms

2011 Hortonworks

1989 Cygnus Solutions

2005 Automattic

2009 MariaDB

Community projects

1987 GCC

1984 GNU Emacs

1993 Debian

1991 Linux kernel

1996 PostgreSQL

2004 CentOS

year

Open Source "Business Models" (Recap)

- Non-profit open source

  - **Community projects** without foundation
  - Open source **developer foundations**
  - Open source **user foundations**

- For-profit open source

  - **Service and support firms**
  - Open source **distributor firms**
  - **Single-vendor** open source **firms**

Open Source Developer Foundations [R10a]

- An **open source developer 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 foundation are
  - Software vendors
  - Service providers
  - Consulting firms

Motivation for Developer Foundations

1. Support and protect people and their projects

   - Protect developers from lawsuits
   - Ensure good open source governance
   - 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

The Apache Software Foundation […] provides an established framework for intellectual property and financial contributions that simultaneously limits contributors potential legal exposure. Through a collaborative and meritocratic development process, Apache projects deliver […] software products […]
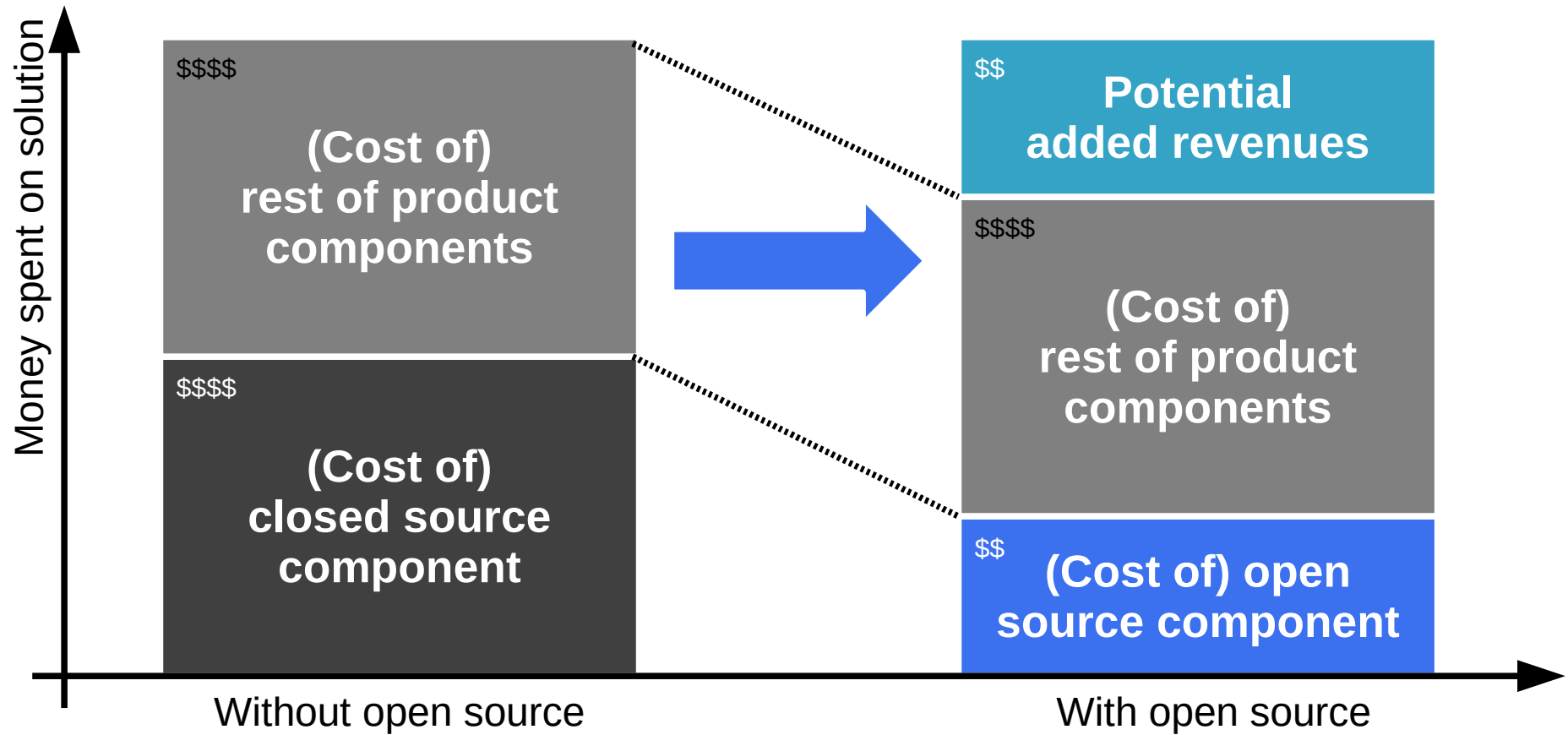
Motivation for Developer Foundations 1 / 3

- Support and protect people and their projects
  - Support and protect people
    - Take administrative work off their hands
    - Protect developers from lawsuits

  - Support and protect projects
    - Ensure good open source governance
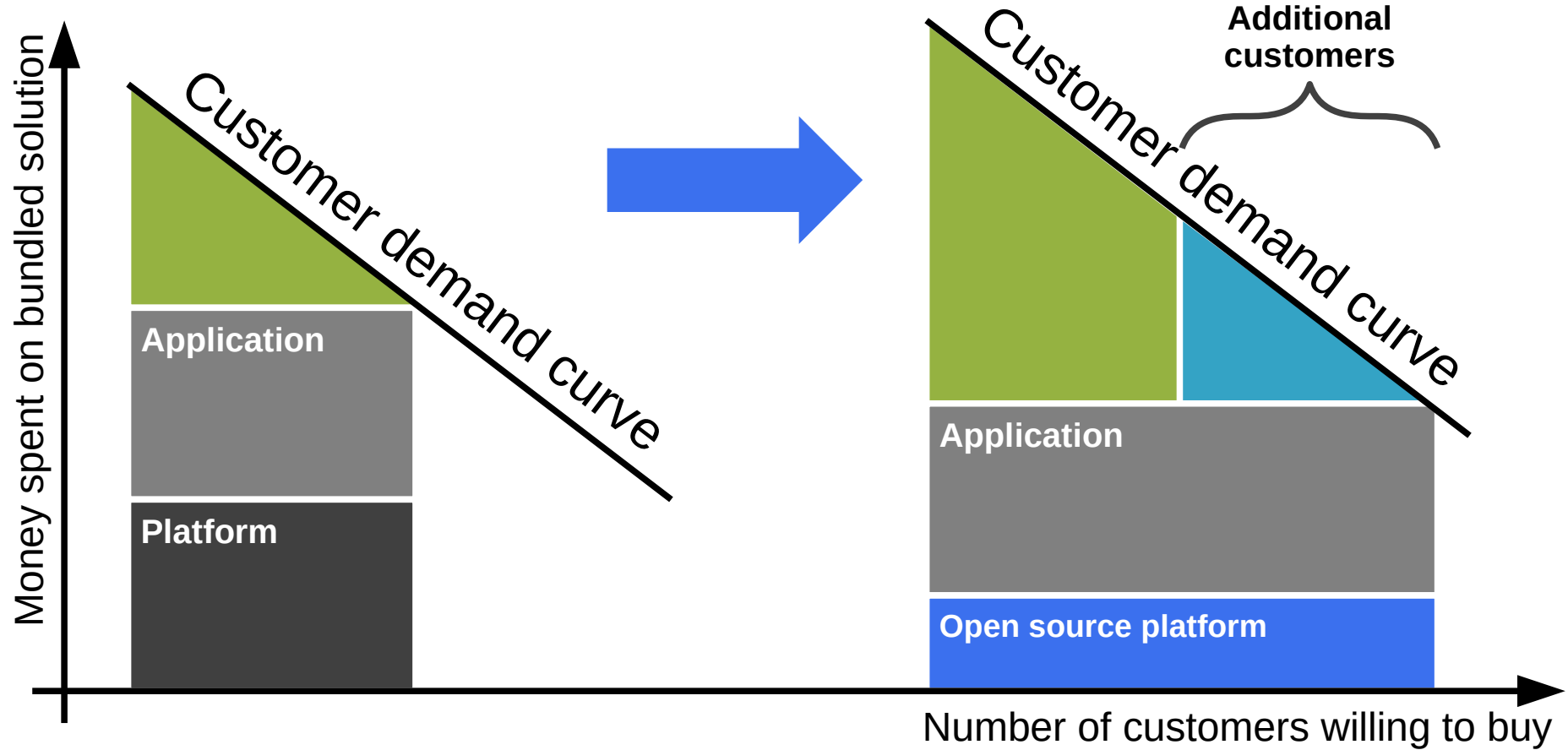   - Legally represent project interests

- Create a stronger more competitive ecosystem [R07] [R10a]
  - Increase share of customer wallet
  - Reach more price-sensitive customers
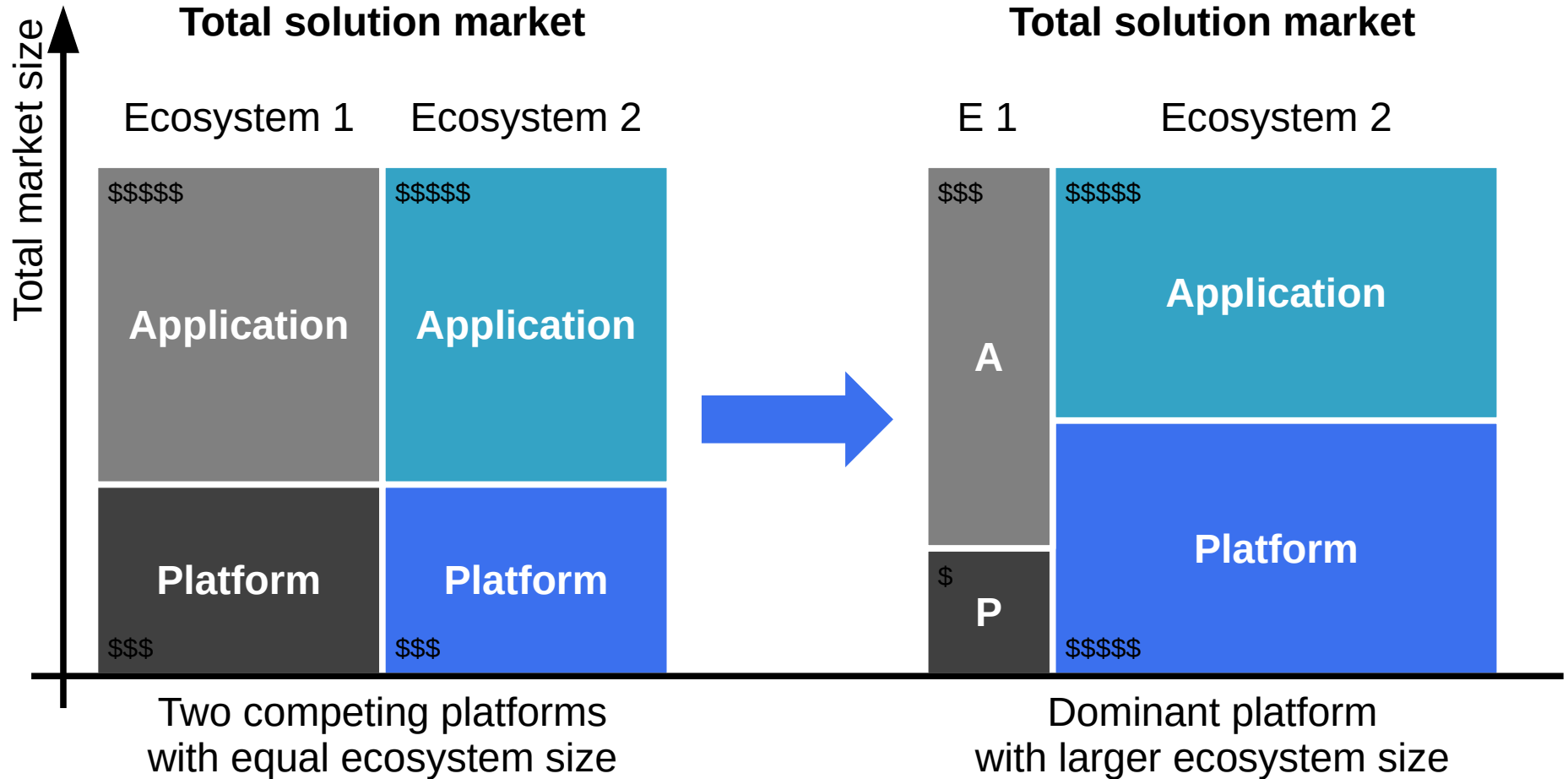 - Grow addressable market

Increase Share of Wallet

Money spent on solution

Without open source

$$$$ **(Cost of) rest of product components**

$$$$ **(Cost of) closed source component**

With open source

$$ **Potential added revenues**

$$$$ **(Cost of) rest of product components**

$$ **(Cost of) open source component**

Reach More Price-Sensitive Customers

- More customers
- More revenue per customer

Grow Addressable Market

Total solution market

Total market size

Ecosystem 1   Ecosystem 2

$$$$$ Application

$$$$$ Application

Platform $$$
Platform $$$

Two competing platforms
with equal ecosystem size

Total solution market

E 1   Ecosystem 2

$$$ $$$$$

A   Application

$ Platform
P $$$$$

Dominant platform
with larger ecosystem size

12

- Create a fair and equal playing field for members

  - Regulate access to intellectual property
  - Regulate access to marketing channels
  - Regulate product roadmap and development

- New open source projects can gain users and contributions fast. How can the project creator maintain control and harass users if they wanted to?
    - Using IP rights
        - Which? How?

    - Using position
        - Which? How?

Steering and Controlling Open Source [R11a]

Component Type		Business Goal		
		Reduce Development Costs	Maximize Customer Exposure	Minimize Competition
	Proprietary Open Source	Social leadership[2]	Domain ownership[2]	Copyright control[2] Trademark control[2] Development process[2]
	Community Open Source	Social leadership[1, 2]	Social leadership[2, 3] Domain ownership[3] Strategic positioning[2, 3]	Trademark control[3] Development process[2, 3] Strategic positioning[2, 3]

[1] Closed source firm   [2] Single-vendor open source firm   [3] Open source distributor

Problems and Solutions

- (Abuse of) copyright and patents → Solved by license
  - A good license contains patent provisions

- (Abuse of) trademarks → Solved by foundation
  - Foundation clarifies access to and usage rights of trademarks

- (Abuse of) social leadership → Can be solved by foundation
  - If necessary, the foundation can employ project leaders

- (Abuse of) process control → Solved by foundation
  - Through by-laws, the foundation can define a proper development process

- (Abuse of) other IP rights → Solved by foundation
  - Foundation can also own other critical IP rights (e.g. domains)

Advantages over Plain Community Open Source

- Clear intellectual property situation

  - Clarifies intellectual property situation (trademarks, patents, licenses)
  - Helps ensure good open source governance ("IP cleanliness")
  - Allows for proper legal representation in the courts

- Improved longevity and survivability

  - Makes software people-independent (somewhat, "bus factor")
  - Establishes good corporate governance, collaboration rules
  - Increases attractiveness of software ecosystem to developers

Advantages over Traditional Consortia

- Established framework
  - Increasingly well-understood legal and governance framework
  - Increasingly well-understood collaboration behavior

- Resulting benefits
  - Faster creation at lower cost, less friction, more trust
  - More legal and collaboration predictability
  - Easier to get skilled developers and firms

- Ultimately, higher likelihood of success

Application vs. Platform Developer Foundation

- An **application developer foundation** is

  - A **developer foundation** focused on developing a particular **application**
  - Examples: Wordpress foundation, Drupal foundation, Blender foundation

- A **platform developer foundation** is

  - A **developer foundation** focused on developing a particular **platform**
  - Examples: Linux foundation, Eclipse foundation, Openstack foundation

- The difference is one of quantity and quality (for now)
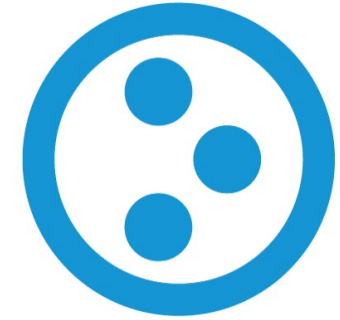
Notes on Application Developer Foundations

- Competition breeds strengthening ecosystems
    - Wordpress vs. Drupal vs. Joomla! vs. Plone
    - KDE vs. Gnome

- Some foundations are host to several projects
    - Mozilla foundation hosts Firefox, Thunderbird, …
    - FSF hosts a garden variety of projects

- Some foundations have a broader mission
    - Wikimedia hosts Mediawiki but mission is Wikipedia etc.
    - Then, perhaps, competitiveness of software suffers

The point of the foundation is to ensure free access, in perpetuity, to the software projects […] As part of this mission, the Foundation will be responsible for protecting the WordPress, WordCamp, and related trademarks. A 501(c)3 non-profit organization […]
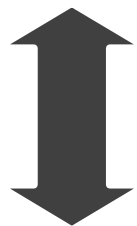
Reasons for Community Fork

- TWiki vs FOSWiki

    - Struggle over trademarks, foundation, and project leadership

- MySQL vs MariaDB

    - Struggle over project future and direction

- ownCloud vs. Nextcloud

    - Struggle over open source philosophy and community

Not all Playing Fields are Created Equal

- Automattic owns wordpress.com / wp.com

  - Creates strong marketing channel for Wordpress services

- Wikimedia Foundation does not have members

  - The Mediawiki ecosystem remains unorganized

- MariaDB is both a company and project name

  - Creates strong marketing channel from .org to .com

Challenges to Building a Partner Ecosystem

- Building a closed source partner ecosystem requires

  - Trust: Inspiring trust in partners; appearing trustworthy
  - Profit: Clearly defining how partners can earn money
  - Risk: Believably promise that behavior will not change

- Building partner ecosystems is difficult and slow

Creating Partner Ecosystems

- Benefits of open source foundation

  - Trust: Conduct business in the open
  - Profit: Clearly defined rules and regulations
  - Risk: Influence through open governance

- Ultimately, still a research hypothesis

Notes on Platform Developer Foundations

- There are less than application developer foundations

  - Many of the "foundations" are projects of larger foundations

- Still, many projects require their own governance

  - The Linux Foundation and Eclipse Foundation are affording this

1993-current   **2000-2007**   **2007-current**
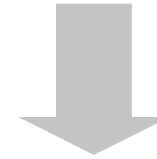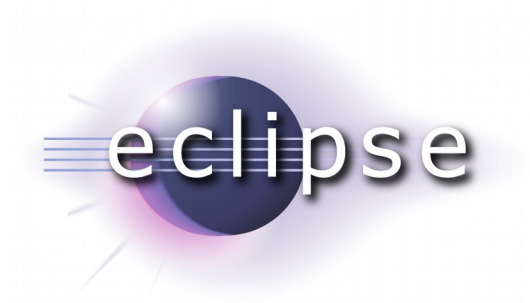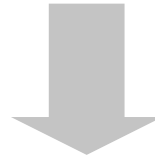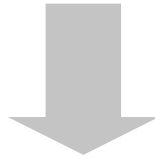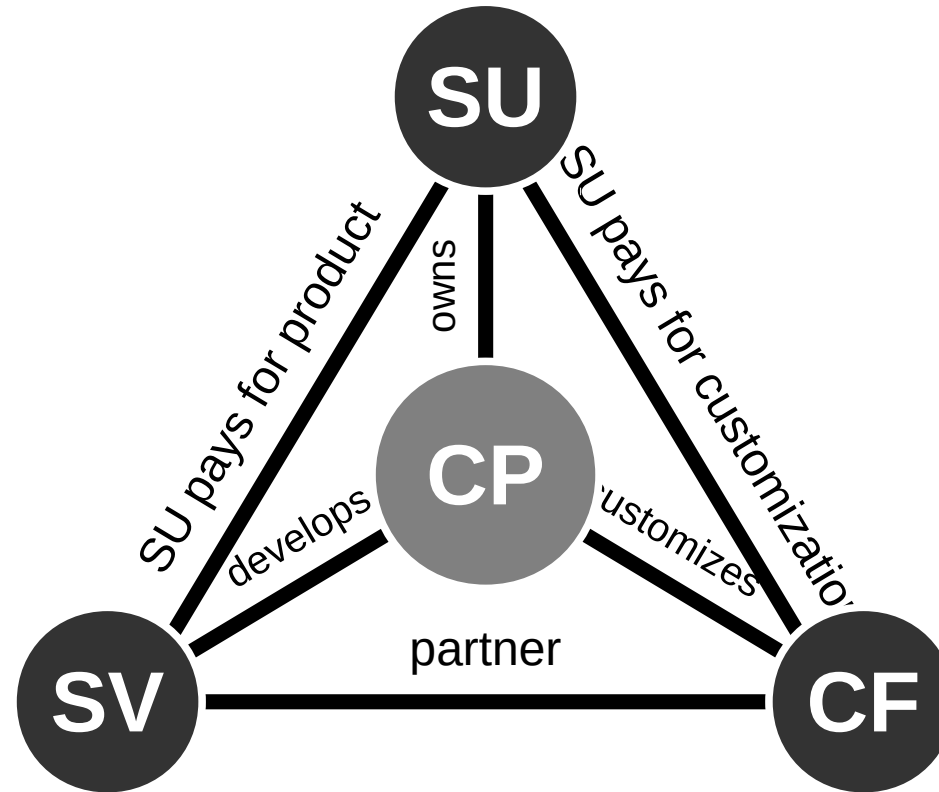
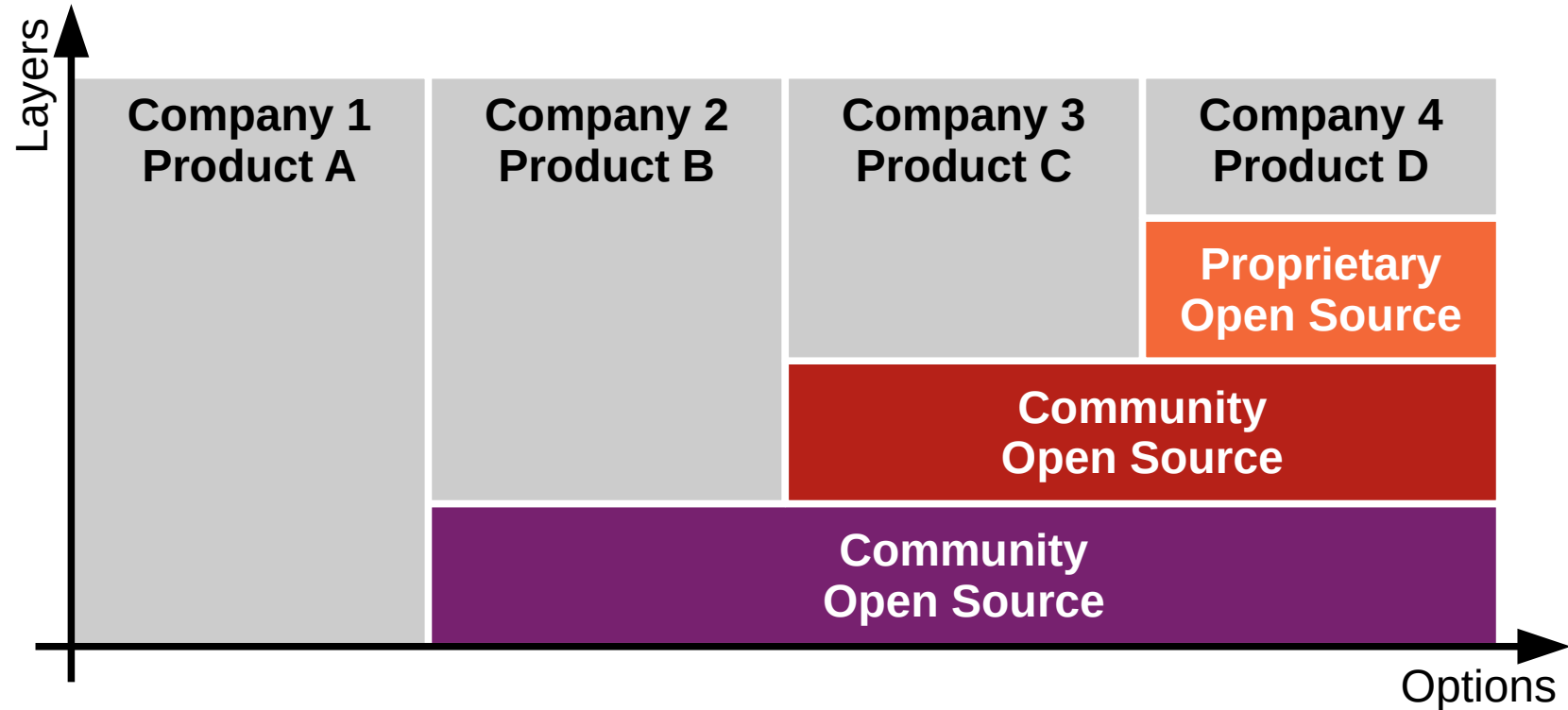# Community Software Development
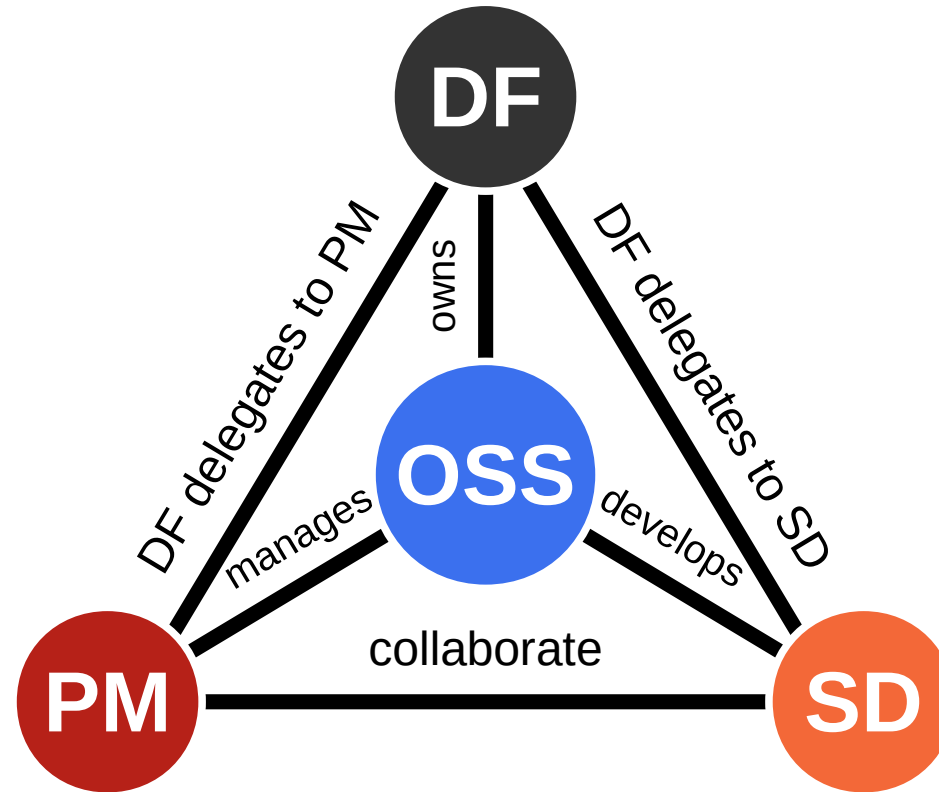
DF = Developer Foundation
PM = Project Management (Committee)
SD = Software Developer (various)
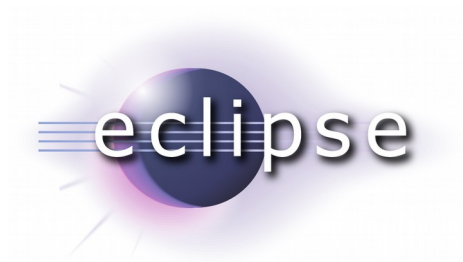
Commercial Product Provision

SU = Customer
SV = Software Vendor
CF = Consulting Firm

Challenge to Product Management

Community over code

Code over community

Apache Software vs. Eclipse Foundation

- Apache Software Foundation
  - Purpose
    - For public benefit

  - Membership
    - Only natural persons

  - Financing
    - Sponsorship

  - Project constraints
    - Any technology or model

  - Project maturation
    - Community over code

- Eclipse Foundation
  - Purpose
    - For member benefit

  - Membership
    - Mostly juristic persons

  - Financing
    - Member fees

  - Project constraints
    - Initially, Java only

  - Project maturation
    - Code over community

1. **Vision + Mission**

2. **Scope / Opportunity**

3. **Business Model**

Blueprint for Developer Foundations

- General
  - Purpose
  - Incorporation
  - Membership

- Philosophy
  - Open source stance
  - Development model

- Intellectual property
  - Project license
  - Patent grant
  - IP ownership

- Governance
  - Board membership
  - Project membership
  - Member careers
  - Member level

- Financing
  - Foundation
  - Projects

- Operations
  - Infrastructure
  - Backoffice

1. Vision

- Vision of organization

2. Beneficiaries

- For public benefit
- For member benefit

3. Incorporation

- No explicit incorporation
- Incorporation, for example, as
  - United States: 501(c)3 or 501(c)6
  - Germany: e.V. or gGmbH

4. Membership

- Natural persons
- Juristic persons, for example,
  - Software vendors
  - Consulting firms
  - User firms

1. Exploitation

   - Community property
     - Implies reciprocal license

   - Partial community property
     - Implies weak reciprocal license

   - Proprietary exploitation
     - Implies permissive license

2. Transparency

   - Open project model
   - Closed project model, e.g.
     - Completeness of artifacts
     - Completeness of history
     - Availability in time

3. Intellectual Property

1. Artifact license

   - Reciprocal license
   - Weak reciprocal license
   - Permissive license

2. Patent handling

   - Patent license
   - Patent retaliation

3. Source code

   - No copyright transfer
   - Contributor agreement
     – With copyright transfer
     – With relicensing rights grant

4. Other rights

   - Trademarks
   - Domains

5. Warranties

   - Required from contributors
   - Explicitly none to user

4. Organization Governance

1. Foundation membership

  - Natural member careers
    - Career stages, for example,
      - User, contributor, committer,
        PMC member, PMC leader, foundation member,
        board member (Apache)

    - Advancement process
      - How to advance?

  - Juristic member levels
    - By type of organization
    - By desired influence, for example,
      - Associate, solution, enterprise, strategic members
        (Eclipse)

2. Board

  - Membership
    - Who may be a member?
    - How to get appointed?
    - When to leave; tenure?

  - Decision process
    - How are decisions made?

1. Project membership

  - Project members
    - Always natural person, even if they represent a company

  - Member roles
    - Typically one of
      - User, contributor, committer
      - PMC member, PMC leader

  - Member role acquisition
    - Decision making process, for example, committer election

  - Member role binding
    - Typically tied to natural person, not employer

2. Software development

  - Project agenda
    - Defines purpose, scope of project

  - Project management
    - Defines development process

  - Decision process
    - Defines how decisions are made

  - Project tooling
    - Defines development tool setup

  - Intervention rights
    - Defines board rights to intervene

6. Financing

1. Financing of organization

   - Through members
     - Commensurate with level

   - Through public grants
   - Through sponsorships
   - Through for-profit subsidiary

2. Financing of projects

   - Through members
   - Through organization funds

7. Operations

1. Project infrastructure
   - Operated by
     - Foundation itself
     - Foundation members
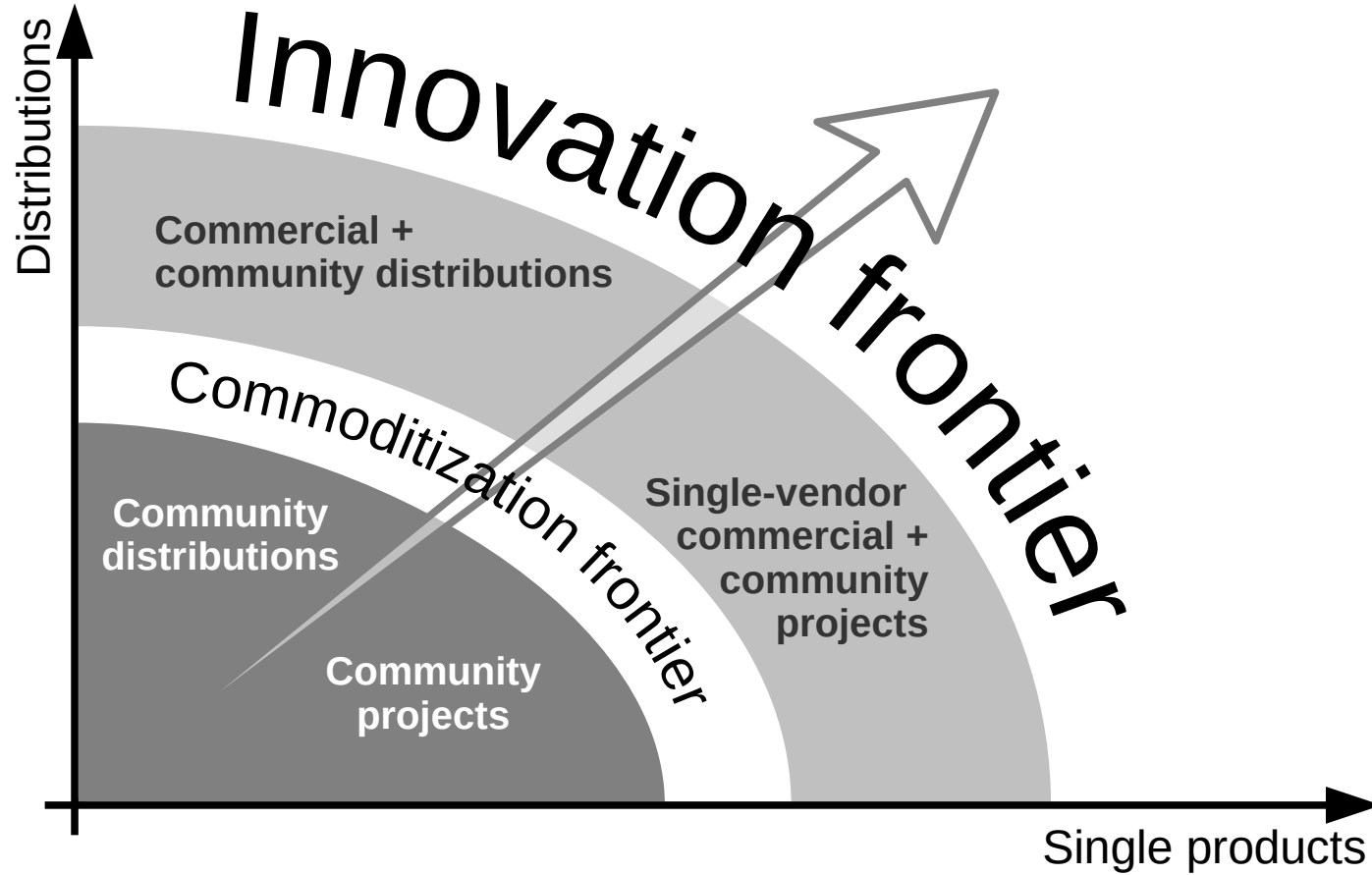   - On a by-project base

2. Foundation processes
   - Operated by
     - Employees
     - Volunteers

Dysfunctions of Developer Foundations

- Misuse of projects as marketing channels
- Abuse of bylaws for ensuring influence

Review / Summary of Session

- Open source developer foundations
    - Definition and examples
    - Motivation for developer foundations
    - Application vs. platform developer foundations

- Governance and inner workings
    - Challenges to developer foundations

Thank you! Questions?

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

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

DR

Credits and License

- Original version

  - © 2012-2019 Dirk Riehle, some rights reserved

  - Licensed under Creative Commons Attribution 4.0 International License

- Contributions

  - ...