Open Source User Foundations

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

BITKOM Forum Open Source

Berlin – 2016-07-05

Professorship of Open Source Software

- Dirk Riehle, professor of computer science
  - Focus is software engineering research incl. open source software
  - At Friedrich-Alexander-University Erlangen-Nürnberg, Faculty of Engineering

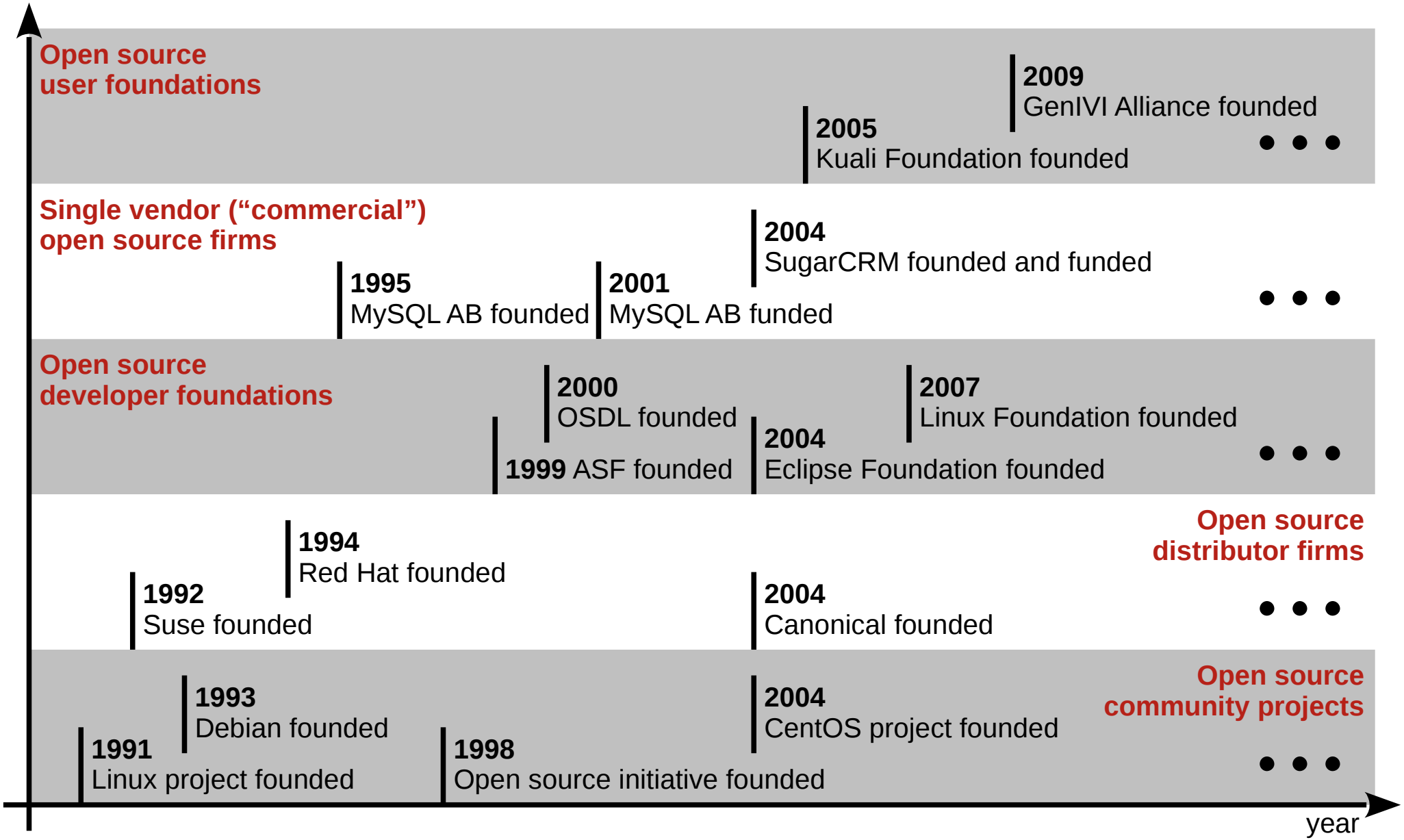
- Previously held research positions at ...
  - SAP Labs (Palo Alto, Silicon Valley) leading the open source research group
  - UBS (Swiss Bank, Zurich) leading the software engineering research group

- Previously worked in development at ...
  - Skyva Inc. (supply chain software startup, Boston) as software architect
  - Bayave GmbH (on-demand business software, Berlin) as CTO

Group Interests and Capabilities

- Open source software
  - **Open source governance**
  - Open source foundations
  - Open source community management
- Software engineering
  - **Inner source**
  - **Continuous delivery**
  - High quality requirements engineering
- Knowledge management

Evolution of Open Source Projects



Not a complete history: Events have been chosen for illustration purposes

Open Source “Business Models”

Non-Profit Open Source

1. Open Source Developer Foundations
2. Open Source User Foundations

For-Profit Open Source

3. Open Source Distributor Firms
4. Single-Vendor Open Source Firms

Open Source User Foundations

- An **open source user foundation** 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**

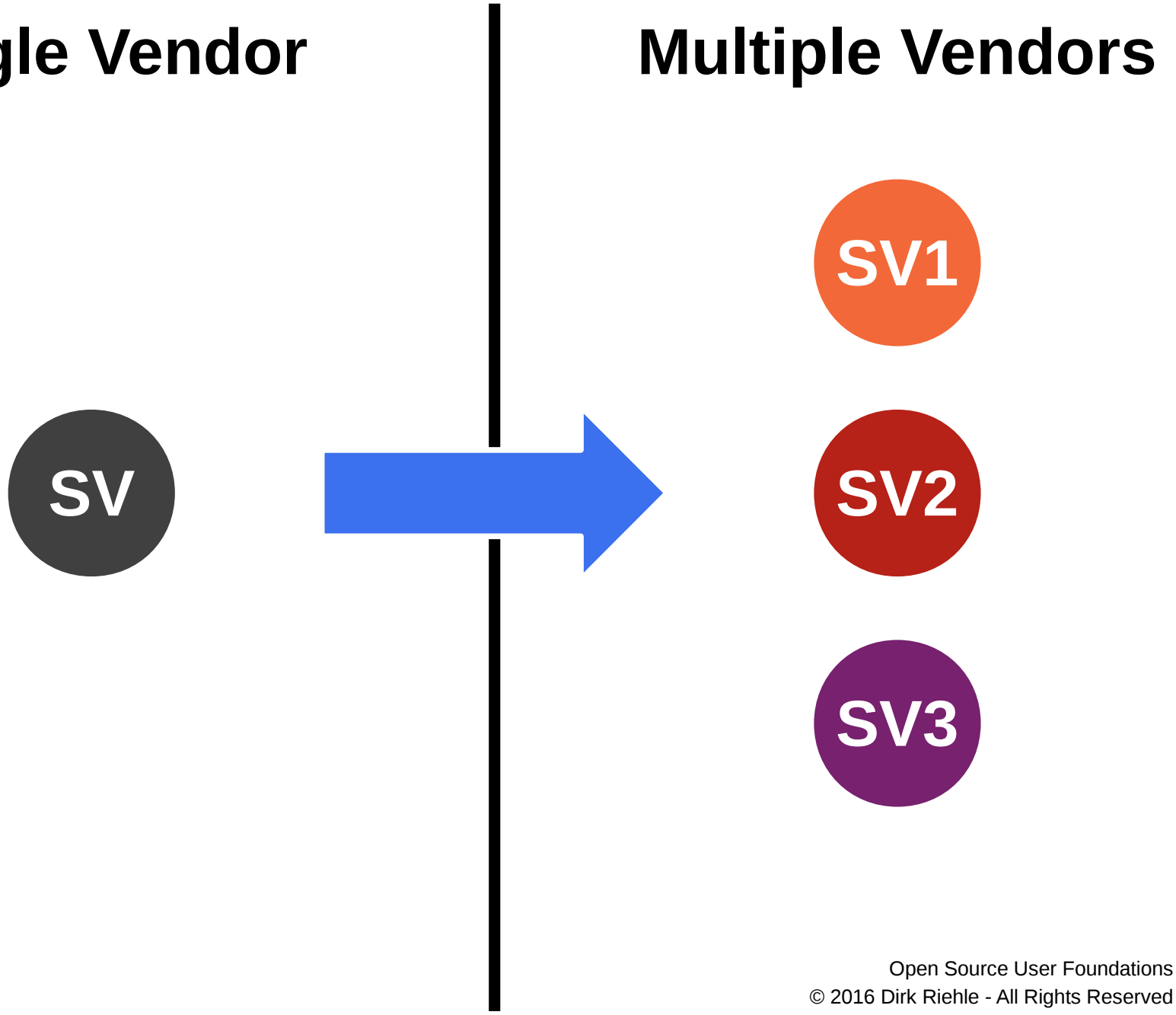
- Typical members of a user foundation are
  - Software user firms
  - Software vendors
  - Consulting firms
  - Service suppliers

Examples of User Foundations

Motivation for User Foundations

To establish a software ecosystem in which vendors and suppliers can provide products and services on an equal playing field.

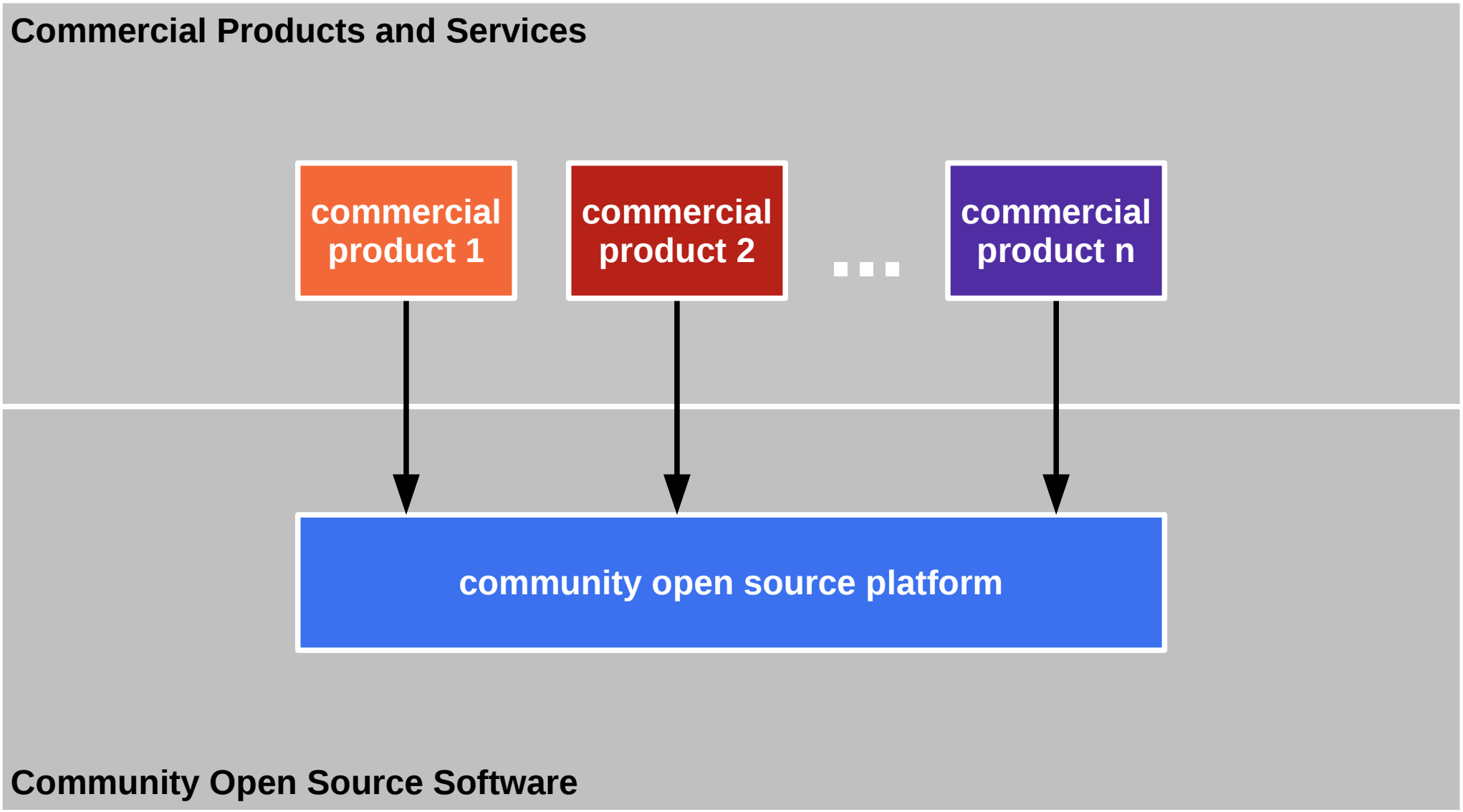
From a Single to Multiple Vendors



Problems with Single Vendor Lock-in

- High total-cost-of-ownership
  - High license fees
  - High customization costs
- No or slow realization of customizations
  - Missed or late product or service innovation
  - Missed or late market opportunities
  - No or late reaction to changing markets
  - Limited predictability of future capabilities
- Increased operational risk
  - What to do if vendor goes out of business?

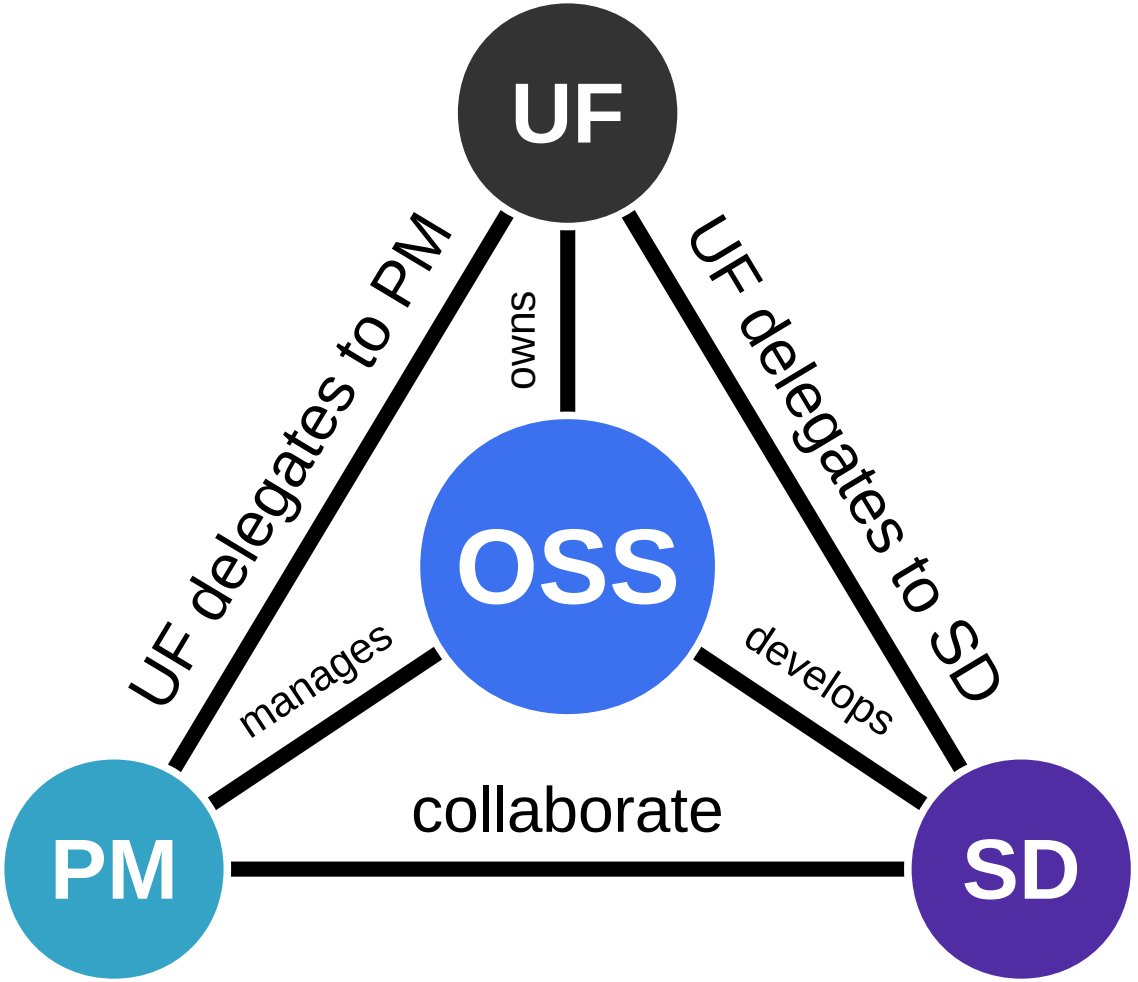
Software and Services Ecosystem



Equal Playing Field

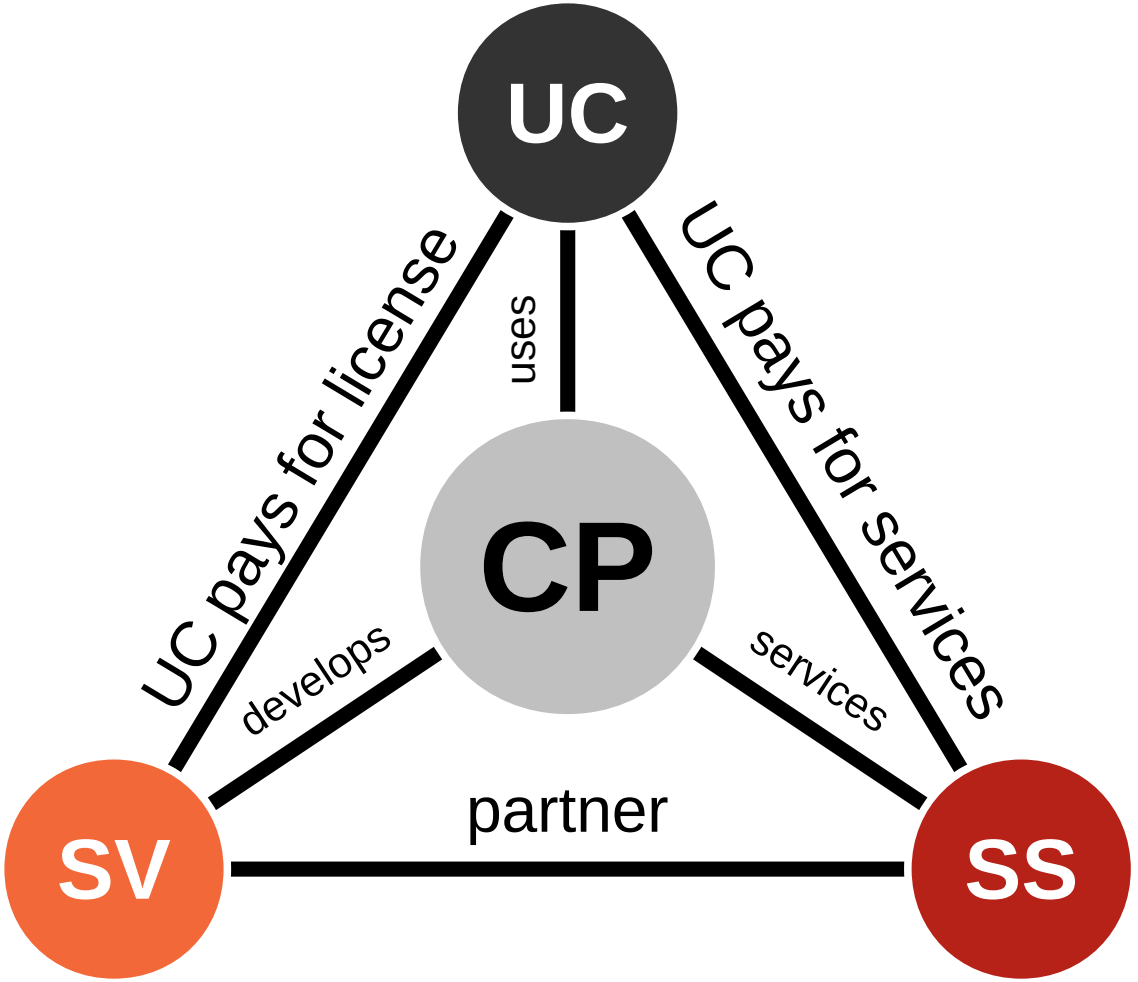
- The software ecosystem needs to be fair
  - Vendors and suppliers need to be able to earn a sufficient living
  - Users want the ability to switch suppliers, avoid lock-in

Community Open Source Software Platform



UF = User foundation
PM = Project management
SD = Software developers

Commercial Product and Services



UC = User company
SV = Software vendor
SS = Services supplier

What Motivates User Foundations

User foundations are typically created when the frustration over suppliers outweighs the (expected) hassles of the foundation.

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

Variants of User Foundations

- Classic stand-alone user foundations
  - Kuali, apereo, Prometheus, …
- As industry working groups
  - Polarsys, OpenMDM, OpenKonsequenz, …
- Strong vendor involvement
  - GenIVI, LocationTech, OpenAPC, …
- Natural-member user foundations
  - OpenStreetMap, OSGeo, OKFN, ...

Simplified Blueprint [RB12]

- Organizational set-up
- Purpose and philosophy
- Intellectual property
- Governance: Members
- Governance: Board
- Governance: Projects
- Governance: Development
- Finances and operations

Kuali Foundation

OpenKonsequenz [H+13b]

open KONSEQUENZ

Motivation for OpenKonsequenz

- Old closed source model not working
  - Strong supplier dependencies, high costs
  - No or little ability to influence direction, functionality
  - Changes and add-ons not possible or error-prone
- New software challenges (smart grid) ahead
  - Smart grid (Energiewende) and other challenges
  - Single monolithic system is not going to cut it
- Purpose and goals of OpenKonsequenz
  - Develop software faster better cheaper
  - Reduce or remove vendor lock-in

Time-Line of OpenKonsequenz

- 2010: First contact between Herr Herdt (N-ERGIE) and Prof. Riehle
- 2011: Initial gathering of local energy distributors, evangelism
- 2012: Feasibility study (result: Let’s do it!)
- 2013: First specification, financing
- 2014: Eclipse IWG founded, RfQ
- 2015: Pilot project starts, currently on-going
- 2016: More specifications, RfQs
- 2017: More implementations

OpenK 1 / 8: Organizational Set-up

- Eclipse Industry Working Group (IWG)
  - Organized through a U.S.-based 501(c)3 non-profit foundation
  - At cost of >= US$ 5000 per year per member
  - In the future, may change
- Steering committee +
  - Project planning committee
  - Architecture committee
  - Quality committee

OpenK 2 / 8: Purpose and Philosophy

- Purpose
  - To develop open source software for the energy sector
  - To motivate and instigate innovation

OpenK 3 / 8: Intellectual Property

- Open source license
  - Eclipse Public License

OpenK 4 / 8: Regular IWG Members

- Different types of membership
  - Driver members
  - User members
  - Service provider members
  - Guest members (incl. non-profits e.g. universities)

- Examples of members
  - Driver members: Energy distributors, e.g MDN, Netring, Westnetz
  - Service provider members: Vendors, e.g. IBM, BTC, SAG
  - Guest members: Non-profit institutions: OFFIS, Univ. Lübeck, FAU

OpenK 5 / 8: Steering Committee Members

- Founding driver members

OpenK 6 / 8: Project Membership

- Projects are open for everyone
  - Within the limits of the Eclipse governance model

OpenK 7 / 8: Software Development

- Project planning
  - Planning leads to module specifications
  - Financing secured from members
- Project initiation
  - Requests for quotations
  - Lowest adequate bidder wins
- Software development
  - Different roles interacting
  - Vendor, architecture, quality
- Final inspection and acceptance

OpenK 8 / 8: Financing and Operations

- Financing
  - Annual membership dues
- Operations
  - Handled by Eclipse Foundation

Summary of OpenKonsequenz

- Organization
  - An industry working group of the Eclipse Foundation
- Purpose
  - To develop open source software for the energy industry
- Motivation
  - Founding members were dissatisfied with closed-source firms
- Development
  - Sponsors development of software through consulting firms

Challenges for User Foundations

- Market size is too small to be sustainable

Dysfunctions of User Foundations

- Over-reliance on one provider creates lock-in

Student Projects with Industry Partners

- **Recruiting**
- **Outsourcing**
- **Innovation**
- **Startups**

- **AMOS** (software tools and components)
- **PROD** (market research, product specs)
- **ARCH** (software architecture analysis)
- **NYT** (interview and data analysis, other)

Thank you! Questions?

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

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