Open Source Software
An inspiration for software ecosystems

Valentin Rothberg

System Software Group Friedrich-Alexander University Erlangen-Nuremberg

January 17th, 2013

Problem Statement

- In the last decade many software ecosystems emerged in the context of free and open source software development
  - Linux running on 60% of all webservers and 90% of super-computers
  - Android with a worldwide marketshare of 75% within 4 years

Problem Statement

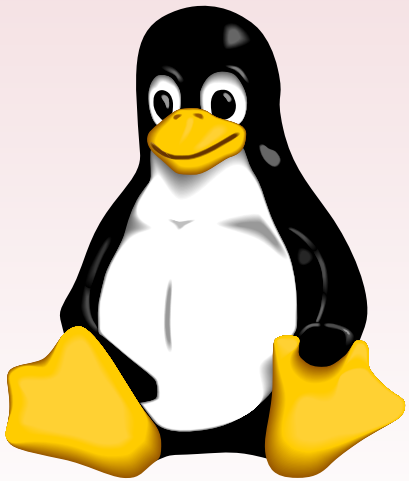
- In the last decade many software ecosystems emerged in the context of free and open source software development
  - Linux running on 60% of all webservers and 90% of super-computers
  - Android with a worldwide marketshare of 75% within 4 years
- Software ecosystems play an ever important role
  - Reduction of costs
  - Common share of resources
  - Reaching more customers

Problem Statement

- In the last decade many software ecosystems emerged in the context of free and open source software development
  - Linux running on 60% of all webservers and 90% of super-computers
  - Android with a worldwide marketshare of 75% within 4 years
- Software ecosystems play an ever important role
  - Reduction of costs
  - Common share of resources
  - Reaching more customers
- Development processes and structures of free and open source software projects can be adopted

Problem Statement

- In the last decade many software ecosystems emerged in the context of free and open source software development
  - Linux running on 60% of all webservers and 90% of super-computers



How can free and open source software development projects help on how to evolve a successful and vivid software ecosystem?

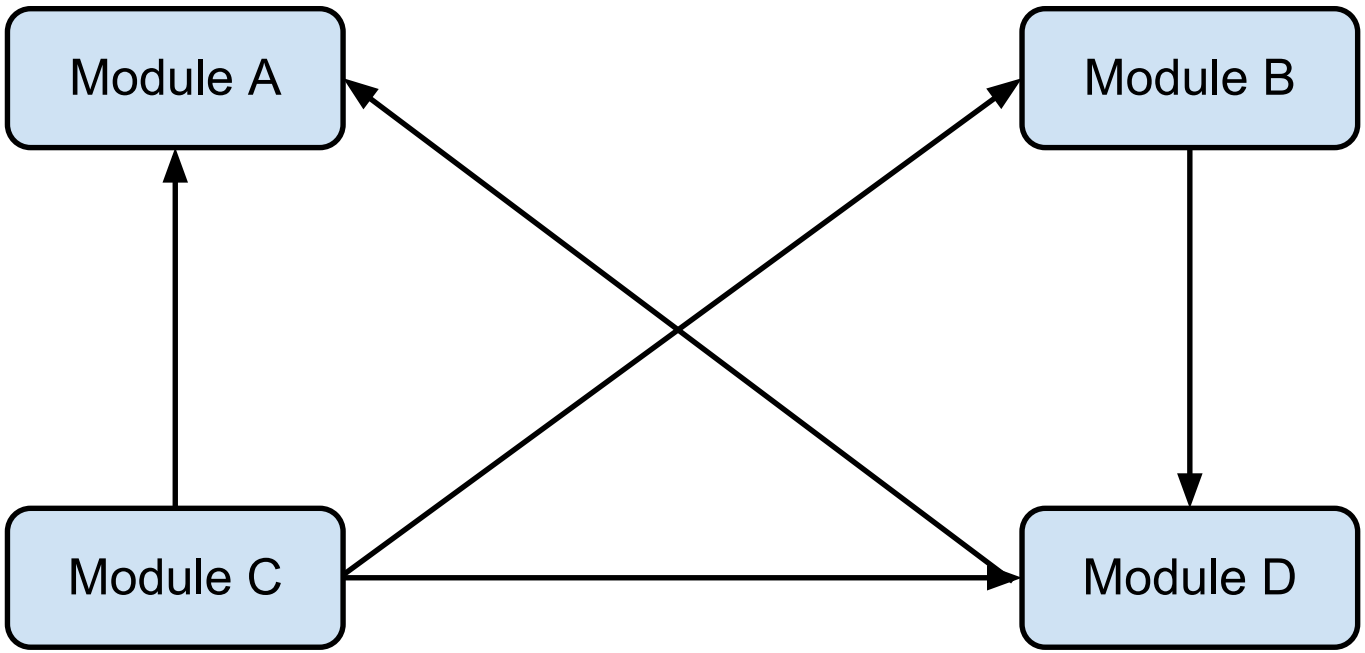
Agenda

- Software Ecosystems
- Architecting Ecosystems
- Interfaces - Central Coordination Mechanisms
- Free and Open Source Software Development
- FOSSD as an Inspiration for SECOs
- Conclusion

Software Ecosystems

- Term emerged during past decade
- Heavily inspired by systems such as iOS or Android
- Defined *"as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them. These relationships are frequently underpinned by a common technological platform or market and operate through the exchange of information, resources and artifacts."*

Architecting Software Ecosystems



Characteristics in Architecting SECOs

- Characteristics stem primarily from two basic principles:
  - Transparency (a pillar in free and open source development)
    - Availability of any type of information
    - Key coordination mechanism
    - Stakeholders can be aware of the evolution of development
  - Modularity (essential in larger software systems)
    - Traditional software engineering principle
    - Decomposing into manageable parts (modules)
    - Minimize technical coupling and dependencies

Challenges in Architecting SECOs

- Both principles also entail several challenges:
  - Transparency
    - Information overload
    - Protection of intellectual property
    - Trade secrets, security and liability concerns
  - Modularity
    - Minimal coordination
    - Integration problems

Interfaces - Key Coordination Mechanisms

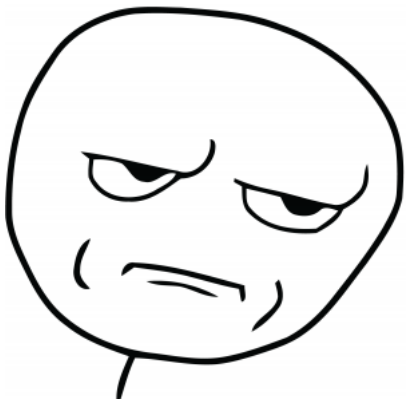
- Impact on how work is devided, coordinated and integrated
- Key coordination mechanism among different development groups
- Once an interface is defined, development can continue independently and in parallel
- Can lead to minimal interaction
  - Transparency
  - Researches address this issue with INTERFACE TRANSLUCENCE

Interface Translucence

- *"... bridges technical and socio-organizational roles of interfaces."*
- Discussed in several papers with one common approach
  - Increase visibility of particular information while hiding others
- Benefits of transparency and modularity under two conditions
  - A priori knowledge of all requirements
  - Modules interact through stable and well known interfaces
  - Translucence addresses both (certain valuation and corresponding URLs)

Interface Translucence

- *"... bridges technical and socio-organizational roles of interfaces."*
- Discussed in several papers with one common approach



Modules interact through stable and well known interfaces ...

Interface Translucence - Valuation

- *Interface uncertainty*
  - Interface a point t_n changes at later point t_{n+1}
  - Leads to heavy integration and quality problems
  - A valutation allows to provide necessary mechanisms
- *Interface complexity*
  - Highly complex interfaces have negative impact
  - *Information-hiding-principle* by Dave Parnas
  - A valuation allows better design decisions and anticipations
- *Activity awareness*
  - Only few (specialised) frameworks allow sound integration of modules
  - Corresponding coordination mechanisms needed

Free Open Source Software Development

- Software that is published under licenses that allow
  - Public access to the source code
  - Free copying and redistribution
  - Open research on the software
- Open Source Software is published under a license from the Open Source Initiative (OSI)
  - More business friendly
- Free Software uses the GNU Public License (GPL)
  - Very strict copylefts (versus copyrights)

Characteristics of FOSSD Projects

- Sustainable and reliable high-quality software systems
- Used by thousands to millions of users
- Typically small group of core developers
- Require skills in one or more application domains
- Many contributors sacrifice leisure time
  - Challenging tasks
  - Improve knowledge and skills
  - Strong communities with shared beliefs
  - "Geek fame"
- Managed by globally distributed *"virtual organisations"*

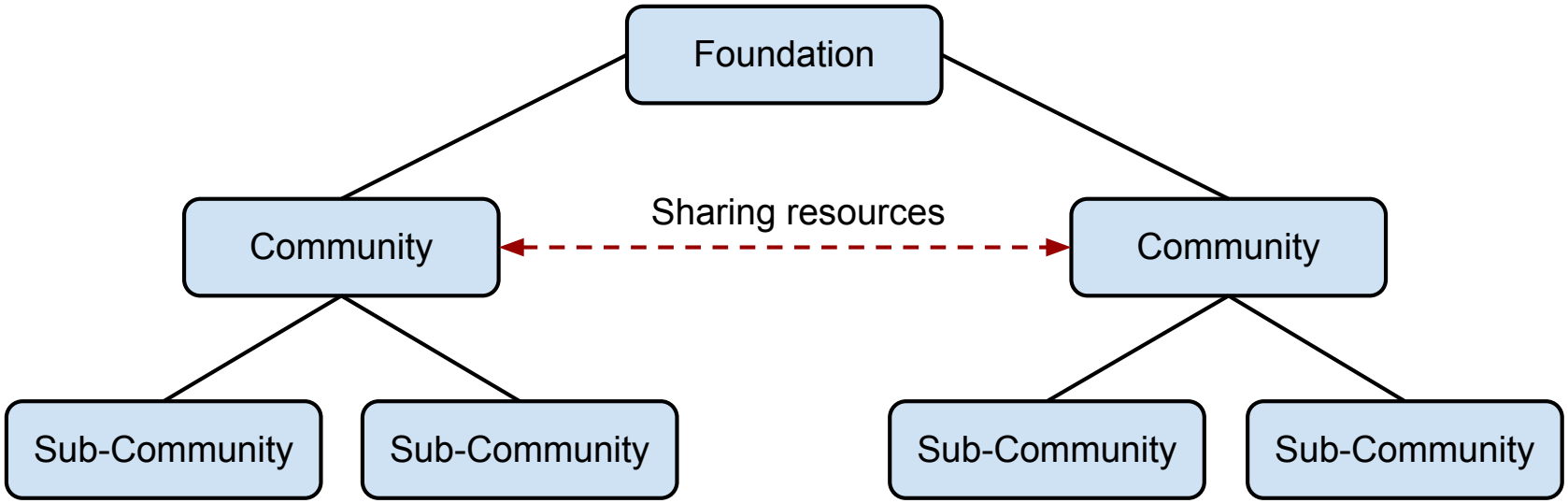
FOSSD Software Informalisms

- Projects need to comprise information repositories
- Software informalisms
  - Used to describe, pro- and prescribe processes within projects
  - Informal narrative resources
  - Can replace formalisms, e.g. requirement specifications
- FOSSD informalisms:
  - Email lists / instant messaging
  - Forums, boards, blogs, or news postings and wikis
  - How-To-Guides, tutorials, and FAQs
  - Traditional system documentations
  - Publications
  - Web sites or portals
  - ...

FOSSD Alliance Formation

- Interdependencies through
  - Networking of developers
  - Development artifacts
  - Common tools and shared infrastructures
- Interdepdencies let projects share
  - Source code (modules)
  - Components
  - (Sub-) systems
  - Developers
- FOSSD systems co-evolve
  - *"A project will not produce and sustain a viable system unless the team reaches a larger critical mass [...]."*
  - Critical mass known to be at $\sim 5 - 15$ core developers

FOSSD Alliance Formation



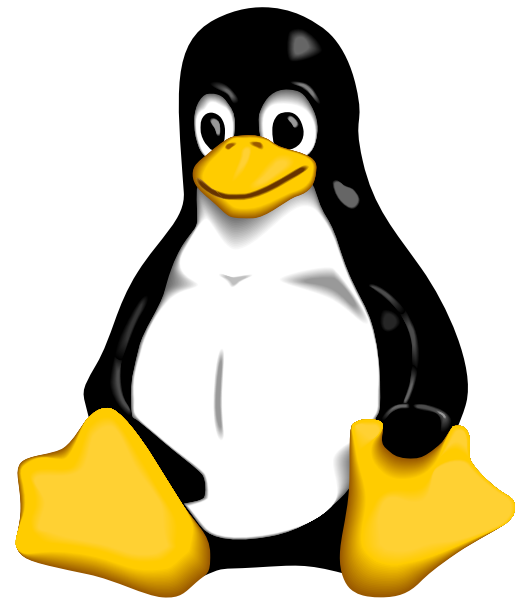
FOSSD Alliance Formation

FOSSD as an Inspiration for SECOs

- Investigation of current software ecosystems
  - Various reasons for the occurance / development of SECOs
  - Two major aspects can be outlined
    - Intentions and market strategies
    - Used license(s)
- I introduce three different models
  - Free Software Model
  - Open Source Model
  - Proprietary Software Model

Free Software Model - Linux

- Most famous, widely known, and most studied FOSS project
- Gave birth to many other ecosystems
  - GNOME, KDE, Android, ...
- Running over 60% of all web servers
- Linux foundation
- Reasons for success
  - Linux is and will always be FOSS software
  - Anybody can use, modify, and redistribute the code
    - In case of distribution, modifications have to be published
    - Great acceptance in industrial field, e.g. server farms
  - Many (big industrial) contributors - win-win-situation



Open Source Model - Android

- Most successful OSS ecosystem
  - Within 4 years to 75% worldwide marketshare
  - 1.3 million activations per day (Sept. '12)
- Main intention was to manifest and expand ongoing business
  - Search engine
  - Advertisement (main source of income)
  - Services, for instance Google Maps or Google Mail
- All in all: *bind present and reach future customers*

Open Source Model - Android

- Acquisition of Android Inc. in 2003
  - Enough qualified core developers
- Open Handset Alliance (OHA)
  - Led and founded by Google in 2007
  - HTC, Sony, Dell, Intel, Texas Instruments, Samsung Electronics
- Certification process
  - Usage of *"Google Android"* implies certification
  - Compatibility Definition Document (CDD)
- Usage of GPL (changes at Linux kernel are to make public)
- Most of Google's code uses Apache License

Open Source Model - Android

- This licensing provides several benefits
  - The internal ecosystem (OHA) is ablte to work as an OSS project
  - Stakeholders benefit through contributions of others
  - Reduction of costs
  - Good image campaign for the system (marketing)
  - Allows closed source development, too
  - Arising of other ecosystems build ontop of Android
    - Amazon's *Fire* devices
    - Android distributions / forks, e.g. *Cyanogenmod*
    - Allows acquisition of qualified developers
- Open source does not imply an open development

Open Source Model - Android

- Several benefits for Google and other stakeholders
  - Contributions let system grow in quality and size
  - Win-win-situation
    - Stakeholders have access to a high-quality system
    - Reduction of costs
    - No need to buy external expertises (OHA)
    - Protect intellectual property
  - Many hardware devices and stakholders
  - Big customer base with a marketshare of 75%!
  - Linux benefits through patches as well

Proprietary Software Model

- No sound research possible
- In the context of software ecosystems there are two possibilities:
  - Software vendors (e.g. Microsoft Windows)
  - Soft- and harware vendors (e.g. Apple OSX / iOS)
- Internal ecosystems can still learn from FOSS development and corresponding research
  - Big companies like Windows, Apple, Dropbox have so called project-weeks
    - Developers can chose challenging tasks and work together in groups
    - Encourages developers to continue working
    - Many developed features are often merged mainline
    - "Spirit" of FOSS development

Conclusion

- FOSSD projects show strong inter-dependencies
- High-quality software
- Linux and Android show impressive stories of success
- Companies are able to evolve vivid ecosystems
- Main advantages
  - Collaboration
  - Own interests while benefiting from others
  - Reduction of costs
  - External knowledge while keeping intellectual property
  - Benefits of ecosystem-development in general

References

- W. Scacchi, "Free/Open Source Software Development: Recent Research Results and Emerging Opportunities," in The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering companion papers - ESEC-FSE companion '07, 2007, p. 459.

- M. Cataldo and J. D. Herbsleb, "Architecting in Software Ecosystems: Interface Translucence as an Enabler for Scalable Collaboration," in Proceedings of the Fourth European Conference on Software Architecture Companion Volume - ECSA '10, 2010, p. 65.

- Others to be found on my paper.

References

- W. Scacchi, "Free/Open Source Software Development: Recent Research Results and Emerging Opportunities," in The 6th Joint

QUESTIONS?