Commercial Open Source Startups and the Cloud

Prof. Dr. Dirk Riehle

University of Erlangen / Bayave GmbH

O4B, Oct 2020

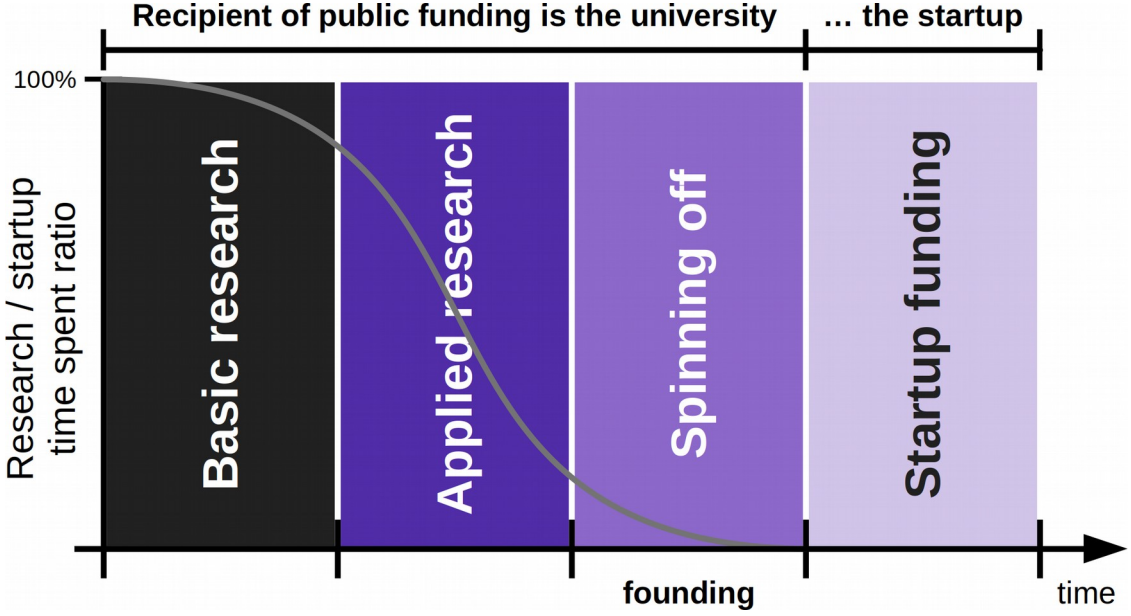
Professorship of Open Source Software

- Professor of Computer Science
  - For software engineering and open source software
  - At the computer science department of the engineering faculty
- Previously held research positions at ...
  - SAP Labs (Silicon Valley) leading the open source research group
  - UBS (Swiss Bank, Zurich) leading the software engineering group
- Previously worked in development at ...
  - Skyva Inc. (supply chain software, Boston) as software architect
  - Bayave GmbH (on-demand business software, Berlin) as CTO
- Researches and teaches commercial open source at
  - Friedrich-Alexander-University Erlangen-Nürnberg
  - University of California, Santa Cruz
- Ph.D. from ETH Zurich, M.B.A. from Stanford GSB

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Commercial Open Source Startups as University Spin-Offs

- We specialize in turning research into commercial open source startups
- Public funding for startups reaches well into a Series A!



- We can also serve as a conduit to public funding as long as you have not yet incorporated

Commercial Open Source by Intellectual Property

- Service and support firms
  - Simply service existing open source software
  - Don’t own any of the IP
  - Don’t attract venture capital
- Open source distributor firms
  - Provide a well working assembly of open source components
  - Own non-core-software IP (configuration data, regression test suites, …)
  - Can attract venture capital; can have outsize returns
- **Single-vendor open source firms**
  - Provide a traditional software product to enterprises
  - Exclusively own (key parts of) the software their business is based on
  - Can attract venture capital; can have outsize returns

Three Generations of Single-Vendor Open Source Firms

- **The pioneers (199x-2002)**

...

- **The second wave (2002-2008)**

...

- **The current breed (since 2008)**

...

Free vs. To-Pay-For

- **Community edition**
  - **Core product**
    - **Core software**
      - **Provided under an open source license**
    - Some complementary artifacts
    - Self-help services

- **Commercial edition**
  - **Core product**
    - Core software
      - **Provided under a commercial license**
    - Additional functionality
    - Complementary artifacts
    - Self-help services
  - **Basic product =**
    - Core product +
    - Fitness for use / certification
    - Indemnification
    - Support services
  - **Whole product =**
    - Basic product +
    - Training
    - Consulting
    - **Operations**

Why the Open Source Strategy?

- **To drive adoption** (of the product in its markets) due to (nearly) **frictionless distribution**
  - To build a **large (not necessarily paying) user base** from which benefits accrue
  - To have an existing base of users to convert to customers
  - To hinder competitors from getting in
- What is not new
  - Revenue sources
- What **is** new
  - Everything else (changes)

The Product Management Challenge

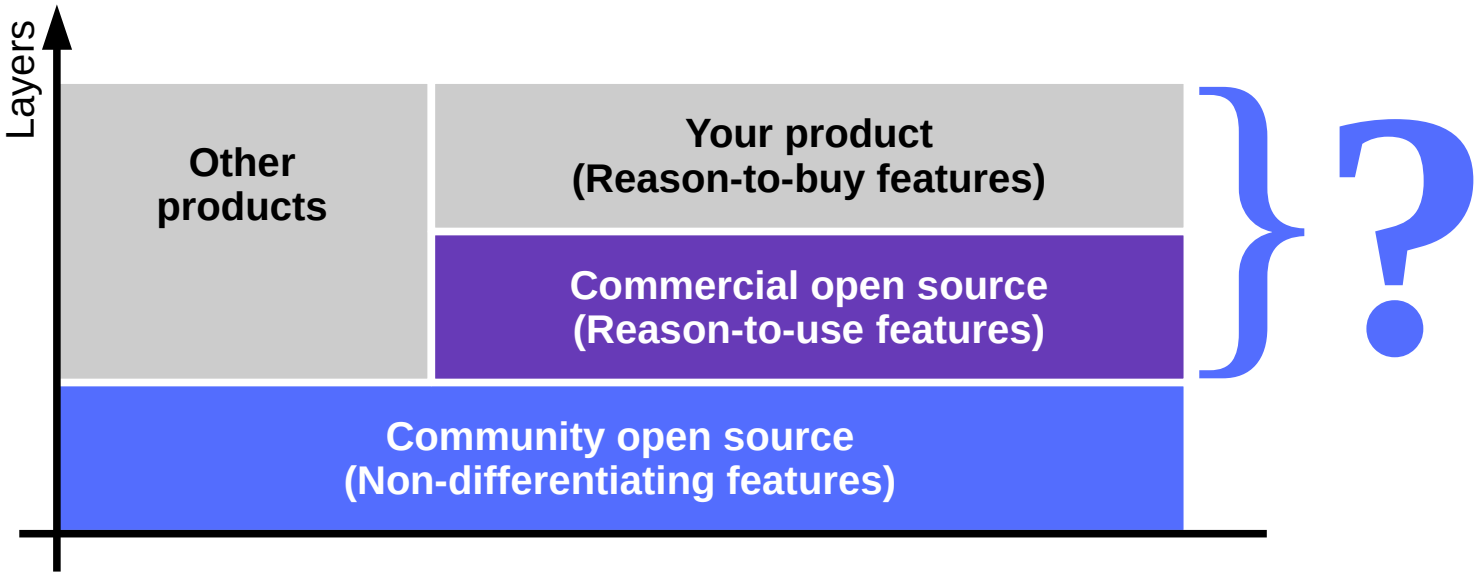
Structure product and services so that you

1. **Maximize conversion to paying customer**
2. **While benefiting from user community**
3. **And keeping the competition at bay**

A Commercial Open Source Feature Classification

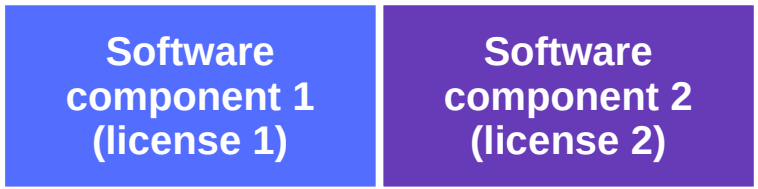
- Non-differentiating
  - The feature is competitively not differentiating and readily available elsewhere
- Reason-to-use (value creation)
  - Users come to your software, because the feature is not ubiquitous
- Reason-to-buy (value appropriation)
  - Users upgrade to paying customers to receive this feature

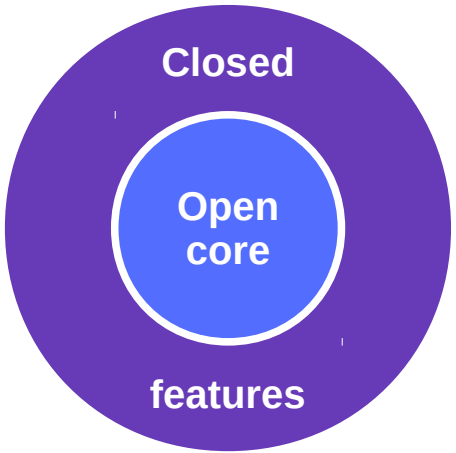
Open / Closed Software Feature Differentiation



Feature Differentiation and IP Modularity

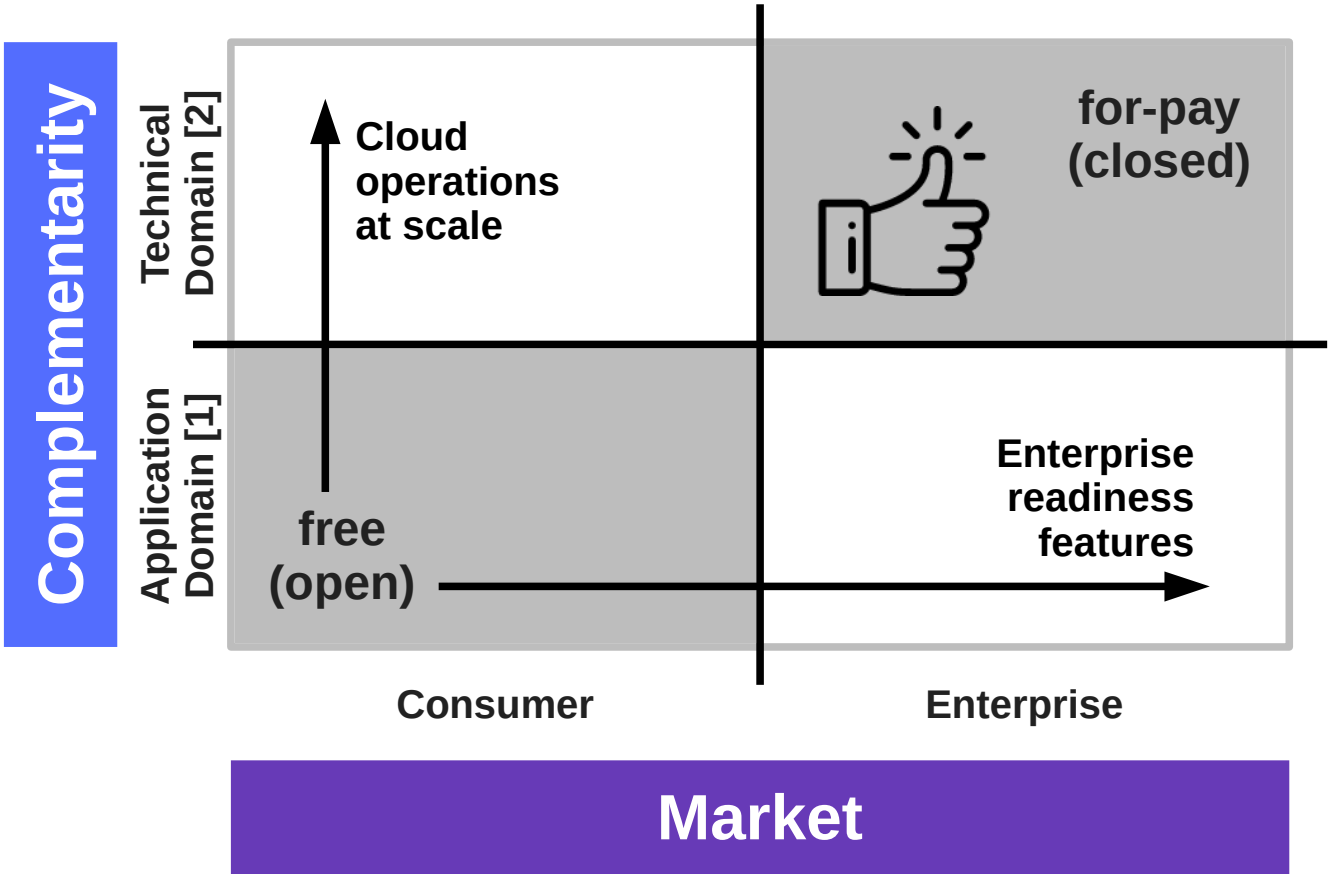
- Intellectual property (IP) modularity
  - The practice of splitting IP into modules of different licenses





- Open core model
  - A particular form of IP modularity where there is
  - An “open” software core available under an open source license and
  - Software extensions of the core are available only under a commercial license

How to Think About Feature Differentiation



[1] Application domain = **business purpose** of software = functional requirements
[2] Technical domain = support infrastructure = non-functional requirements like costs of operations

The Product Management Challenge (Again)

Structure product and services so that you

1. **Maximize conversion to paying customer**
2. **While benefiting from user community**
3. **And keeping the competition at bay**

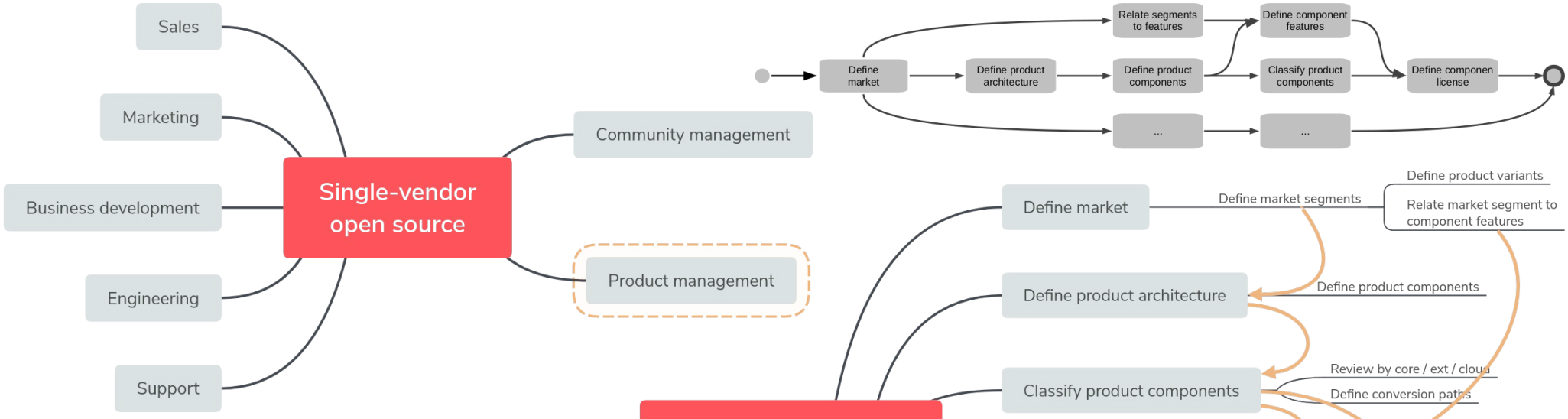
The Move Into the Cloud

- A tectonic shift
  - (Almost) everything is moving into the cloud
- Open source
  - Becomes an on-ramp to the cloud
- Conversion is
  - From self-hosted to vendor-hosted
- The hyperscalers
  - Are the new competition

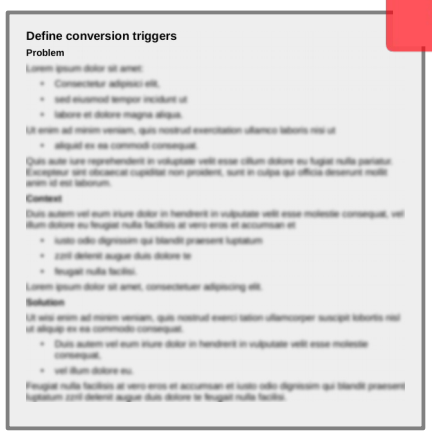
The 2018 MongoDB License Change

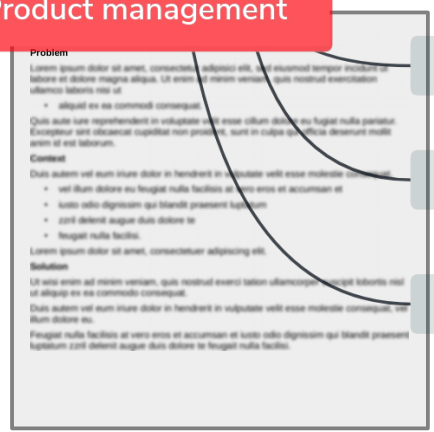
Component	From-License	To-License
Community server	**AGPLv3** (and commercial)	**SSPL** (and commercial)
Connectors and drivers	**Apache 2.0** (and commercial)	**Apache 2.0** (and commercial)
Cloud management	**Commercial** (only)	**Commercial** (only)

The Commercial Open Source Playbook




Define product architecture
Problem


Define conversion triggers
Problem



How to Raise Funds Without Losing Equity

#	Phase	Program	# Persons	Amount [PM p. P.]
1	Basic research	DFG, ERC	1-3 (up to 6)	36
2	Applied research	BMWi (various), EU H2020	1-4	18-36
3	Spinning off	EXIST Forschungstransfer	3-4	18
4	Starting up	EXIST II	3-4	6
5	...	KMU Innovativ	...	12-24

Know the Difference.

EDITIVE Scout

Includes functions for text comparison using HD-Diff.

Know more

Make the Difference.

EDITIVE Collaborate.

EDITIVE Scout functions expanded to include collaboration.

Make more

Thank you! Questions?

dirk.riehle@fau.de – https://oss.cs.fau.de

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