

# 开源软件概述

周明辉

zhmh@pku.edu.cn



北京大学  
PEKING UNIVERSITY

# 纲要

- 开源软件的定义
- 开源软件的历史和现状
- 开源社区的结构和治理
- 面向开源的研究

什么是开源软件？

# 开源软件

- 开源软件是一种**源代码可以自由获取**的计算机软件。
- 发布开源软件需要附带**开源许可证**：
  - 开源许可证是对开源软件的知识产权进行规范和约束的法律合同：甲方是软件版权所有者，乙方是软件用户。
  - 软件的版权持有人在开源许可证的规定之下允许用户使用、修改以及分发该软件。许可证定义了开源软件用户的权利和义务。
  - 软件知识产权：版权，专利，商标。
- 开源许可证通常符合**开源的定义**的要求。

# 开源的定义

- 开源不仅意味着可以访问源代码，开源软件的分发条款必须满足下述条件：

- (1) 自由再发行；
- (2) 程序必须包含或方便取得源代码；
- (3) 许可证必须允许更改和派生程序；
- (4) 保护作者源代码的完整性；
- (5) 无个人或团体的歧视；
- (6) 无领域歧视；

.....

- (8) 许可证不能限制其他软件；
- (10) 许可证需要是技术中立的。



<https://opensource.org/osd>

# The Open Source Definition

-- <https://opensource.org/osd>

## Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

### 1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

### 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost, preferably downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

### 3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

### 4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

### 5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

### 6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

### 7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

### 8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

### 9. License Must Not Restrict Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

### 10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

Last modified, 2007-03-22

# 开源许可证需要符合开源的定义的要求

- 开源社区存在大量不同类型的许可证，OSI 认证的开源许可证已有 126 个<sup>[1]</sup>，在开源项目中的使用率 > 80%。
  - 宽松型，Apache/MulanPSL，关键特点：分发可以闭源
  - 传染型，GPL/MulanPubL，关键特点：分发必须开源

木兰宽松许可证(宽松型许可证): <http://license.coscl.org.cn/MulanPSL2>

木兰公共许可证(传染型许可证): <http://license.coscl.org.cn/MulanPubL-2.0>

均采用中英文双语表述，以中文为准

【1】 [https://spdx.org/licenses/?ivk\\_sa=1024320u](https://spdx.org/licenses/?ivk_sa=1024320u), 2021/03/07

# 开源软件的历史和现状



# 开源软件的历史：创新的历史

大公司参与和主导开源社区，**混合模式**兴盛；开放众包、群智**创新**，开源延展到硬件、教育等领域

**开源 (Open Source)** 一词在1998年2月3日提出，在1998年开源峰会达成共识，开源**正式成为旗帜**，开源运动迅猛发展，影响力迅速扩大

Bill Gates推动**软件商业化**，软件单独售卖，源码不再免费可得；R.M. Stallman发起**自由软件free software**运动；各种开源社区兴起

软件商业化尚未出现或尚未成熟，软件跟硬件搭售，随源代码发布，学术共同体**自发开放和共享源代码**

多元生态发展阶段  
(商业模式多元化)

原始萌芽阶段



1980

多家争鸣阶段



1998

共识达成阶段



2005

融合发展阶段

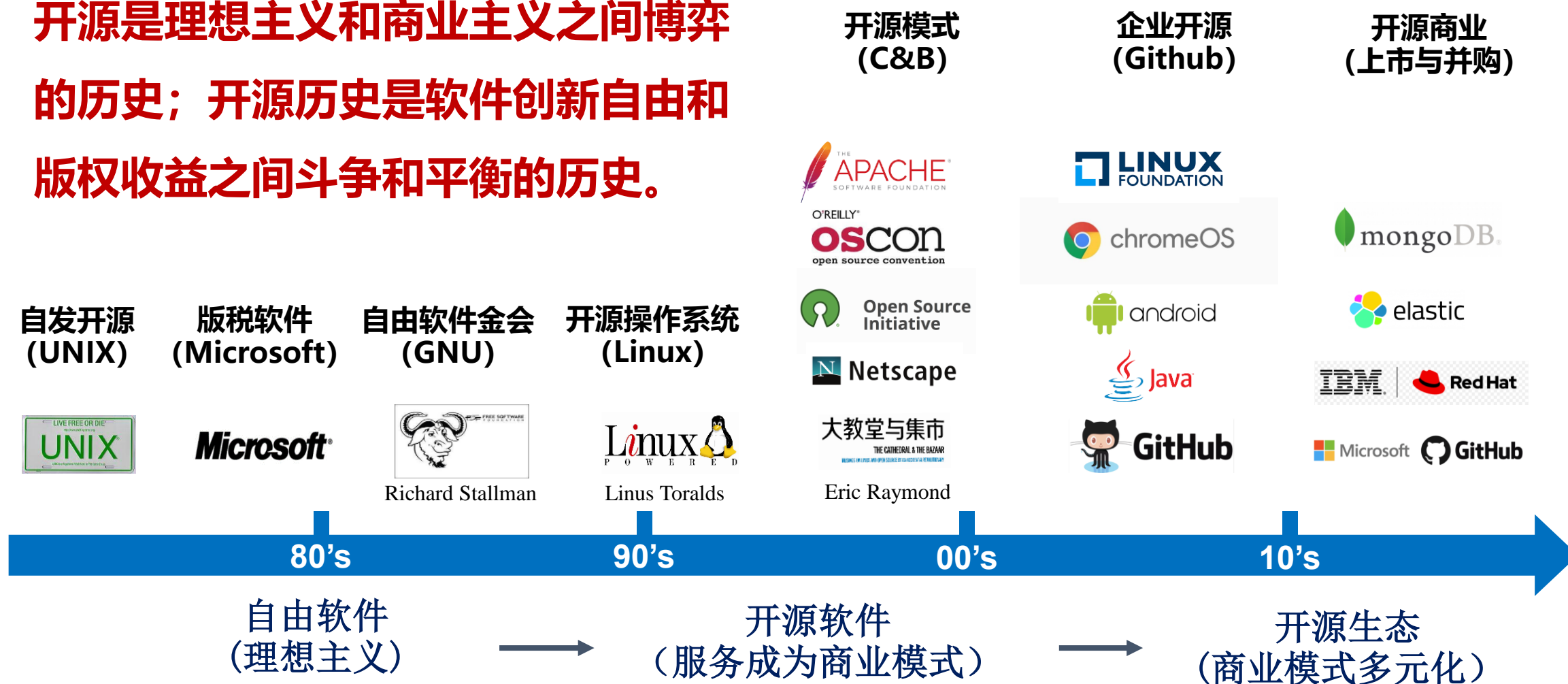


2021

软件从跟硬件搭售到单独售卖，是**专有**推动了创新；  
从商业许可发展到开源许可，是**共享**推动了创新；  
从开源发展到混合模式，是**共享/专有的融合**创新

# 开源软件的历史：自由和商业博弈的历史

开源是理想主义和商业主义之间博弈的历史；开源历史是软件创新自由和版权收益之间斗争和平衡的历史。



# 开源软件的历史：英雄们的历史



## 0. 1970年 启蒙年代

- Ken Thompson 和 Dennis Ritchie 缔造 Unix



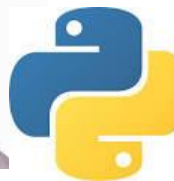
## 1. 1984年 自由软件运动

- AT&T 将UNIX商业化
- Bill Gates的微软发布DOS和Windows
- Richard Stallman 发起GNU项目



## 2. 1991年 Linux 内核系统

- Linus Torvalds 发起并发动社区
- 完善了GNU项目



## 4. 著名开源软件项目

- Larry Wall, perl, 1987
- Guido V. Rossum, Python, 1991
- R. Ihaka & R. Gentleman, R, 1993

## 3. 1998年 开源峰会

- Netscape 宣布开放Navigator浏览器的源代码
- “开源软件”替代“自由软件”，并广泛传播

自由软件

开源软件的诞生与发展

# 开源软件的商业模式

商业模式	简介	特点	代表企业
<b>Support 支持服务</b>	<ul style="list-style-type: none"><li>用户只需为技术支持及咨询服务买单</li></ul>	<ul style="list-style-type: none"><li>人工外包作，利润率偏低</li><li>工作可复制性低，scale较难</li><li>客户转换率低，通常&lt;1%</li></ul>	 
<b>Hosting 托管</b>	<ul style="list-style-type: none"><li>供应商将其开源软件作为服务托管在云上，通过收取每月/每年的托管和服务费获利</li></ul>	<ul style="list-style-type: none"><li>该模式成为了部分云厂商打包开源项目赚取利润的途径</li></ul>	 
<b>Restrictive Licensing 限制性许可</b>	<ul style="list-style-type: none"><li>通过提供一个带有稍带限制的开源许可证来激励使用者进行付费</li></ul>	<ul style="list-style-type: none"><li>许可证定义模糊，需要法院判决</li><li>部分公司禁止使用该商业模式下的开源软件</li></ul>	 
<b>Open-core 开放核心</b>	<ul style="list-style-type: none"><li>该模式下的大部分代码是开源的，而少数代码（针对企业用户）是专有的，需要收费</li><li>专有部分可以打包成与开源基础部分连接的单独模块或服务，或者在分叉版本中分发</li></ul>	<ul style="list-style-type: none"><li>该模式可以避免云厂商打包开源项目赚取利润</li><li>难以拿捏开源范围的尺度</li><li>很难将代码中的开源与专有特性完全分离</li></ul>	 
<b>Hybrid Licensing 开放核心+混合许可</b>	<ul style="list-style-type: none"><li>最新的模式，在开放核心基础上进行了改进</li><li>混合许可在同一个代码库中混合了开源代码和专有代码</li><li>用户可以选择只使用开源代码，或者同时使用开源代码和专有软件代码</li></ul>	<ul style="list-style-type: none"><li>代码在同一个代码库中，使管理和开发变得更容易</li><li>允许用户方便升级到付费模式</li><li>允许外部社区（比如GitHub）成员对专有软件功能模块进行改进</li></ul>	 

目前，开放核心+混合许可逐渐成为主流的商业模式，其原因在于：

- 开源软件商能够轻松管理代码库而不必拿捏开源的尺度
- 客户能够方便的从免费开源模式切换到付费模式（不需要额外部署，也不需要和销售沟通）
- 外部的开源社区也能对专有付费模块代码进行改进，降低了开发成本

来源：云启资本



# 开源的理念扩展到了软件行业以外

- 硬件产品是一种设计的实现，硬件的设计，以及由这些设计创造物理产品的许可，也可以以一种开放许可的方式向公众提供
- 例如：今天RISC-V的开源，中国业界深度参与

## 开源硬件 (open source hardware)

硬件制品的设计信息，以及硬件的软件驱动程序，全部以 FOSS 的方式发布。硬件设计信息包括机械图、系统图、电路原理图、材料配单、PCB 布局、HDL 源码以及集成电路布局等。



# 今天的国际开源状况

## 全民开源

### GitHub

- 2020 年总活跃项目数量约 **5,373 万个**，相较于 2019 年的约 3,972 万增长约 **35.3%**；
- 2020 年 GitHub 总开发者数为 **5,600 多万**，总活跃开发者数量约 1,446 万，相较于 2019 年的约 1,190 万增加了约 **21.2%**。

Let's look back at the code and communities built on GitHub this year...  
Based on the data collection range of October 2019 - September 2020.

56 M+  
total developers on  
GitHub

72 %  
of Fortune 500 companies  
use GitHub Enterprise

60 M+  
new repositories created  
in the last year

1.9 B+  
contributions added  
in the last year

《2020 community report》，GITHUB

## 企业参与

- 微软2018年**收购GitHub**，2020年**收购**全球最大的JS包管理平台**npm**；Prosus2021年**收购Stack Overflow**。
- Redhat调查了全世界1250位全球IT领导者，**90%**正在使用企业开源软件。《**2021年企业开源现状**》，Redhat，2021.3
- Synopsys分析了17个行业1,500+个代码库，其中**98%**包含了开源代码。平均每个应用包含的开源组件数量**5年内增长259%**。《**2021年开源安全与风险分析报告**》，Synopsys，2021.4

# 中国的开源现状

- 受益开源，贡献国际项目，开源自有项目

- 操作系统：openEuler，openHarmony，
- 数据库：TiDB，OceanBase，openGauss，
- AI框架：paddlepaddle，mindSpore，
- 应用技术：Vue.js，echarts

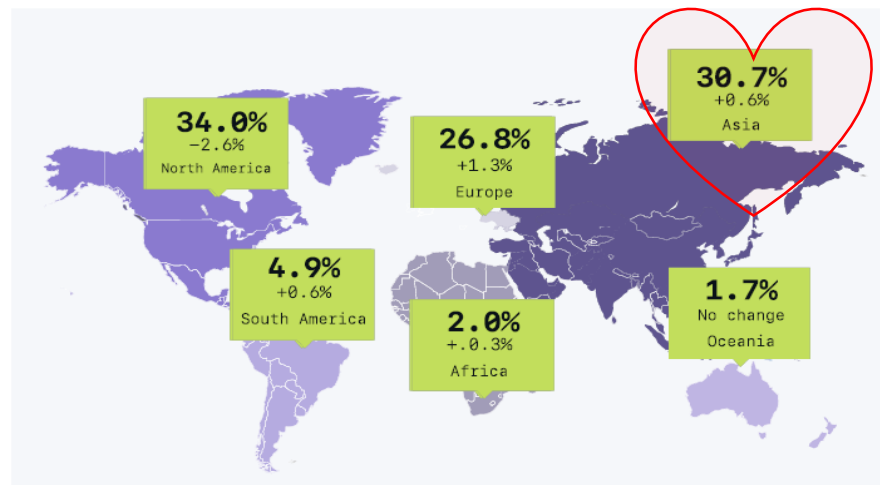
- 开源基础设施逐步发展和成熟

- 代码托管平台：Gitee，code.china，Trustie
- 第一个开源基金会：开放原子开源基金会
- 第一个OSI approved的中英文开源许可证：MulanPSL2.0 -- 40k+个项目

- 开源商业化

- 一批开源软件独角兽或准独角兽创业企业，如PingCAP等，获得风险投资

Geographic distribution of active users  
Current and absolute change relative to 2019



# 木兰许可证系列：中英文双语

## 大背景：

本土企业需求，中文开源社区的发展和成长需求。

## 具体需求：

中文解释权，少/无风险的开源项目和产品的发展需求。

## 第一个获得OSI认可的本土开源许可证：木兰宽松许可证MulanPSL2



国内社区广泛支持



17k+ 项目采用了 MulanPSL-2.0，覆盖云计算、大数据、AI、OS 等，例如，openEuler，openGauss，香山(RISC-V处理器)。

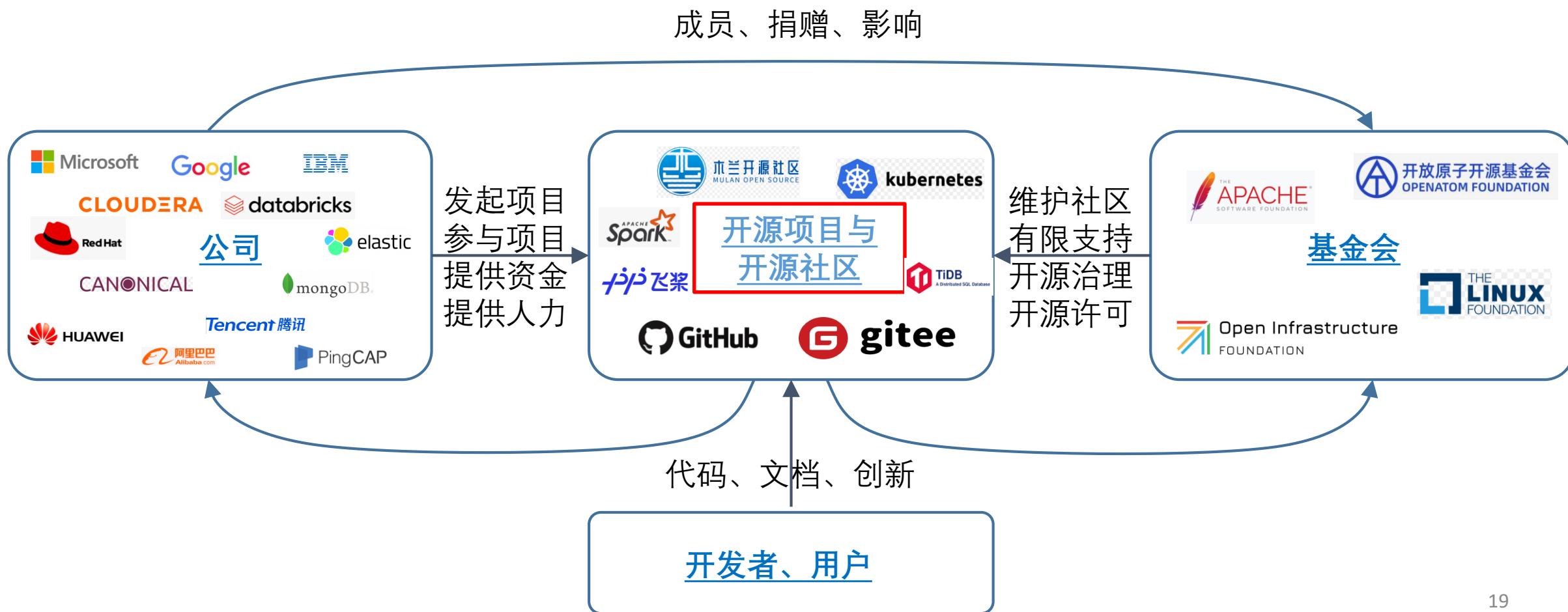


# 中国的开源：机会和挑战并存

- 国家战略： 开源创新被列入十四五规划和2035远景目标
  - “支持数字技术开源社区等创新联合体发展，完善开源知识产权和法律体系，鼓励企业开放软件源代码、硬件设计和应用服务。”
- 本土的优势： 市场大（多样化需求）、人多（人才储备强）。
- 存在的不足： 奉献精神不够，协作共赢欠缺，难以形成生态。以竞争、主导为出发点的“开源”仍然不少。

# 开源社区的结构与治理

# 开源软件生态的结构



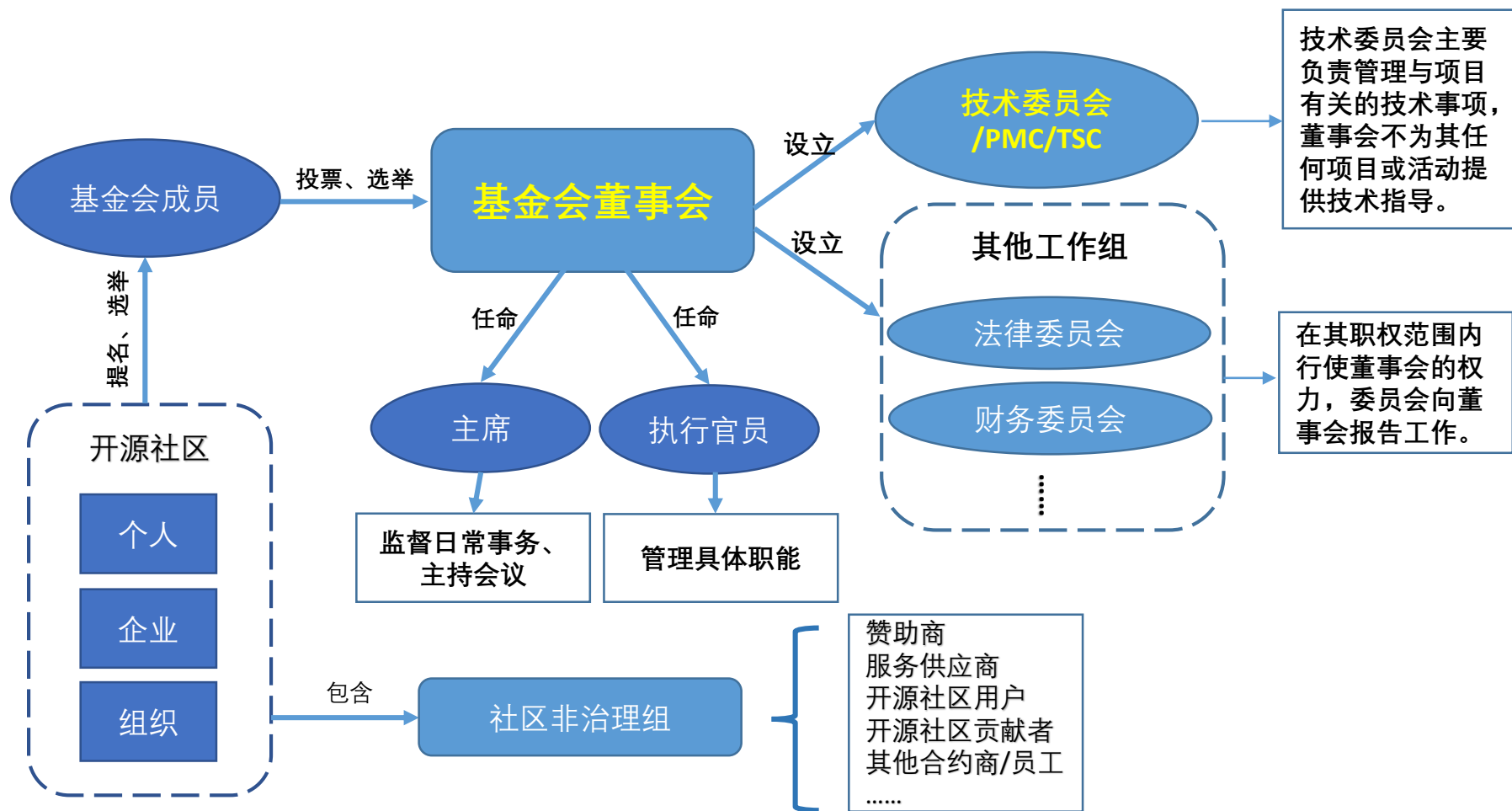
# 开源项目运作的核心框架

- **代码托管**：选择许可证，选择托管平台，（选择基金会），开放源代码
  - 软件的生态定位
- **社区使能**：定义系统化贡献指南，经营多样化社区
  - 定义指南：CONTRIBUTING.MD，为新手任务打标签
  - 运营：生态的形成需要有三类角色：软件/产品/服务的生产者、提供者和消费者
- **版本发布和迭代**：新特征开发和发布、老版本持续维护的周期
- **项目可持续**：社区和市场的平衡

# 开源基金会的角色

- 任何人都可以在Internet上发布项目的源代码，但是要构建一个可持续的项目社区，仅靠代码是不够的，而基金会就扮演了“社区看门人”的角色。
- 开源基金会作为管理和推广开源项目的非盈利机构，为开源项目提供基础设施、活动、培训以及法律、商业、技术等服务。

# 开源基金会的组织架构



# Apache基金会(Apache Software Foundation, ASF)



成立于1999年

涉及大数据、云、搜索和CMS、DevOps和构建管理、物联网和边缘计算、机器人和深度学习、服务器、Web框架等领域。



一个项目要进入Apache基金会，需要按照基金会的要求进行“孵化”。Apache孵化器就是为那些想要进入Apache基金会的项目提供服务的，孵化一般需要一年半的时间，满足一系列质量要求之后方可毕业，通过孵化毕业的项目要么成为顶级的ASF项目，要么成为其他顶级项目的子项目。



面向开源的研究



# 面向开源的经典研究

- 经济学
  - 为什么人们会无偿做一件事情?
- 组织学
  - 一个松散无序的社区怎么能高效生产高质量产品?
- 管理学
  - 开源社区的创新原则是什么? 用户创新!
- 软件工程
  - 开源开发了跟商业软件媲美的高质量软件, 其开发方法、技术和工具是什么?
- 企业实践
  - 怎么借鉴开源做全球分布式开发? 内源!



# 数据时代带来巨大机会



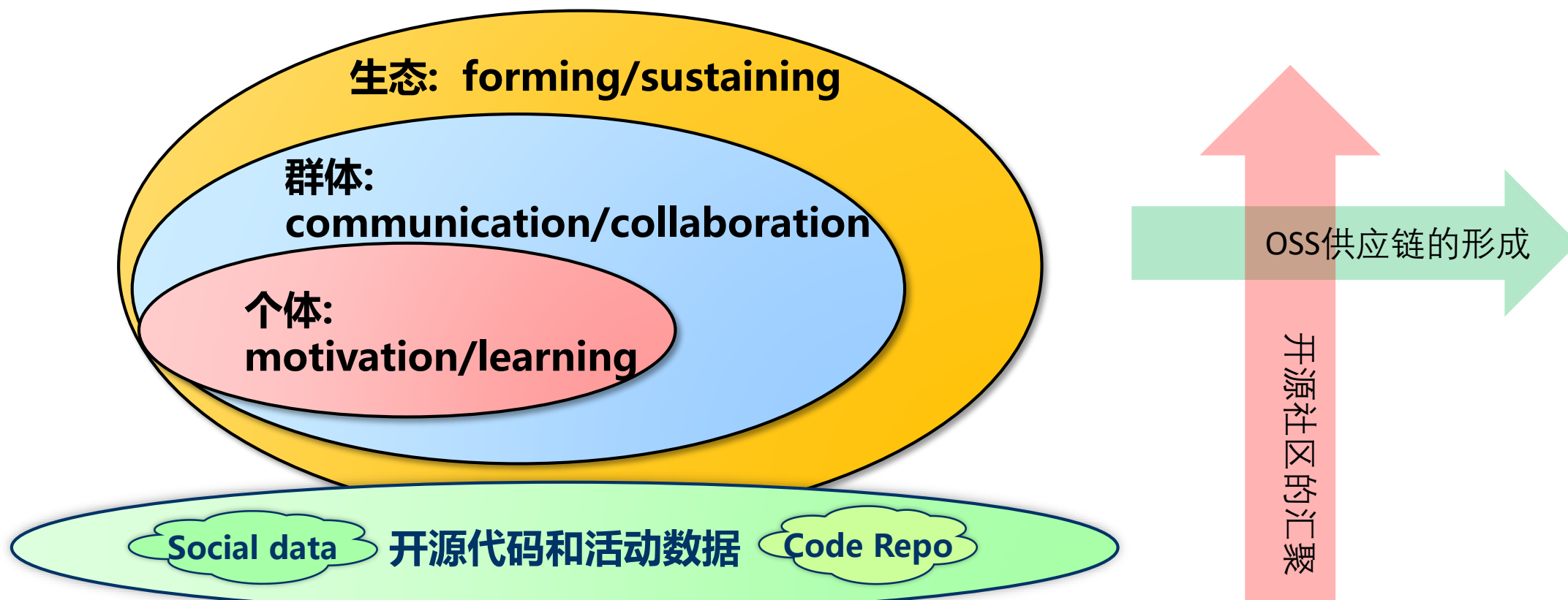
# 理解开源生态：数据驱动的实证主义



开源数字社会学

# 开源数字社会学

- **基础**: 软件活动数据
- **方法**: 数据挖掘技术、统计学、学科交叉
- **问题**: 软件自动化、群体协作、开源治理的效率与质量



# 开源环境：人人有责，从我做起

- 国内的开源环境
  - 企业：云计算（大）厂商受益开源不贡献开源
  - 大学：缺乏学生开源组织/缺乏开源相关课程
  - 程序员/学生：工作太忙/功课太多/缺乏开源精神和实践
- 解决之道：人人有责，从我做起
- 北大计算机系开源开发课程（in process）
  - <https://code.educoder.net/projects/minghui/OSSDevelopment/>
  - <https://github.com/osslab-pku/OSSDevelopment/>
  - <https://gitee.com/osslab/ossdevelopment>



周明辉

[zhmh@pku.edu.cn](mailto:zhmh@pku.edu.cn)