

操作系统软件包组件引用情况及治理边界讨论

sig-compliance

目录

- 背景与现状
- 软件包组件引用情况分析
- 治理边界讨论

背景与现状

openEuler社区组成-代码仓

openEuler社区自
研开源软件

如isula、stratovirt

openEuler社区基
于开源衍生软件

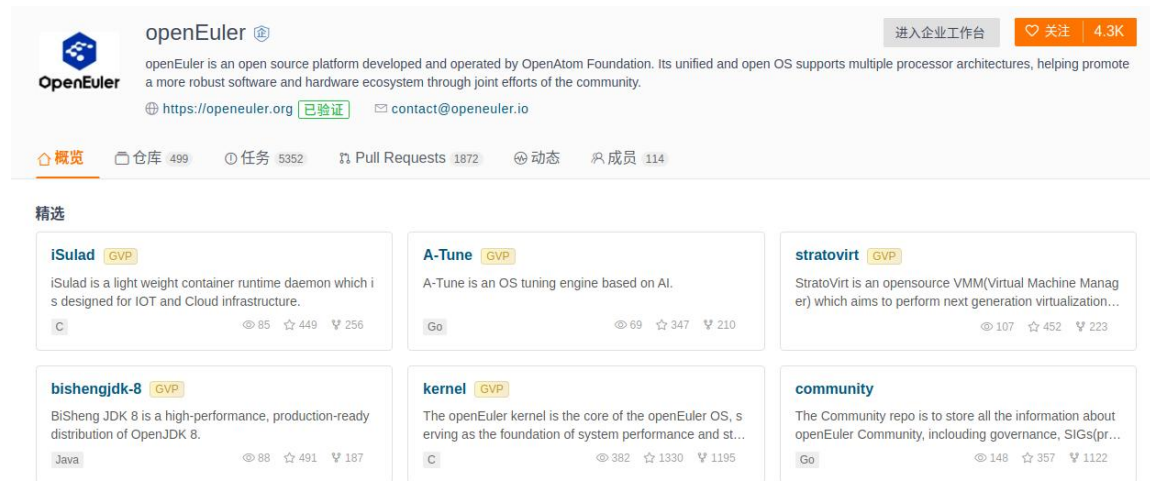
如 bishengjdk

openEuler社区重
度维护开源软件

如 kernel、qemu

共计400+

 **gitee openEuler**



The screenshot shows the openEuler GitHub repository page. At the top, the repository name 'openEuler' is displayed with a verified badge. Below it, a description states: 'openEuler is an open source platform developed and operated by OpenAtom Foundation. Its unified and open OS supports multiple processor architectures, helping promote a more robust software and hardware ecosystem through joint efforts of the community.' Links for 'https://openeuler.org' and 'contact@openeuler.io' are provided. The repository statistics show 499 repositories, 5352 tasks, 1872 pull requests, and 114 members. A '精选' (Selected) section lists several repositories:

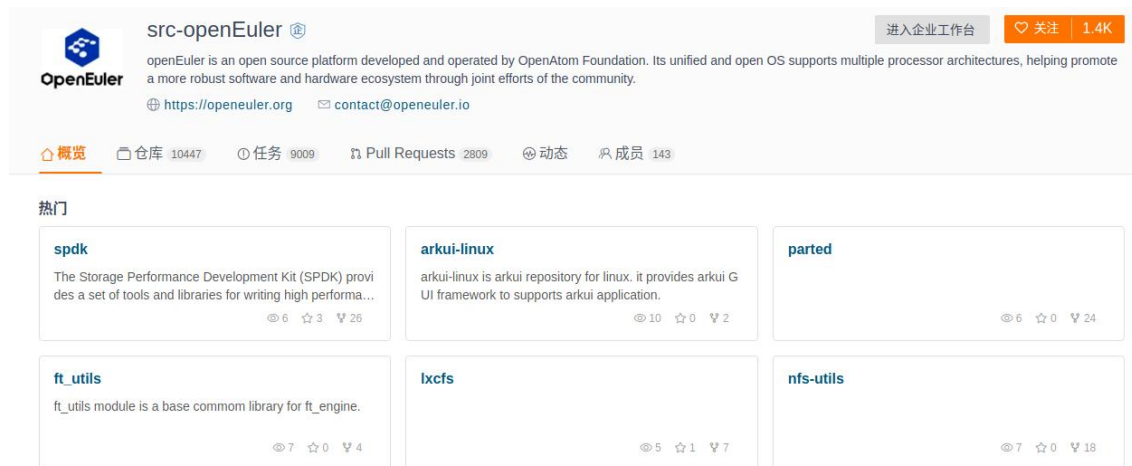
Repository Name	Language	Stars	Forks	Issues
iSulad	C	85	449	256
A-Tune	Go	69	347	210
stratovirt	Go	107	452	223
bishengjdk-8	Java	88	491	187
kernel	C	382	1330	1195
community	Go	148	357	1122

openEuler社区组成-软件包仓

openEuler代码仓为上游的开源软件

其他开源社区或开源项目为上游的开源软件

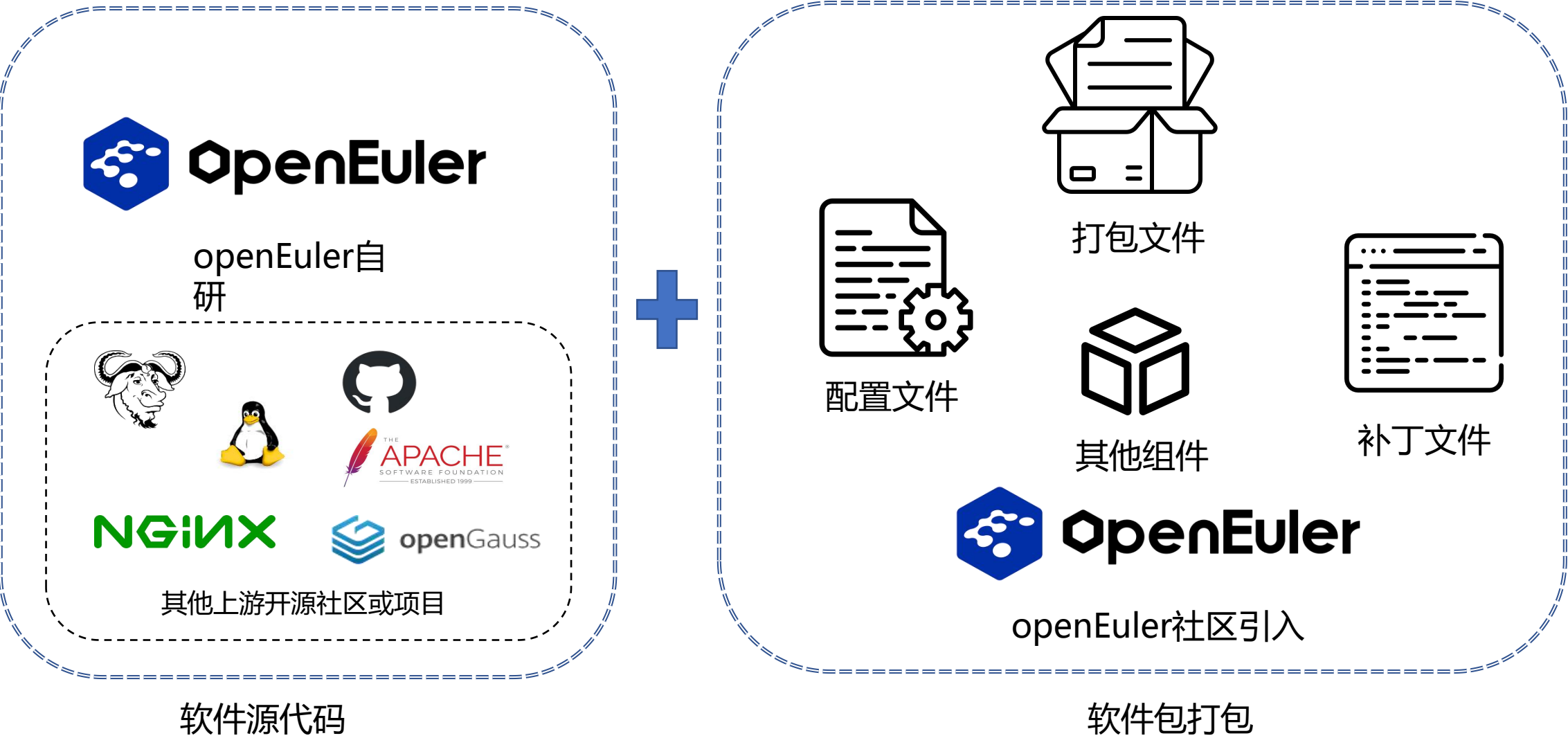
共计
10000+
 **gitee** src-openEuler



根据选型策略、维护质量等要求，选择其中的部分软件包形成操作系统发行版，俗称“进版本”。



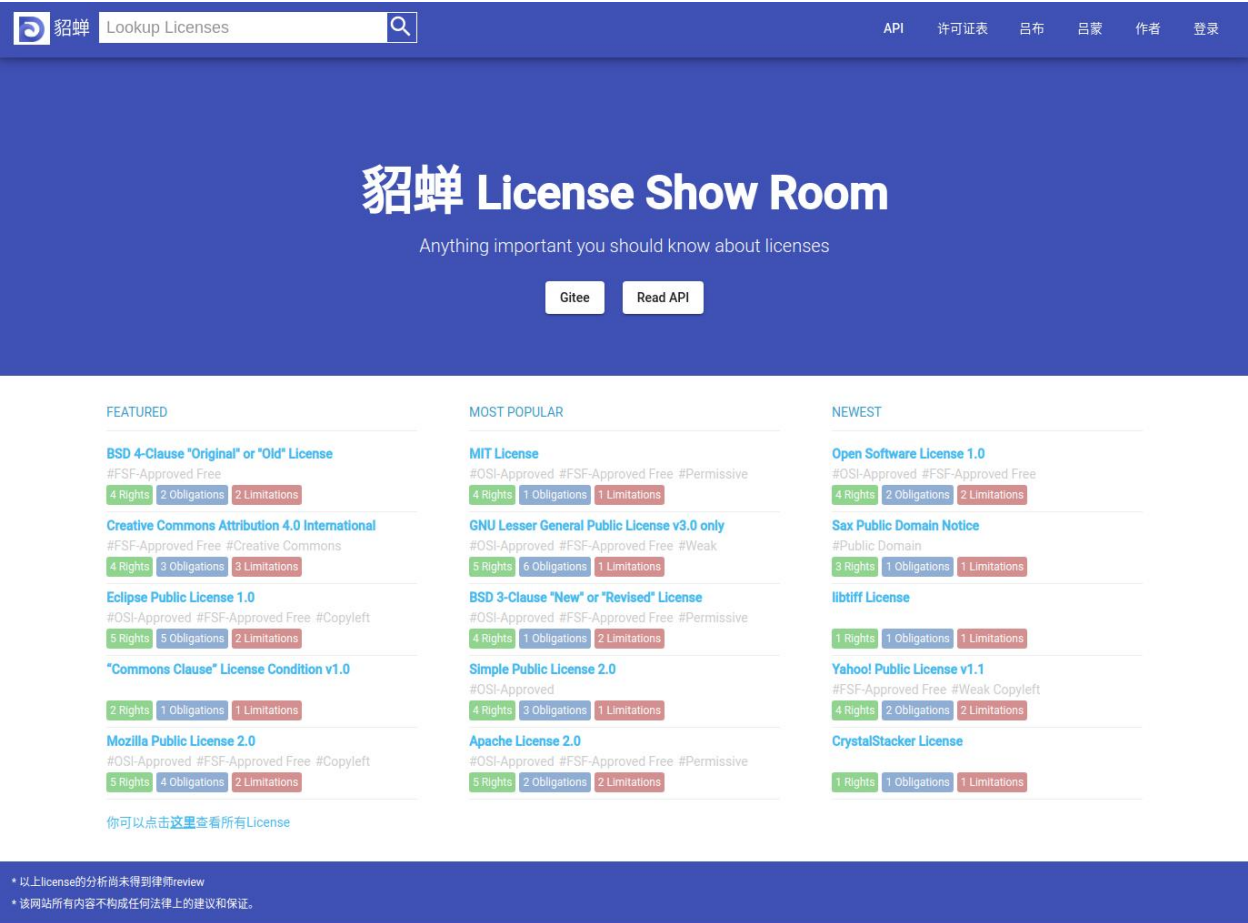
openEuler软件包成分



openEuler合规治理成果-基础能力

- License
 - 貂蝉网站提供License基础信息
 - 吕蒙提供文本识别许可证能力
 - 吕布提供开源许可证兼容性判断能力
 - Compliance SIG对社区引入License进行审核判断
- 源代码片段引用治理
 - 对源代码片段引用风险分析较为完善
 - 有相应落地工具实践

源代码片段(Snippet)引用风险分析						2022 开源季
风险分析：关键点-不要修改被引用片段的License&Copyright声明						
<ul style="list-style-type: none">我们需要关注4个要素：本地项目的License/Copyright, 本地文件的License/Copyright, 远程项目（被引用的开源项目）的License/Copyright, 远程文件的License/Copyright如果文件级的License或Copyright未声明，则它会继承项目级的License/Copyright声明						
本地文件L/C声明	远程文件L/C声明	本地项目的声明和远程项目的L/C声明	本地文件“实际”L/C 声明	远程文件“实际”L/C 声明	关注的风险	合规措施
空	空	LPL (Local Project License)	LPL, LPC	RPL, RPC	LPL 大概率不等于RPL LPC 一定不等于RPC	本地文件声明，并
空	有	LPC (Local Project Copyright)	LPL, LPC	RFL, RFC	LPL 大概率不等于RFL LPC 一定不等于RFC	本地文件不要删除
有	空	RPL (Referenced Project License)	LFL, LFC	RPL, RPC	检查：本地文件的L/C声明是否远程项目信息L/C一致。	如有风险：修改本地文件的L/C声明
有	有	RPC (Referenced Project Copyright)	LFL, LFC	RFL, RFC	检查：LFL = RFL and LFC = RFC	如有风险：修改本地文件的L/C声明



openEuler合规治理成果-社区治理

- 代码仓
 - 已规范项目许可证、版权声明
 - 已初步具备片段引用治理能力
- 软件包仓
 - 已实现SPEC文件许可证声明与源码声明一致性门禁检查

openeuler-ci-bot

拥有者

11月25日 17:36

Check Name		Build Result	Build Details
check_binary_file		✔ SUCCESS	#218
check_package_license		⚠ WARNING	
check_package_yaml_file		✔ SUCCESS	
check_consistency		✔ SUCCESS	
check_spec_file		✖ FAILED	#176
aarch64	check_build	✔ SUCCESS	
	check_install	✔ SUCCESS	
x86_64	check_build	✔ SUCCESS	#217
	check_install	✔ SUCCESS	

表情

回复

MaJun

工作台

代码

合规

开源片段

发布

漏洞

版本扫描PR扫描

社区openEuler代码仓请选择代码仓

● 风险数据看板可在帮助中心,查看“风险治理政策”

社区告警总数: 317619社区待处理告警总数: 174918

代码仓	最新扫描时间	任务状态	分支	CommitId	告警总数	总数	待处理告警数			已处理告警数		
							License引用风险	Copyright风险	License&Copyright风险	总数	人工分析数量	自动分析数量
A-FOT	2023-12-05 16:1...	成功	master	c8dbbe1fa3d83fab4f517a873582da9ba01502	0	0	0	0	0	0	0	0
A-Ops	2023-12-05 16:2...	成功	master	08dc9631b6908bea0813c6c10ac0001912126	22	0	0	0	0	22	0	22
A-Tune	2023-12-04 15:3...	成功	master	086085b9c9e2520bd18e6ad779bc2239ef03467	703	43	17	19	7	660	0	660
A-Tune-Collector	2023-12-05 16:2...	成功	master	5472c1671e2436d826b5d05694a41169263ae753	35	6	0	2	4	29	0	29
A-Tune-UI	2023-12-05 16:3...	成功	master	6a8acac4889ead04884780c537d67b5e3735d0d	8	0	0	0	0	8	0	8
BIOSK	2023-12-11 20:5...	成功	master	4876a8cce7ed485eca81786c93965e056129f4f	57	2	0	2	0	55	0	55
BiShengCLangui...	2023-11-08 13:5...	成功	master	ac05c6c17a630838b282246921b885799af720b	0	0	0	0	0	0	0	0
Kmesh	2023-12-07 11:5...	成功	master	33290654ea888024be762215f0e4c89bf4c0ddc1	3012	1126	52	1074	0	1886	0	1886
KubeOS	2023-12-12 17:3...	失败	master	3000a0c97b843e7ed13227a2a9ad121ec839247b	3994	1250	56	1192	2	2744	0	2744
PilotGo	2023-11-10 11:4...	成功	master	296412ee128b809645726bb1988082a89f8812	3449	3449	0	0	0	0	0	0
RISC-V	2023-12-06 09:2...	失败	master	9a964be7583548f57a86307a8eb7cb8cb6708d32	0	20	0	9	0	0	0	0
aops-apollo	2023-12-05 16:4...	成功	master	3ba21bc9d9ab0d195f4c784dc2c0b6e6da27e17	0	0	0	0	0	0	0	0
aops-ceres	2023-12-05 22:1...	成功	master	#2b1d8ae1c0e546139a0e8022319319aba803	7	1	0	1	0	6	0	6
aops-diana	2023-12-05 17:0...	成功	master	2ea71f57889c0f076abb4e56f6c0dbf3c75e9	0	0	0	0	0	0	0	0
aops-hermes	2023-12-05 22:3...	成功	master	88829c586464143bbe105995fba08844b033efc	16	1	0	1	0	15	0	15
aops-vulcanus	2023-12-05 17:1...	成功	master	39505a909163793b65f0e22f3b32be9303767a5	0	0	0	0	0	0	0	0
aops-zeus	2023-12-05 21:1...	成功	master	a9fa7a913e2c702e22166b71763ed931e18d7	2	1	0	1	0	1	0	1
async-libfuse	2023-12-05 17:3...	成功	master	30c779a2b01117ef407aa86f48db13642dc88b	33	15	1	14	0	18	0	18

共 104 条50条/页123>前往 1 页

软件包仓仍需进一步更细粒度治理，为此需要梳理软件包组件引用情况，从而确定开源合规治理策略和切入点

软件包组件引用情况分析

软件包内部成分分析



软件包仓每个仓库或操作系统发行版提供的src包解压后，可能会存在上述每类不定项个成分，这些成分满足了功能业务需求的同时，可能会从不同来源、以不同形式引入与目标打包软件版权和开源许可不同的成分，这也为开源合规治理带来了更大的复杂度。

软件源代码包引用情况

片段引用

文件引用

组件引用-
Vendor

组件引用-
包配置管理
文件

openEuler / Kmesh

Watch 16 Star 33 Fork 18

代码

Issues 10

Pull Requests 1

Wiki

统计

流水线

服务

master 分支 3 标签 3

+ Pull Request + Issue 文件 Web IDE 克隆/下载

openeuler ci bot !141Reconstruction to decouple cgroup_con... 3329065 2个月前 405 次提交	
api	modify Kmesh file permission which only root can use. 5个月前
bpf	support L4 traffic manager 2个月前
build	enable mda feature in kmesh program 4个月前
cmd	modify license 9个月前
config	kmesh compile macros init 3个月前
daemon	fix get kmesh-daemon help failed when kmesh.service start 9个月前
depends/include	adapt openEuler23.03 10个月前
docs	!140Readme optimization 3个月前
kernel	support L4 traffic manager 2个月前
mk	Kmesh: fix can not find bpf and api-c-v2 pkgconfig when compile go file 9个月前
oncn-mda	sync some changes from 23.09 branch 3个月前
pkg	!141Reconstruction to decouple cgroup_conn and sockops bpf logic 2个月前
release/Kmesh/docker	update docs 12个月前
test	support aarch64 3个月前
vendor	support aarch64 3个月前
.gitignore	enable pkg/bpf 2年前
LICENSE	modify license 9个月前
Makefile	optimize build.sh and Makefile 4个月前
README.en.md	Initial commit 1年前
README.md	docs: Readme optimization 3个月前
build.sh	kmesh compile macros init 3个月前
go.mod	update go.mod and vendor for go build 1年前
go.sum	update go.mod and vendor for go build 1年前
kmesh.spec	modify Kmesh file permission which only root can use. 5个月前
kmesh_bpf_env.sh	kmesh compile macros init 3个月前
kmesh_macros_env.sh	kmesh compile macros init 3个月前

简介

Kmesh (kernel mesh) is a data plane software for service grids. It is dedicated to providing infrastructure for service communication and service governance for cloud applications, provides better latency and noise floor performance.

sig-ebpf

C 等 6 种语言

Apache-2.0

发行版 (3) 全部

release kmesh-0.3.0 4个月前

openEuler_sig-eb... ⑦

贡献者 (13) 全部

近期动态

24天前评论了任务 #16SFF2 是否可以和 cilium比较一下，有哪些差异和优劣?

24天前加入了仓库

24天前被 openeuler ci bot 移出了仓库

SPEC打包文件引用情况



构建依赖



运行依赖

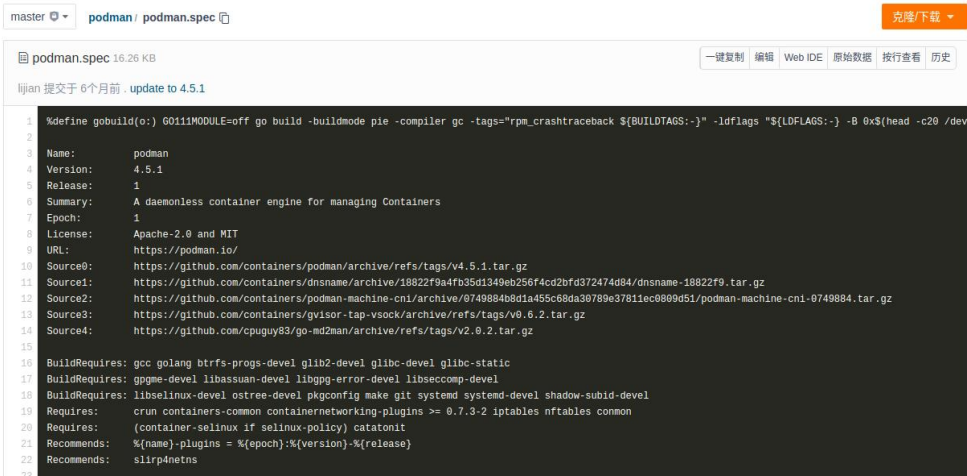
Patch文件引用情况

Patch来源 Patch引入组件	自研	上游项目 补丁回合	第三方补丁引入
片段引用			
文件引用			
组件引用			

其他组件源代码包引用情况

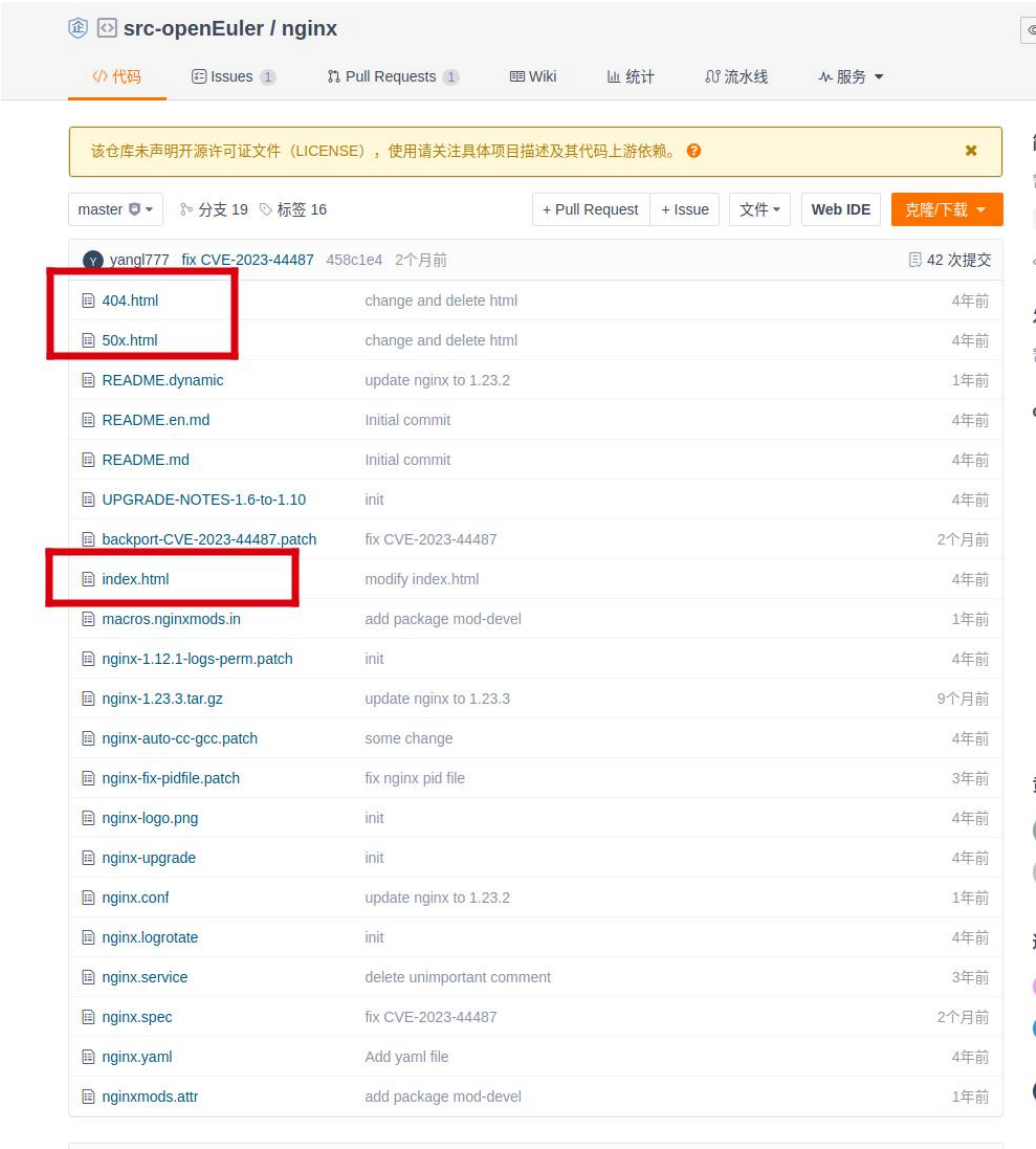


理论上操作系统软件包中只应当存在一份源代码包，然而开发过程中存在此现象，应当判断该引入组件源代码包作用，最佳实践该组件应当单独作为软件包引入社区。



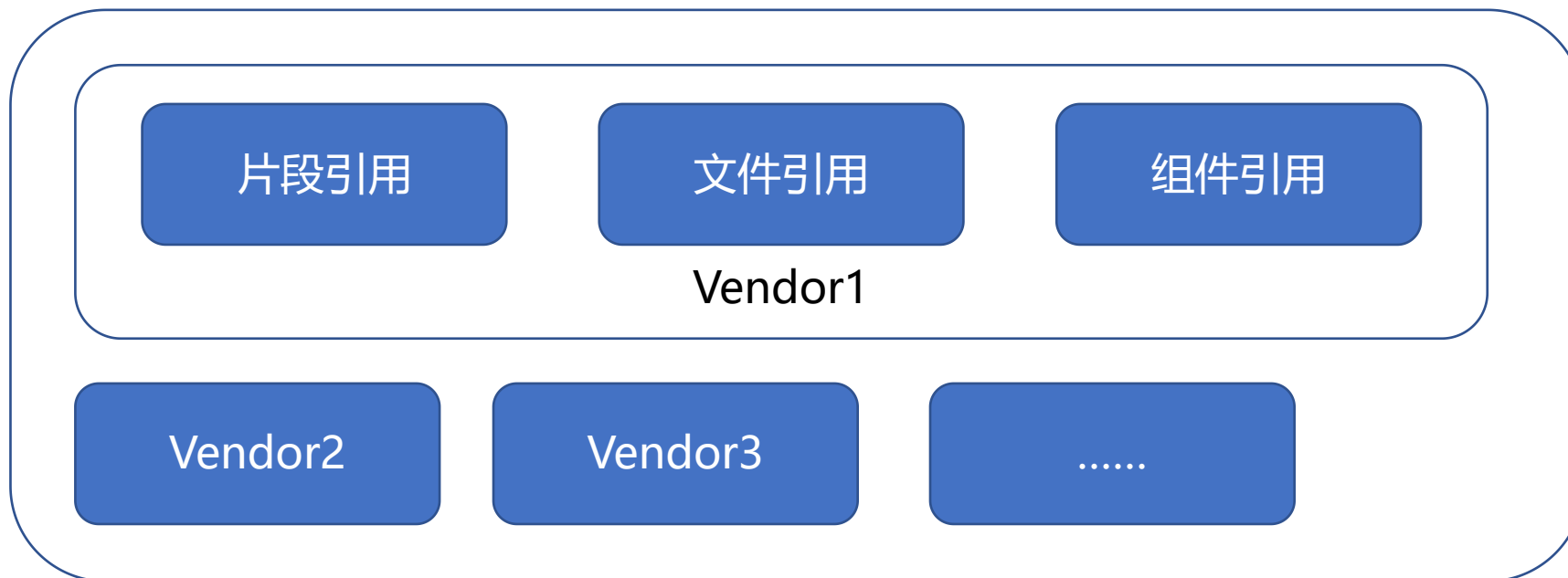
配置文件引用情况

绝大多数情况配置文件不作为版权及开源许可的声索依据，不会引入新的版权或开源许可证问题，但需考虑部分特殊情况可能会涉及商标等问题。



src-openEuler / nginx		
该仓库未声明开源许可证文件 (LICENSE)，使用请关注具体项目描述及其代码上游依赖。		
master 分支 19 标签 16		
+ Pull Request + Issue 文件 Web IDE 克隆/下载		
yangl777 fix CVE-2023-44487 458c1e4 2个月前 42 次提交		
404.html	change and delete html	4年前
50x.html	change and delete html	4年前
README.dynamic	update nginx to 1.23.2	1年前
README.en.md	Initial commit	4年前
README.md	Initial commit	4年前
UPGRADE-NOTES-1.6-to-1.10	init	4年前
backport-CVE-2023-44487.patch	fix CVE-2023-44487	2个月前
index.html	modify index.html	4年前
macros.nginxmods.in	add package mod-devel	1年前
nginx-1.12.1-logs-perm.patch	init	4年前
nginx-1.23.3.tar.gz	update nginx to 1.23.3	9个月前
nginx-auto-cc-gcc.patch	some change	4年前
nginx-fix-pidfile.patch	fix nginx pid file	3年前
nginx-logo.png	init	4年前
nginx-upgrade	init	4年前
nginx.conf	update nginx to 1.23.2	1年前
nginx.logrotate	init	4年前
nginx.service	delete unimportant comment	3年前
nginx.spec	fix CVE-2023-44487	2个月前
nginx.yaml	Add yaml file	4年前
nginxmods.attr	add package mod-devel	1年前

Vendor包引用情况



理论上操作系统软件包中应当将目标软件所需vendor同样打包，转换为软件包依赖，然而开发过程中存在此现象，如golang、rust等语言社区目前无组件引入计划，应用软件打包只能以开发者自行压缩Vendor目录的形式提供。此种方式对安全和开源合规造成较大隐患。

治理边界讨论

抛砖引玉

- 1.社区对操作系统软件包引入组件的开源合规治理边界？
- 2.社区软件包开源合规治理与功能业务需求冲突时该如何平衡？
(例如vendor打包问题)
- 3.开源合规治理中检测到问题由上游引入，该如何处理？

THANKS