Pouch 容器的演进与实践

阿里巴巴系统软件事业部-沈陵

目录
CONTENTS

- Pouch在阿里内部使用现状
- 阿里容器技术的演进和实践
- Pouch的开源路线和未来规划

Pouch

Pouch在阿里内部的使用现状

- 本意育儿袋，隐喻贴身呵护应用
- 始于2011年
- 基于LXC
- 阿里内部容器技术产品，并于当年上线
- 2015年初开始吸收Docker镜像功能
- 容器结合阿里内核，大幅提高隔离性
- 大规模部署于阿里集团内部

Pouch在阿里内部的使用现状

规模：

- 覆盖集团大部分BU
- 2017年双11百万级容器
- 在线业务100%容器化

覆盖场景：

- 运行模式
- 编程语言
- 技术栈

覆盖业务：

- 蚂蚁&交易&中间件
- B2B/CBU/ICBU/1688/村淘
- 合一集团（优酷）
- 菜鸟&高德&UC（接入中）
- 广告（阿里妈妈）
- 阿里云专有云输出
- 集团测试环境
- ……

目录
CONTENTS

- Pouch在阿里内部使用现状
- 阿里容器技术的演进和实践
- Pouch的开源路线和未来规划

Pouch

从物理机到容器

- 架构演变
  - 运行：从集中式到分布式
  - 运维：从分散到云化

- 资源使用的演变
  - 从物理机到VM
  - 从VM到容器

- 容器的要素--阿里内部运维和应用视角
  - 有独立IP
  - 能够ssh登陆
  - 登陆后能够看到一个独立的，隔离的文件系统
  - 资源隔离—使用量和可见性

和物理机的
使用体验一致

Pouch容器发展轨迹

- 容器的要素--阿里内部运维和应用视角
  - 有独立IP
  - 能够ssh登陆
  - 登陆后能够看到一个独立的的文件系统
  - 资源隔离—使用量和可见性

- 手工Hack实现容器要素
  - 虚拟网卡，网桥
  - sshd
  - Chroot (pivot_root)
  - CGroup，Namespace

T4
- 引入LXC（Linux Container）
- 内核可见性隔离Patch
- 内核磁盘空间配额Patch

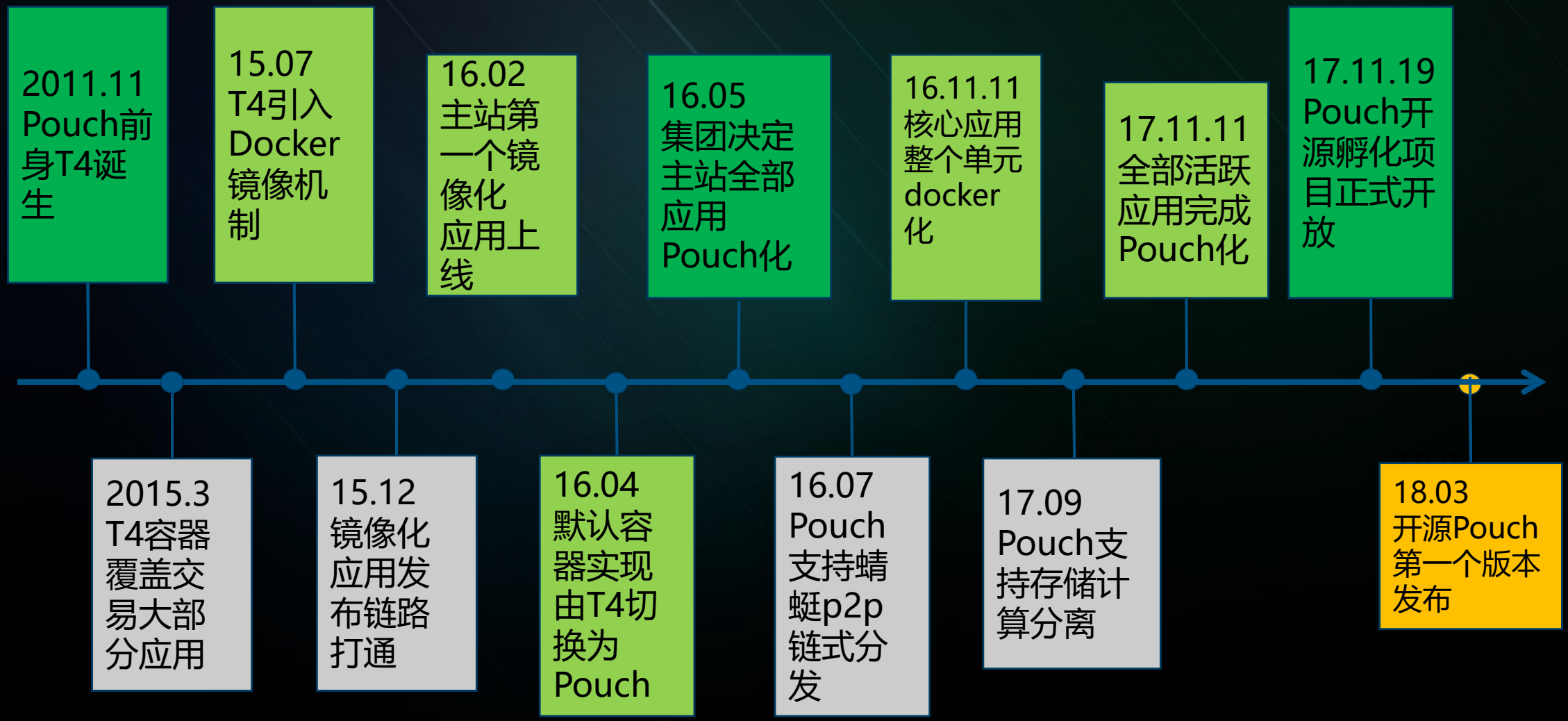
引入Docker

T4和Docker的整合

	T4	Docker
cpu隔离	cgroup.cpuset,cpu,cpuacct	cgroup.cpuset,cpu
内存隔离	cgroup.memory	cgroup.memory
进程隔离	pid namespace	pid namespace
文件系统隔离	chroot	chroot
磁盘空间隔离	dirquota	无
网络带宽隔离	飞天的netqos模块	无
网络模式	网桥+独立IP	网桥、Host
镜像系统	只有简单的t4模板	完备的Docker镜像系统
执行引擎	LXC	LXC，libcontainer
文件系统分层	overlayfs	aufs，device mapper，overlayfs
资源可见性	内核补丁隔离	无隔离
支持的内核和OS	alios5u,6u; 2.6.32-358; ali1172	7u+, 3.10+

Pouch-研发和运维

Pouch Roadmap



研发关注的要素

- 传统要素
  - 功能与性能
  - 稳定性
  - 可扩展性
  - 可测试性
- DevOps要素
  - **可运维性**
  - **运维成本**

DevOps八荣八耻

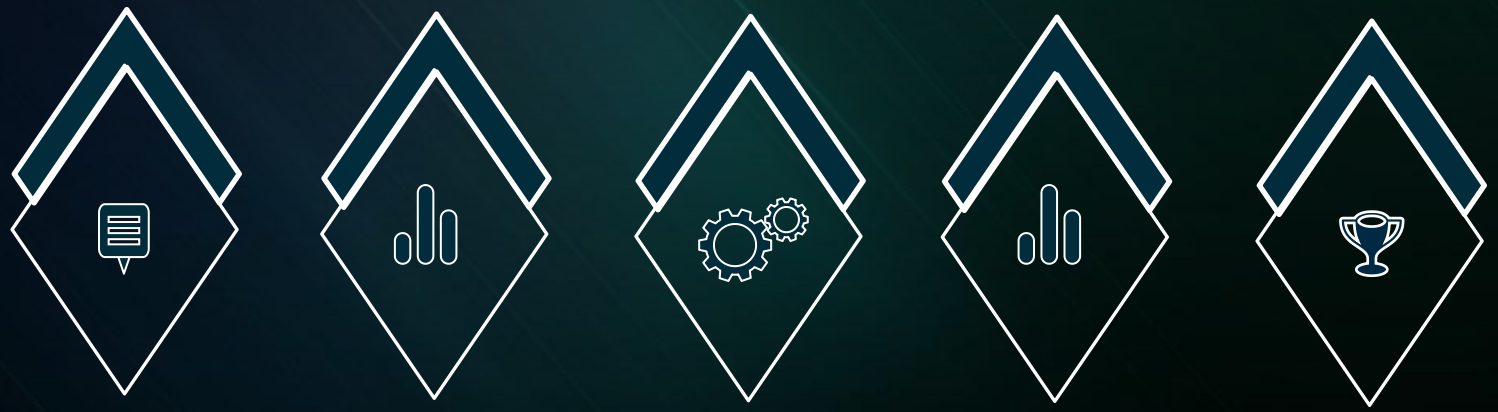
- **以可配置为荣，以硬编码为耻**
- **以系统互备为荣，以系统单点为耻**
- **以随时可重启为荣，以不能迁移为耻**
- **以整体交付为荣，以部分交付为耻**
- **以无状态为荣，以有状态为耻**
- **以标准化为荣，以特殊化为耻**
- **以自动化工具为荣，以人肉操作为耻**
- **以无人值守为荣，以人工介入为耻**

目录
CONTENTS

- Pouch在阿里内部使用现状
- 阿里容器技术的演进和实践
- Pouch的开源路线和未来规划

Pouch

Pouch的技术优势

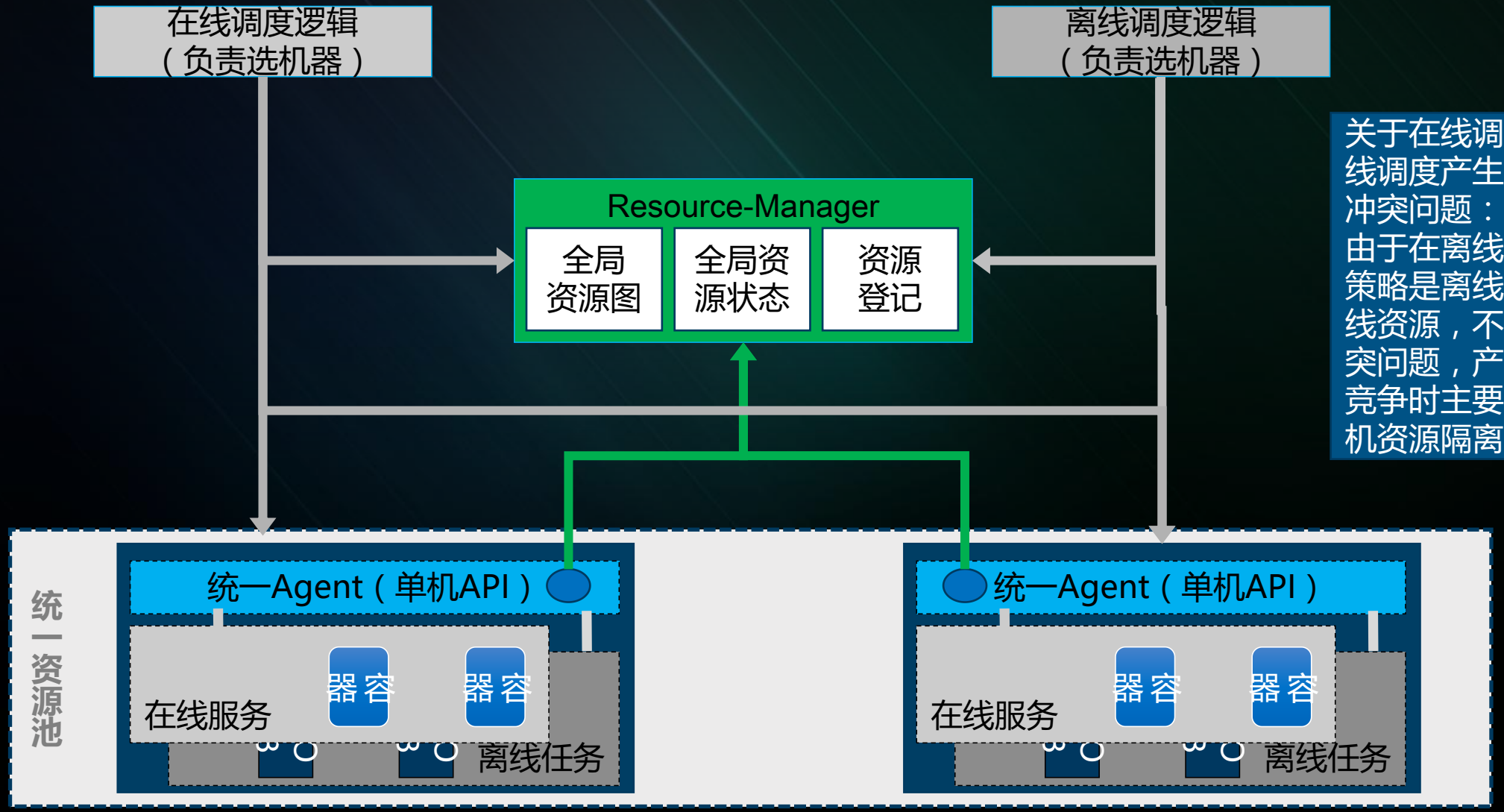


隔离性	P2P镜像分发	富容器	规模化考验	内核兼容性

丰富的隔离性

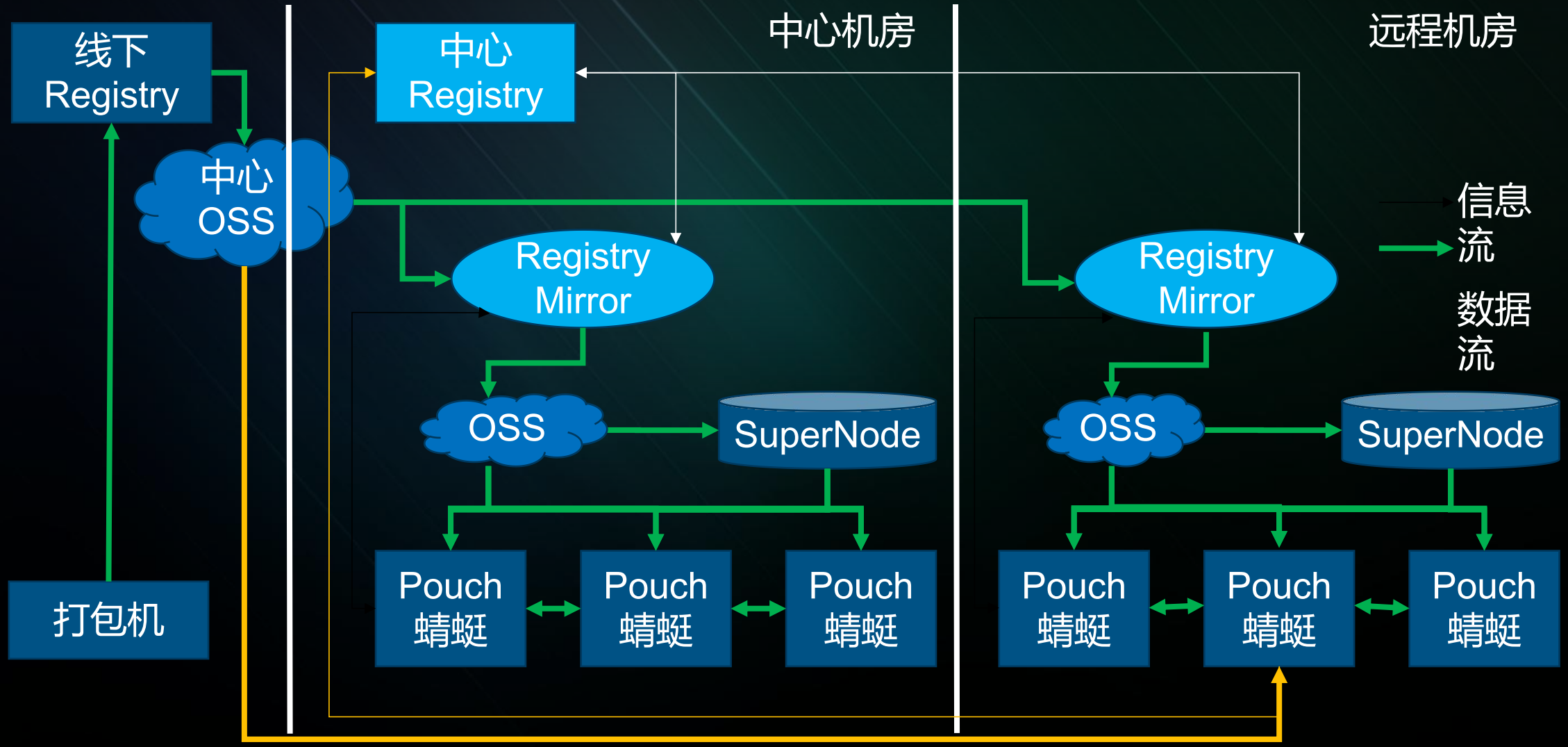
- 传统容器的隔离维度：namesapce，cgroup
- 更优的容器可见性隔离：内核patch，lxcfs
- 额外隔离维度：磁盘，网络等：diskquota
- 基于Hypervisor的强容器隔离
  - runV
  - clear container
  - Kata
  - ECS

离在线混部



关于在线调度和离线调度产生的资源冲突问题：
由于在离线混部的策略是离线共享在线资源，不存在冲突问题，产生严重竞争时主要通过单机资源隔离来控制。

基于p2p的镜像分发架构



Pouch开源计划

2017.10.10
合作伙伴共同孵化
外部开发者邀请内测

2017.11.11
后正式开源
与生态共建Pouch

2018.03.31
发布第一个
大版本

Pouch内部版本的体系结构

CNTC SEL

运维支持

富容器 | 进程管理 | 容器内重启 | Staragent | SSHD

发布模式 | 镜像热更新 | 登录信息保留

发布模式 | 权限系统 | 应用规范 | 监控系统 | 多套环境

插件体系

网络插件 | Alinet | Overlay | 公网下沉 | Sriov | vpc

volume插件 | alilocal | nvidia | tmpfs | 盘古

Graph插件 | Overlay 2 | 盘古 | Ceph

网络模型

Host | Nat | Bridge | Vlan | Vxlan | Sriov | VPC | Overlay | DHCP | IPAM

资源隔离

资源可见性 | CpuAcct | LXCFS

CPU | LLC | CFS

网络IO | 金银铜牌

磁盘IO | BufferIO | DirectIO

磁盘空间 | DirQuota | VolumeQuota

OS适配

Libnetwork | Overlayfs | 5u | 2.6.32 | sysVinit

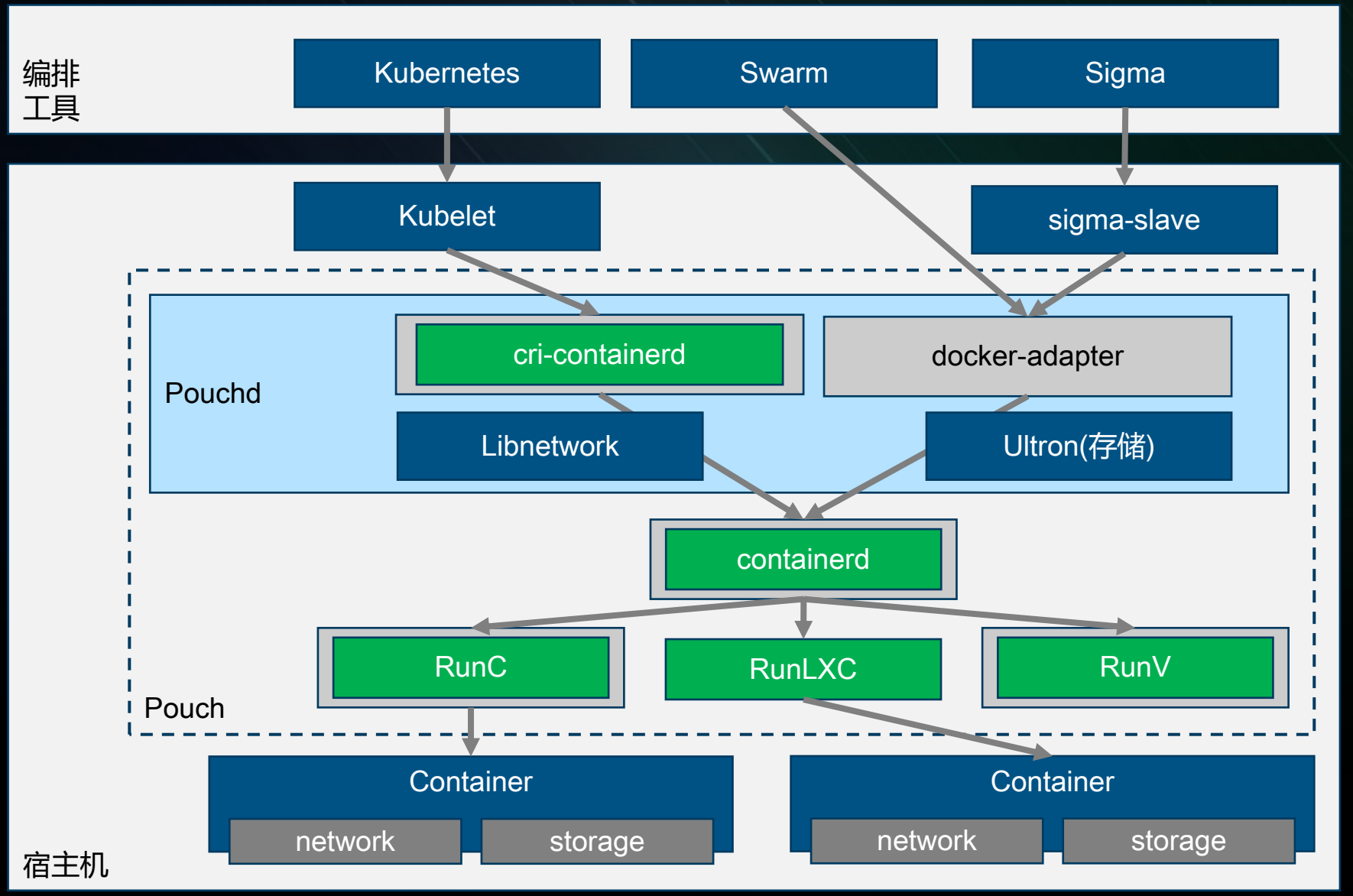
Libnetwork | Overlayfs | 6u | 2.6.32 | sysVinit

cgroup ns | RunLXC | 7u | 3.10/4.9 | systemd

宿主机

服务保活 | 镜像清理 | P2P分发 | 物理机/ECS | 安全控制 | 权限管理 | 环境检查

Pouch系统结构



社区标准

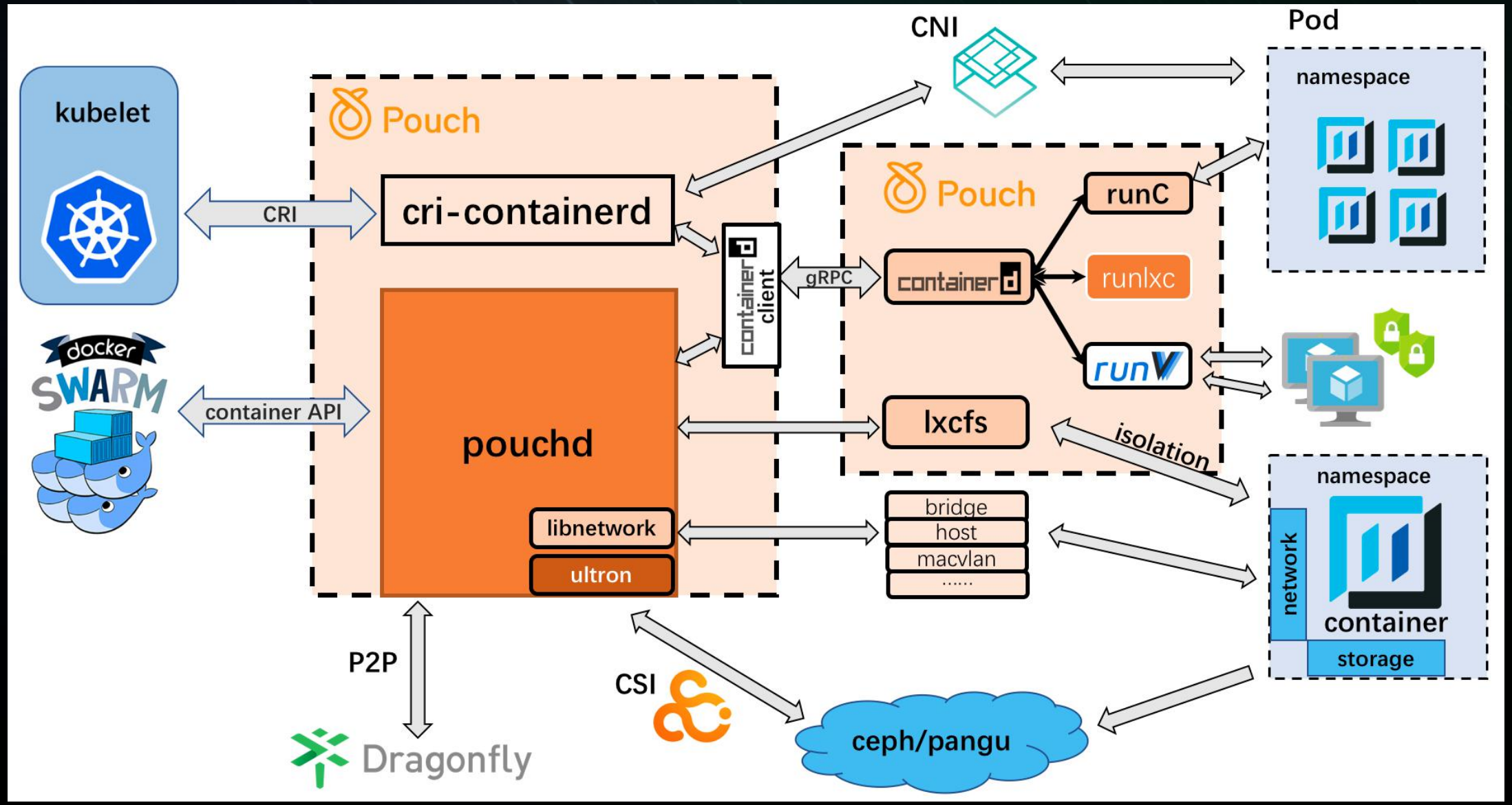
CRI

OCI

CNI

CSI

Pouch社区生态


CNI
Pod
kubelet
Pouch
namespace
CRI
cri-containerd
Pouch
runC
container client
gRPC
container
runlxc
runV
docker
SWARM
container API
pouchd
lxcfs
isolation
libnetwork
ultron
bridge
host
macvlan
.....
namespace
network
container
storage
P2P
CSI
Dragonfly
ceph/pangu

如何参与

- 在你的组织中使用Pouch
- 布道与宣传
- 贡献回你的bug修复、功能扩展以及文档
- 日常贡献 -> maintainer
- 说服你的朋友贡献新技术
- https://github.com/alibaba/pouch/blob/master/CONTRIBUTING.md

CNTC SEL
Software Engineering Lab
Zhejiang University

感谢聆听

Thanks ！