
Architect
SACC
2022 中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2022
激发架构性能 点亮业务活力
云上会议 网络直播 | 2022年10月27-29日
ITPUB

Cloud Native Computing and Intel Cloud Client

Intel

Cloud Software Engineering Manager

Zhao Juan(Amy)

Agenda

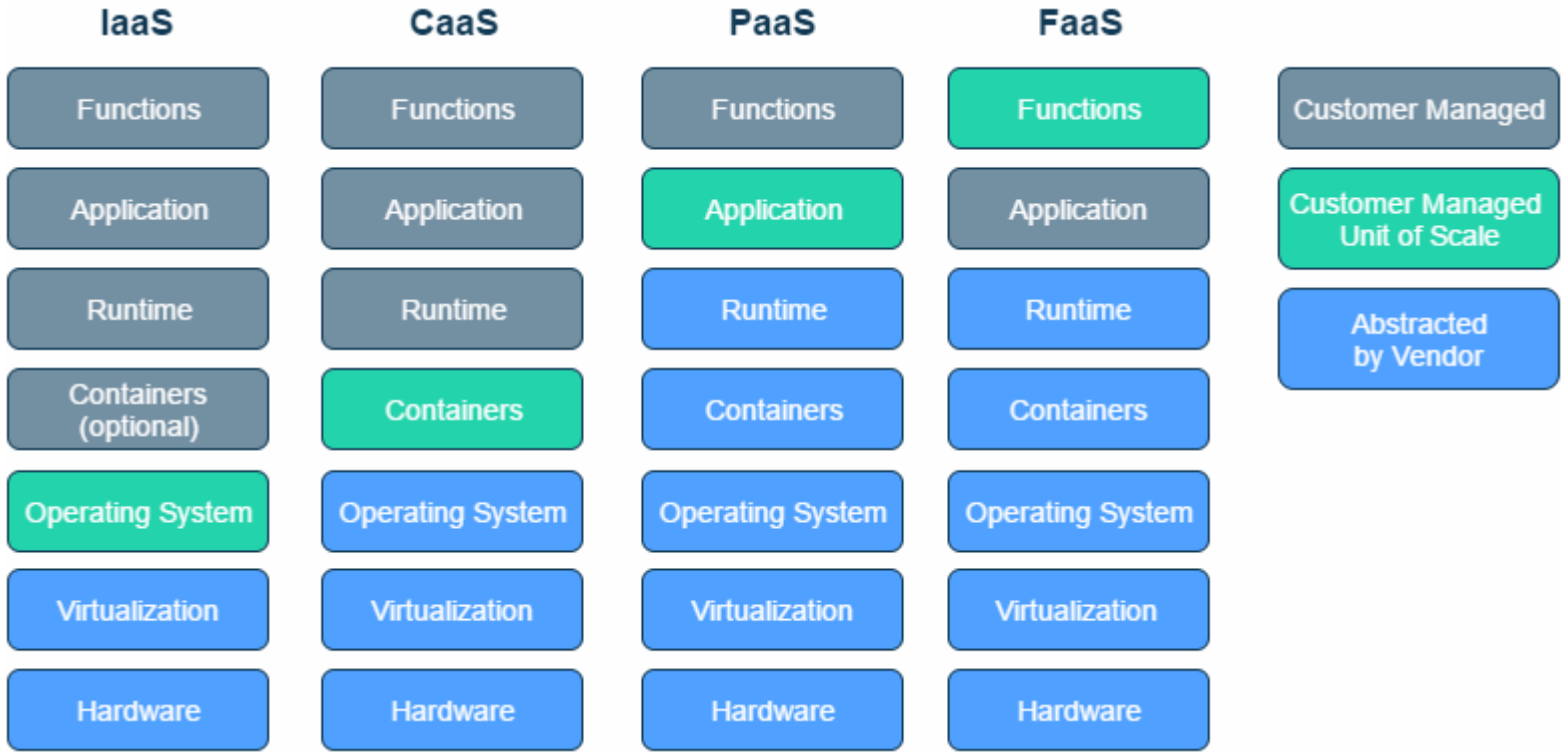
1. 云原生及其基础技术
2. Cloud2PC计算和云计算的异同
3. Cloud2PC 关键技术 – 网络，计算，安全，性能指标
4. 更多展望

云原生及其基础技术

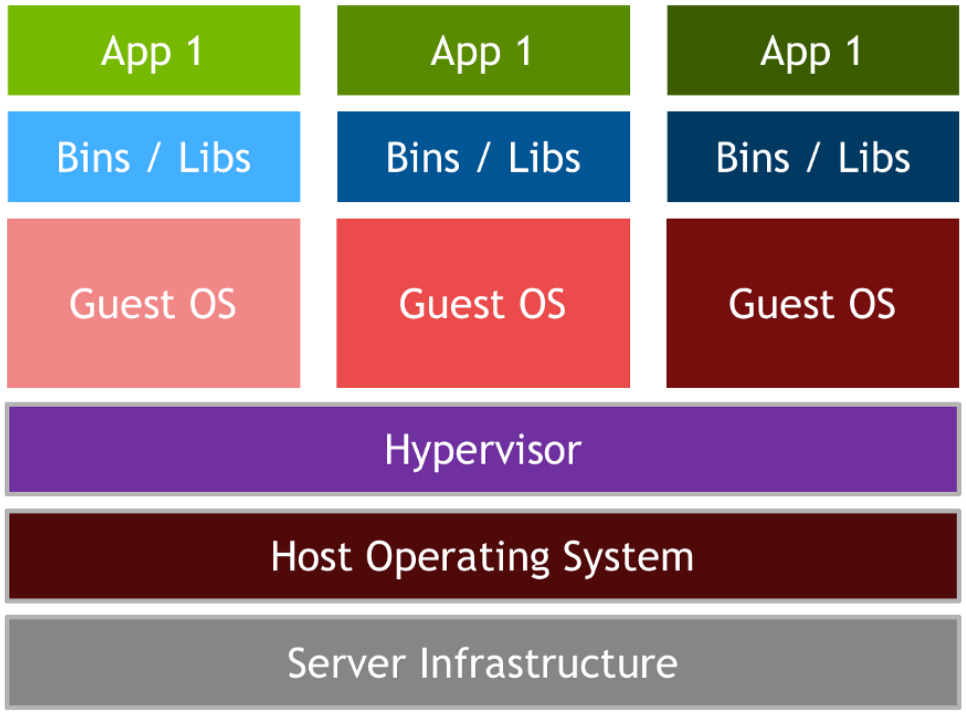
- 云计算一览
- Golang语言
- Kubernetes套件
- Ingress举例Nginx
- 基础： TLS, TPM, SGX, TDX
- 基础： Kubernetes中的Limit, Request, LimitRange
- 基础： DNS, DPDK

云计算一览

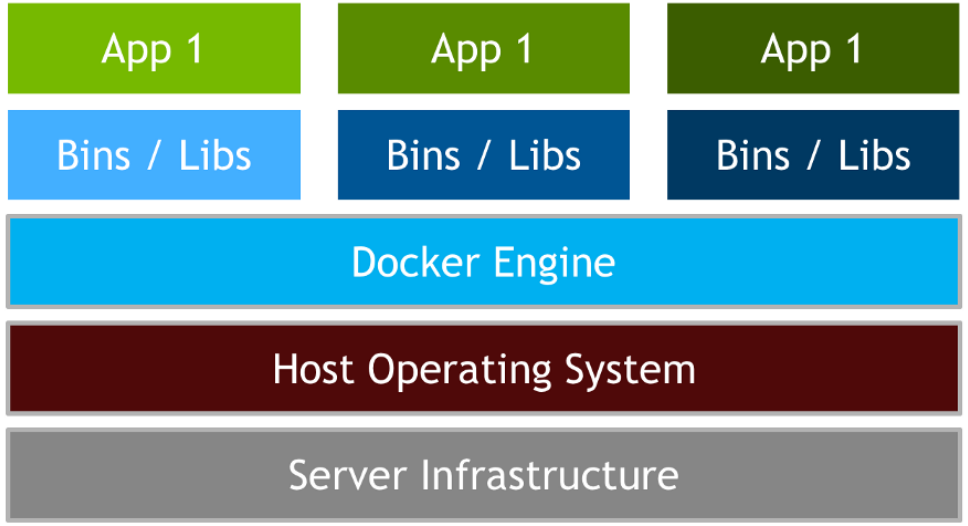
IaaS, CaaS, PaaS, SaaS, FaaS



虚拟机和容器

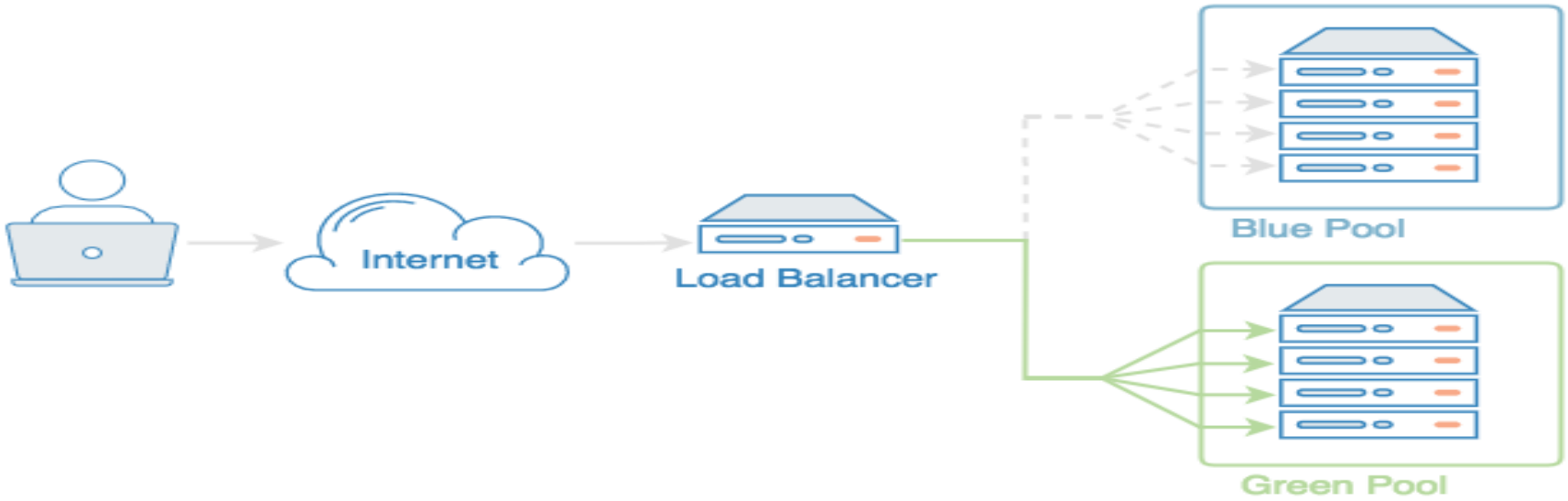


VIRTUAL MACHINES

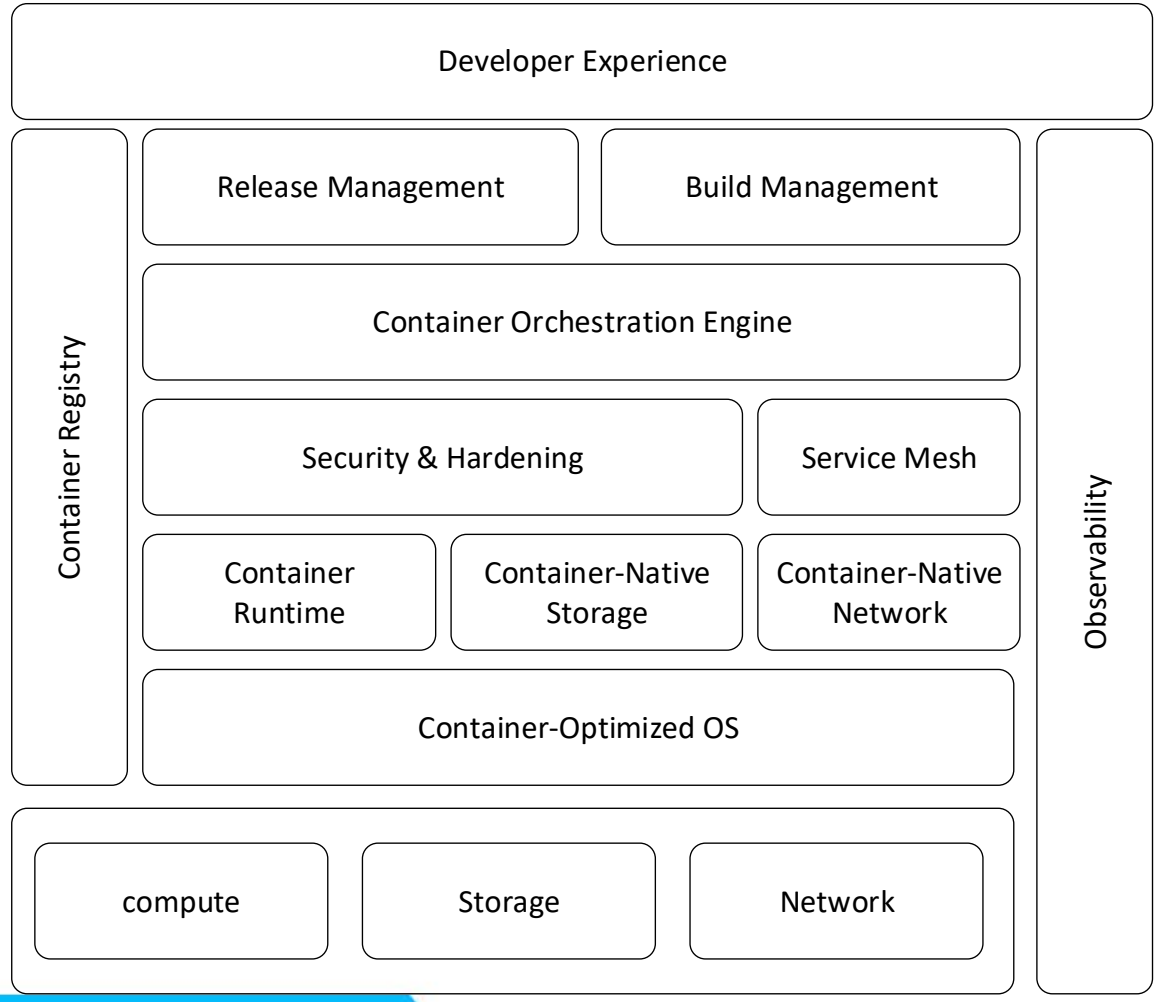


CONTAINERS

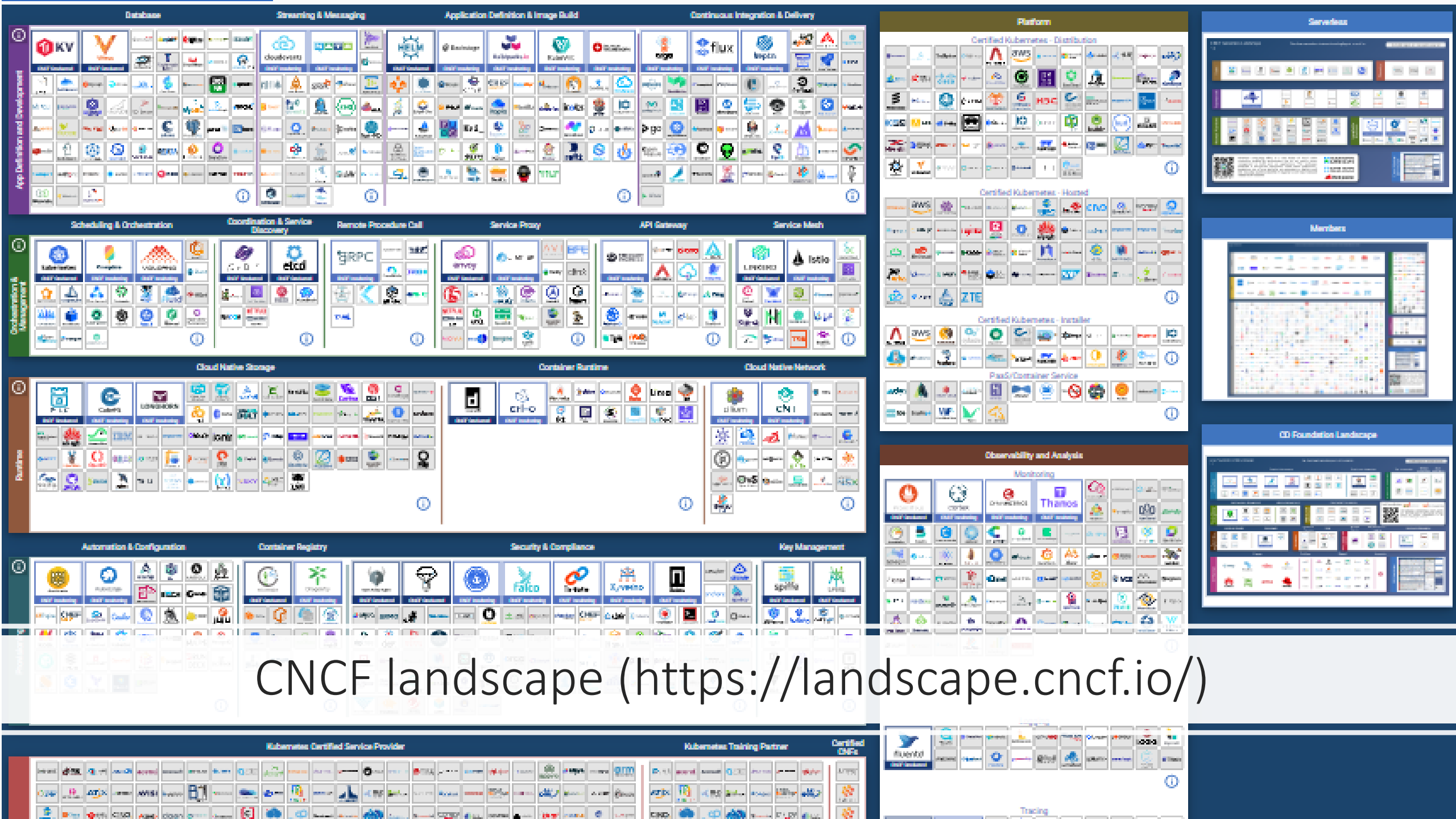
Load Balancer



The Cloud Native Stack



The Cloud Native Stack



CNCF landscape (https://landscape.cncf.io/)

Golang

Why Golang?

- 云原生的通用语言
- 非常健全的生态
- 强大而好用的工具链
- 特殊的协程和接口更好的支持并行和异步

Easy to try it out on
https://go.dev/play/?v=goprev

协程举例

```
func main() {
        task(parameters…) //在主进程中的执行函数

        go task1(parameters….) //在协程中执行的函数

        go task2(parameters定义) {
                //task2 contents
                ….
        }(parameters…)
}
```

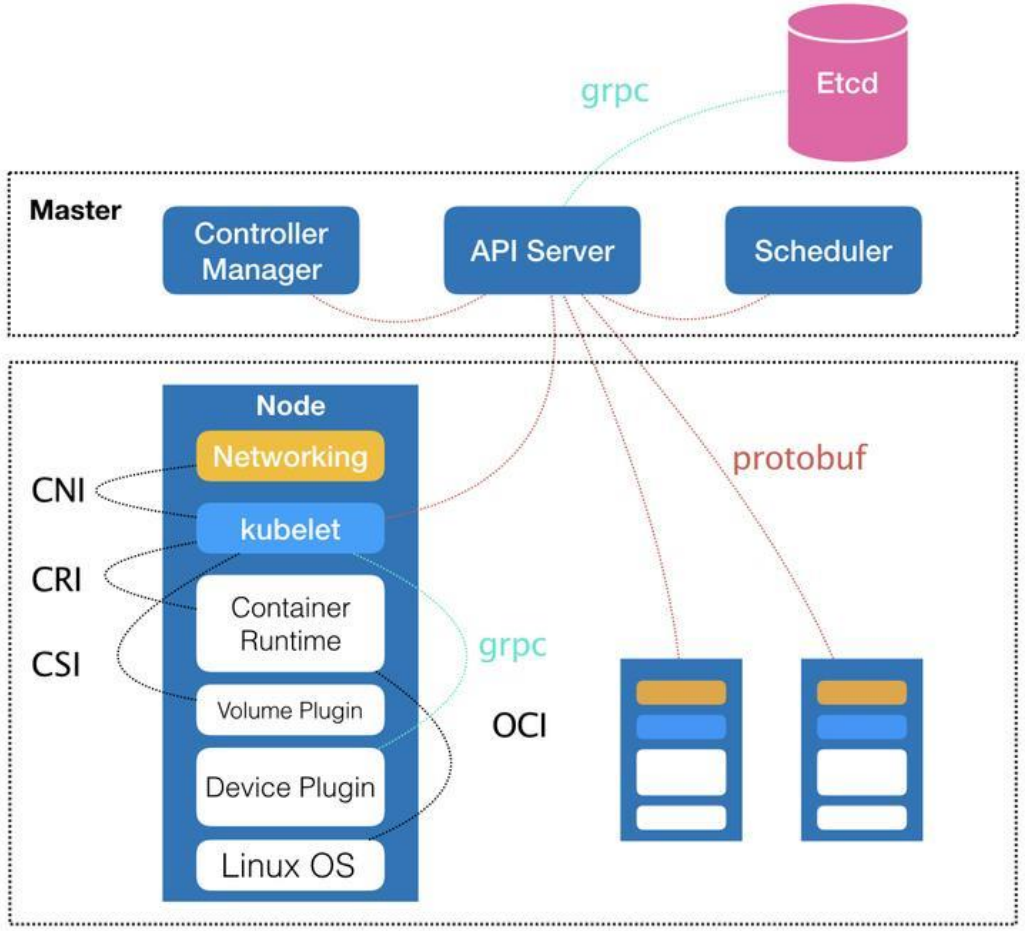
```
func main() {
        lock := &sync.Mutex{}

        for i := 0; i < 10; i++ {
                go Count(lock)
        }
        for {
                lock.Lock() // 上锁
                c := counter
                lock.Unlock()    // 解锁
                runtime.Gosched() // 出让时间片
                if c >= 10 {
                        break
                }
        }
}
```

协程，线程，进程

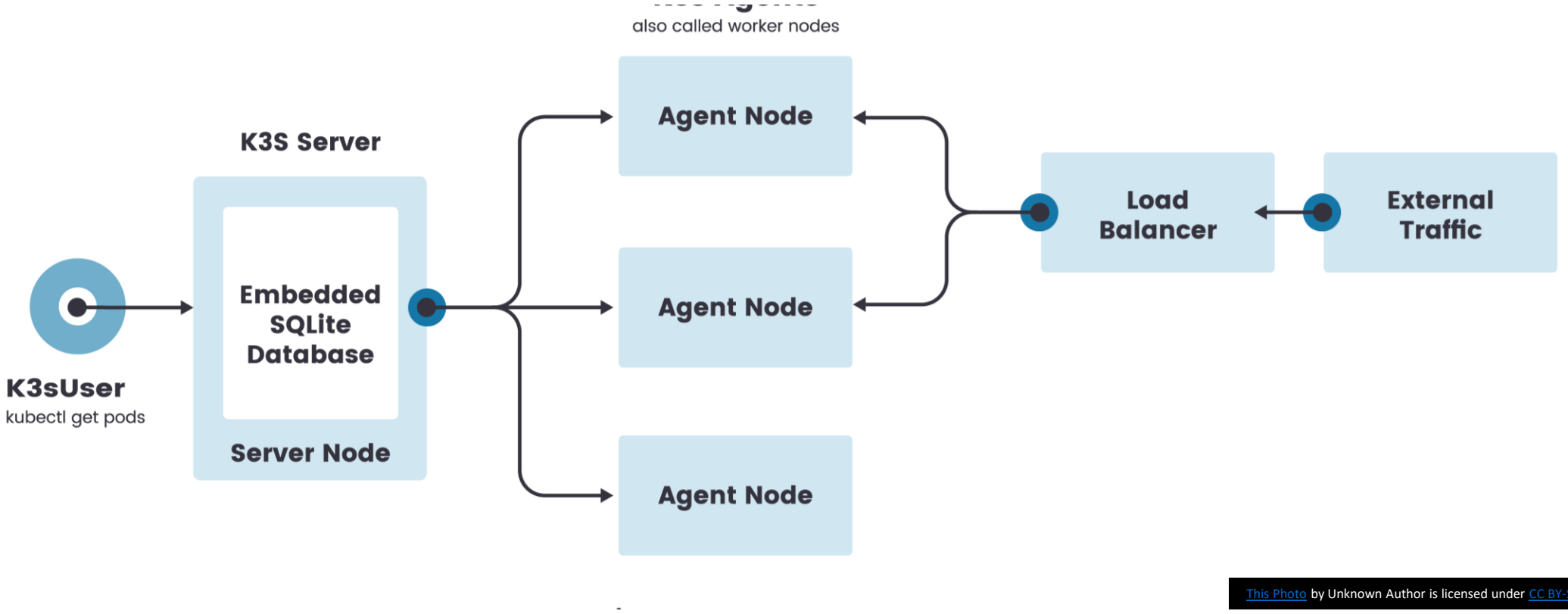
- 进程和线程是由内核进行调度，有CPU时间片的概念，进行抢占式调度
- 协程（用户级线程）对内核透明，是完全由用户自己的程序进行调度的，进行的是协作式调度，需要协程自己主动把控制权转让出去之后，其他协程才能被执行到
- 用于协程间通信的Channel

https://www.cnblogs.com/liang1101/p/7285955.html

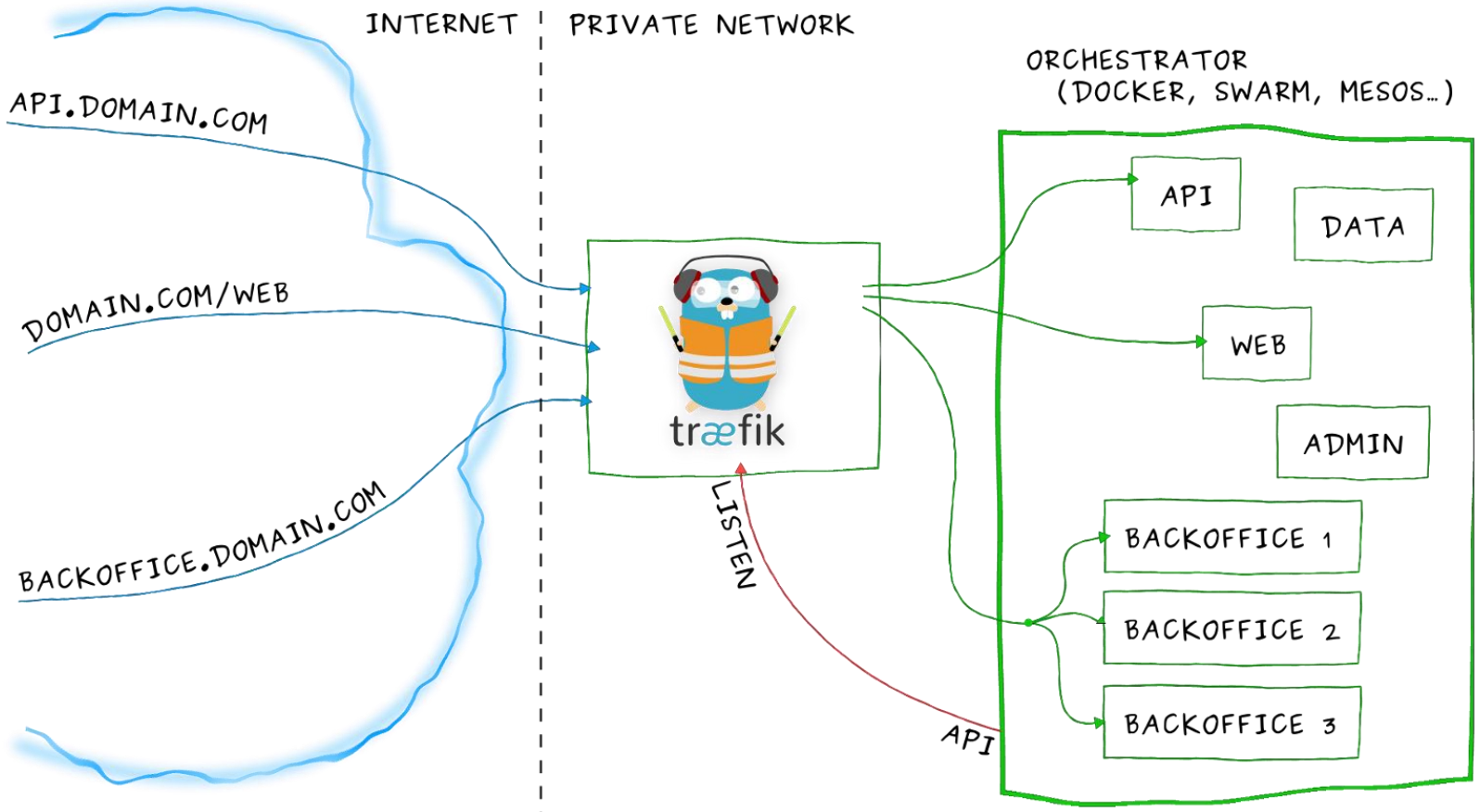
Kubernetes套件



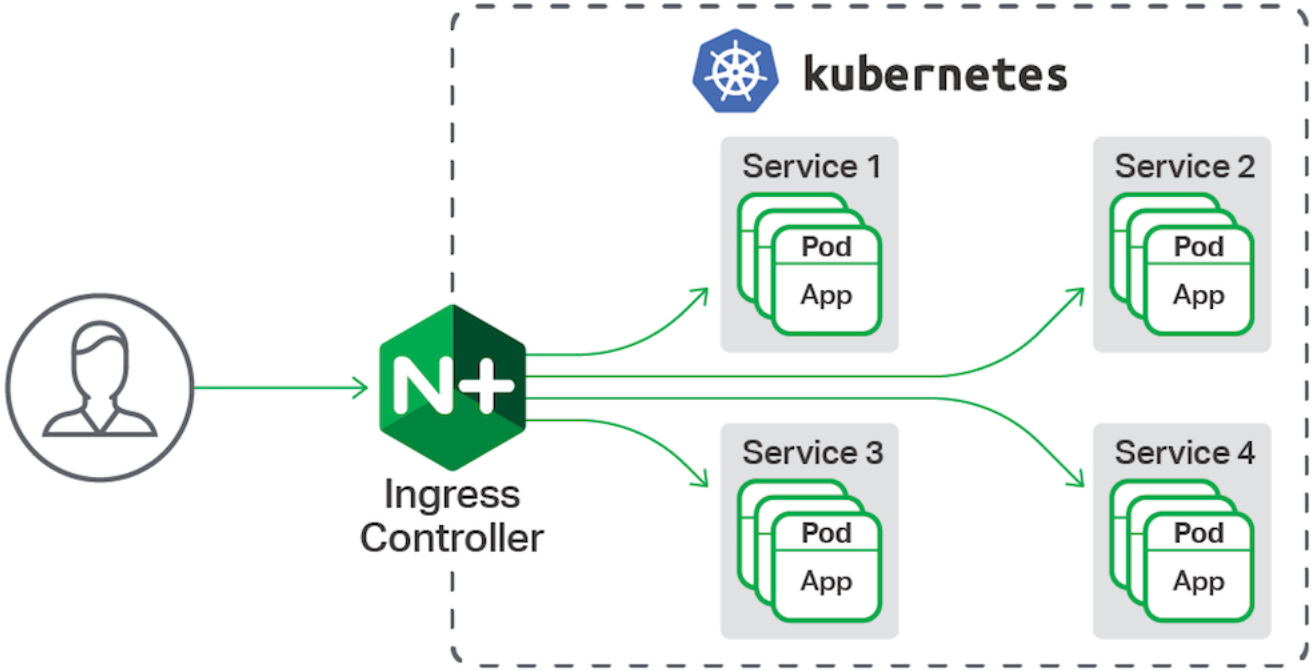
K3s Architecture



Ingress举例： traefik



Ingress举例：nginx



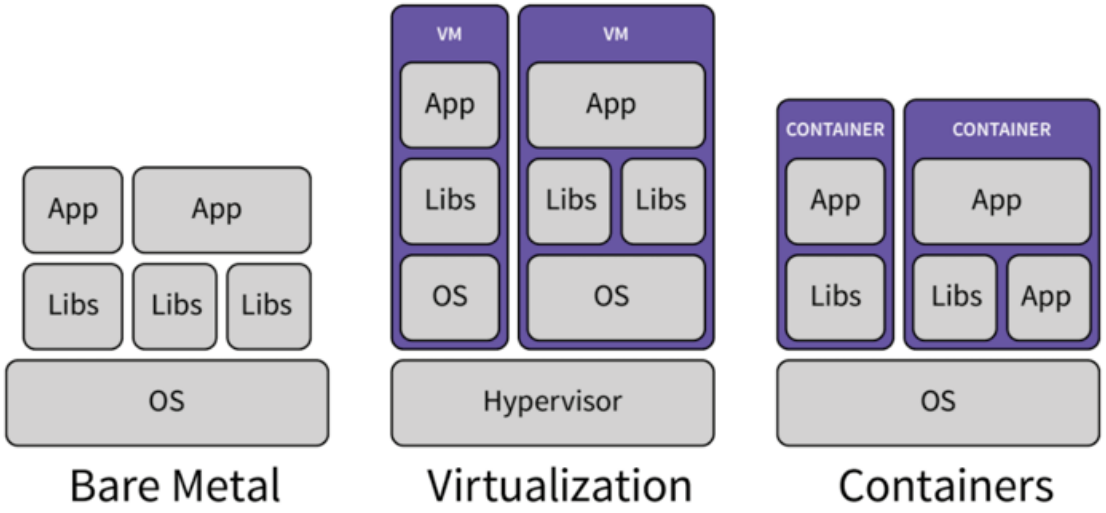
Boltdb (Key-Value Database)

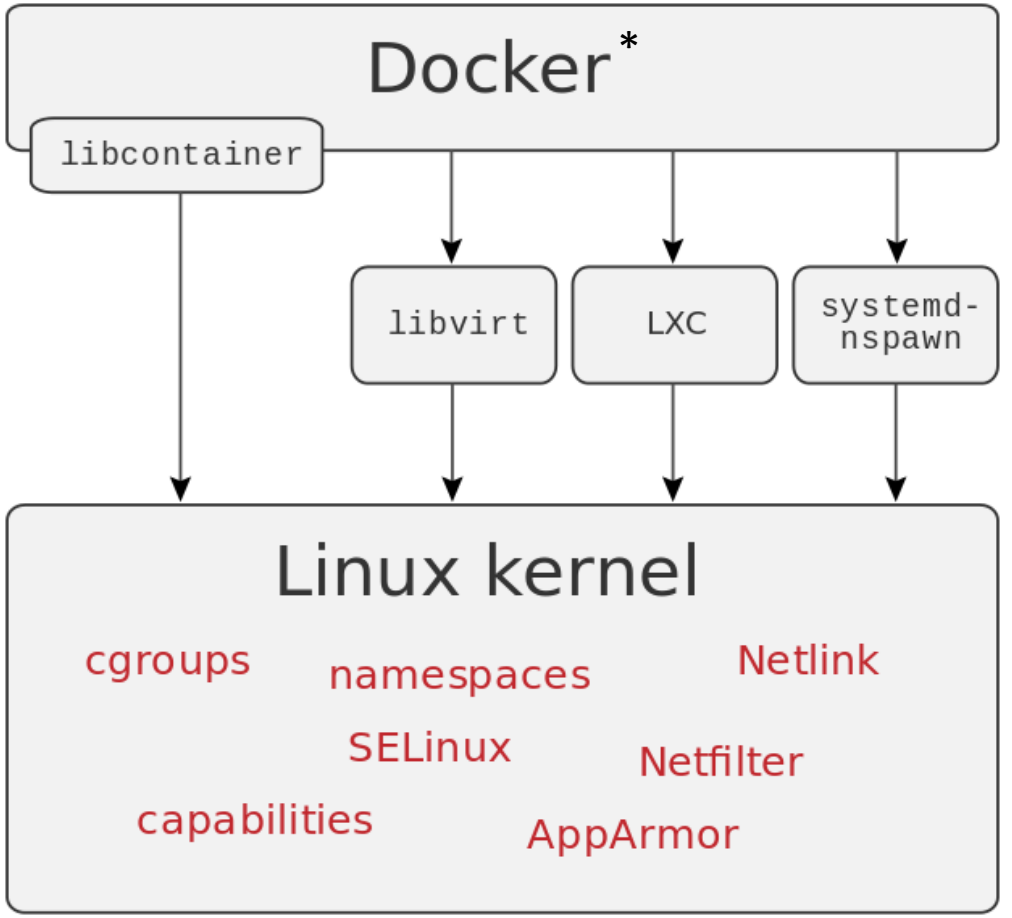
- Bolt 是受 Howard Chu 的 LMDB 项目启发的纯 Go 键/值存储。 该项目的目标是为不需要完整数据库服务器（如 Postgres 或 MySQL）的项目提供简单、快速且可靠的数据库。

- 由于 Bolt 旨在用作此类低级功能，因此简单性是关键。 API 会很小，只专注于获取值和设置值。 而已。

https://db-engines.com/en/system/BoltDB%3BRedis

Container Technology





*Other names or brands may be claimed by the property of others

Image Source and Credits: https://delftswa.github.io/chapters/docker/

Cloud2PC计算和云计算的异同

- Cloud2PC的原则和理念
- 编排和部署在云计算和客户端的异同

The Principle of Architecture

Best client experience for cloud services by harnessing …

…The Power Of Cloud…

- Service Scale and Reach
- Cross-device Experience
- Collaboration and Sharing
- Subscriptions and Monetization
- Deployment/Manageability
- Security

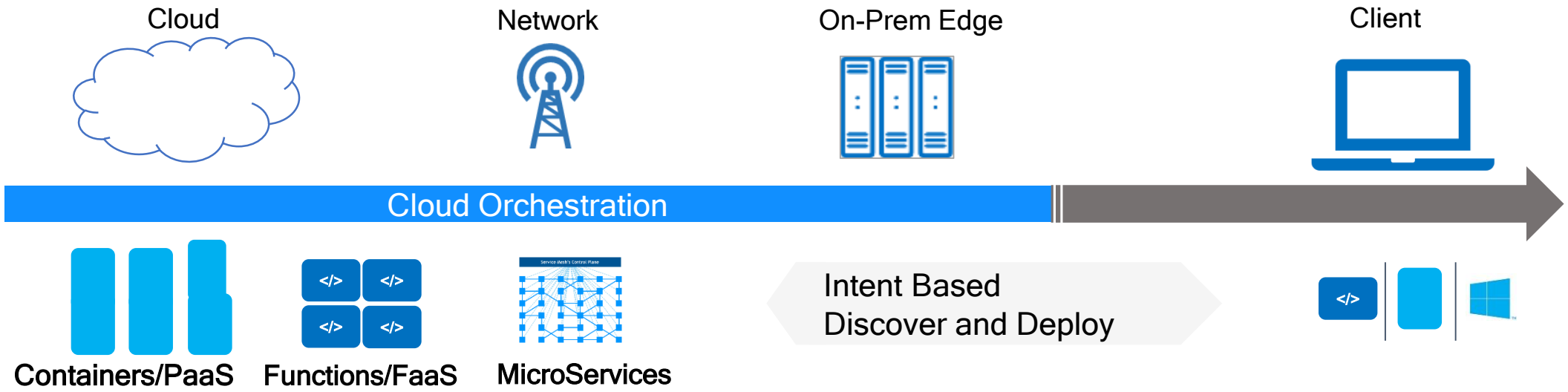
…and Experience Of Client

- Responsiveness
- Offline Use
- Data Privacy and Control
- Lower Cost and Less Data Plan Use
- Human Interactivity and Context
- Form Factors

…We Need to…

- Extend cloud services to use client capabilities
- Shift workloads to where they run best
- Write once and run everywhere for cloud developers

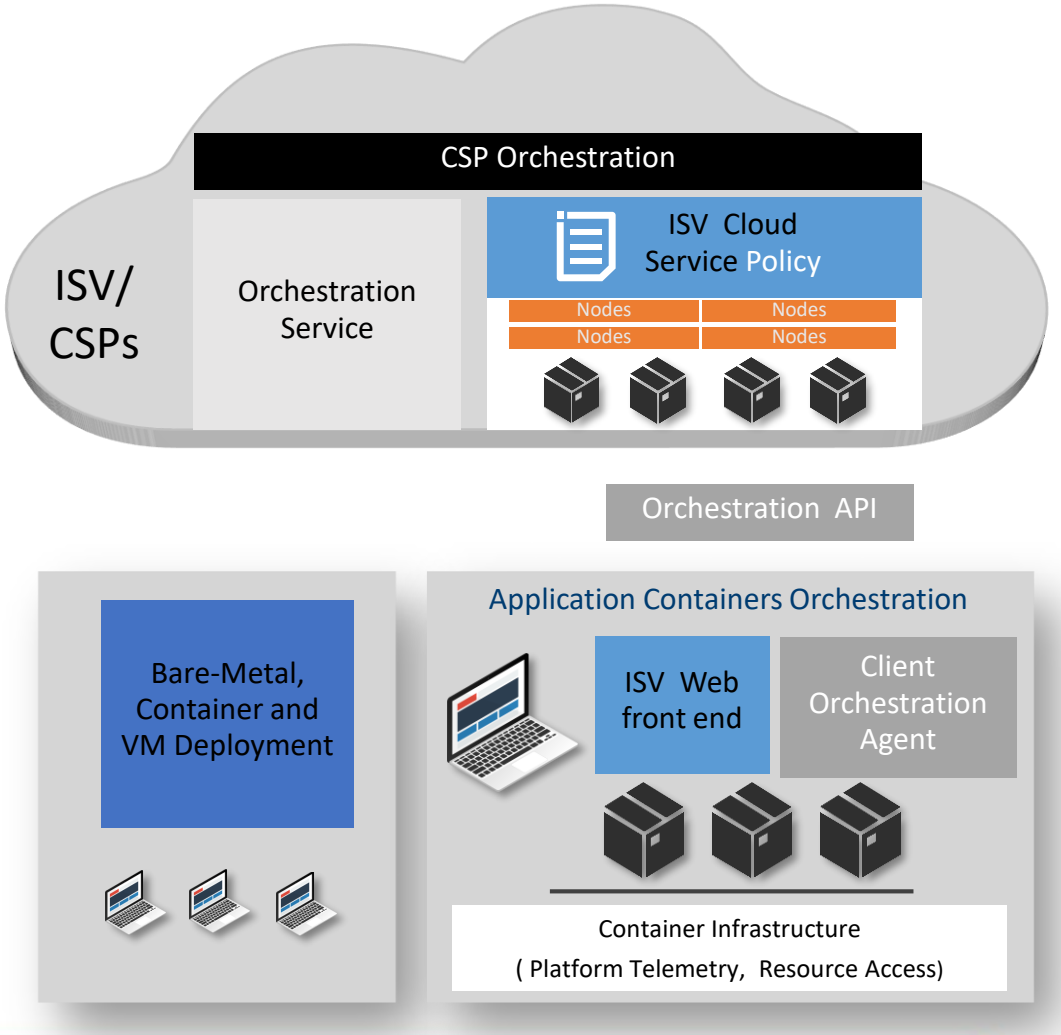
Cloud Client Orchestration



- ✓ Build on Cloud-Preferred Orchestration Frameworks – Standards and open-source reference
- ✓ Deploy Application Containers to edge/clients for best experience

- Dynamically discover **client capabilities**
- Dynamically discover E2E network telemetry, priority classification, and QOS
- Complement cloud's orchestration frameworks with client's
  - to handle **intermittently connected** clients and nodes
  - to handle dynamically changing IP configurations
  - to handle firewalls, VPNs and NAT layers

Distributed Intelligence



VALUE PROPOSITION

- Better User Experience
- Reduced TCO
- Ease of use for developers

FUNCTIONAL DEMO

- Dynamic Orchestration working on Windows and Linux client platforms
- Heavy-lifting done by Orchestrator, easy to configure by ISV
- Local container delivers better experience

Provided by Arvind Kumer@Intel

编排和部署在云计算和客户端的异同

	云原生	Cloud2PC
编排单元	粗粒度为主，主要以节点为单位 CPU cores Memory Extended resources	细粒度为主，单节点内的资源分配 CPU cores Memory Extended resources
OS	Linux为主	Windows 为主
网络	跨结点	结点内
主导	云端	客户端
存储	云为主	本地为主
隐私保护	云端隔离	客户端隔离

Cloud2PC 关键技术

Cloud2PC 关键技术 – 网络

- 网络问题的本质是把“对于外部访问的流量转到本地的容器中”
- 在单节点可用的ingress是nginx, traefik等等
- 在客户端中，需要配合DNS， IPTable以及ingress来达到流量转发的目的

Cloud2PC 关键技术 – 计算

- 云端的主流OS是Linux， 客户端的主流OS是windows
- 不同OS之间的整合，最好的方法是使用虚拟机
  - 以windows为例，最主流的虚拟机是WSL
  - 所以在Windows上，计算主要是在WSL内部进行的
  - 虚拟机guestOS是Linux，Linux容器就可以在GuestOS上无缝的进行

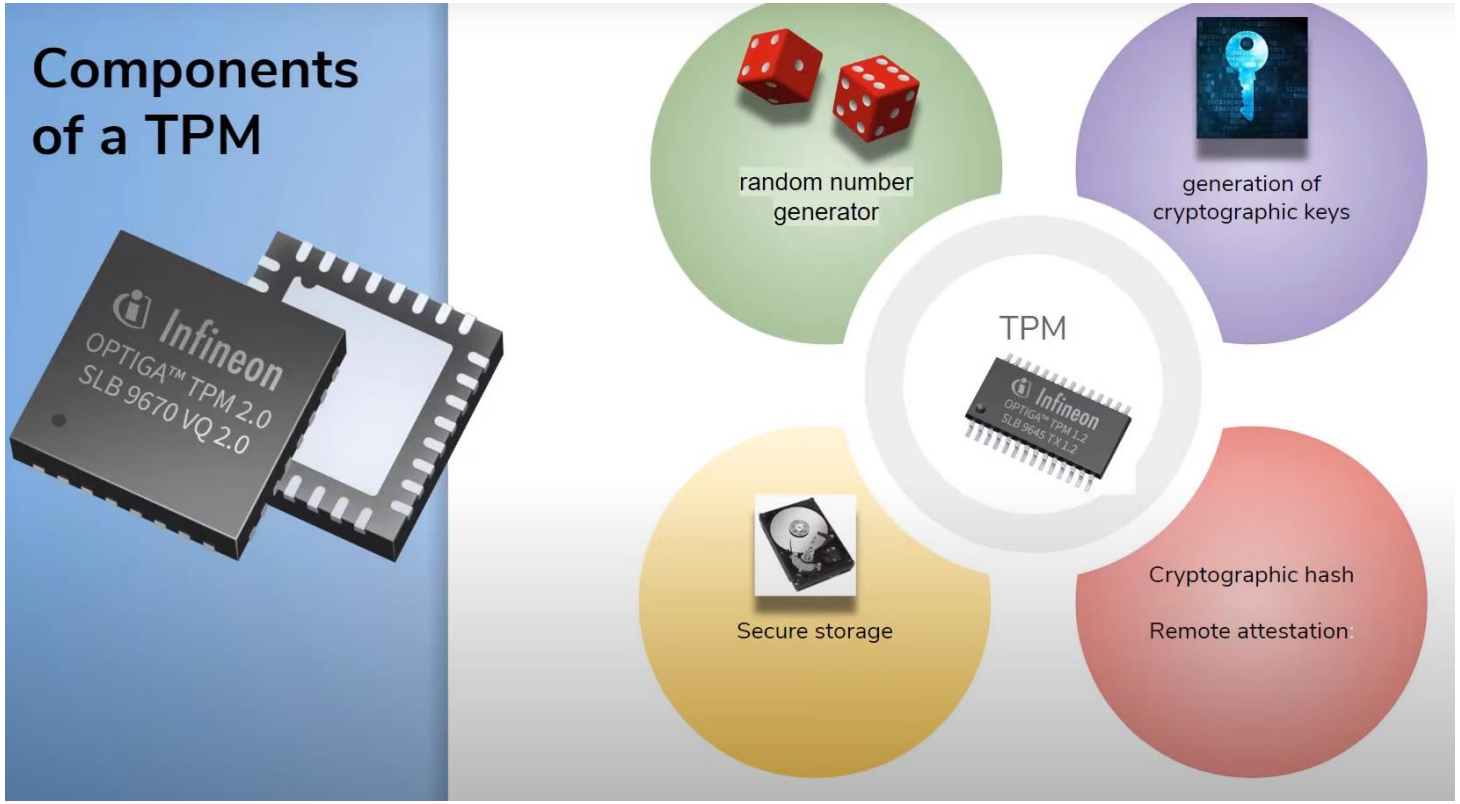
https://www.intel.com/content/www/us/en/events/on365/cloud-containers-client-platforms.html

Cloud2PC关键性能指标

- Latency (比较数据传到云端处理后再发送回来)
- CPU 占用率
- Memory
- Cgroups的使用
- Limit/request, LimitRange, Resource quota的使用

Cloud2PC -- 安全探索

- TLS & 安全基础
- TPM to protect
- TDX possibilities
- The magic localhost



Cloud2PC 更多展望

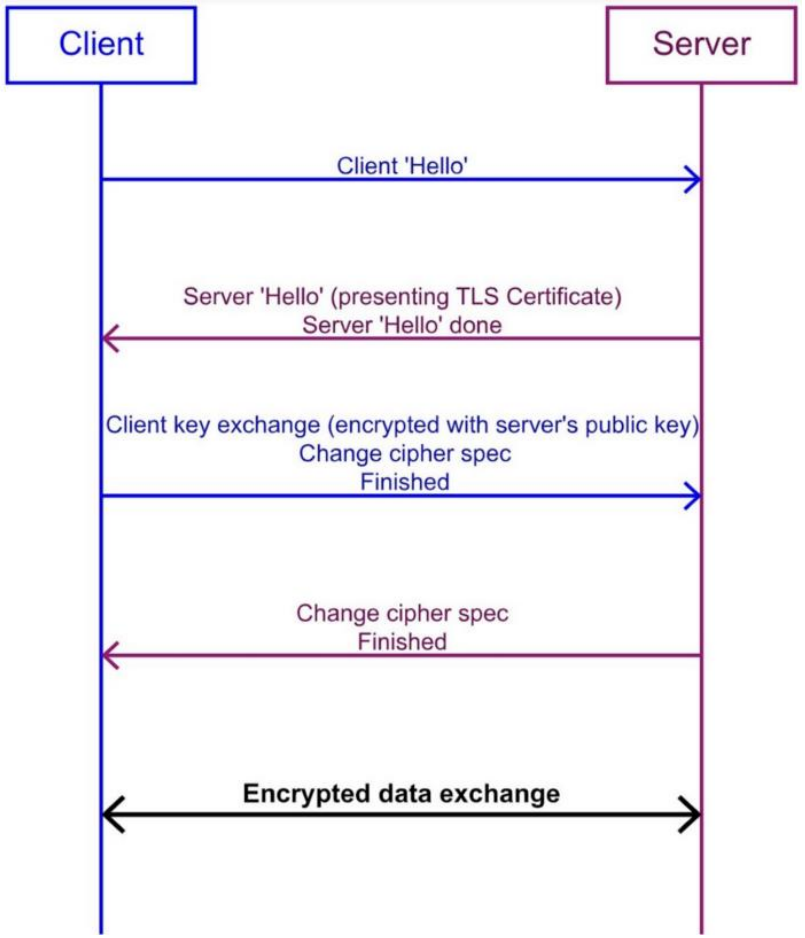
- WSL默认支持了systemd，这让Cloud2PC的自动启动更灵活
- 存储的隔离让用户的隐私得到了更好的保护
- 需要共同创建Cloud2PC的生态
- 分布式智能的可能
- 由PC组成的Hybrid Cluster
- 联邦学习等等

THANKS

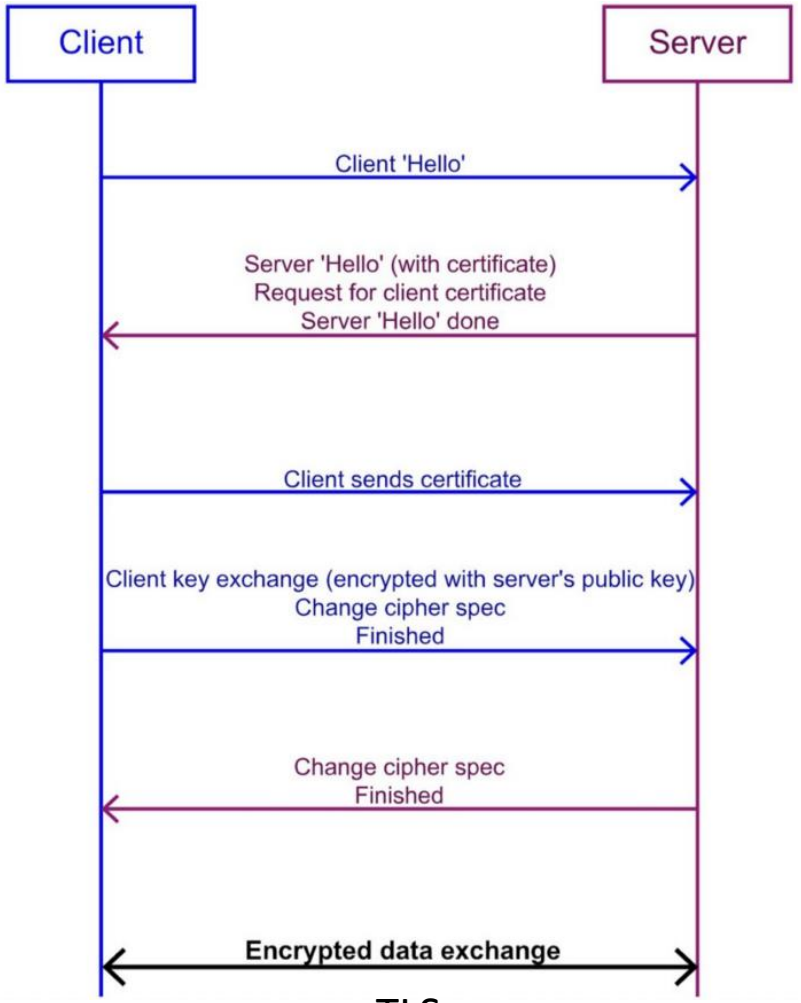
SSL vs TLS

SSL (Secure Socket Layer)	TLS (Transport Layer Security)
It was developed by Netscape.	It was developed by Internet Engineering Taskforce (IETF).
SSL was first released in 1995 (SSL 2.0).	The first version (TLS 1.0) was released in 1999.
SSL's three versions include SSL 1.0, SSL 2.0 and SSL 3.0.	TLS's four versions include TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3.
All versions have been deprecated due to security flaws.	TLS 1.0 and 1.1 have been deprecated since 2020. TLS 1.2 and 1.3 are in use.
It uses a port to set up explicit connection.	It uses protocol to set up implicit connection.
SSL uses Message Authentication Code (MAC) to authenticate messages.	TLS uses HMAC (Hash-based Message Authentication Code) to authenticate messages.

TLS vs mTLS

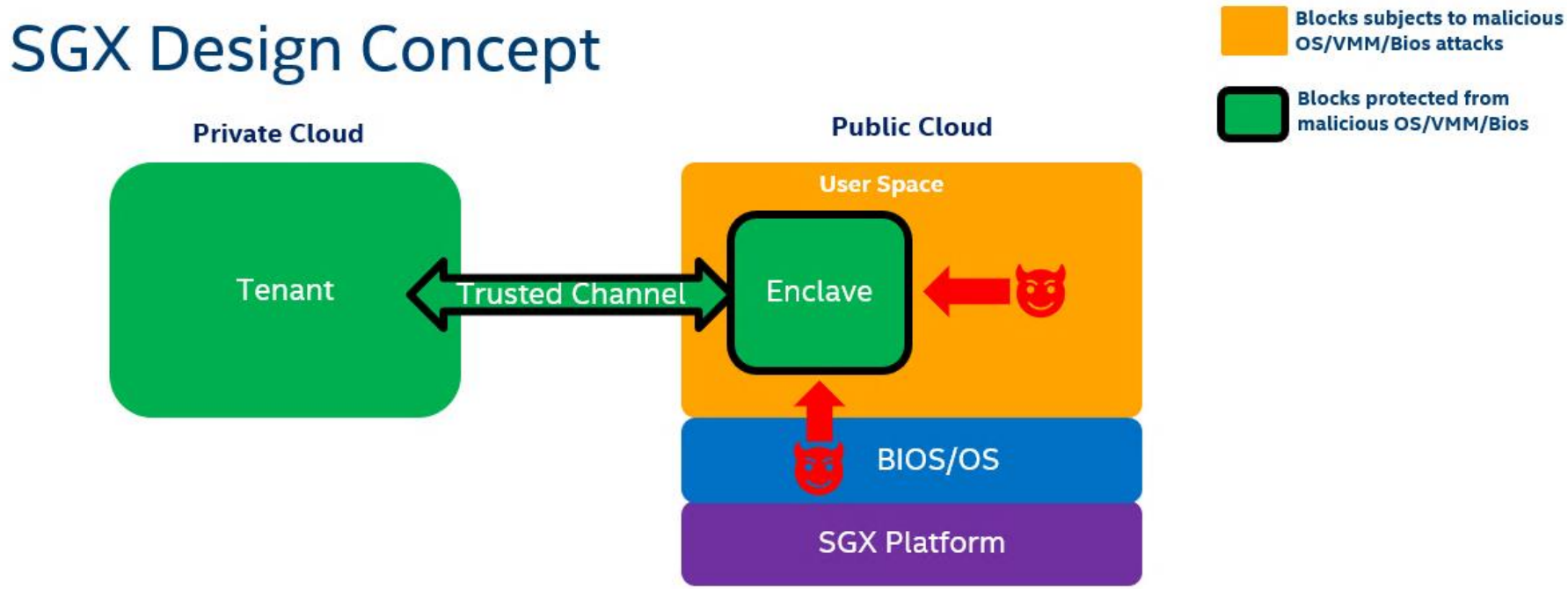


TLS



mTLS

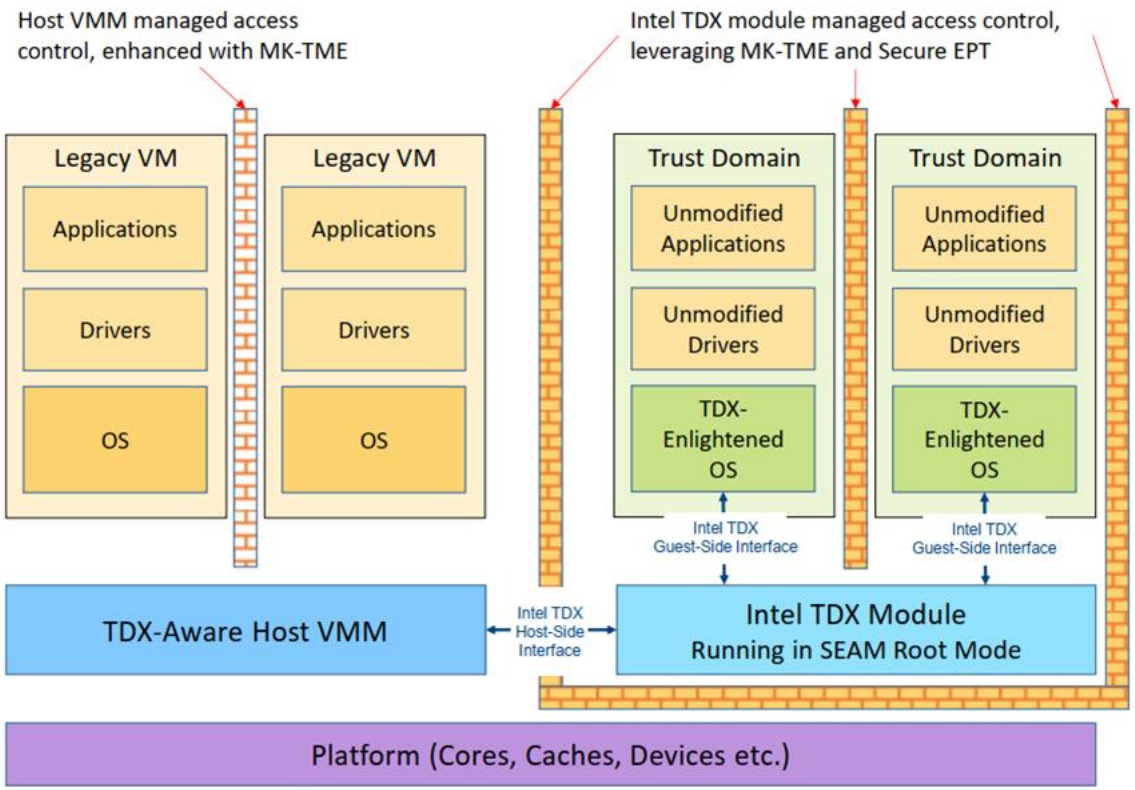
SGX



- Data confidentiality and integrity protection in untrusted environment.
- Protect against attacks from malicious OS, VMM, BIOS.
- Use scenarios:
  Transfer workload from private cloud to public cloud

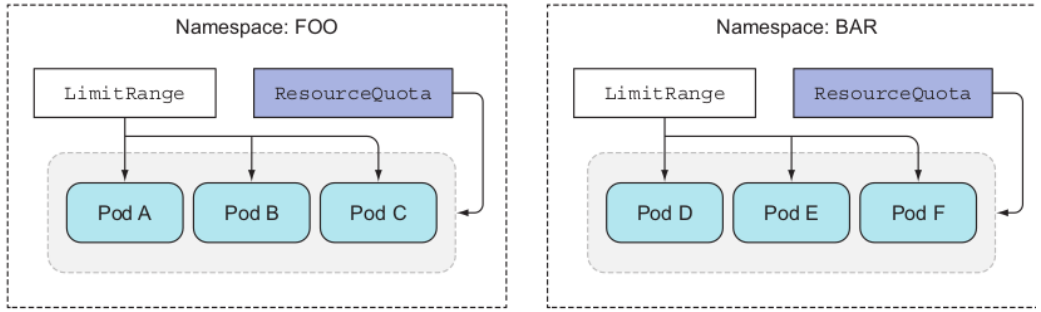
Check with
fan.du@intel.com

TDX

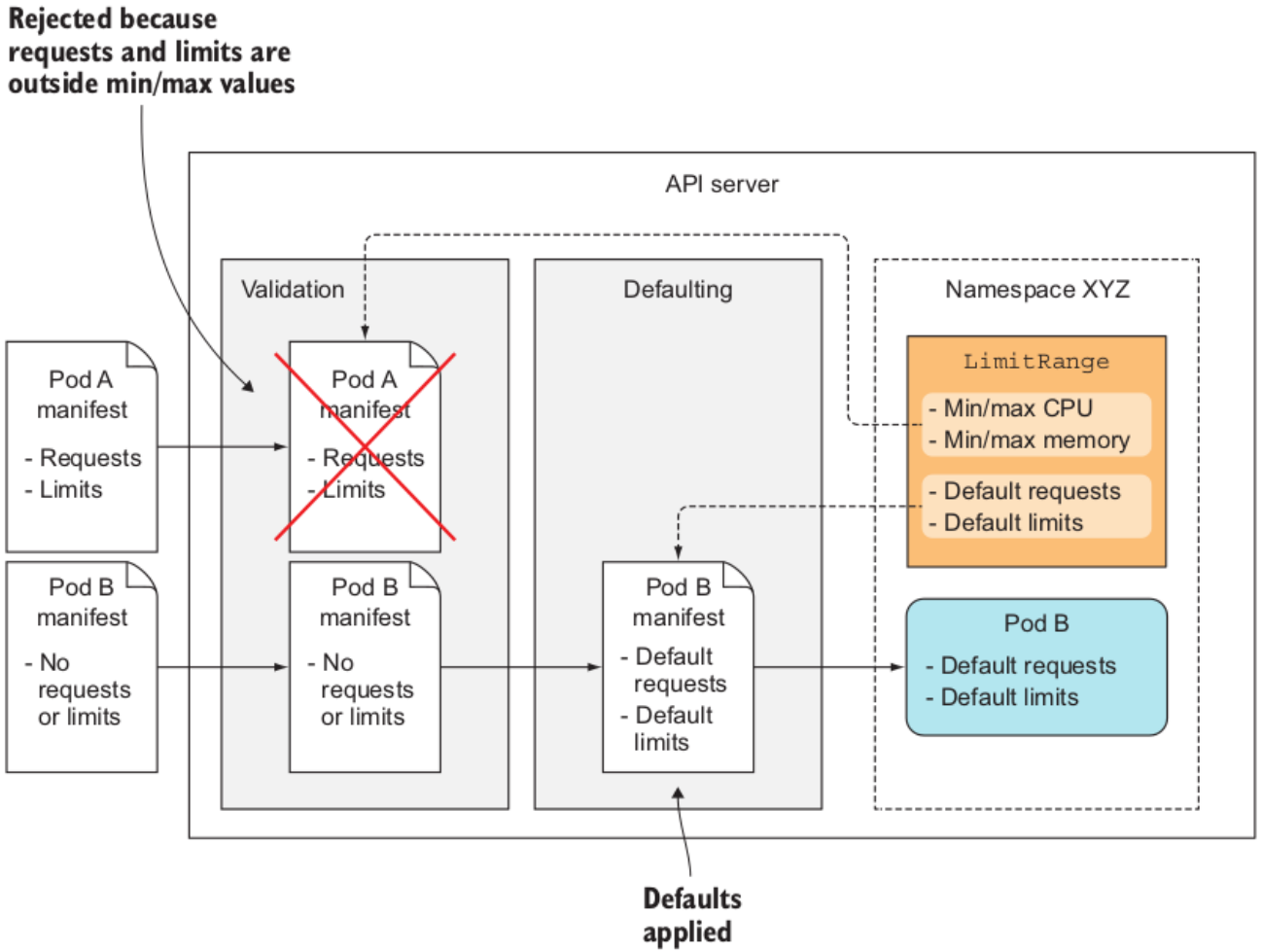


[Intel® Trust Domain Extensions]

Kubernetes中的Limit, Request, LimitRange, ResourceQuota



ResourceQuota is
the total amount of memory and CPU that can be used by all Containers running in a namespace
LimitRangeis
if ... the Container does not specify its own CPU limit, then the Container is assigned the default CPU limit



通信加速—情景DPDK

- APP和被使用的服务再同一个pod内

小结
方案名称：用户态CNI（openEuler高性能网络sig已经在设计/开发这款软件：Gazelle CNI）

用户态CNI可以基于DPDK、AF_XDP两种技术实现，两种技术各有优劣，互补关系，所以需要考虑同时支持DPDK、AF_XDP两种技术。

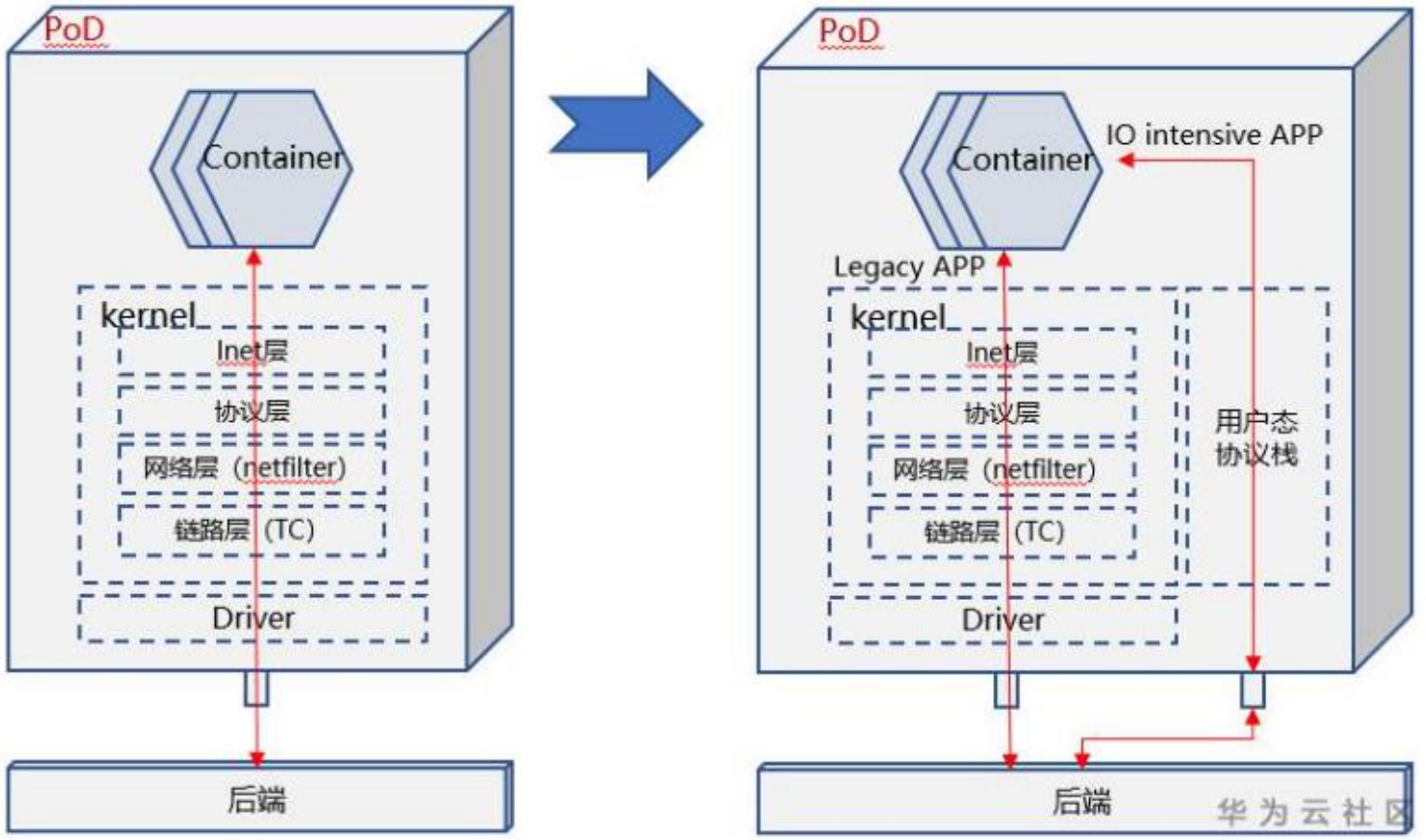
在控制面/数据面分离、Legacy/IO intensive APP并存、多租户隔离等场景，用户态CNI存在应用价值。

PoD内通信加速

PoD内主要数据路径集中在内核协议栈，内核协议栈本身一直在研究并持续的性能优化，但是其性能还是远不如DPDK、AF_XDP这类用户态协议栈。

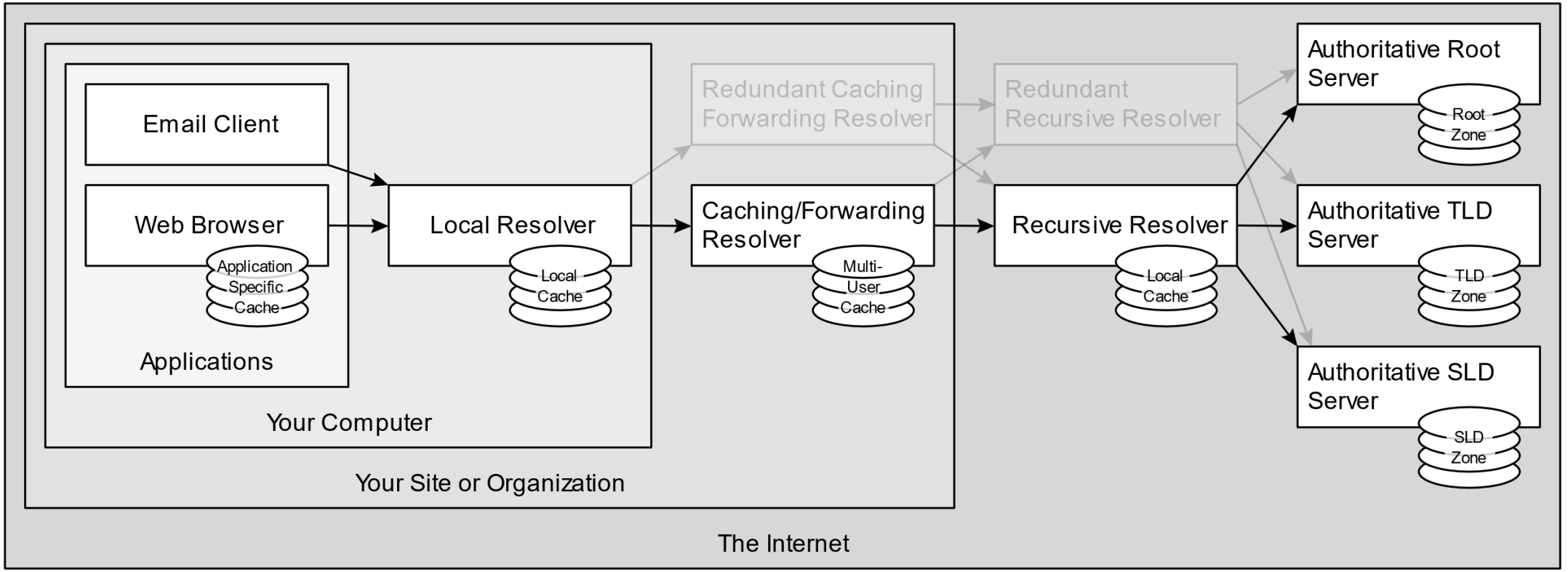
提速方案

PoD内引入数据多平面，当PoD内存在Legacy/IO intensive APP并存时，后者可以走高性能数据面，提速的同时可以避免对其他APP的影响。

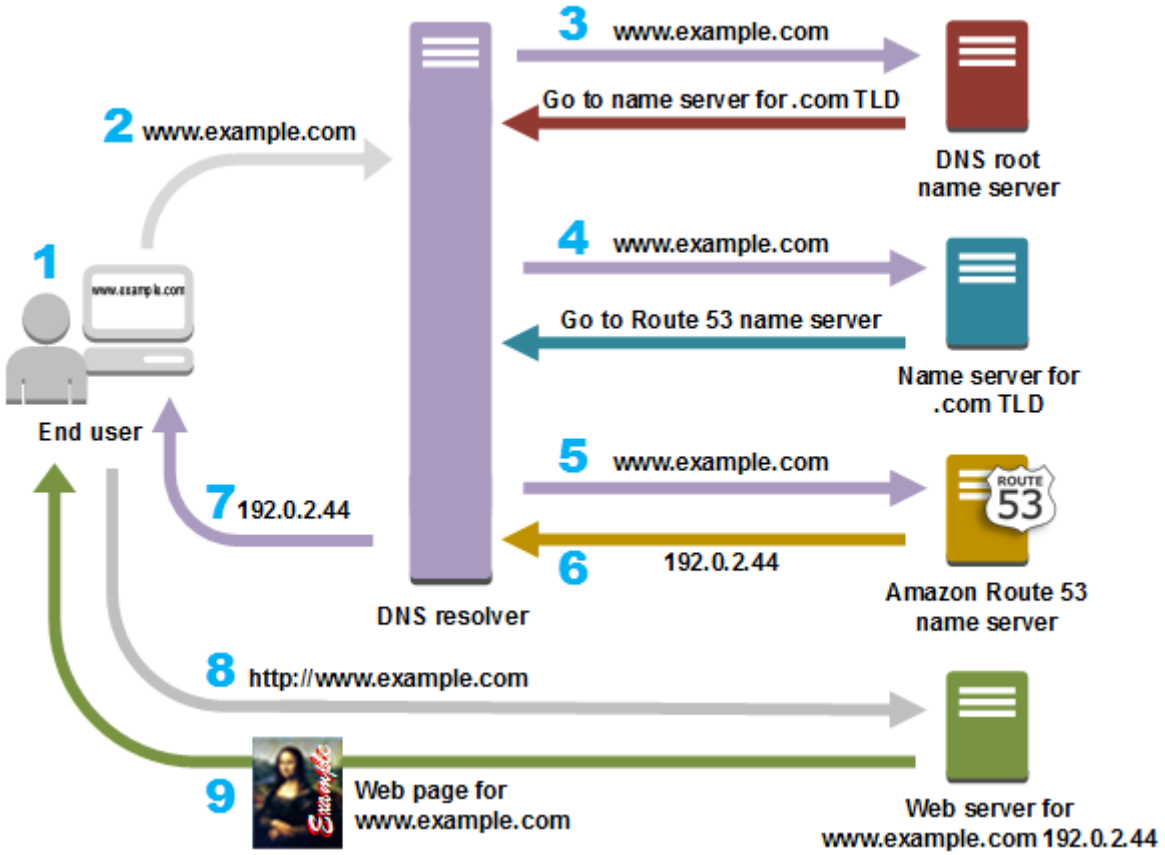


https://bbs.huaweicloud.com/forum/forum.php?mod=viewthread&tid=95490

DNS basics


Email Client
Web Browser
Application Specific Cache
Applications
Your Computer
Local Resolver
Local Cache
Redundant Caching Forwarding Resolver
Caching/Forwarding Resolver
Multi-User Cache
Your Site or Organization
Redundant Recursive Resolver
Recursive Resolver
Local Cache
Authoritative Root Server
Root Zone
Authoritative TLD Server
TLD Zone
Authoritative SLD Server
SLD Zone
The Internet

DNS basics



aws: what is dns.