



51CTO 传媒

WOT 2015 互联网运维与开发者大会

■ 2015年04月10日-11日 ■ 北京珠三角JW万豪酒店

轻松玩转云端存储

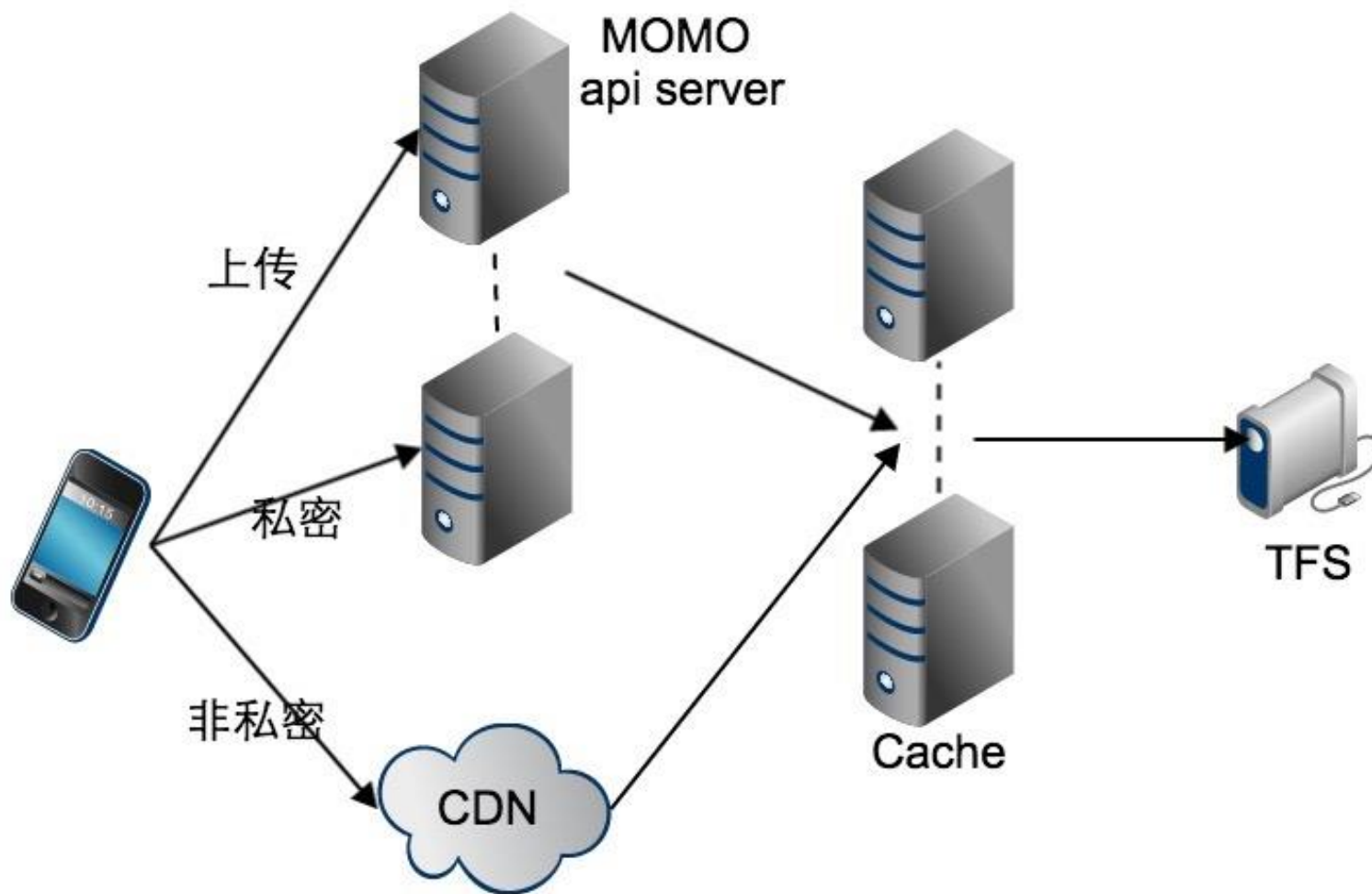
—轻运维时代的云存储平台

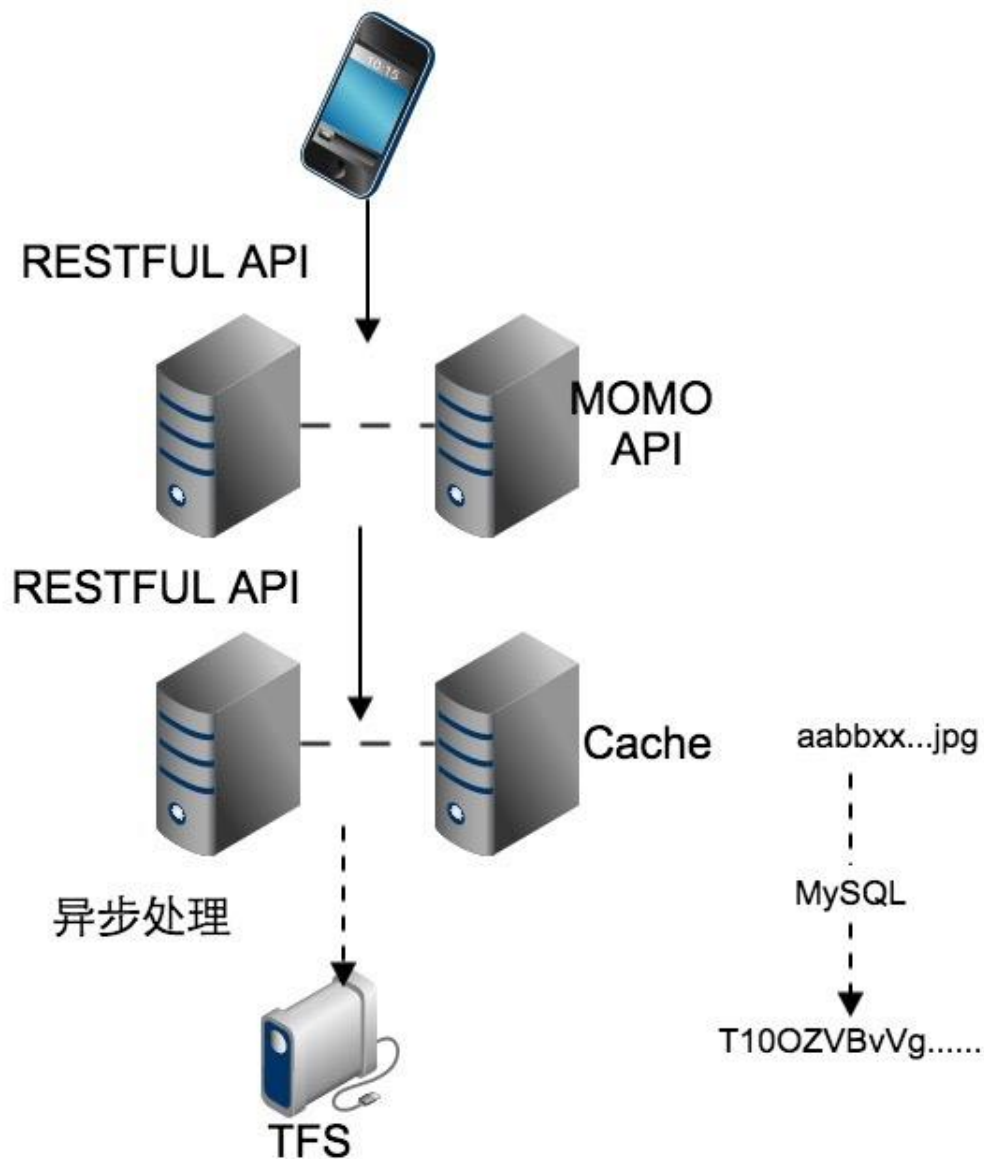
陈湛翀

陌陌存储平台

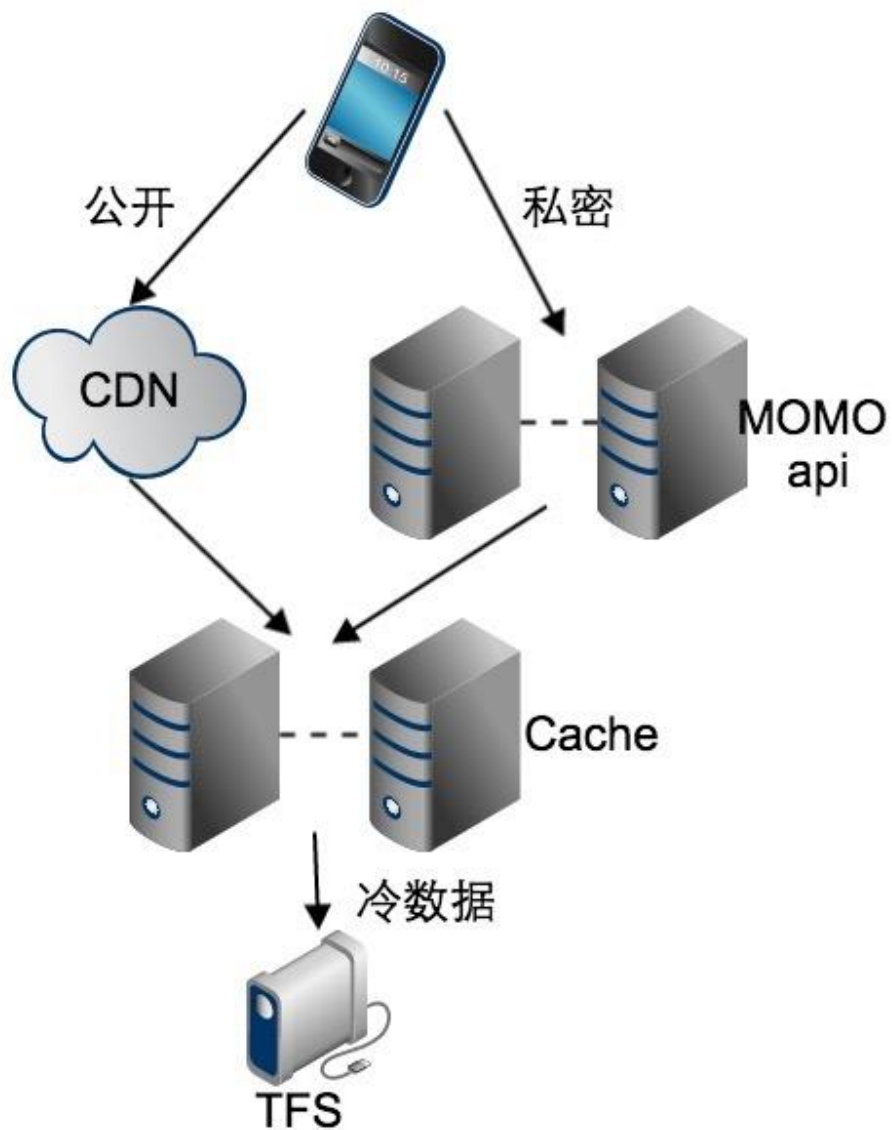
- 图片、语音、视频等
- 近 1P 的存储容量
- 超过 300 亿的文件数
- 每天超过 1T 的增速

自建存储平台





- 用户首先上传到 MOMO 的 API Server，进行业务逻辑处理
- API 层再上传到存储平台的 Cache 层，存储平台与业务逻辑分离
- Cache 层异步上传到 TFS，并作一定的图片处理
- 业务文件名与 TFS 的文件名的对应关系使用 MySQL 维护
- Cache 层区分冷热数据存放

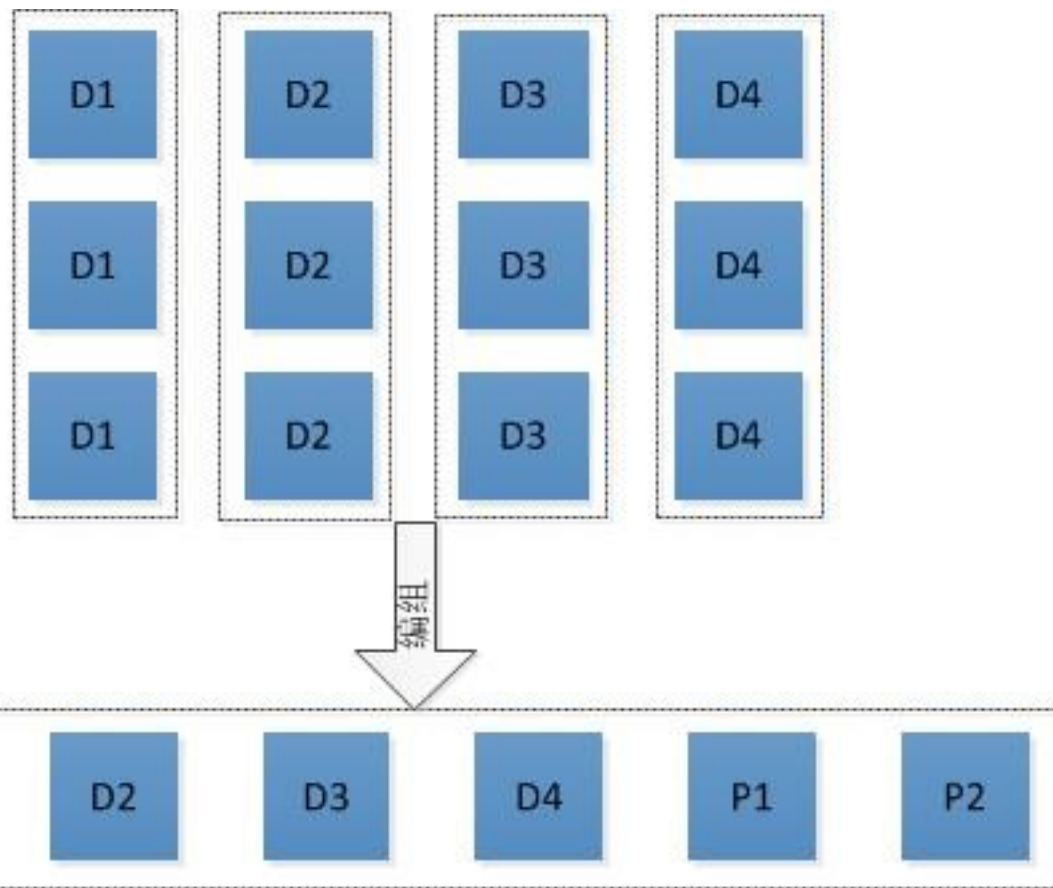


- 可公开的数据和私密的数据不同入口
- Cache 层是提供访问的主要部分
- 冷数据才会穿透到 TFS
- 由于业务基于 LBS，没有热点内容，长尾效应
- Cache 层还支持动态处理数据
- CDN 也是两层缓存

持续优化

- 用 RocksDB 替代 MySQL
- 升级 TFS 以支持[纠删码](#) (erasure code , EC)
- 使用 Ceph 来替代 TFS
- more and more ...

纠删码 Erasure Code



- n 个数据, m 个校验
- 可以容忍 m 个坏盘
- 假设, $n = 28, m = 4$. 使用 1.14 倍的容量, 代替了之前 3 倍的容量, 并且具有更高的可靠性

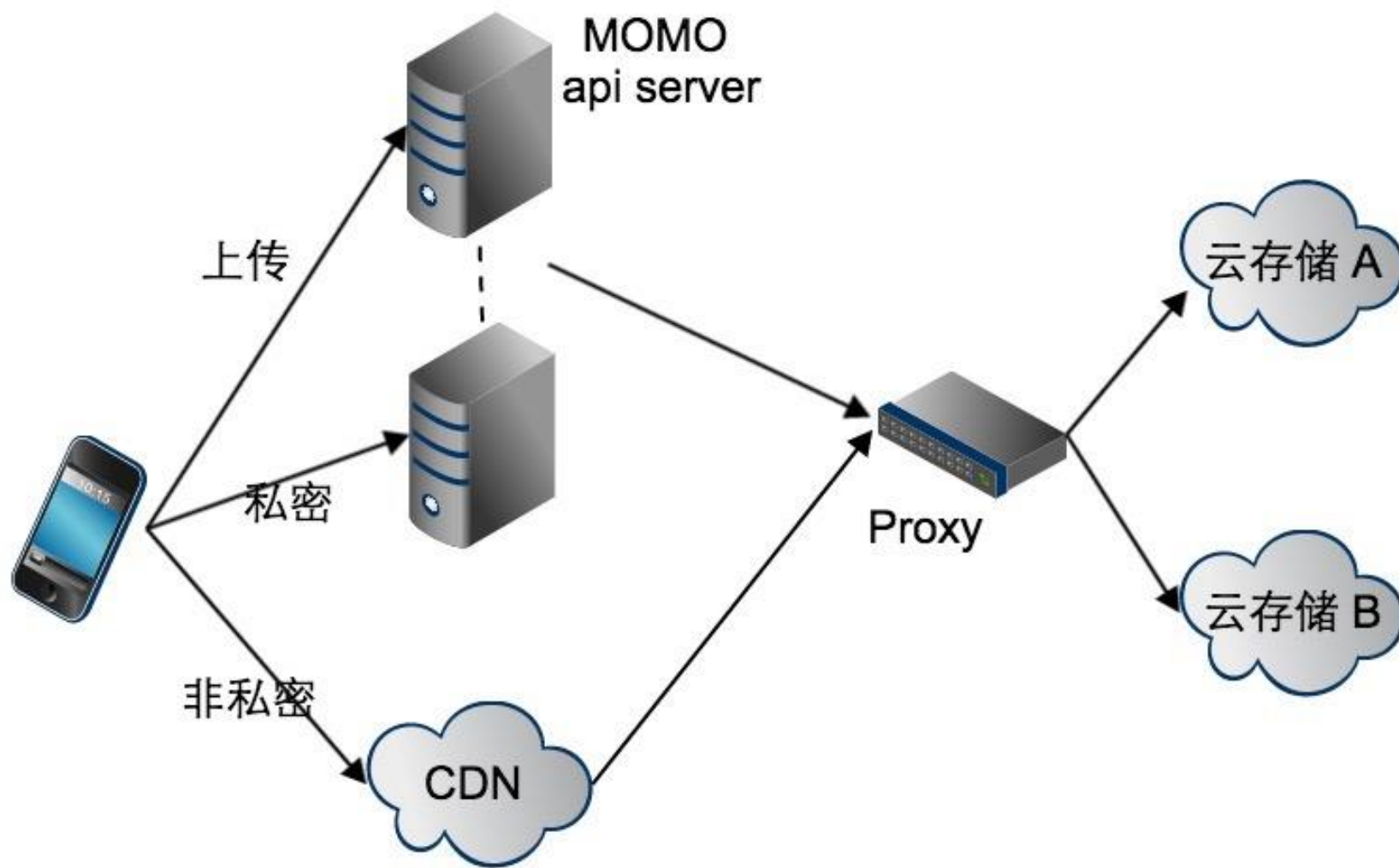
Pros

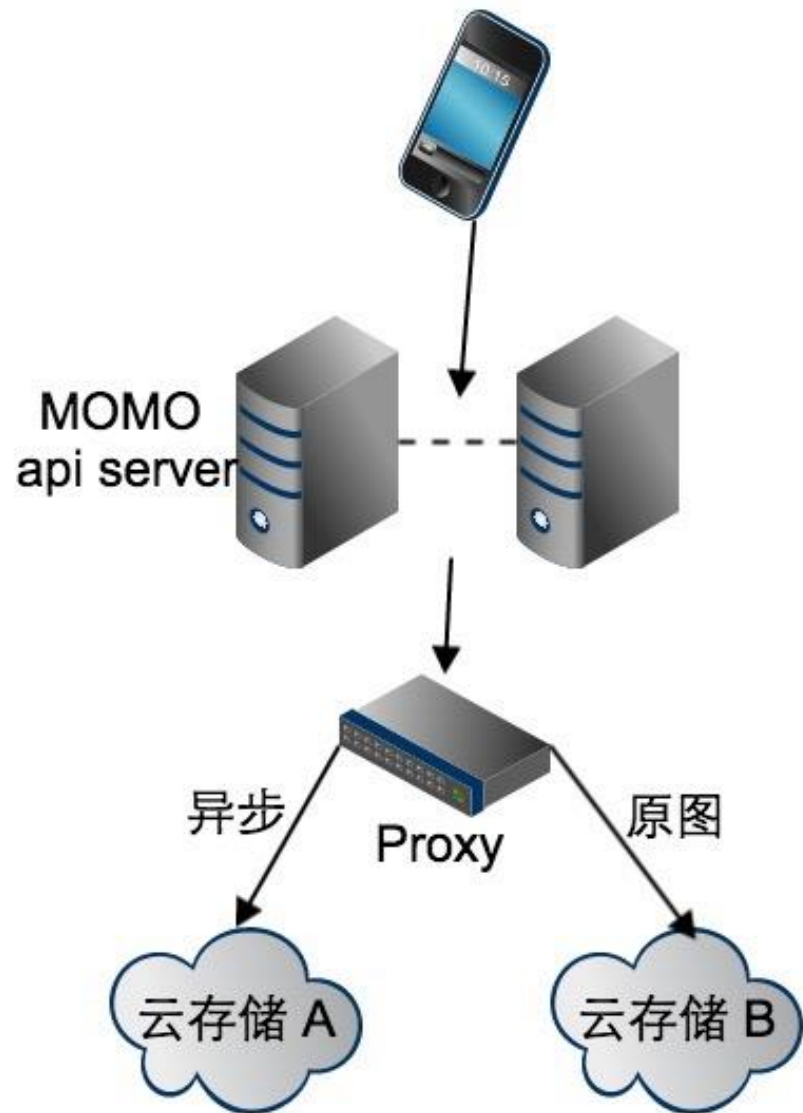
- 数据安全
- 可控性

Cons

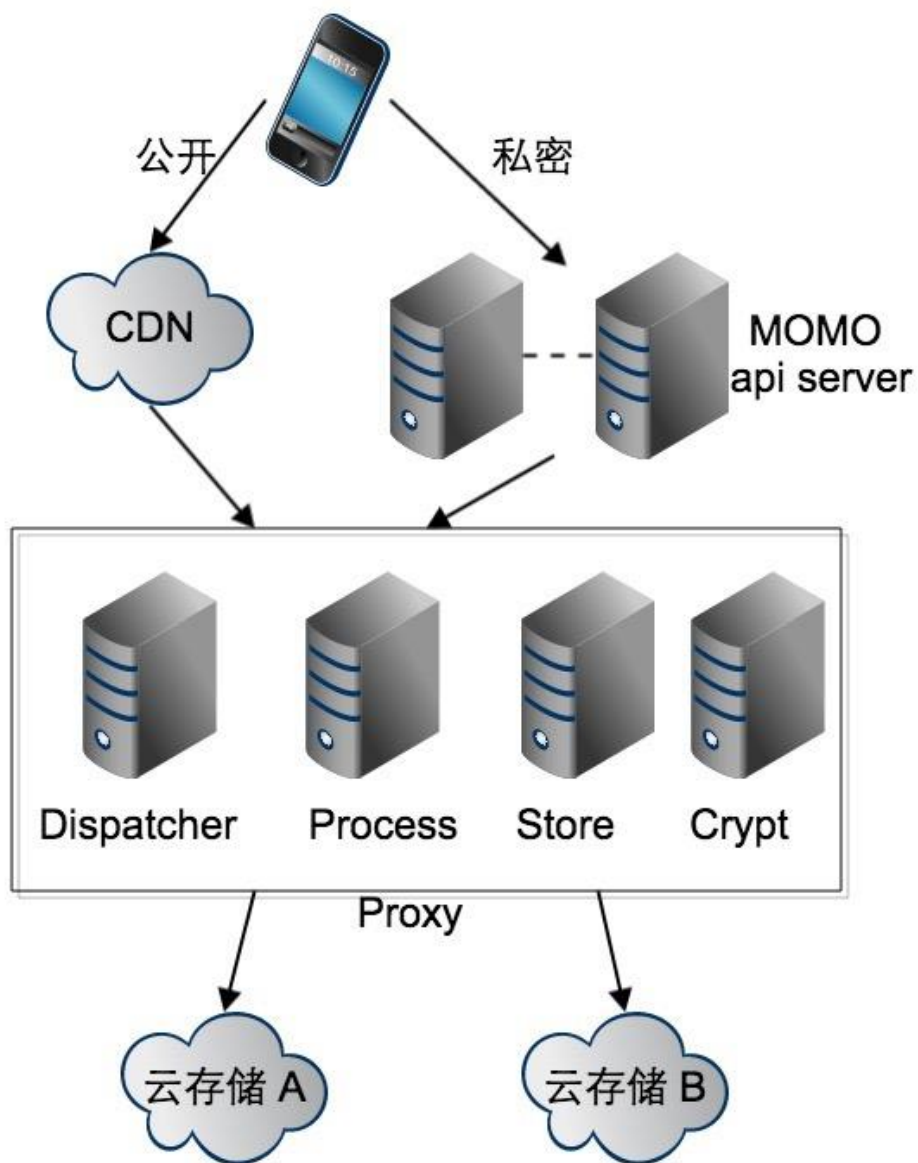
- 硬件成本（机房托管、设备、带宽等）
- 人力成本（软件、硬件）

云存储平台





- 上传过程更轻
- 两个云存储实现高可用
- 异步上传原图到云
- 私密图片通过 GPG 加密后上传到云



- proxy 层存在多个处理 module
- docker 部署各个 module
- dispatcher 负责调度请求
- store 层存放一天的原数据
- crypt 层负责解密从云拉回来的数据
- process 层负责代理到云存储，同时负责动态处理图片
- 优先代理到云存储，云存储自带丰富的多媒体处理功能。当原数据还没有异步上传到云的时候，才会读取 store 层的原数据

GPG 加密

- 非对称加密
- 公钥可以存放在客户端，或云服务商
- 私钥由服务器端保管

Python & Golang Example



Python Code

```
from PIL import Image

im = Image.open("origin.jpg")
width, height = im.size
delta = abs(width - height)
if width <= height:
    box = (0, delta / 2, width, width + delta / 2)
else:
    box = (delta / 2, 0, height + delta / 2, height)
out = im.crop(box).resize((150, 150))
out.save("new.webp", "webp", quality=80)
```

Golang Code

```
import "gopkg.com/magick.v1"
```

```
...
```

```
    im, _ := magick.DecodeFile("origin.jpg")
    delta := int(math.Abs(float64(im.Width() - im.Height())))
    var box magick.Rect
    if im.Width() <= im.Height() {
        box = magick.Rect{0, delta / 2, uint(im.Width()), uint(im.Width())}
    } else {
        box = magick.Rect{delta / 2, 0, uint(im.Height()), uint(im.Height())}
    }
    temp, _ := im.Crop(box)
    out, _ := temp.Resize(150, 150, magick.FBox)
    outputFile, _ := os.Create("new.webp")
    defer outputFile.Close()
    info := magick.NewInfo()
    info.SetFormat("webp")
    info.SetQuality(80)
    out.Encode(outputFile, info)
```

```
...
```

Python & Golang Example Performance

- Intel(R) Xeon(R) CPU E5-2420 0 @ 1.90GHz
- Python 2.6.6 , go 1.4.2
- Python 处理 1000 张图片需时 1m45s
- Go 处理 1000 张图片需时 1m

Golang vs. Python

- 部署更简单
- 表达能力不差
- 标准库较完善
- 性能好

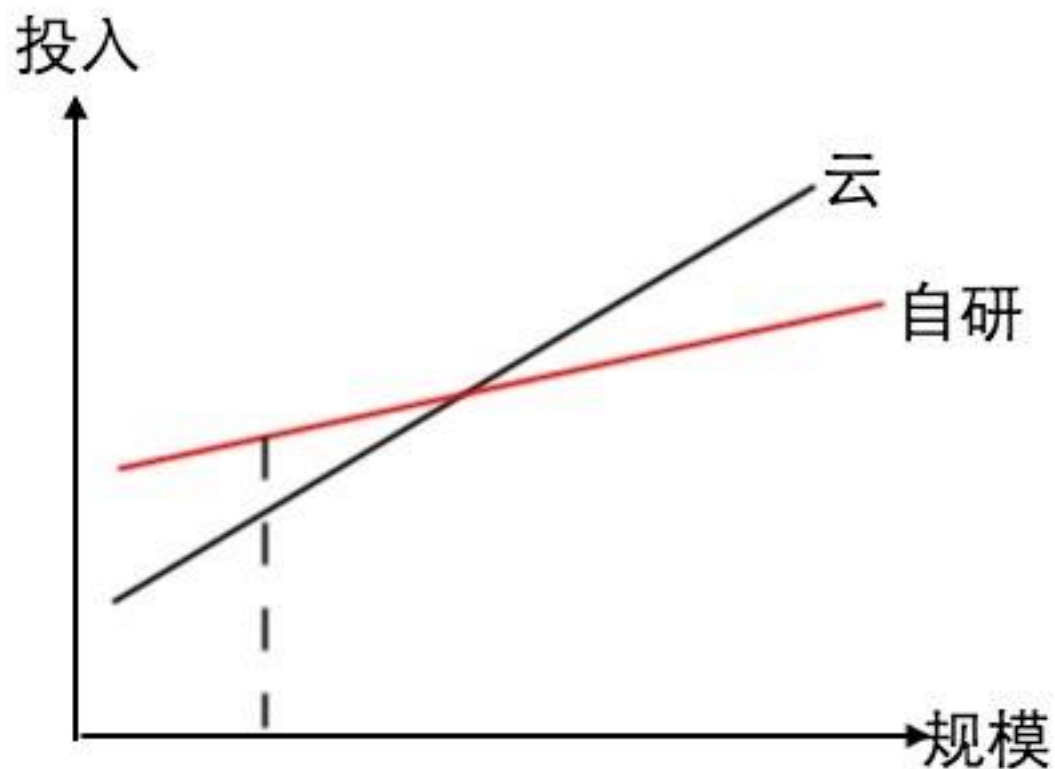
Python vs. Golang

- 更强的表达能力
- 丰富的第三方库
- 开发效率高

Python & Golang

- Python: 开发效率
- Go: 性能

云 vs. 自建



云与自研的经济效益对比

云 vs. 自建

- 成本大幅降低
- 业务响应能力大幅提升

谢谢大家