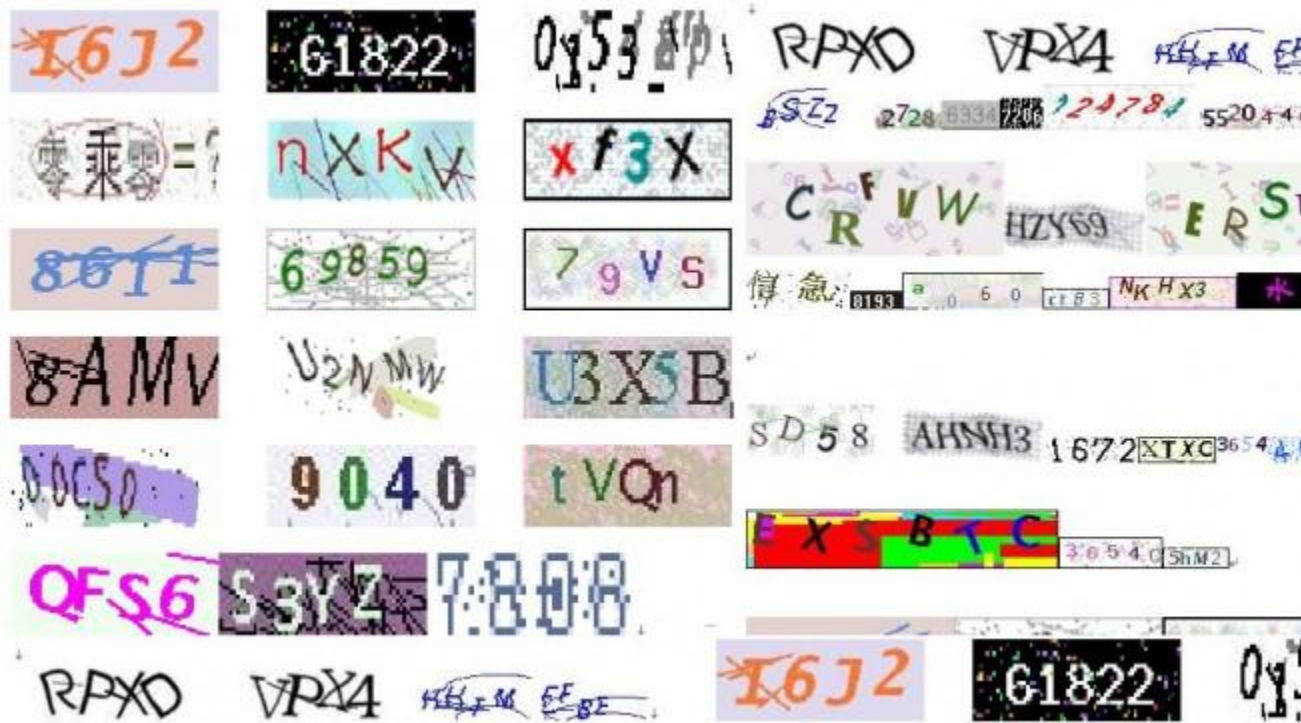


# 极验技术架构 演变之路

**陈渊** 极验验证服务端架构负责人

# 千奇百怪的验证码



您现在的位置: 客运首页 > 登录

温馨提示:

- 1、12306.cn网站自3月16日起启用图形验证码
- 2、12306.cn网站每日07:00~23:00提供服务
- 3、在12306.cn网站购票、改签和退票须不晚于开车前2小时

登录名:

密码:

验证码:

刷新

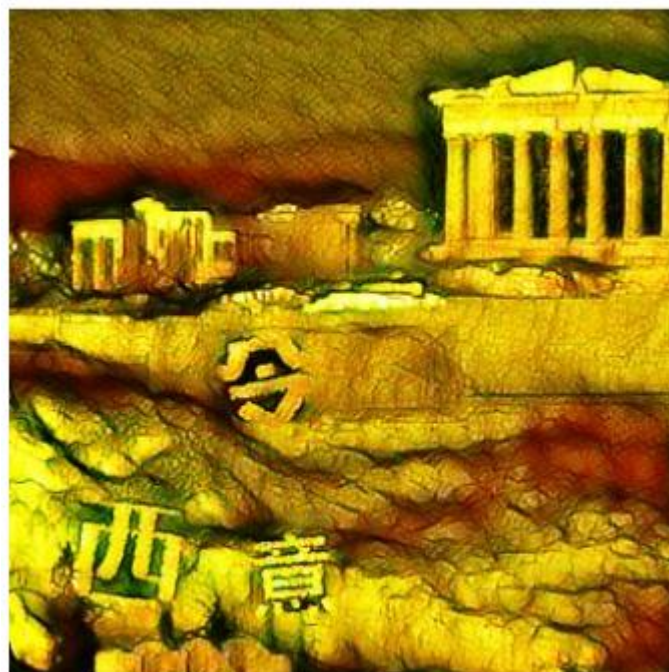
登录 快速注册

忘记密码/密码?

验证码如何使用?

# 极验的验证码

请在下图依次点击：西 意 令



## 一键通过模式

用户名：

密码：

完成验证： 点击按钮进行验证

# 关于极验

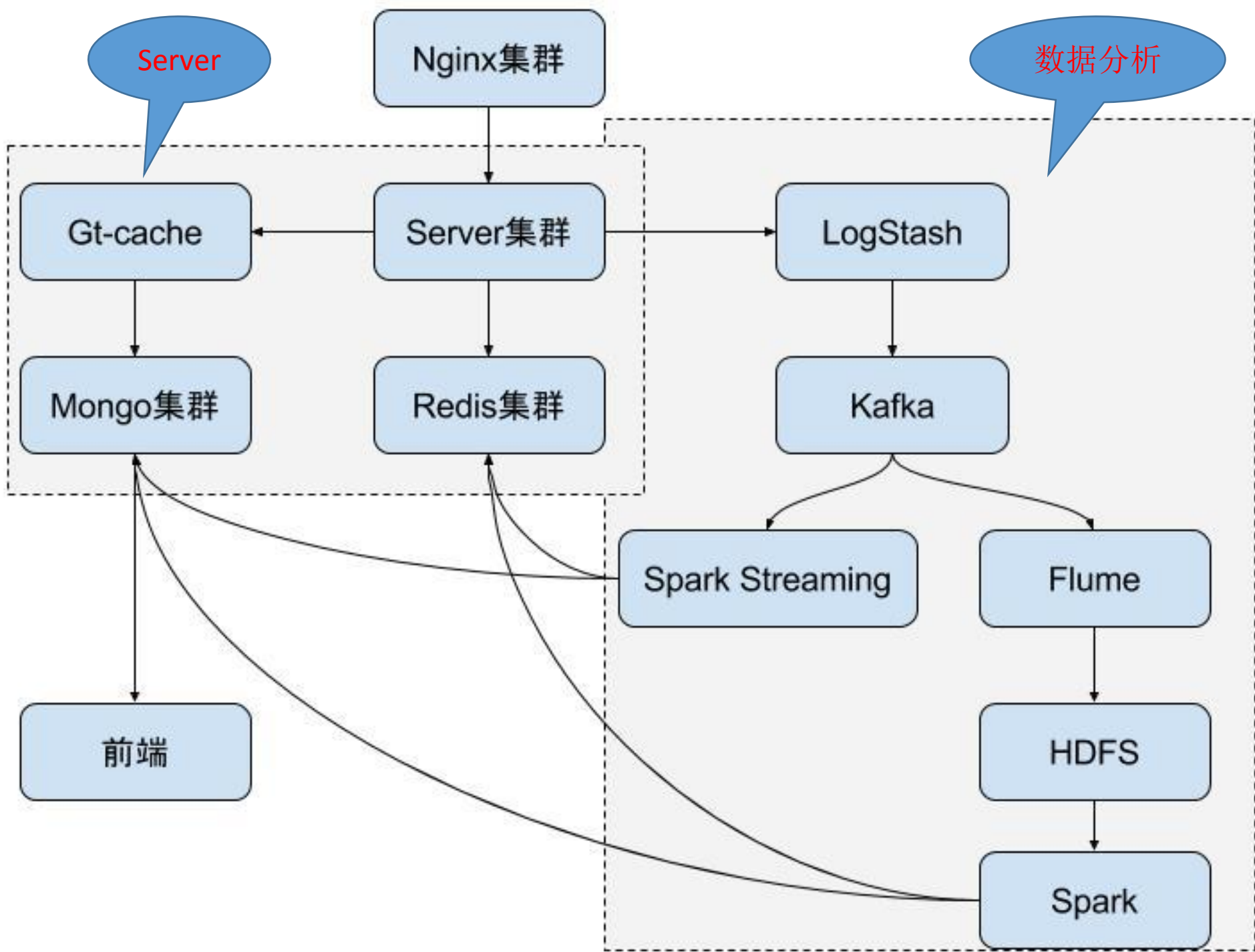
- 全球首次提出“行为式验证”概念，并获得多项知识产权
- 超过16万家网站使用极验的验证产品
- 每天提供超过5亿次的验证安全防护服务
- 对自然人识别正确率：98.5%，机器人识别正确率：98%
- 产品演示：<http://www.geetest.com/demo/>

# 极验的特点

- 产品小巧精致，简单就是美
- 业务逻辑简洁、稳定，核心业务逻辑迭代慢，API少
- 服务要求：高并发、高响应、高可用
- 主要技术分布：服务端、数据平台、数据挖掘、运维



# 当前服务端架构



# 提纲

## Server

- 语言、框架选型
- 数据库
- 缓存
- 数据存储和交互
- 容错
- 安全

## 数据分析

- 日志收集和存储
- Spark
- Mesos
- Tensorflow
- 图片处理

# Server : 语言、框架选型

## ➤ Python or Go

- 过去: Python2.7, 逻辑简单、流量小
- 现在: Python3.5 + Scipy + C++, 逻辑复杂、流量大、交互多
- 将来: Python3.5 + Scipy + C++ + Go(或其他), 私有化、解耦

## ➤ Tornado or Django or Other

- 极验: 只有Tornado
- 其他公司: Django似乎更多



# Server : 数据库

## ➤ 变迁过程

单机MongoDB → MongoDB集群 → MongoDB + InfluxDB

## ➤ 关注点

- 高可用，MongoDB副本集，多机备份
- 高速，MongoDB + gt-cache
- 时序性强，InfluxDB
- 海量存储，HDFS + *HBase*

# Server : 缓存

## ➤ 数据库缓存: gt-cache

基于mmap, 使用C语言实现的进程间共享内存模块, 将较小规模的数据映射到本地固定的内存空间, 只提供get和put操作, 非常高效

## ➤ gt-cache VS redis

gt-cache

- 无网络IO, 更快;
- 每台server一份, 冗余, 耗费内存
- 功能简单, 需要自己开发维护

redis

- 有网络IO, 稍慢;
- 集中管理, 可用内存很大, 无冗余
- 功能丰富, 维护简单

# Server : 缓存

Redis集群：自建gedis

➤ 主要用途：缓存proof结构体，记录每一次验证过程变化状态，需要100GB+的内存

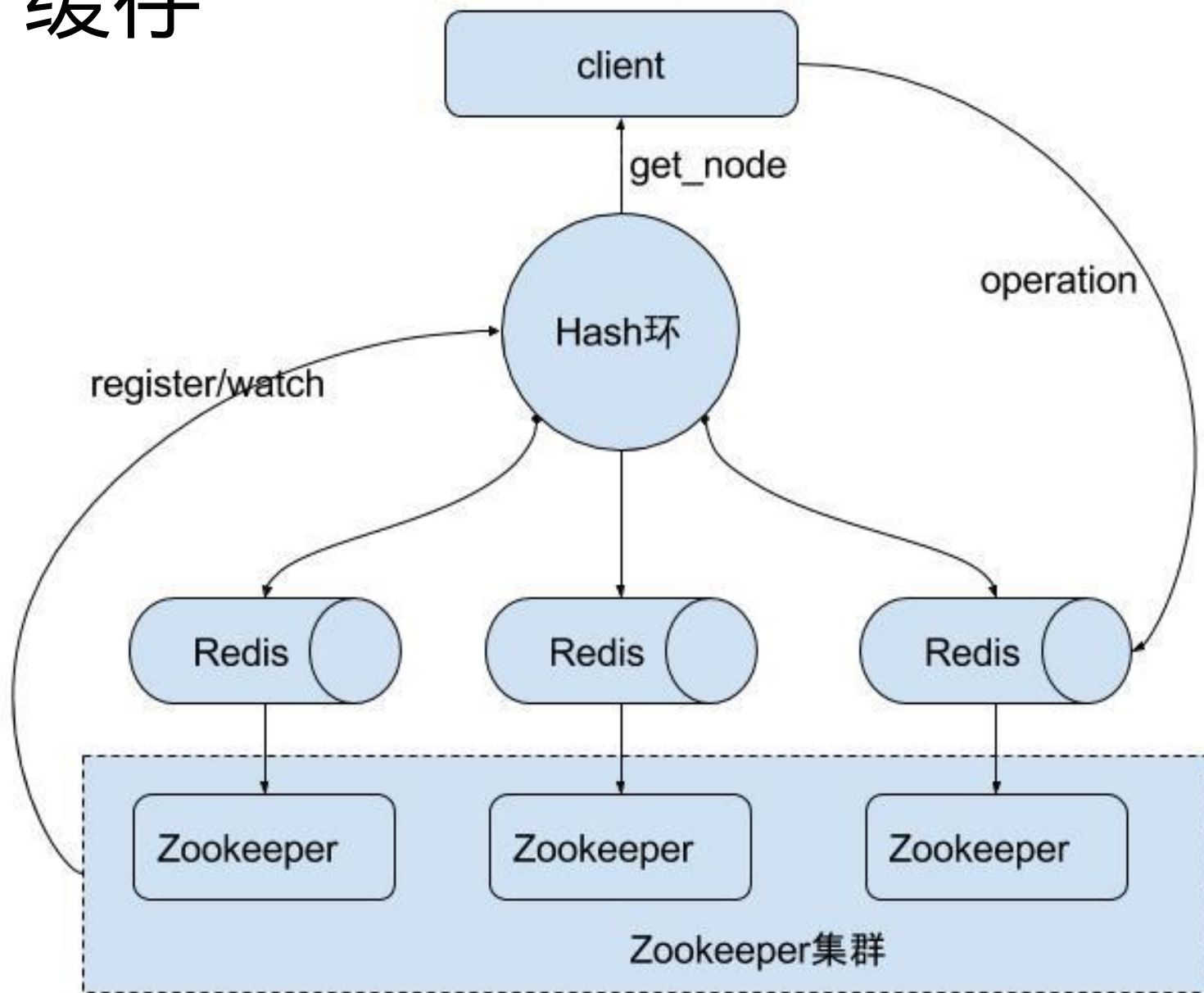
➤ 实现：zookeeper + 一致性hash算法 + redis

➤ 效果

- 稳定：稳定A(zookeeper+一致hash) + 稳定B(redis稳定) = 整体稳定
- 高可用：set双写，zookeeper动态增删节点

# Server : 缓存

Gedis实现



# Server : 缓存

## ➤ 变迁

单机redis → 单机redis + codis(fail) → 单机redis + gedis → 全gedis → *codis(maybe)*

## ➤ gedis VS codis

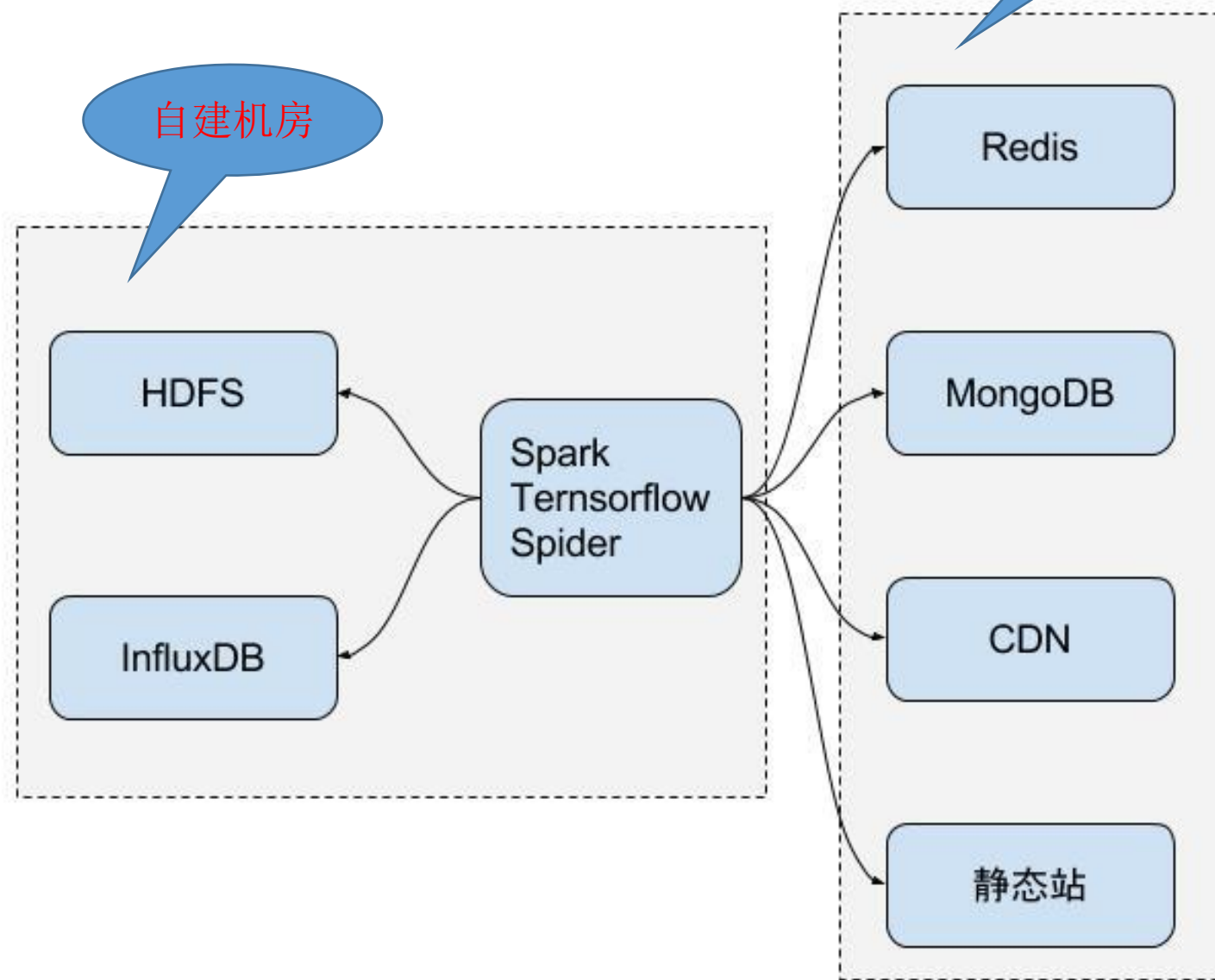
gedis

- 稳定可靠可定制
- 功能不完备，如pipeline

codis

- 越来越成熟
- 开源，功能越来越丰富

# Server : 数据存储和交互



# Server：数据存储和交互

## 问题

- 跨机房的数据读写
- InfluxDB集群、热备
- 大数据存储和查询
- 图片防抓取

# Server : 容错



极验云安全服务

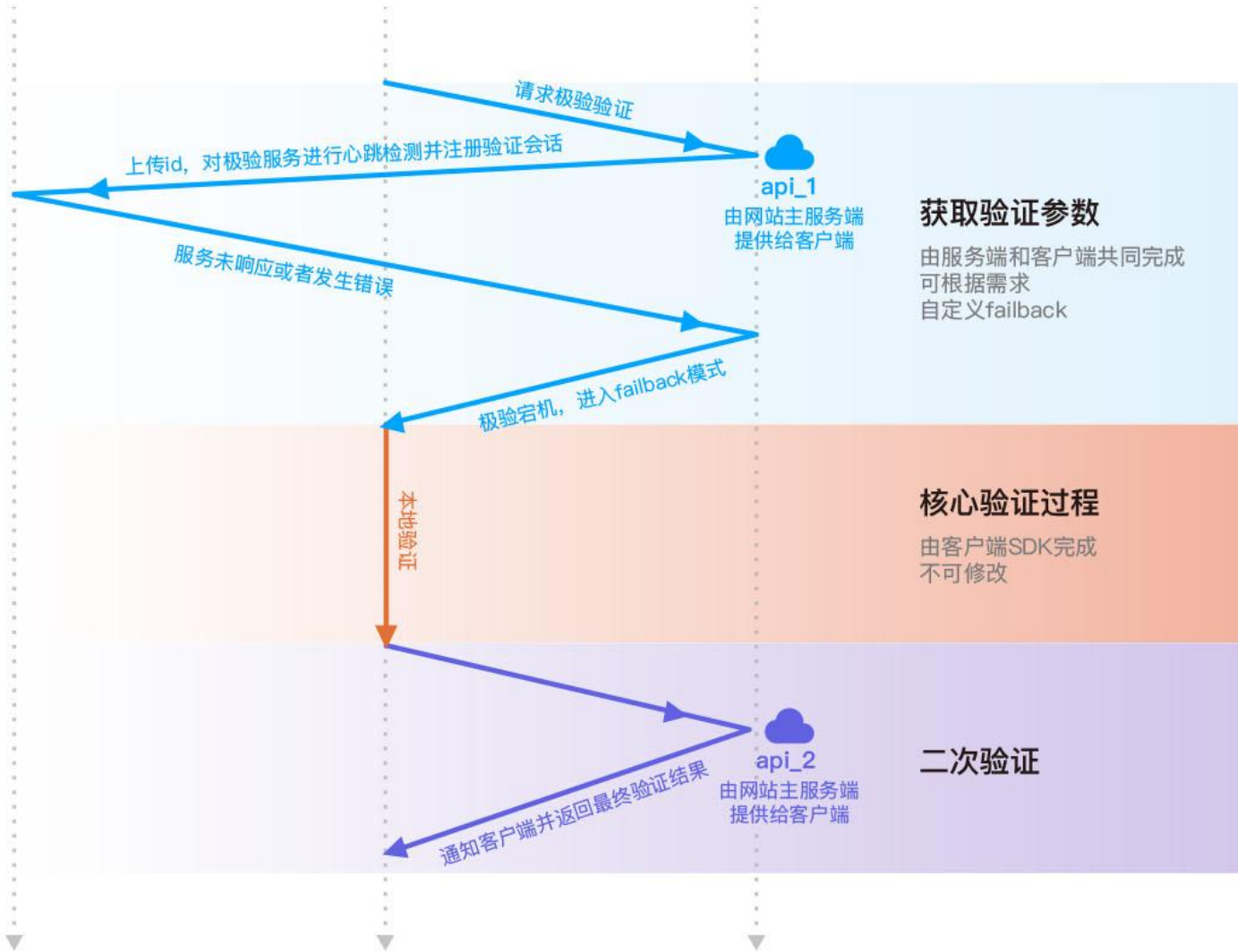


网站主的客户端



网站主的服务器

register

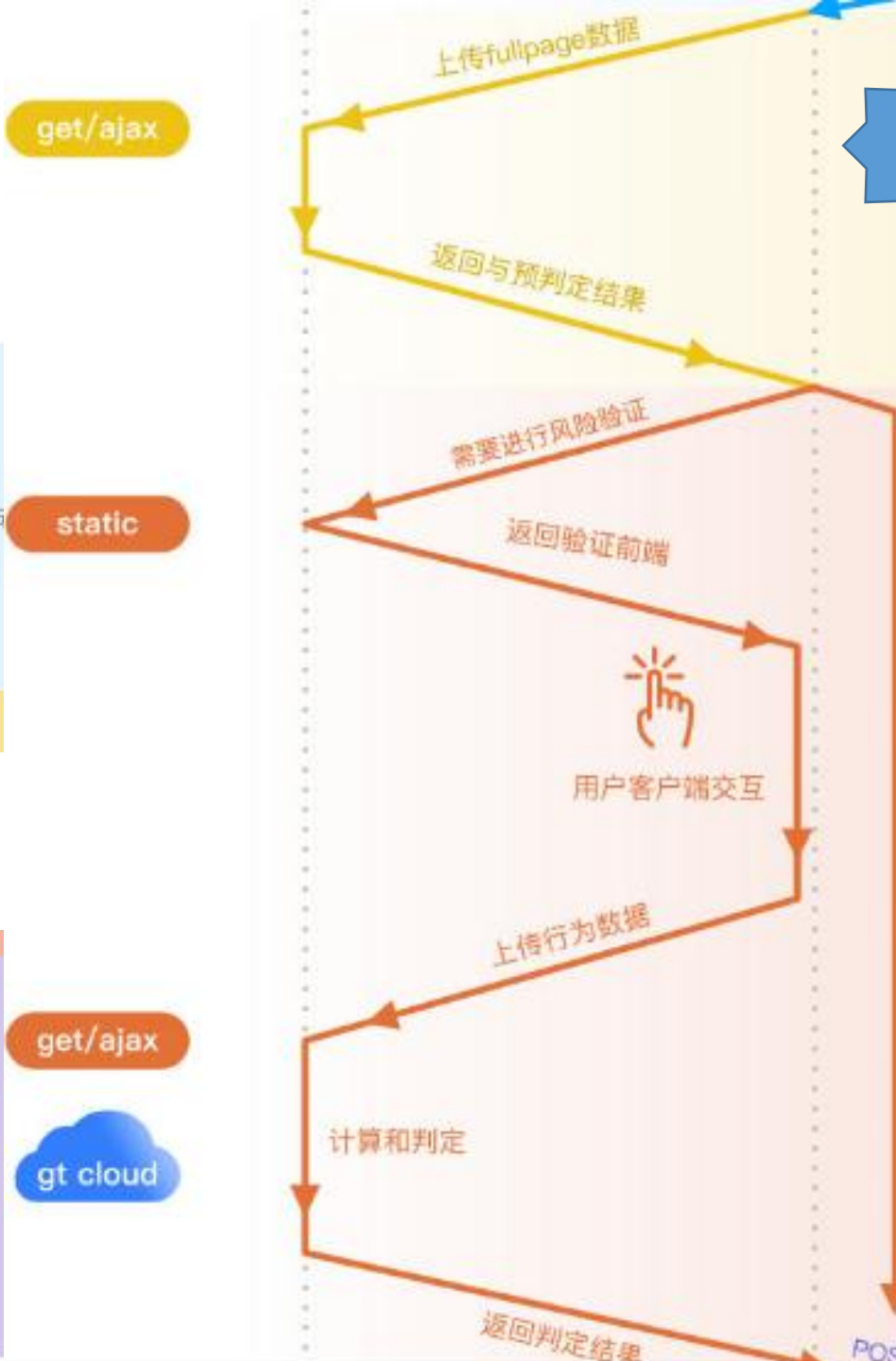
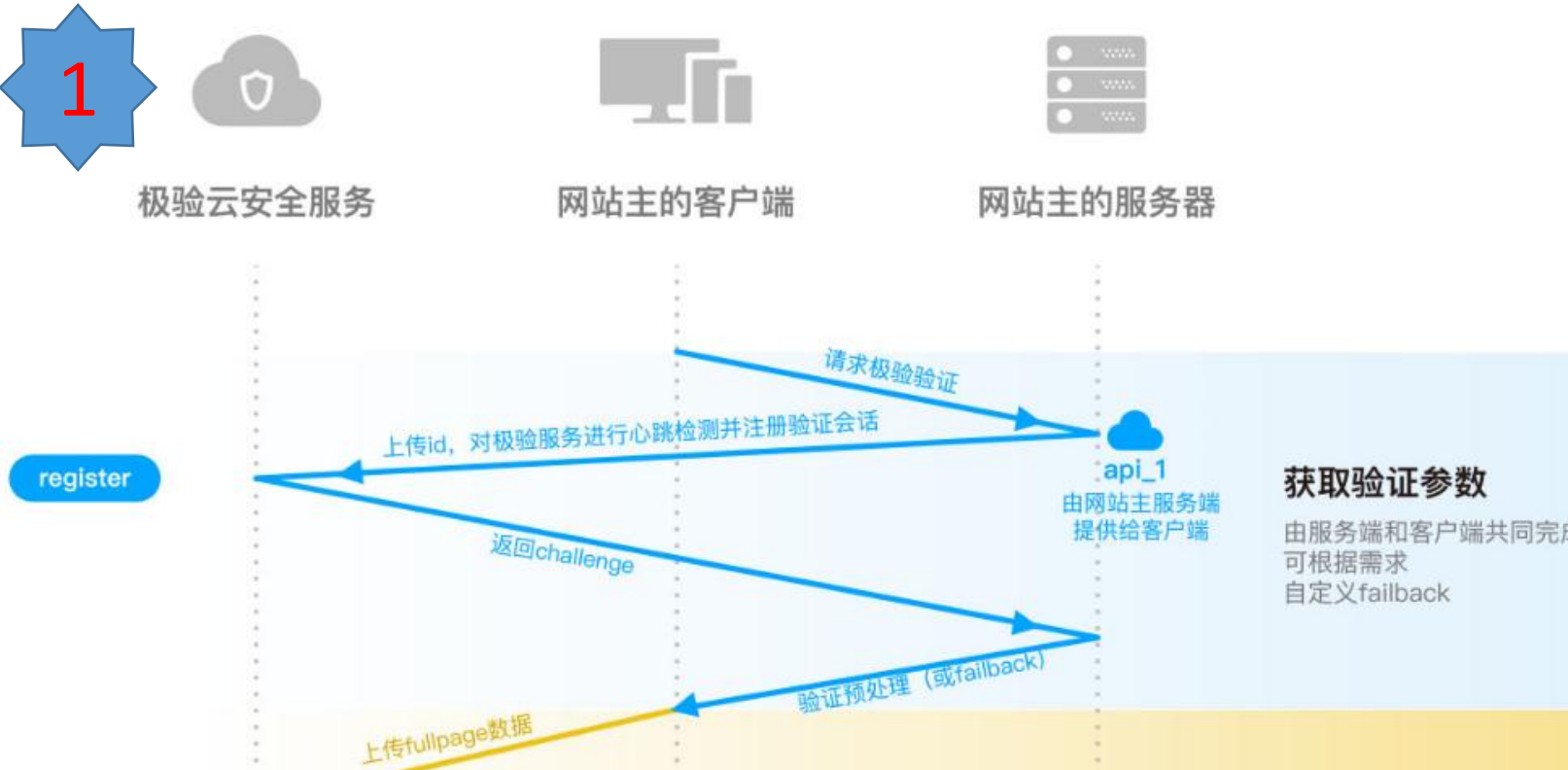


Server failback

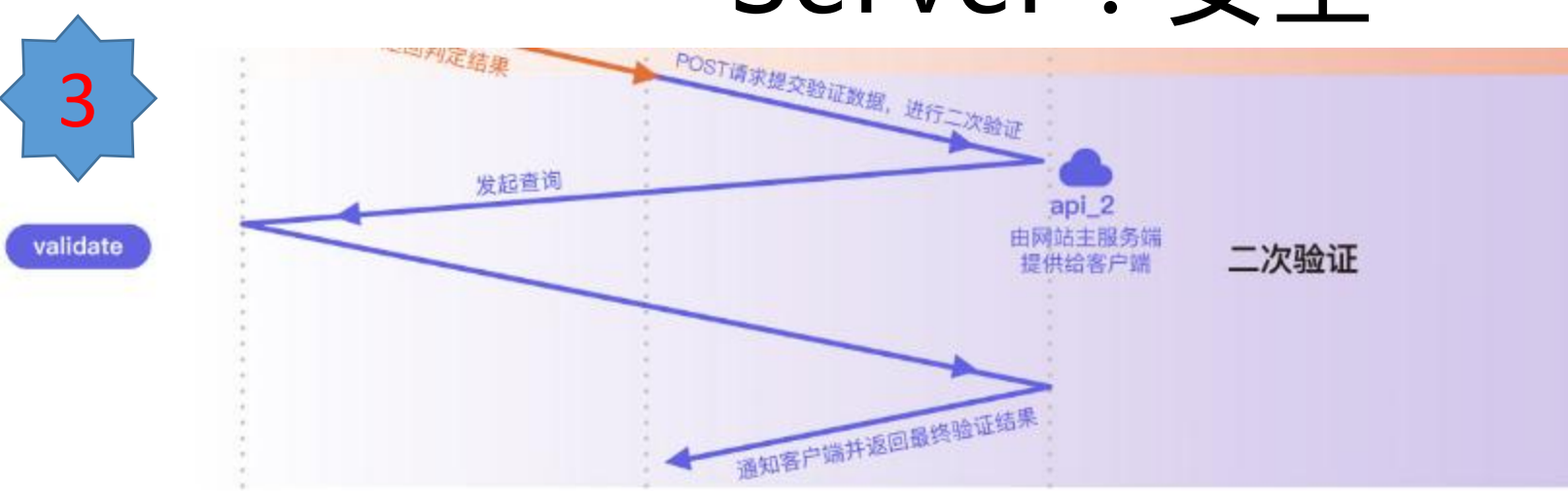


# Server : 容错

- MongoDB副本集
- Redis集群动态伸缩, 双写
- 整体服务异地机房热备, 甚至缓存热同步
- 双CDN(七牛/阿里云) ↔ 自建图片服务器 热切换



# Server : 安全



# Server : 安全

- 加密：图片加密、数据加密、答案加密
- 一次性：验证不能重复使用，图片都可能是一次性的
- 客户服务端、极验云端相互校验

# 数据分析：日志收集和存储

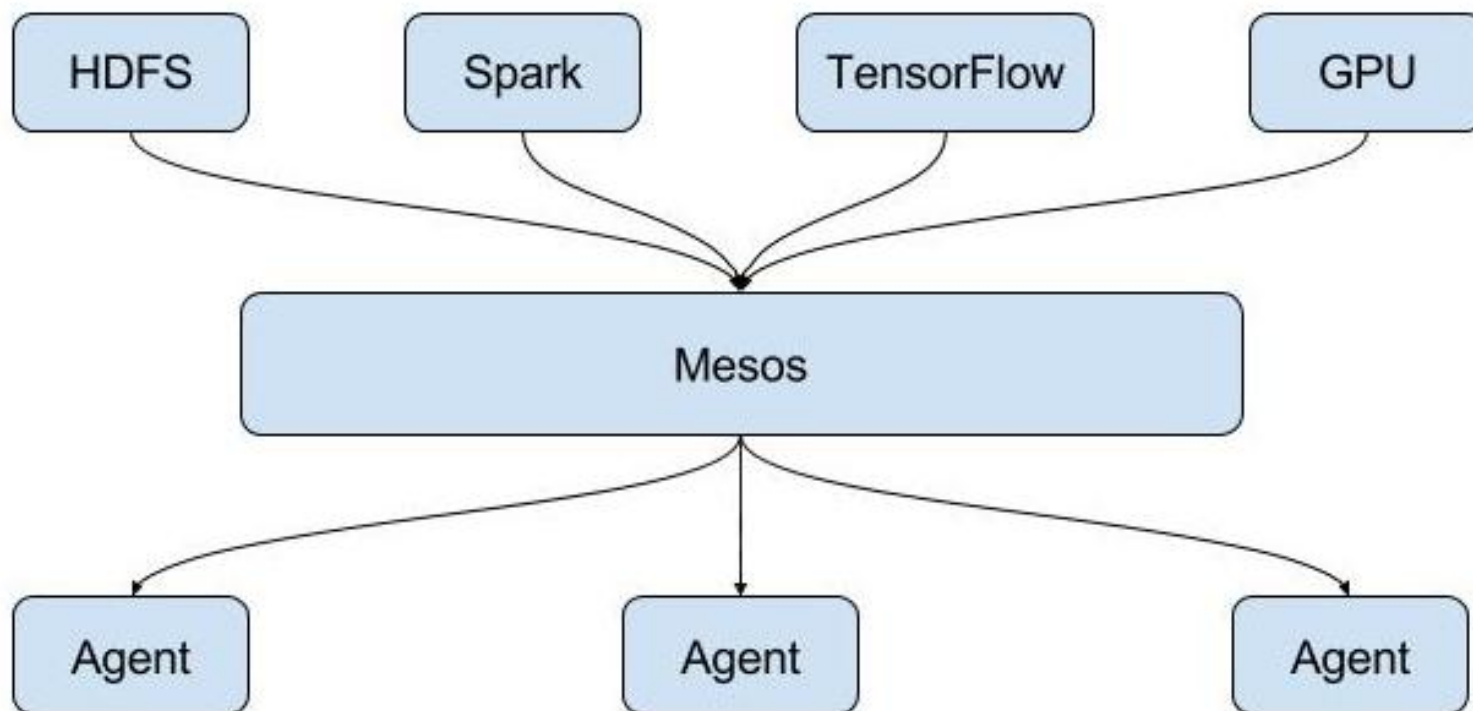
- 结构：LogStash + Flume + Kafka + HDFS
- Kafka：供Spark Streaming实时计算
- HDFS：持久化，离线计算

# 数据分析：Spark

- Spark、Storm、Hadoop
- Spark与Spark Streaming
- Spark输入、输出(HDFS、DB)
- 存在的问题：
  - Streaming在数据量太大时还是不够快
  - 与数据库的交互还存在问题

# 数据分析：Mesos

- Mesos是Apache下的开源分布式资源管理框架，它被称为是分布式系统的内核。我们用来复用服务器资源。



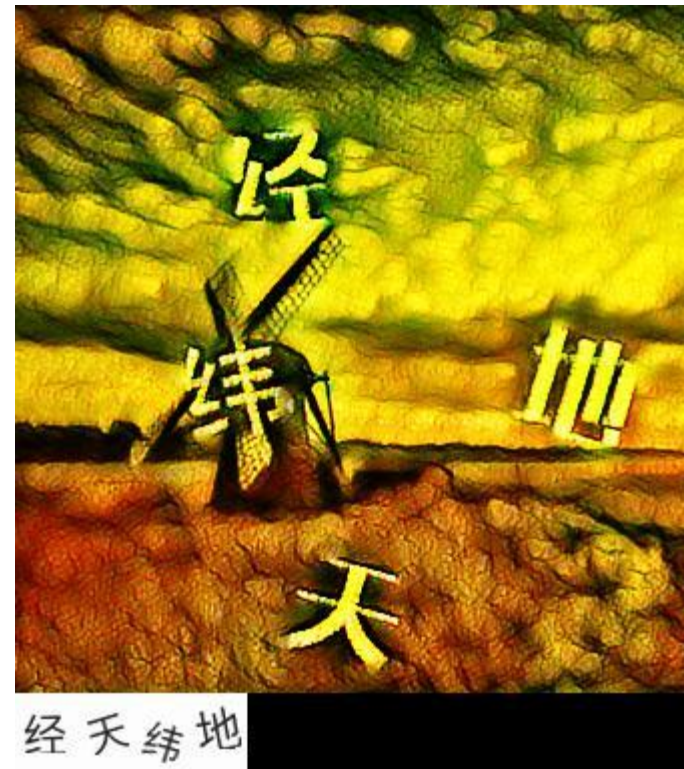
# 数据分析：Tensorflow

➤用途：图片处理、模型训练

➤历程：

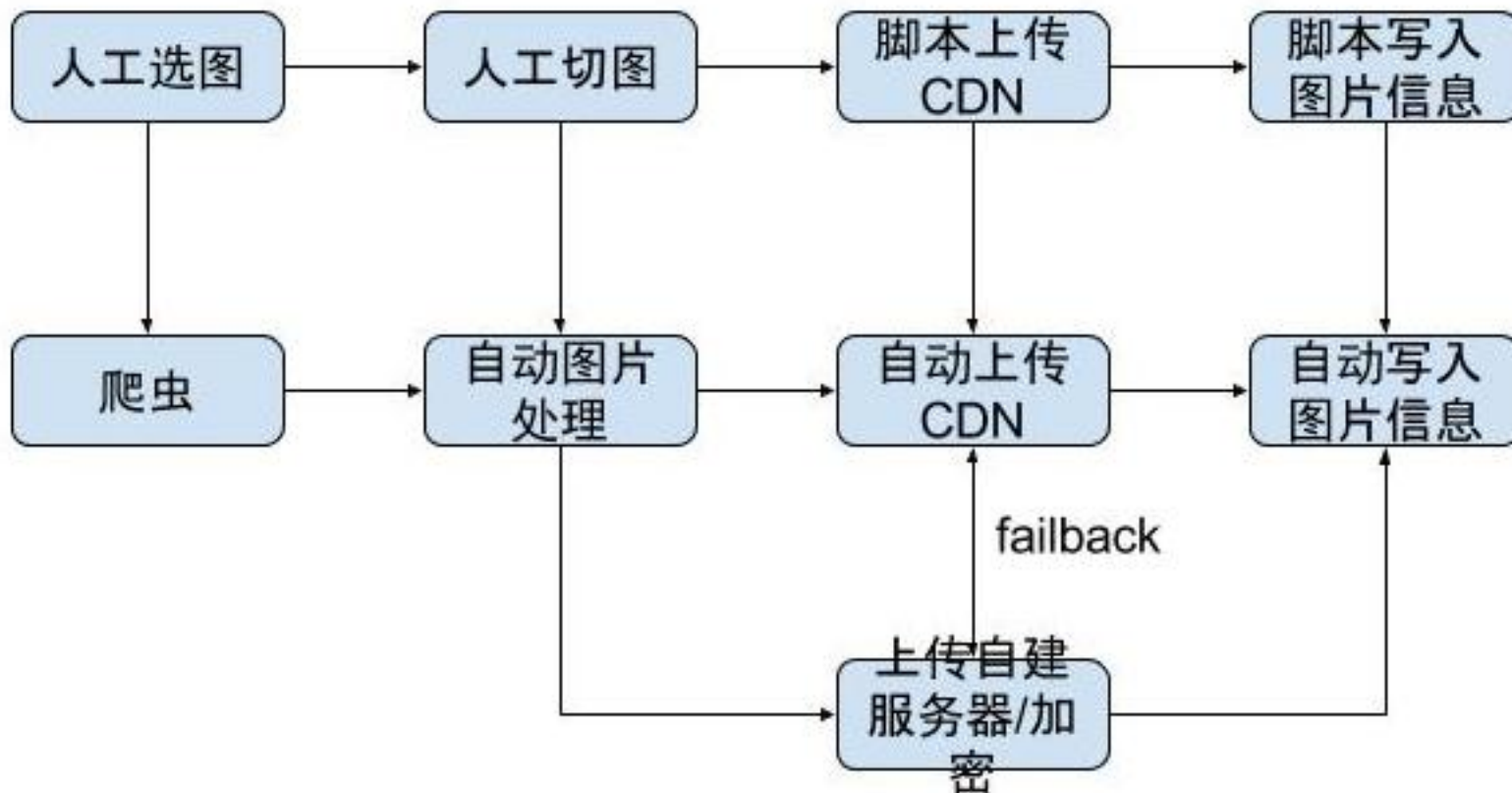
CPU → 单GPU → 单机多GPU

→ 分布式GPU调度(gt-mesos)



# 数据分析：图片处理

➤ 历程





# 未来的路

- Server、数据挖掘代码解耦
- 私有化方向中整体架构如何重构
- 重要板块优化：数据库、缓存、容错
- 大数据存储和查询
- 计算平台高效利用，甚至对外开放
- 海量验证图片，“阅后即焚”

...

# 总结

- 由单机走向集群
- 由人工、脚本、小数据量走向自动化、项目化、大数据
- 由全部使用开源软件走向部分使用自研软件
- 由关注业务到关注高可用、高响应
- 由独立、随意到平台、复用、规范

Thank you

Q & A