分布式系统游戏应用的后台架构设计

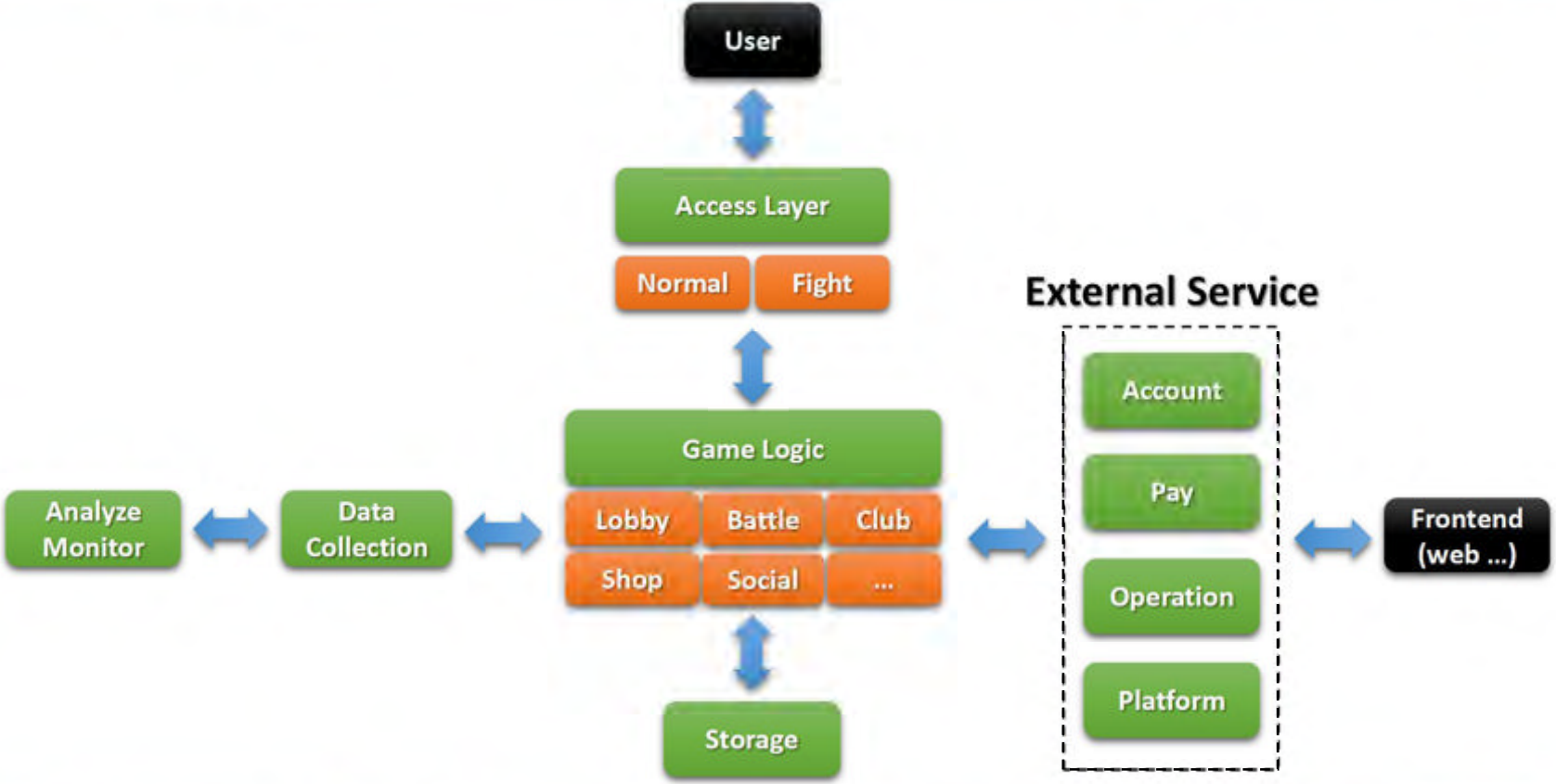
陶文质，腾讯互娱事业部高级工程师

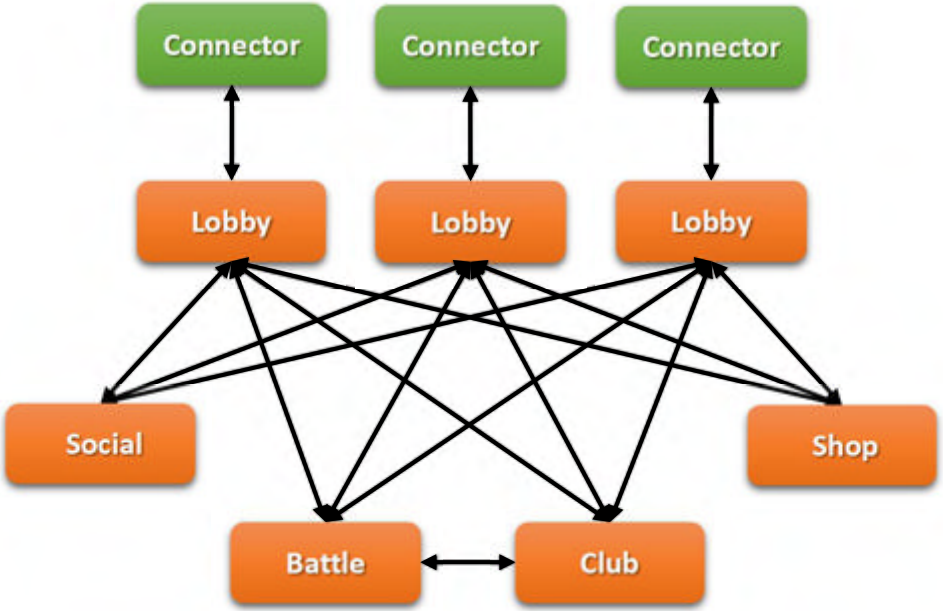
概要

- 传统架构的局限
- 分布式设计
- 应用场景

趟过的那些坑

- 游戏系统基本组成
- 传统实现
- 困惑和局限


User
Access Layer
Normal
Fight
External Service
Account
Pay
Operation
Platform
Game Logic
Lobby
Battle
Club
Shop
Social
...
Analyze Monitor
Data Collection
Frontend (web ...)
Storage



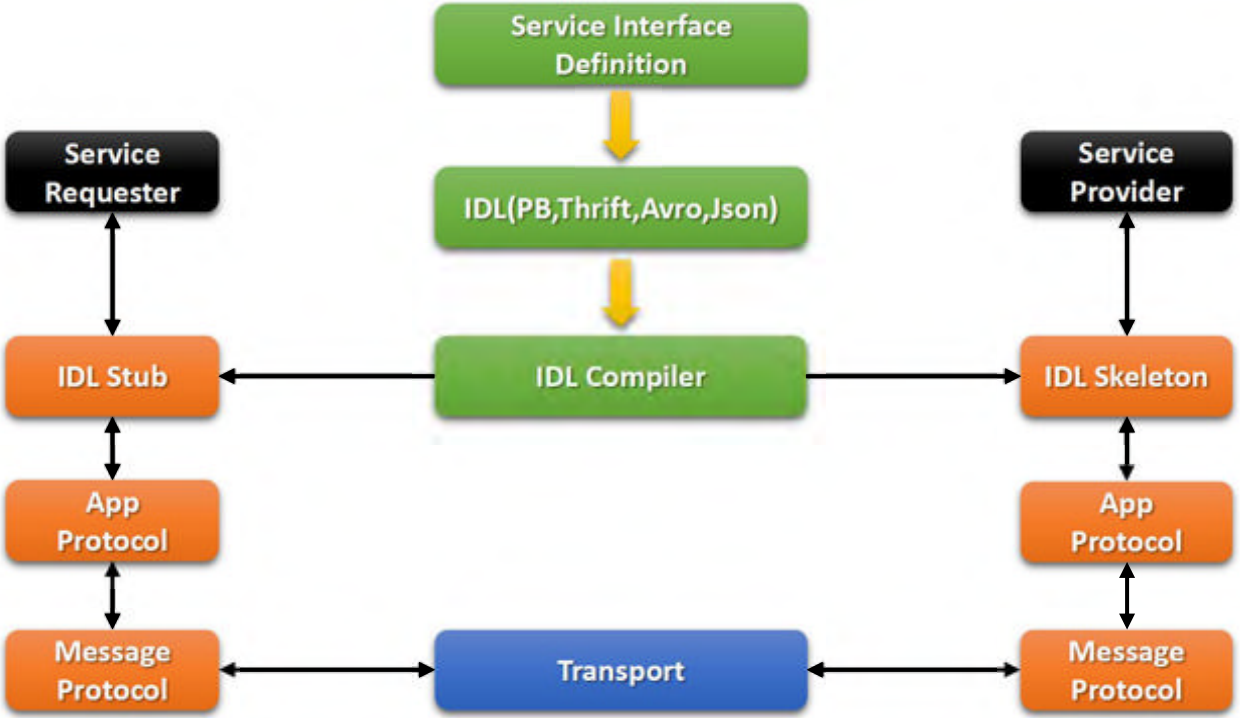
- 静态角度看
  - 如何知道通信对端地址
  - 耦合度较高
  - 配置管理复杂
  - 通信链路多而杂
- 动态角度看
  - 怎么扩容
  - 怎么容灾
  - 怎么运行时变更

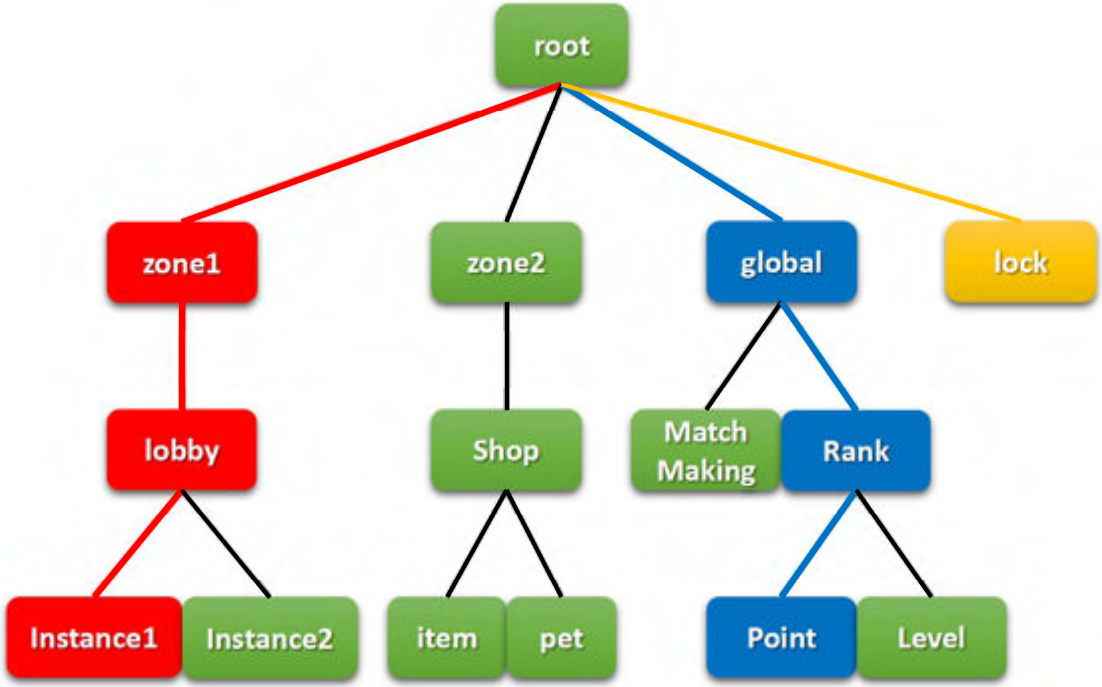
分布式设计

- 面向服务
- 基础设计
- 高层需求与实现

面向服务

- 抽象与解耦
- 内聚与自治
- 定义服务内容与交互行为
- 可发现、可协调


Service Interface Definition
IDL(PB,Thrift,Avro,Json)
IDL Compiler
Service Requester
IDL Stub
App Protocol
Message Protocol
Service Provider
IDL Skeleton
App Protocol
Message Protocol
Transport

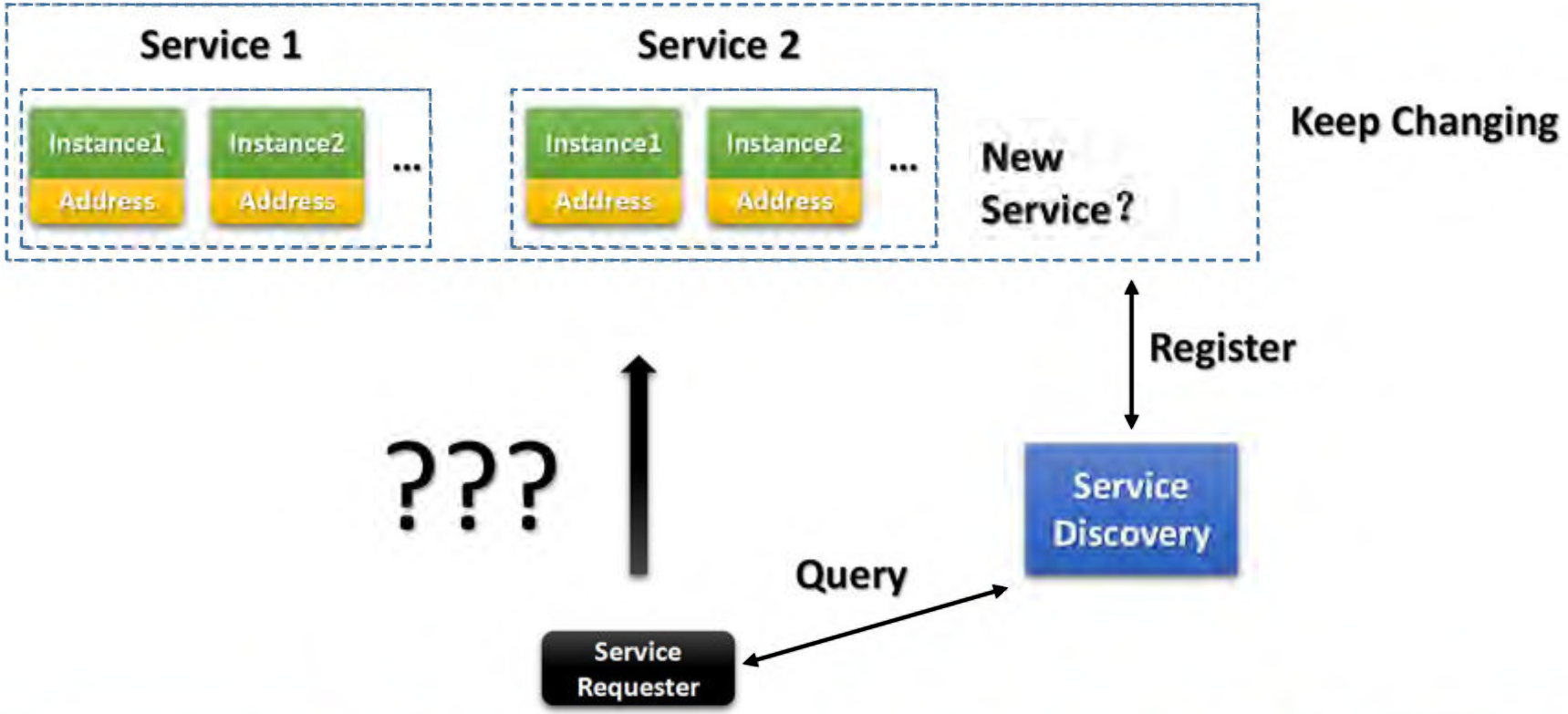


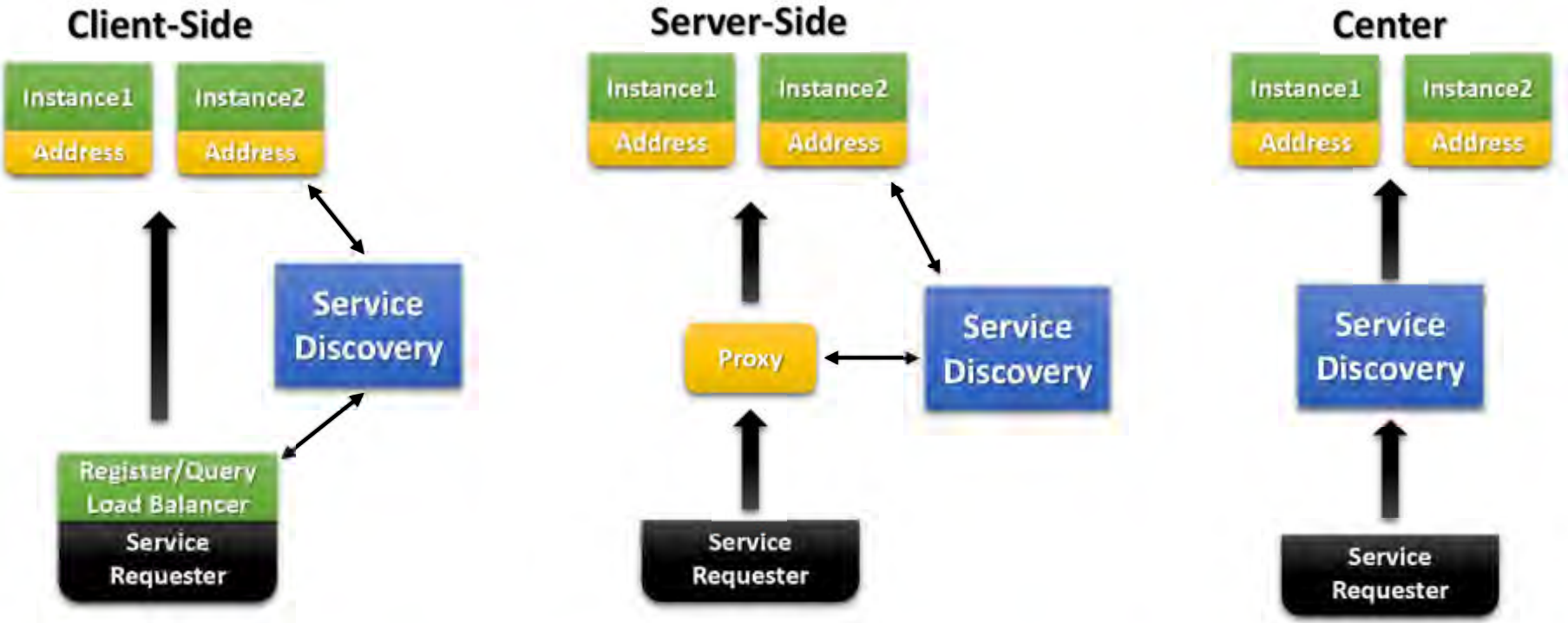
/zone1/lobby/Instance1

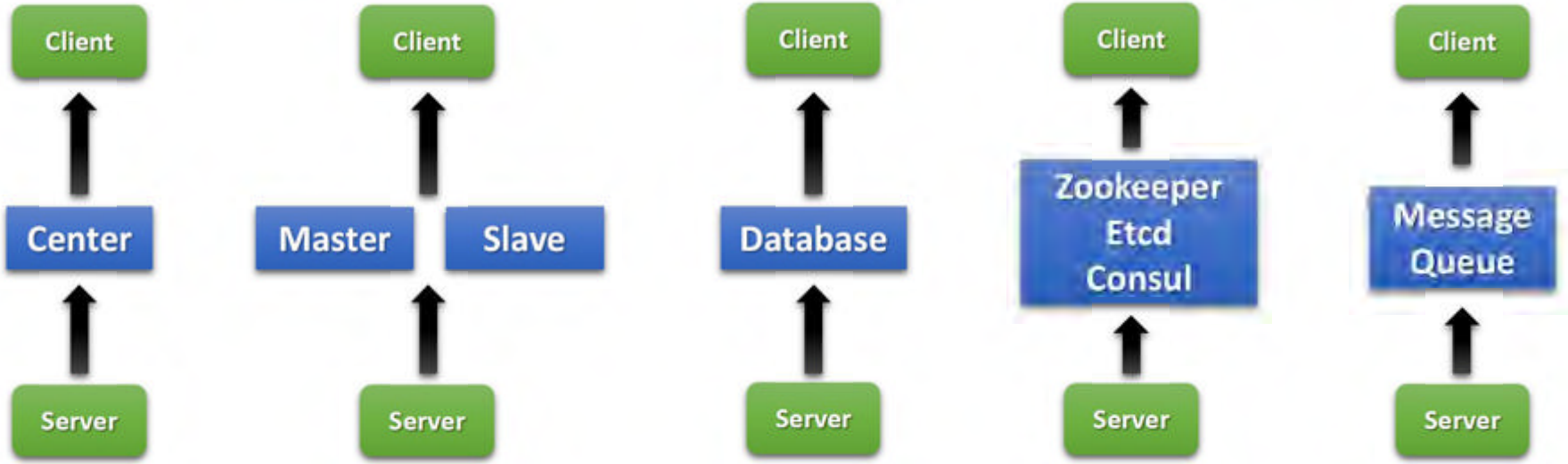
/global/Rank/Point

/lock


Single Instance
Item
Trade
Homogeneous
Item
Trade1
Item
Trade2
Item
Trade3
Heterogeneous
Trade
Item
Pet
Item
Instance1
Item
Instance2
Cat
Dog


Service 1
Service 2
Instance1
Address
Instance2
Address
...
Instance1
Address
Instance2
Address
...
New Service?
Keep Changing
Register
???
Service Discovery
Query
Service Requester


Client-Side
Instance1
Address
Instance2
Address
Service Discovery
Register/Query
Load Balancer
Service Requester
Server-Side
Instance1
Address
Instance2
Address
Proxy
Service Discovery
Service Requester
Center
Instance1
Address
Instance2
Address
Service Discovery
Service Requester



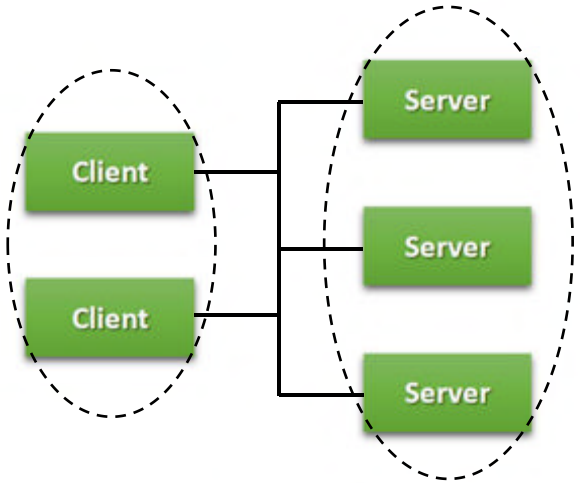
Watch, Load Balancing, Availability, Scalability

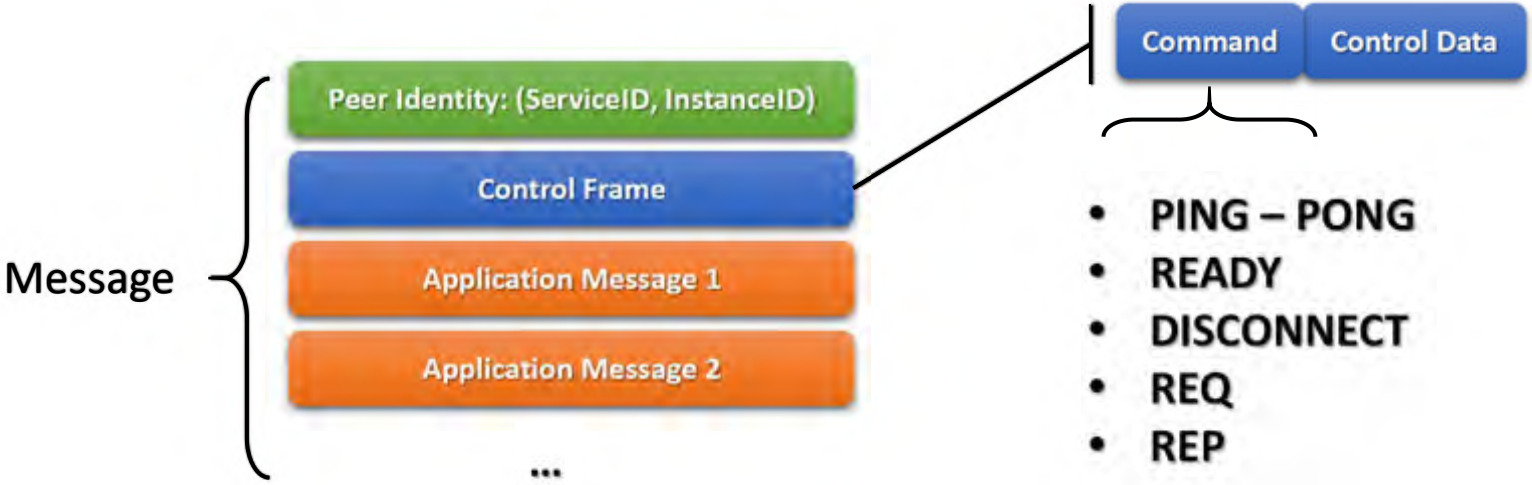
基础设计

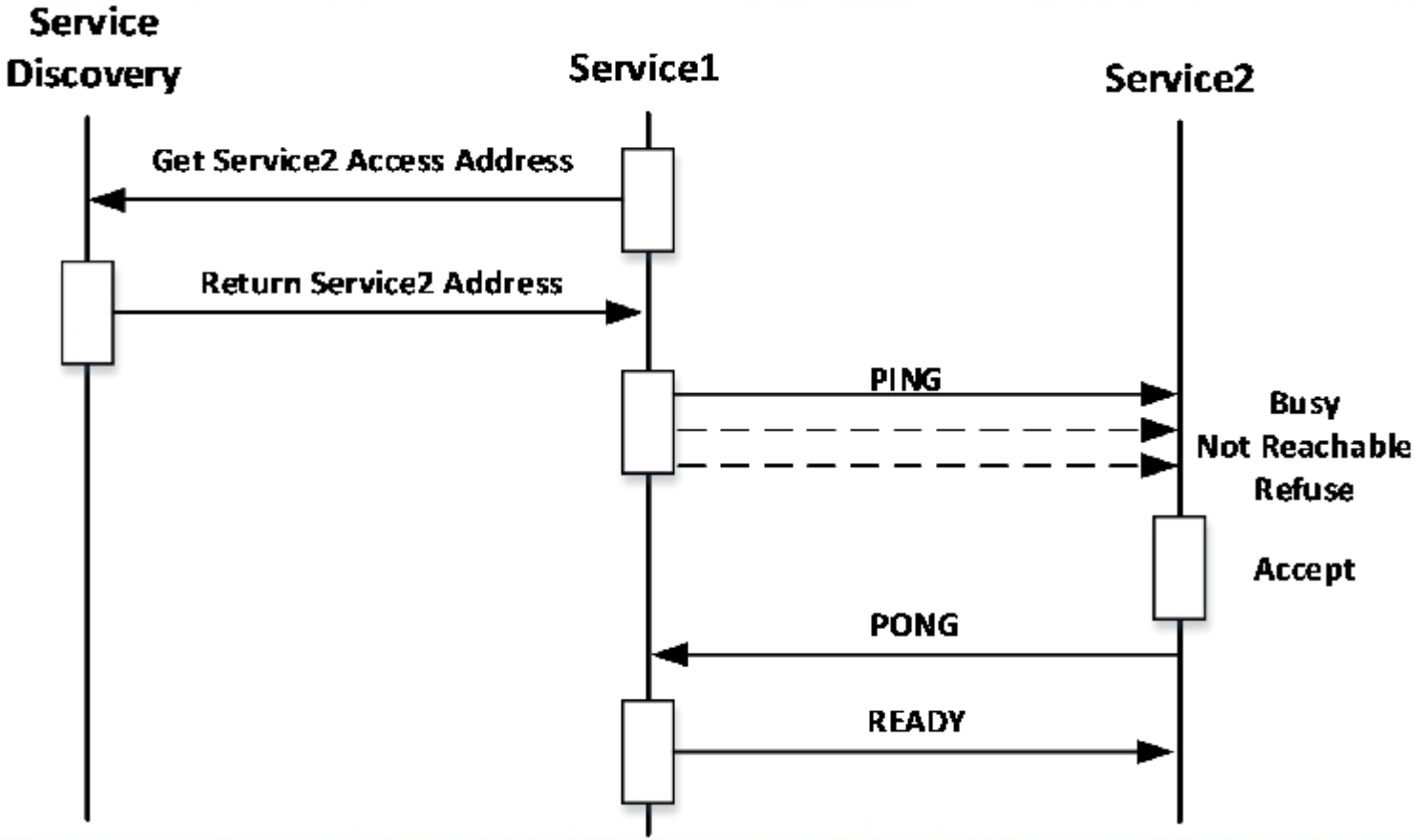
- 最简模式——两个集群之间的通信
- 协议设计
- 结构设计

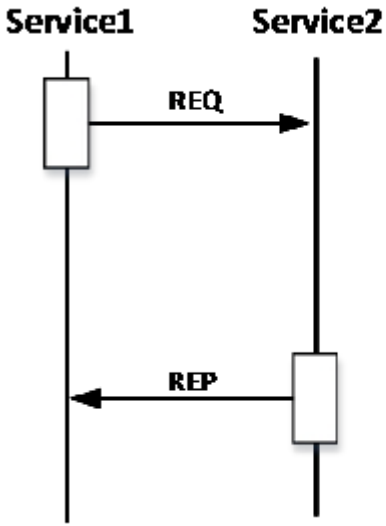
最简模式基本功能

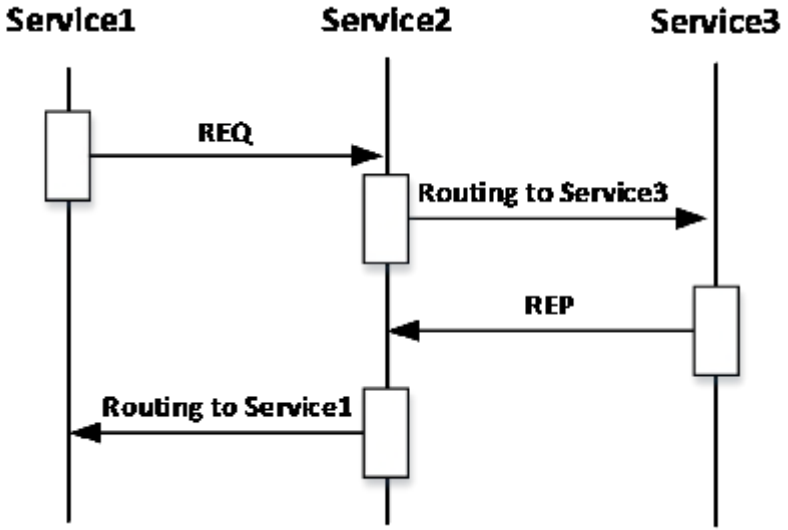
- 探测
- 保活
- One Way, Request-Reply
- 广播, 路由
- 负载均衡

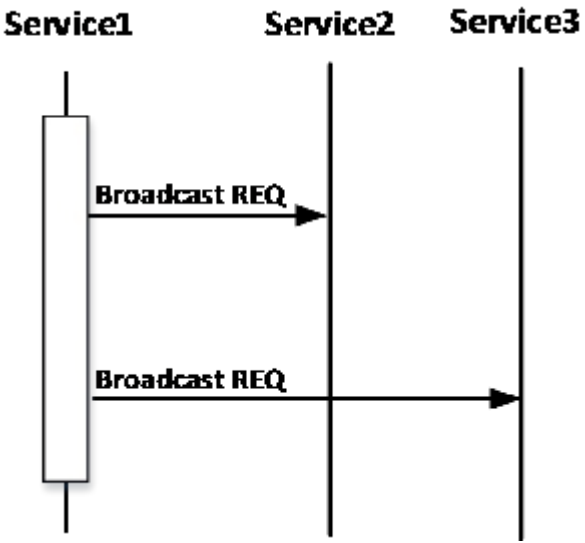



Message
Peer Identity: (ServiceID, InstanceID)
Control Frame
Application Message 1
Application Message 2
...
Command
Control Data
PING – PONG
READY
DISCONNECT
REQ
REP

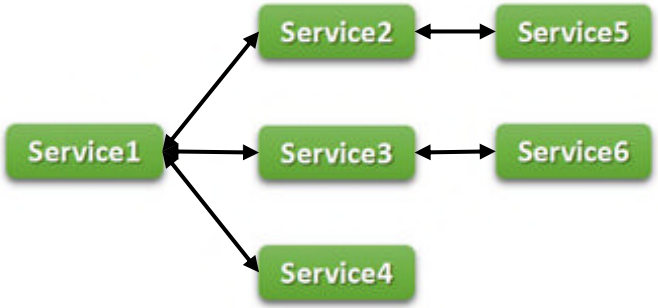

Service Discovery
Service1
Service2
Get Service2 Access Address
Return Service2 Address
PING
Busy
Not Reachable
Refuse
Accept
PONG
READY

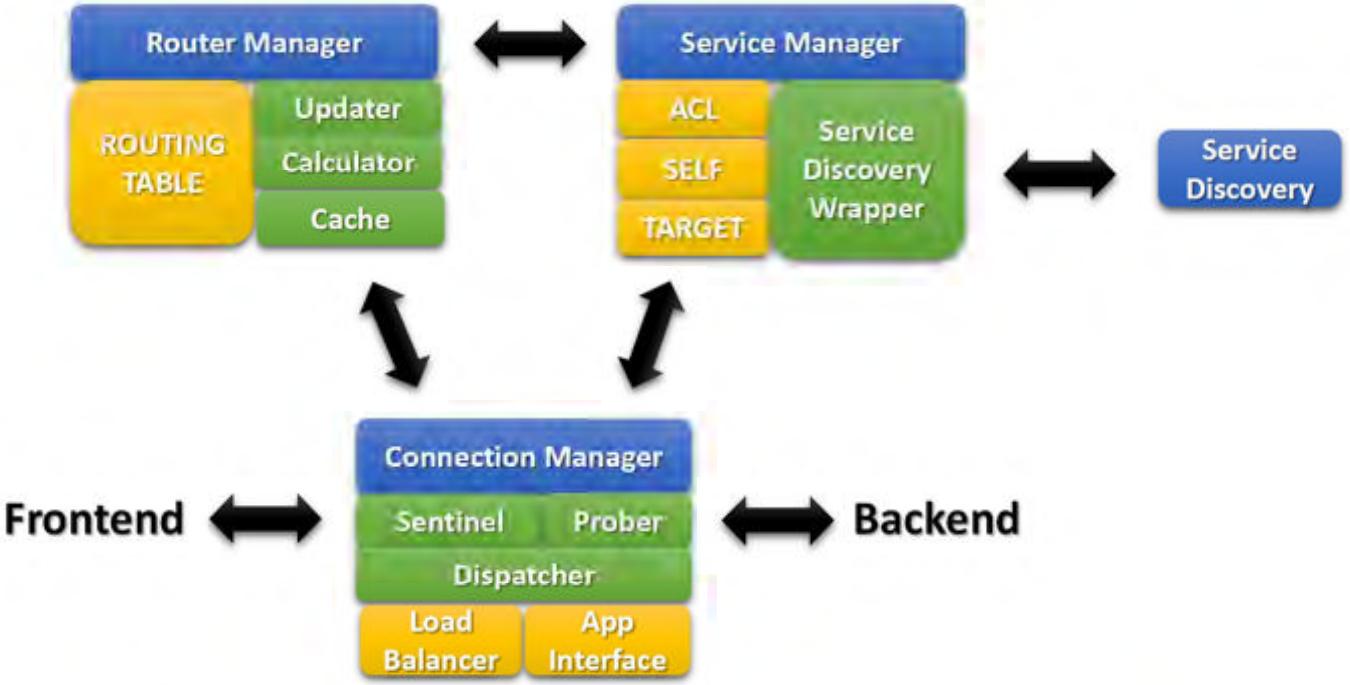

Service1
Service2
REQ
REP


Service1
Service2
Service3
REQ
Routing to Service3
REP
Routing to Service1


Service1
Service2
Service3
Broadcast REQ
Broadcast REQ

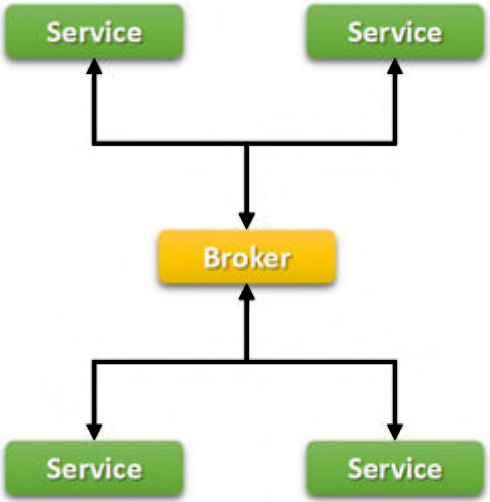
Routing Table	
Service1	Service2
Service1	Service3
Service1	Service4
Service2	Service5
Service3	Service6

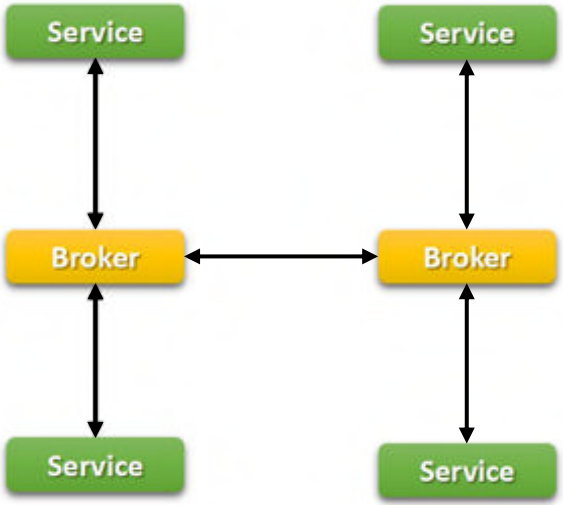


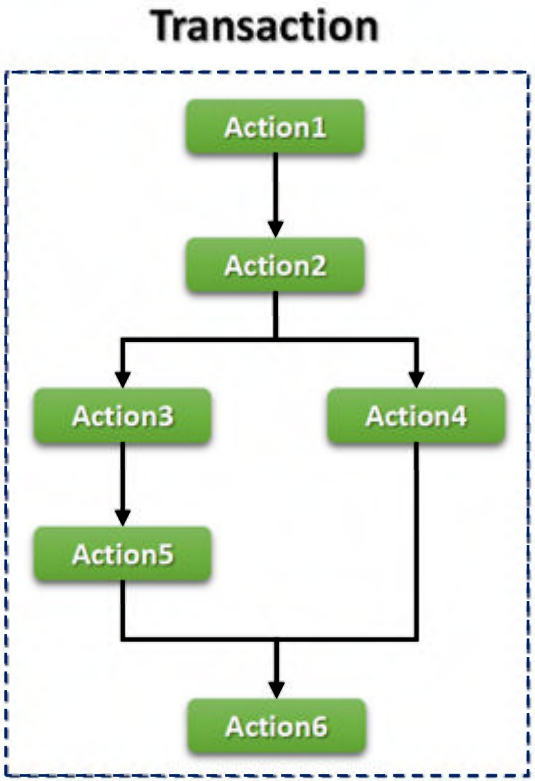

Router Manager
ROUTING TABLE
Updater
Calculator
Cache
Service Manager
ACL
SELF
TARGET
Service Discovery Wrapper
Service Discovery
Frontend
Connection Manager
Sentinel
Prober
Dispatcher
Load Balancer
App Interface
Backend

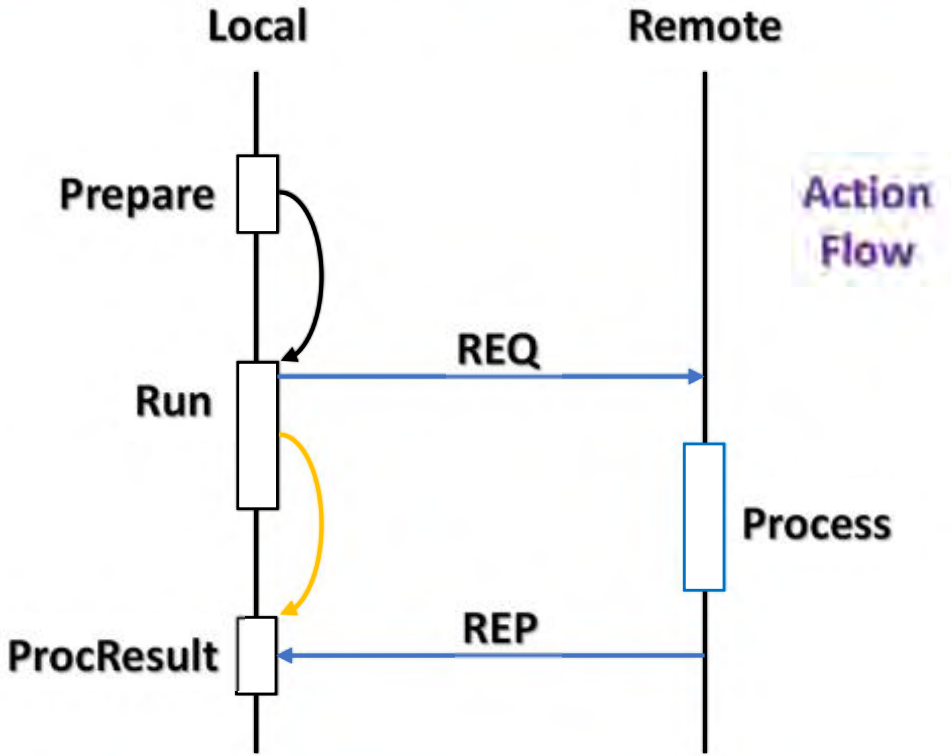
高阶设计举例

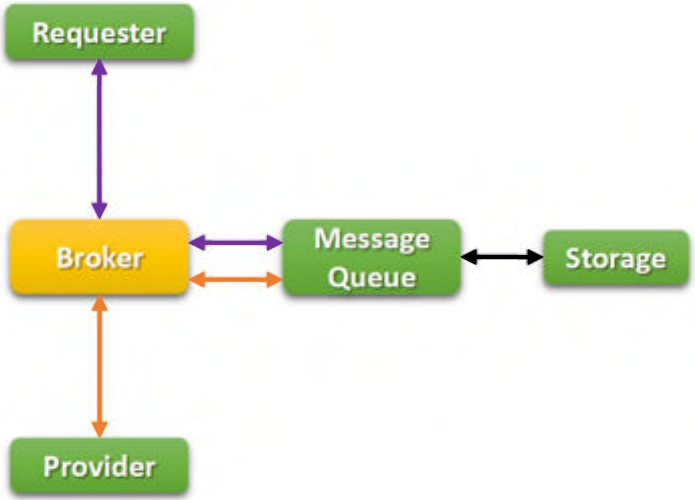
- Broker-Based通信
- 简单事务
- 消息队列

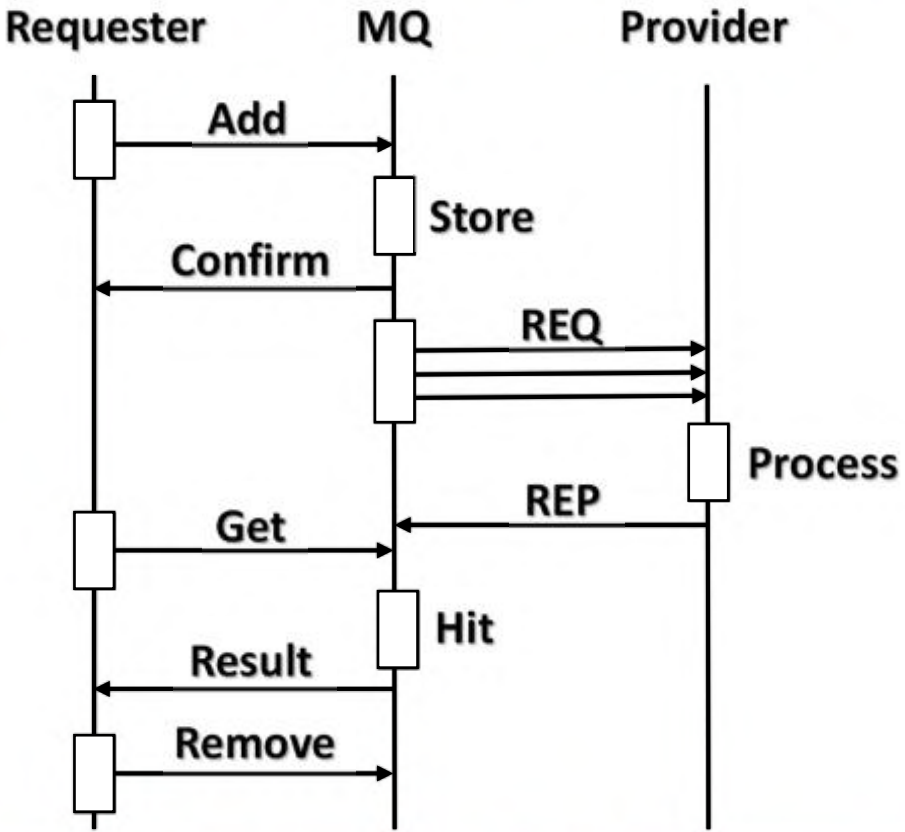

Service
Service
Broker
Service
Service

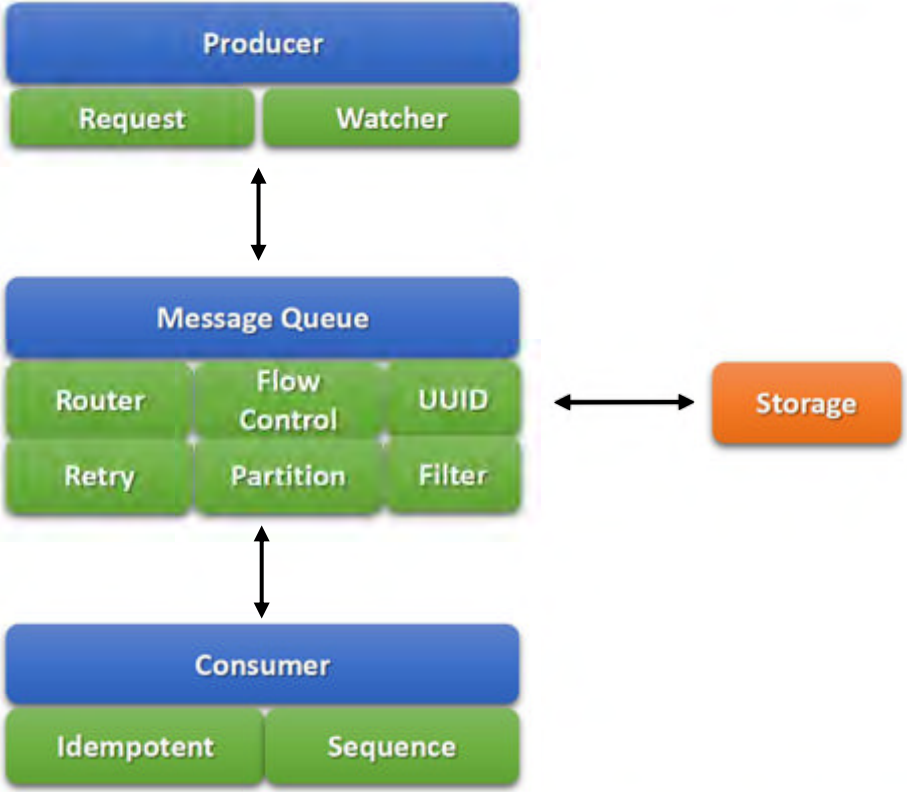

Service
Service
Broker
Broker
Service
Service


Transaction
Action1
Action2
Action3
Action4
Action5
Action6


Local
Remote
Prepare
Action
Flow
REQ
Run
Process
REP
ProcResult


Requester
Broker
Message Queue
Storage
Provider

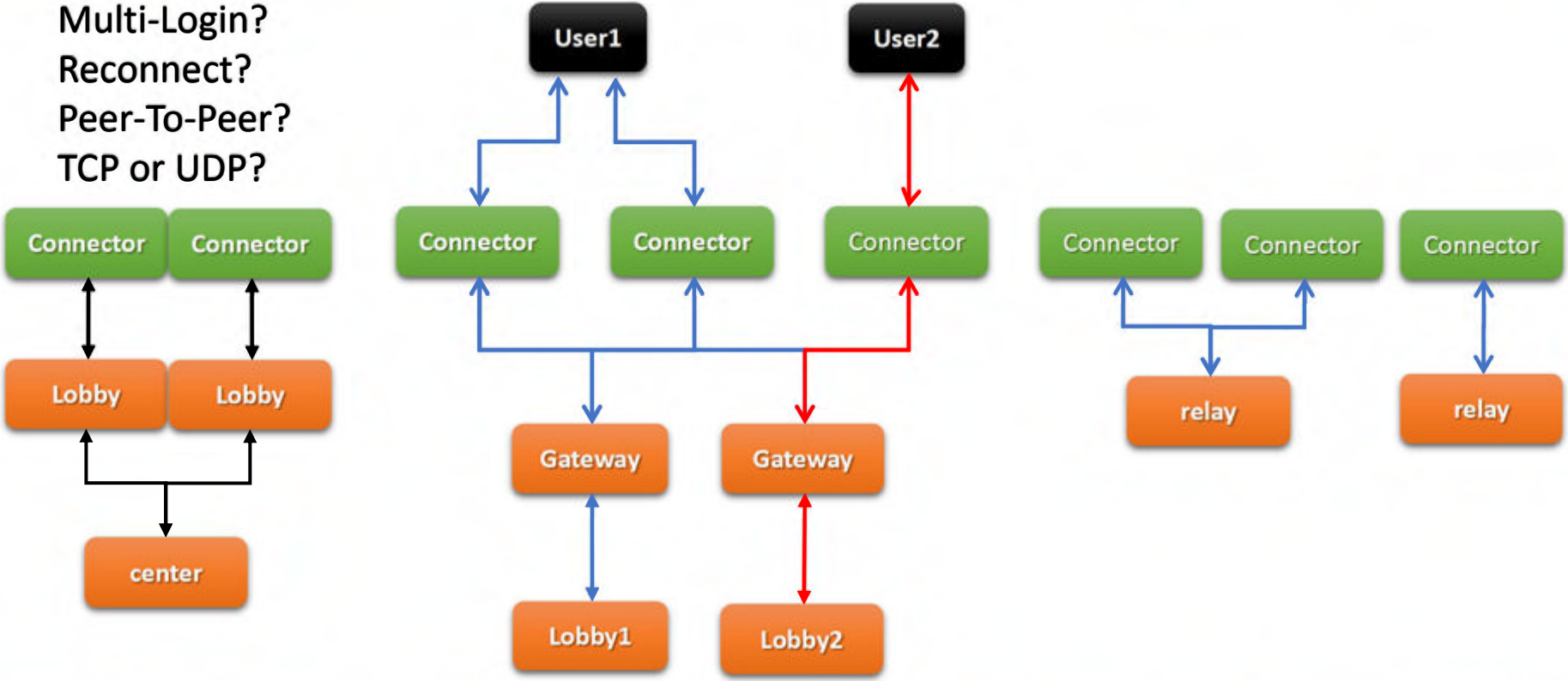

Requester
MQ
Provider
Add
Store
Confirm
REQ
Process
REP
Get
Hit
Result
Remove

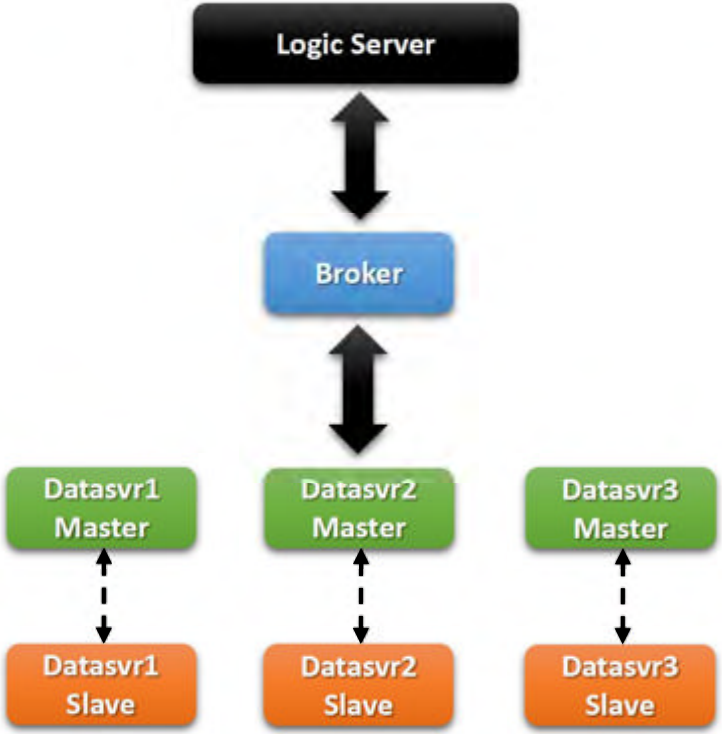

Producer
Request
Watcher
Message Queue
Router
Flow Control
UUID
Retry
Partition
Filter
Storage
Consumer
Idempotent
Sequence

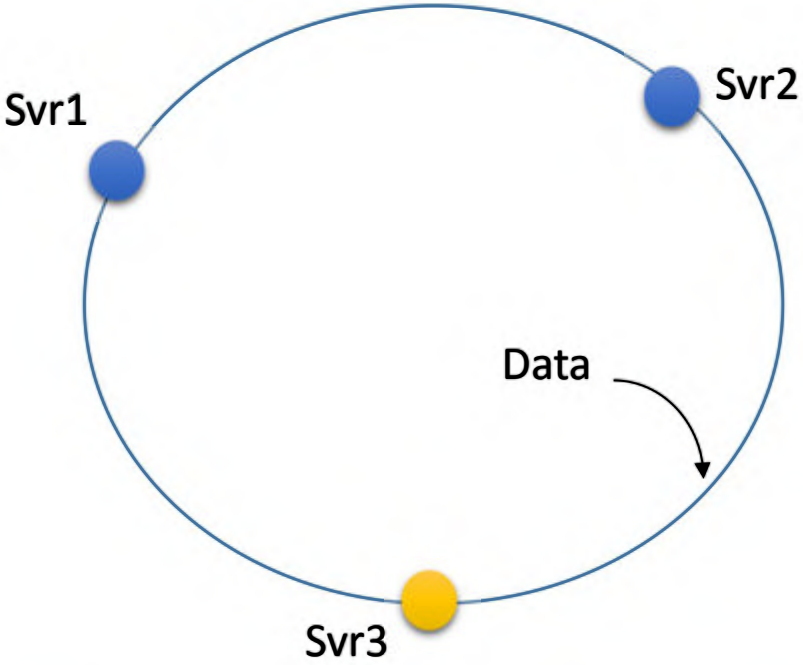
案例说明

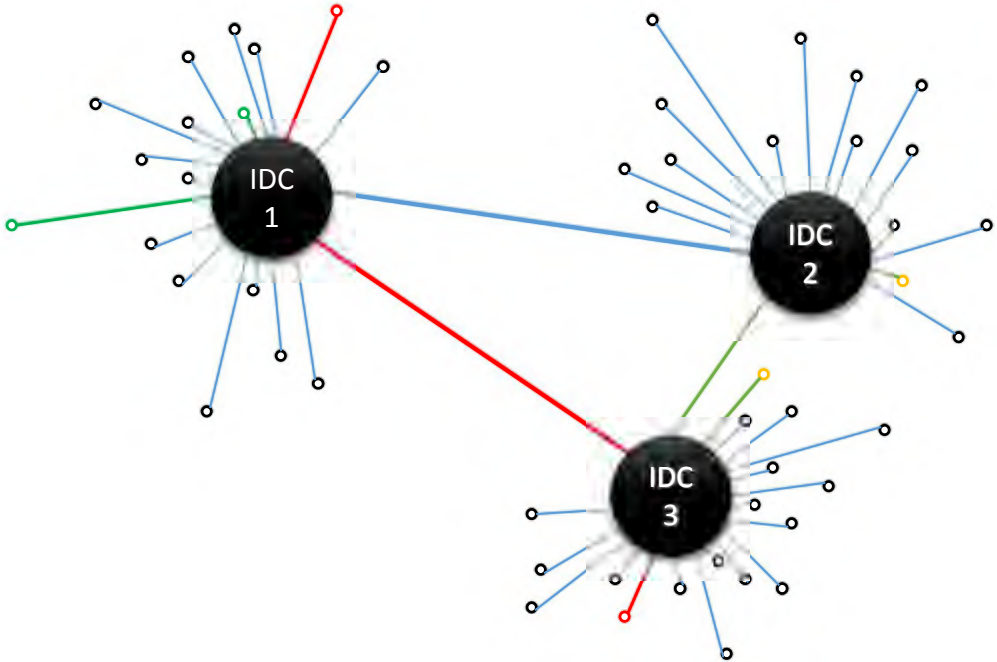
- 网络接入
- 数据存储
- 部署
- 逻辑去中心化
- 消息队列

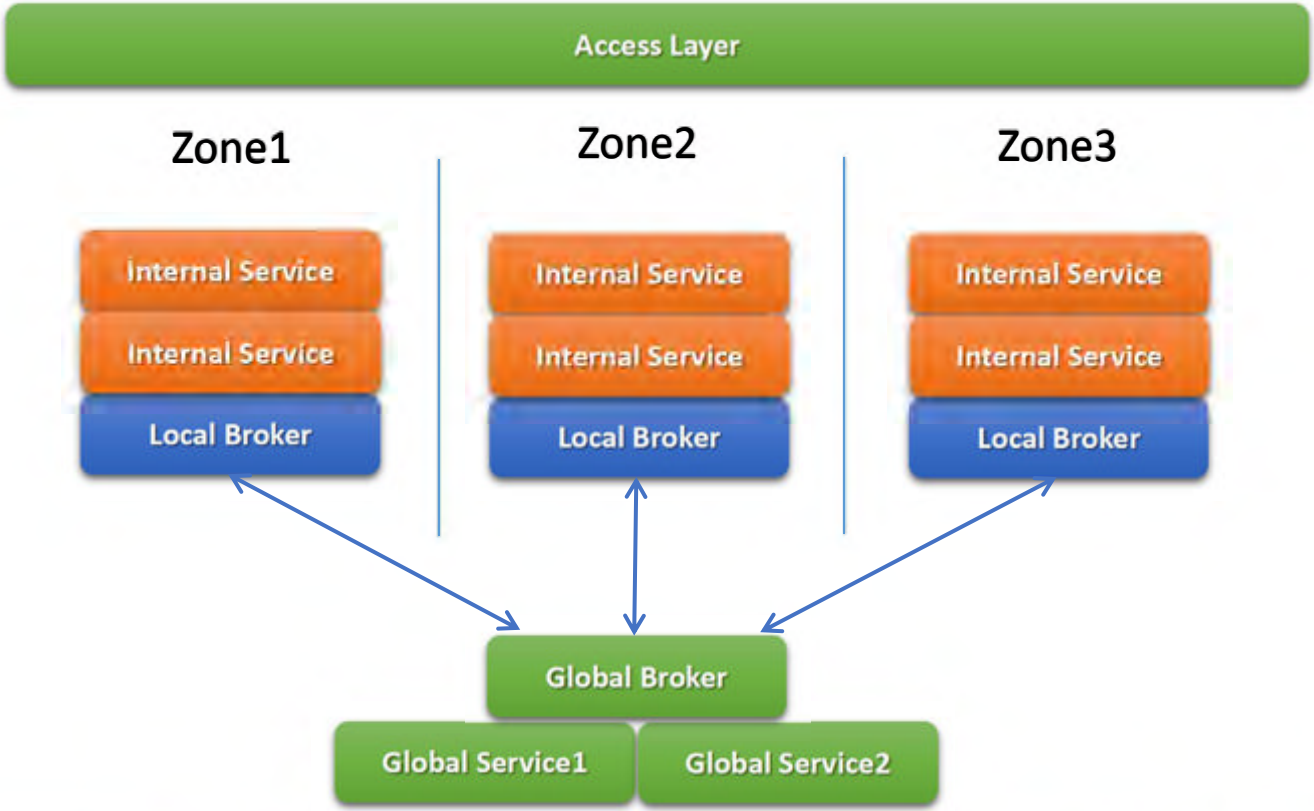
Multi-Login?
Reconnect?
Peer-To-Peer?
TCP or UDP?



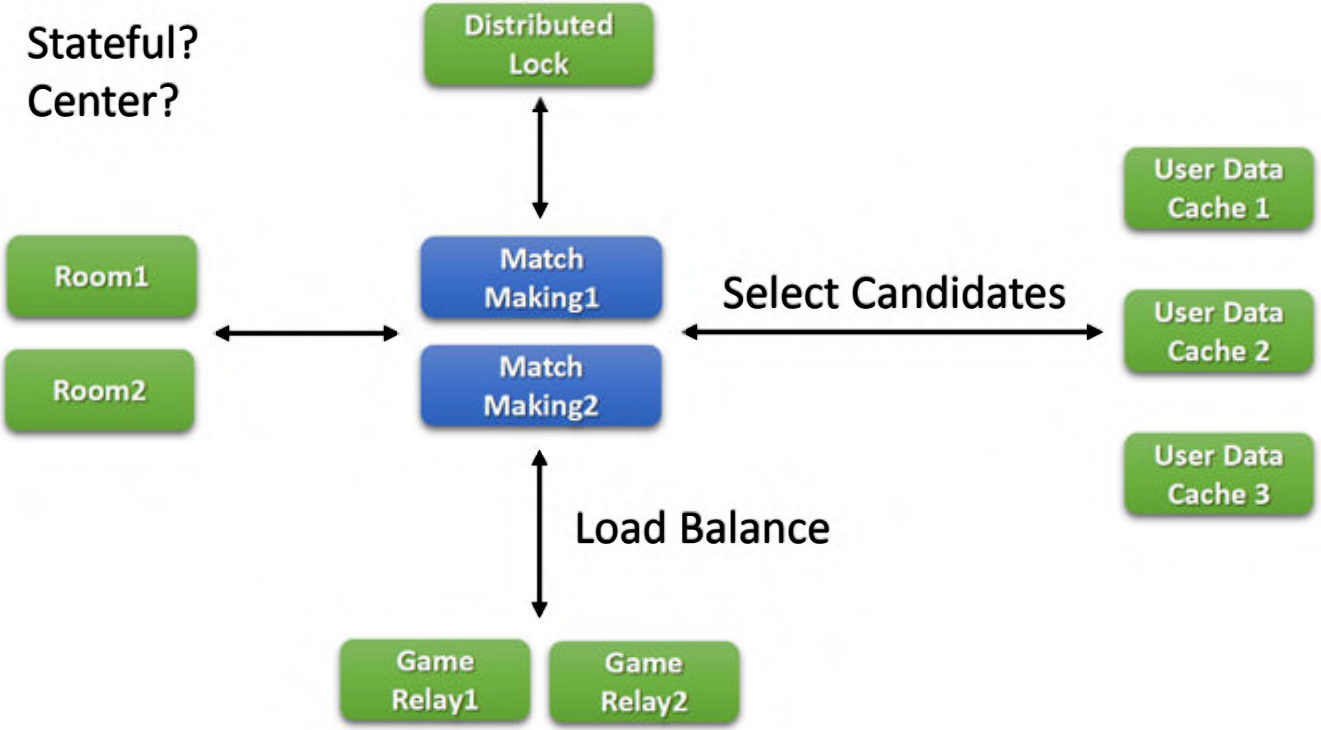

Logic Server
Broker
Datasvr1 Master
Datasvr2 Master
Datasvr3 Master
Datasvr1 Slave
Datasvr2 Slave
Datasvr3 Slave

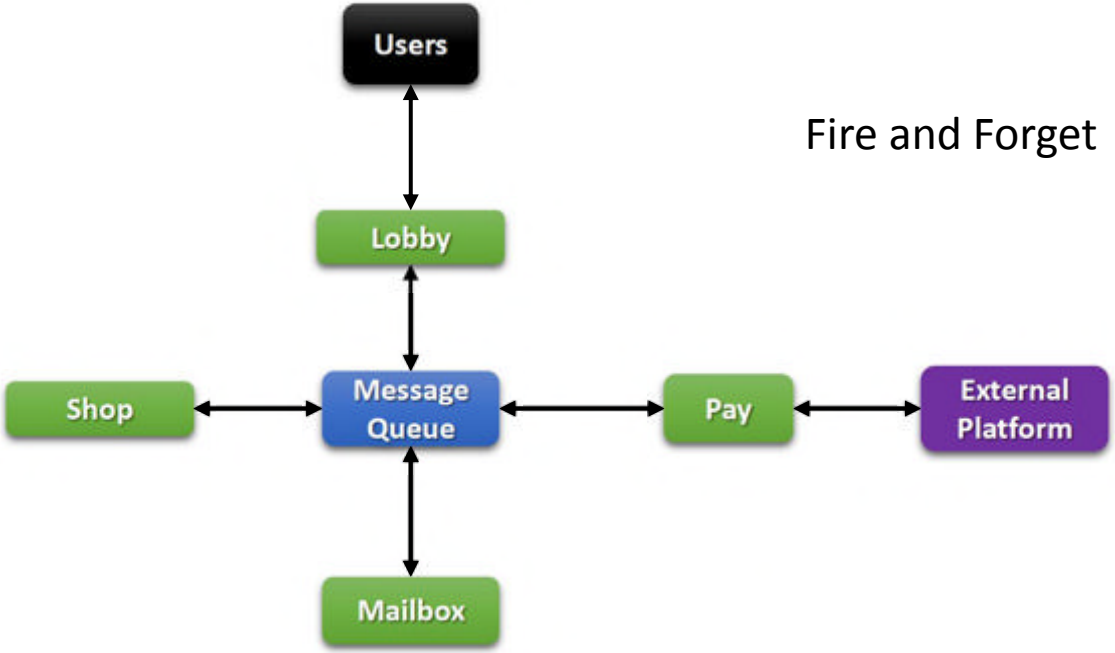

Svr1
Svr2
Data
Svr3


IDC
1
IDC
2
IDC
3


Access Layer
Zone1
Zone2
Zone3
Internal Service
Internal Service
Local Broker
Internal Service
Internal Service
Local Broker
Internal Service
Internal Service
Local Broker
Global Broker
Global Service1
Global Service2

Docker?


Stateful?
Center?
Distributed Lock
Room1
Room2
Match Making1
Match Making2
Select Candidates
User Data Cache 1
User Data Cache 2
User Data Cache 3
Load Balance
Game Relay1
Game Relay2



Fire and Forget


THANKS