乱斗西游开发经验分享

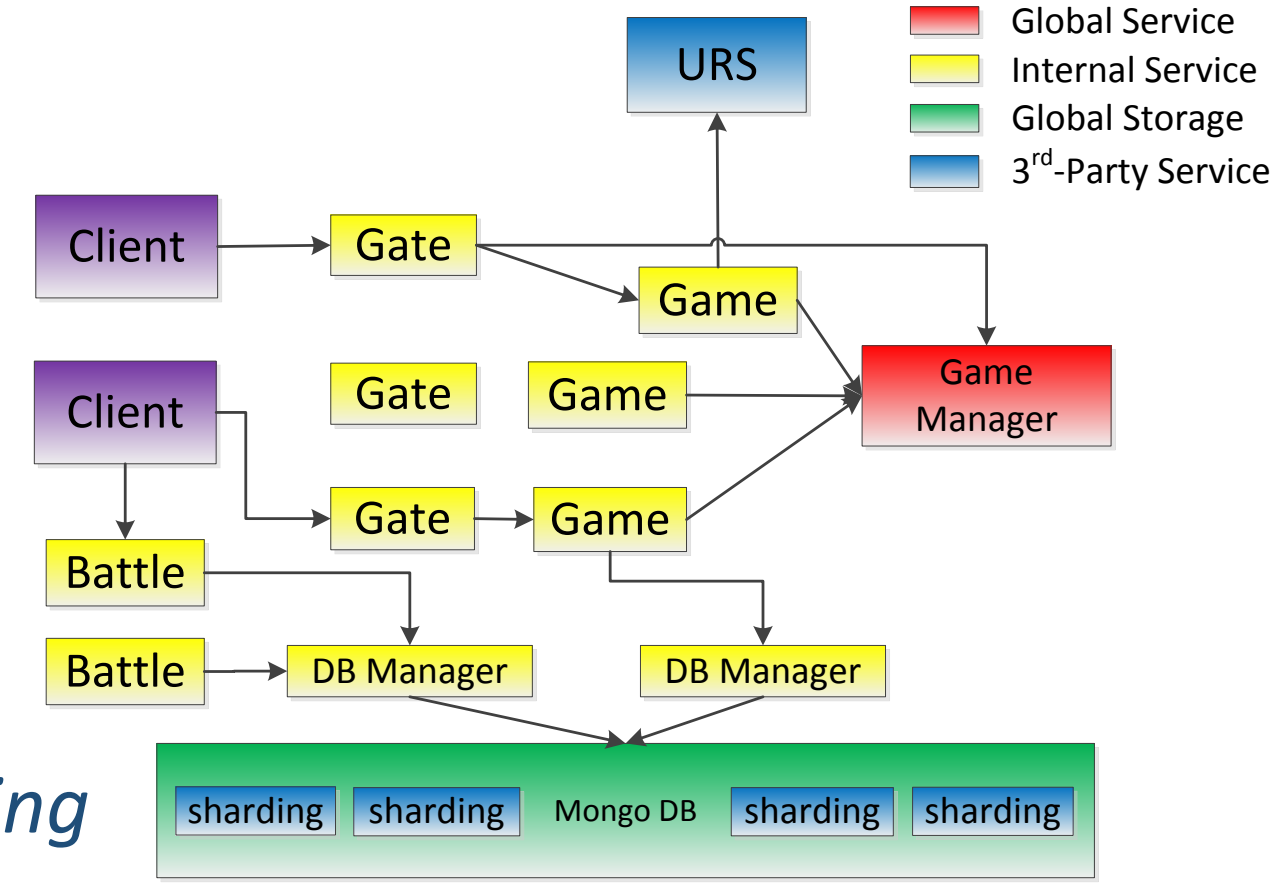
陈伊力
网易游戏 技术经理

游戏开发者大会•中国
GAME DEVELOPERS CONFERENCE CHINA
SHANGHAI INTERNATIONAL CONVENTION CENTER
SHANGHAI, CHINA · OCTOBER 25-27, 2015

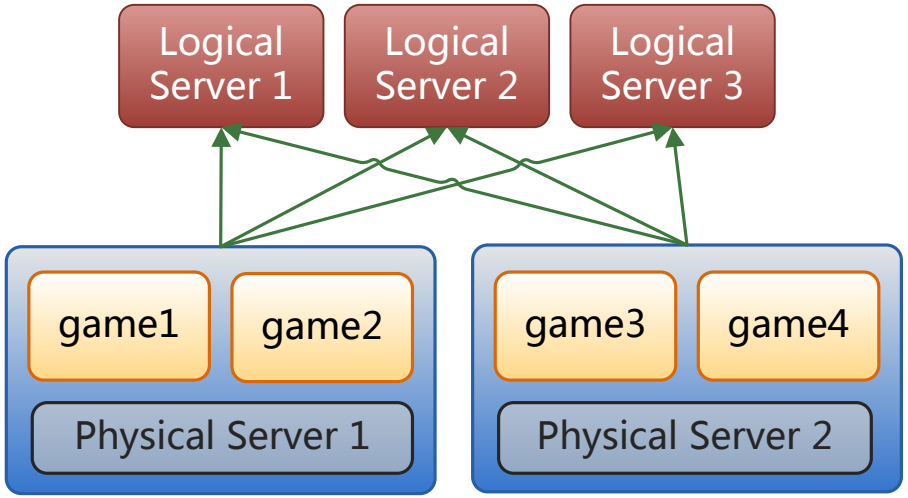
- 概要 *Outline*
  - 服务器架构 *Server Architecture*
  - 同步技术剖析 *Synchronization Techniques*
  - 总结 *Summary*

服务器架构 *Server Architecture*

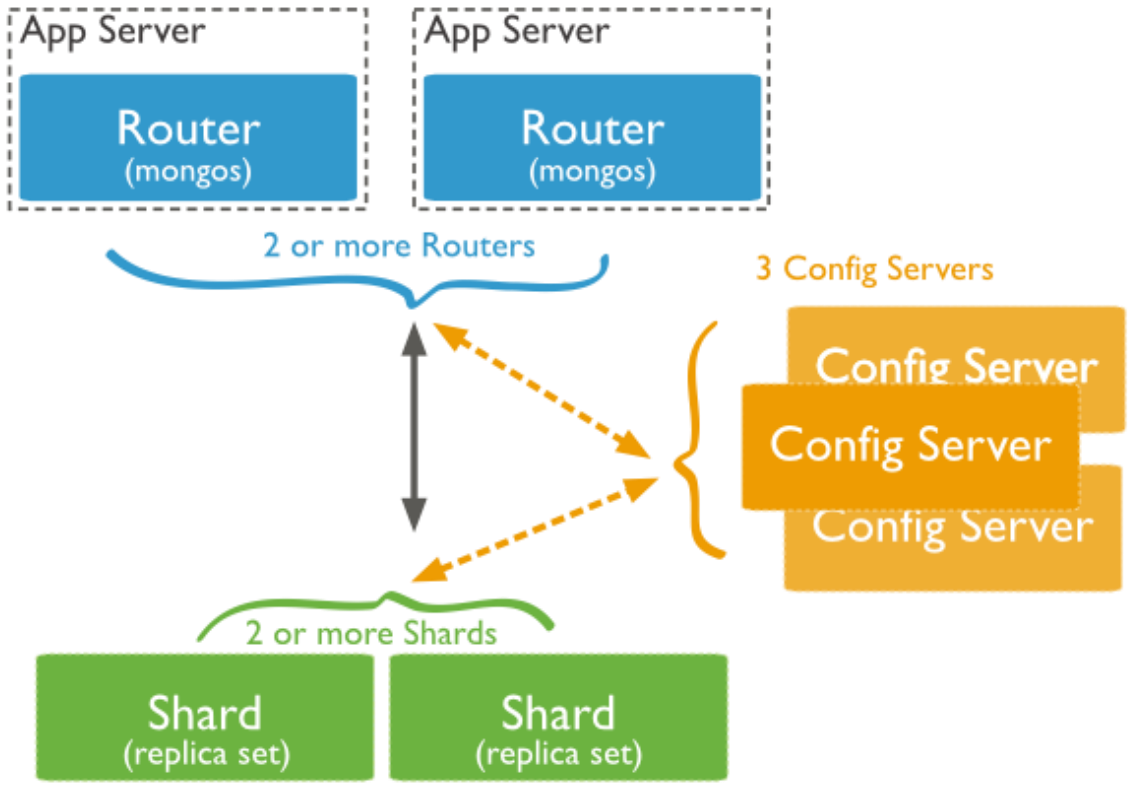
- 多服架构 *Distributed server*
  - 基于C++与Python编写
    - *Written in C++ & Python*
  - 大量应用无锁编程
    - *Lock-Free & Multi-Threaded*
  - 基于Mongo DB的存储
    - *Mongo DB based*
  - 原生支持跨服
    - *Native Cross-Server Supporting*



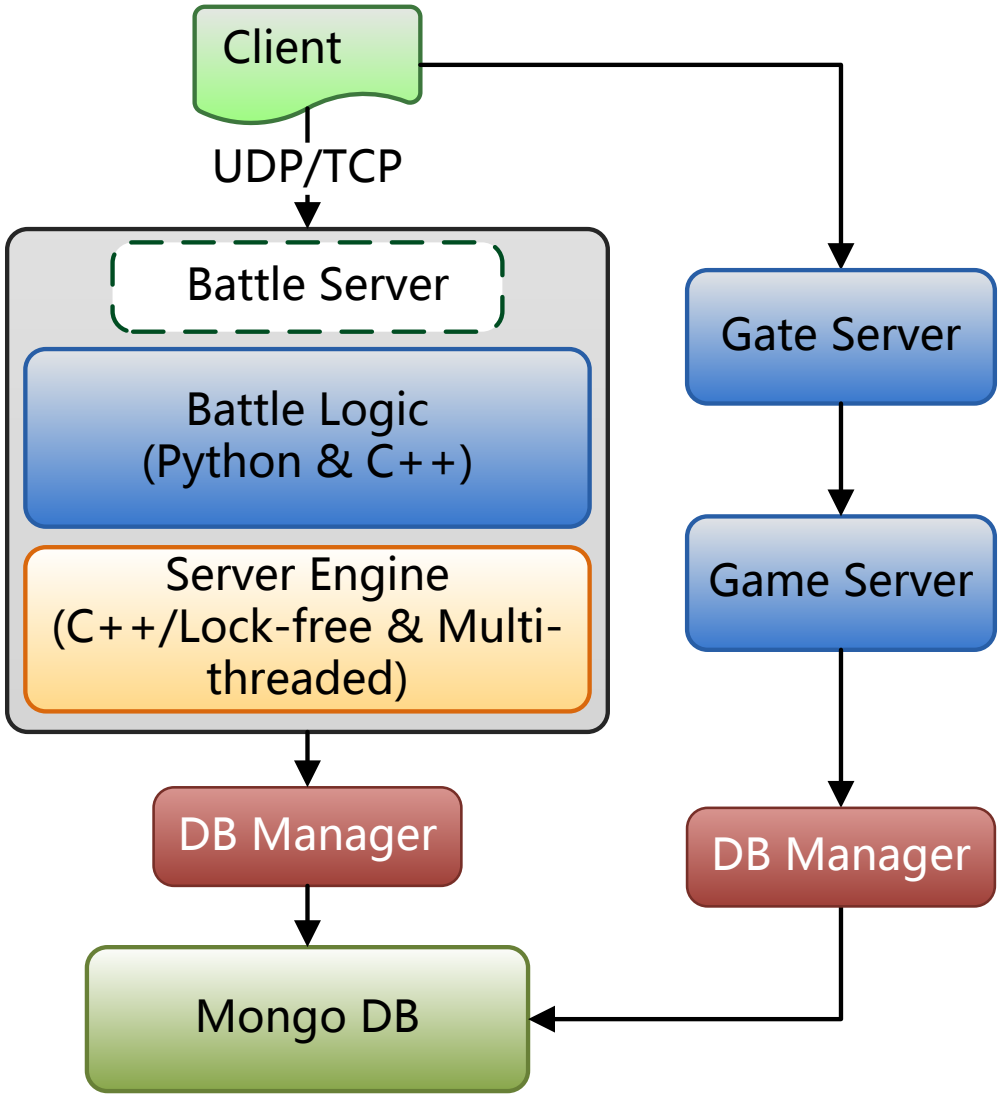
- 逻辑/物理服 *Logical/Physical server*
  - 最大化硬件利用率
    - *Maximize hardware utilization*
  - 跨服玩法更易制作
    - *Easier to cross-server*
  - 开服更加迅速
    - *Easier to extend for new server*
  - 难点—广播消息
    - *The broadcasting problem*



- 数据存储 *Data Storage*
  - Mongo DB: Sharding + Replica set
    - *Mongo DB: Sharding + Replica set*
  - 将数据完全切片（重要！）
    - *Shard everything (important!)*
  - 用于搭建部分数据管道
    - *Constructing data pipeline*
  - 隔离核心库与边缘库
    - *Data isolation*



- 战斗服 *Battle Server*
  - 全服共享战斗服务
    - *Cross-server shared battle service*
  - 基于数据库进行数据交换
    - *Database-based data exchanging*
  - 双协议—UDP/TCP
    - *Dual-protocol: UDP/TCP*



同步技术剖析 *Synchronization Techniques*

- 问题背景 *Background*
  - 行业趋势—手游的实时PVP
    - *Real-time PVP in mobile games*
  - 糟糕的网络环境
    - *Terrible network in China*
  - 重走端游的老路
    - *Just like the PC Online game's old time*
  - 始料未及的需求
    - *An unexpected task*

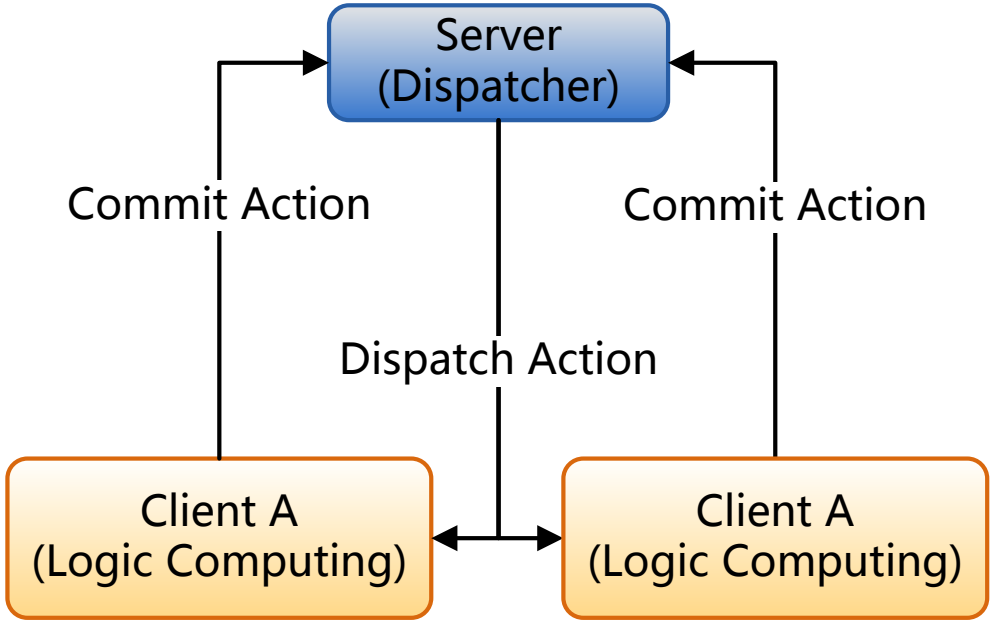
- 项目难点分析 *Project Difficulty Analysis*
  - 节奏快—行走速度快/攻击间隔短
    - *Quick-fighting*
  - 大量强制位移技能
    - *Lots of enforced-displacement skills*
  - 严格的技能释放仲裁
    - *Strict skill releasing rules*

- 需求分析 *Task Analysis*
  - 手感要好 → 客机先行
    - *Pleasing controlling feedback* → *Pre-moving*
  - 能够适应3G/4G网络 → 优化流量/战胜丢包
    - *Works on 3G/4G network* → *Overcome package loss*
  - 一致的战斗结果 → 服务器结算/帧同步
    - *Result consistency* → *Server calculation/Lockstep*

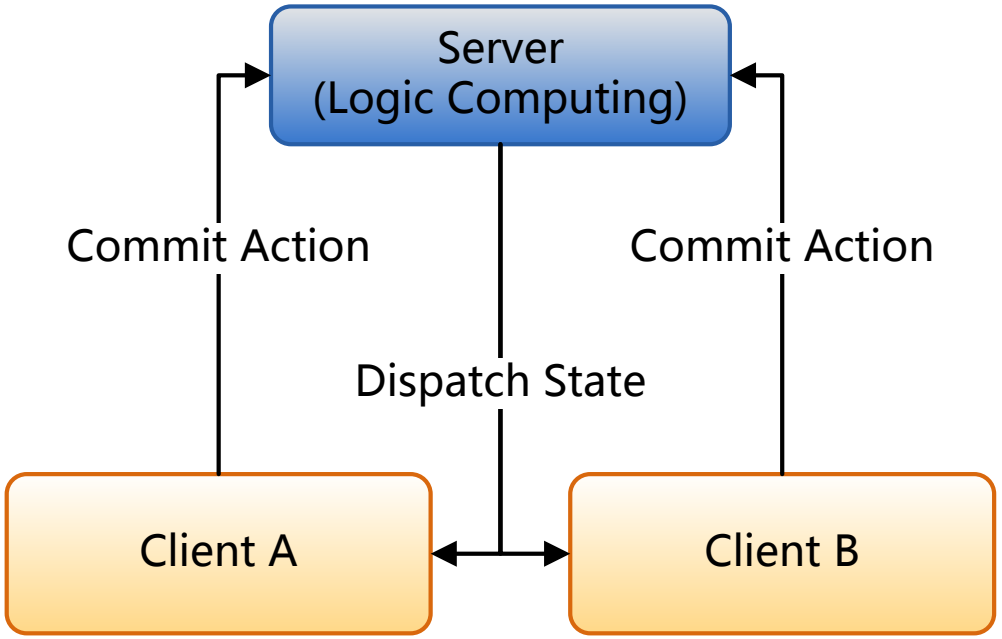
•同步机制 *Synchronization Mechanism*

•帧同步 or 状态同步？

•*Lockstep or State-Sync?*

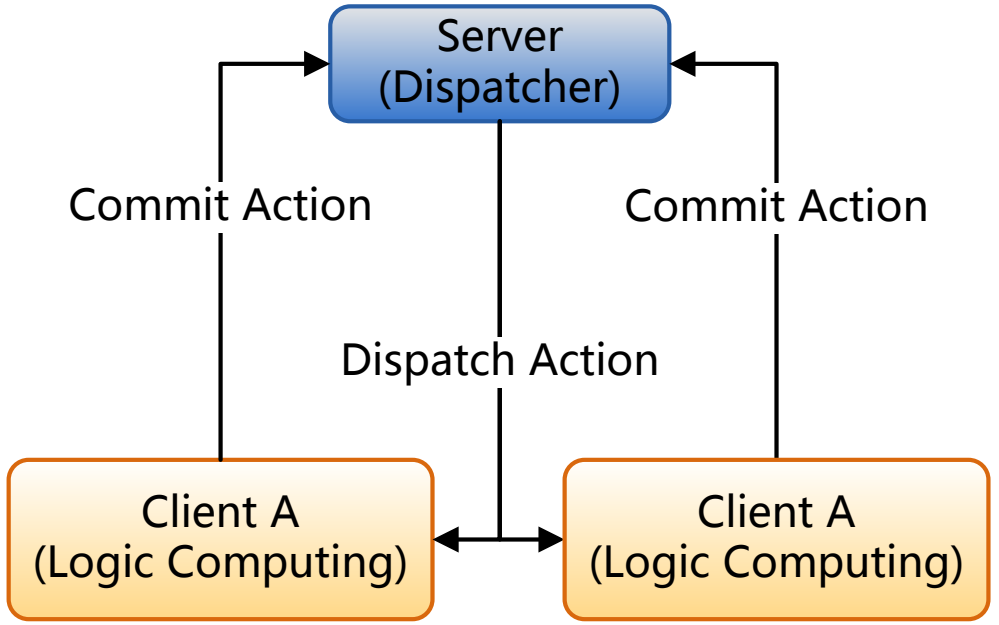


Lockstep



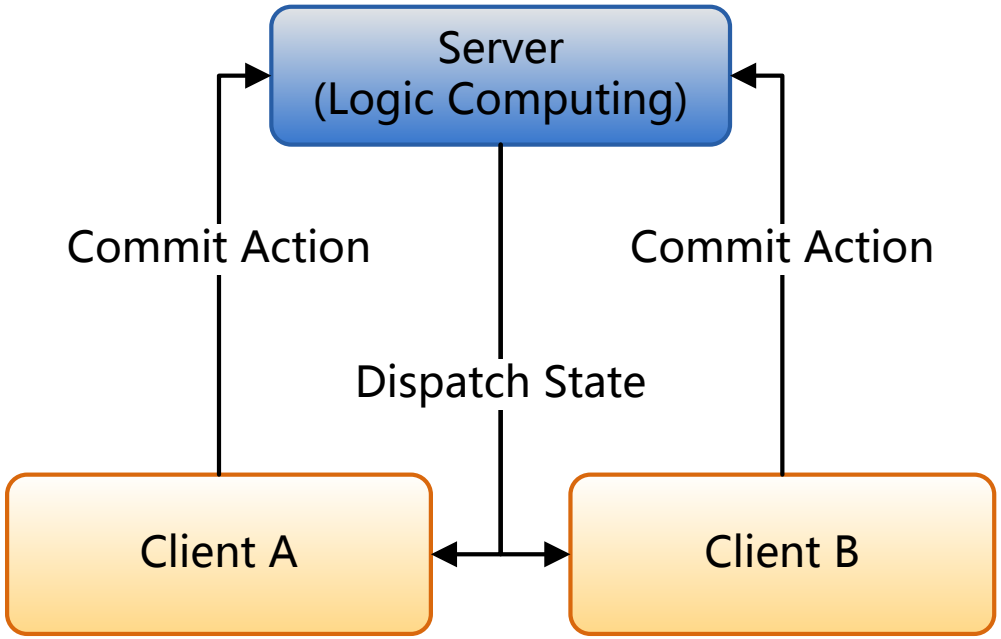
State-Sync

- 帧同步 *Lockstep*
  - 由指令驱动所有客户端
    - *Action-driven client logic*
  - 像写单机游戏一样写网游
    - *As simple as a single player game*
  - 流量消耗非常小
    - *Low network requirement*
  - 断线重连是噩梦
    - *Network dropping is a nightmare*
  - 随机因素是噩梦
    - *Zero tolerance for random*



Lockstep

- 状态同步 *State-Sync*
  - 服务端承担所有逻辑计算
    - *Action-driven client logic*
  - 流量消耗比帧同步略高
    - *Higher network requirement*
  - 断线恢复易于实现
    - *Easy to recover from network failure*
  - 天然防作弊
    - *No cheating possibility*



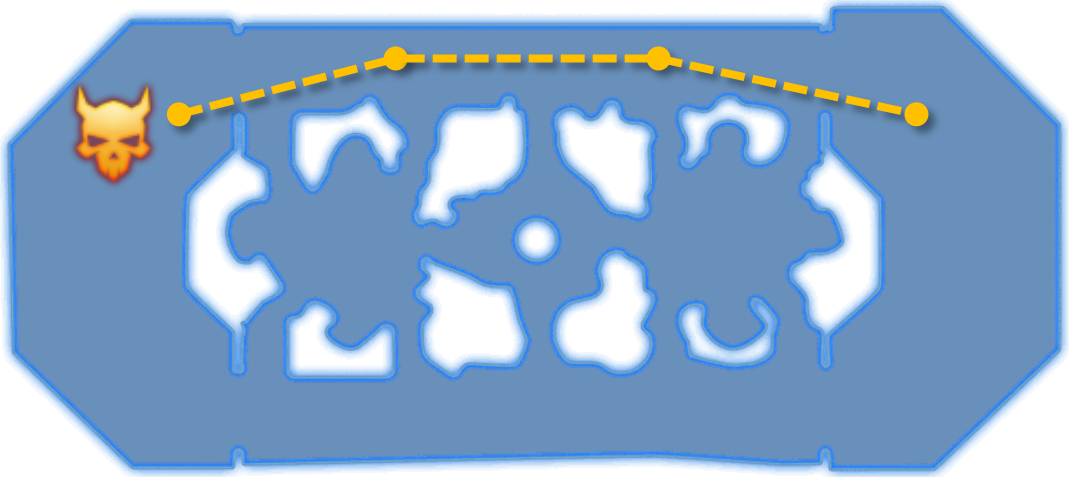
State-Sync

- 体验优化 *User Experience Optimization*
  - 客户端移动先行
    - *Client side pre-moving*
  - 客户端需要有完整的寻路系统
    - *Require client side path-finding*
  - 根据角色状态决定是否可先行
    - *Enable pre-moving according to state*

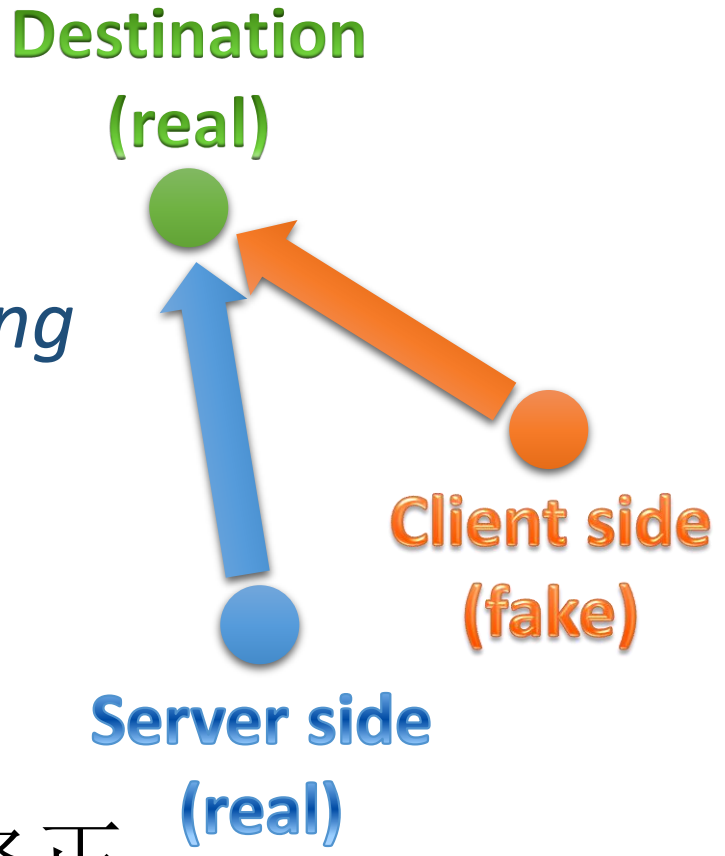
•体验优化 *User Experience Optimization*

- •宽技能释放判定
  - •*Relaxed skill releasing rules*
- •小心作弊者
  - •*Anti-cheating is necessary*
- •根据网络延迟动态调整Margin
  - •*Calculate the margin from latency*

- 体验优化 *User Experience Optimization*
  - 使用路点代替连续坐标
    - *Use way points instead of continuous coordinates*
  - 根据误差缩放行走速度
    - *Scale the moving speed according to position error*

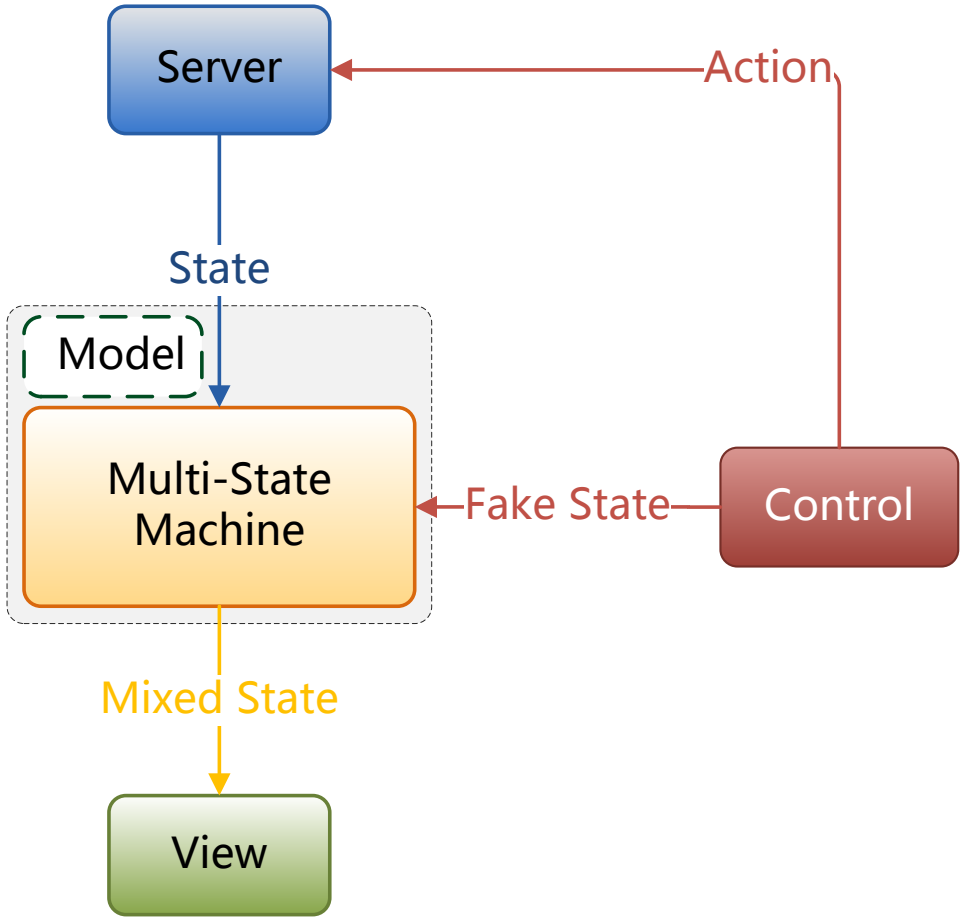


- 位置修正 *Position Correction*
  - 在角色遭受强制位移时修正
    - *Correct the position when enforced-displacing*
  - 在角色释放技能时部分修正
    - *Correct the position when releasing skill*
  - 结论：当用户合理失去控制权时修正
    - *Correct it when the user is losing control*
  - 原因：只有这时用户心理才接受你的修正
    - *It's not about rationality, it's about feeling*



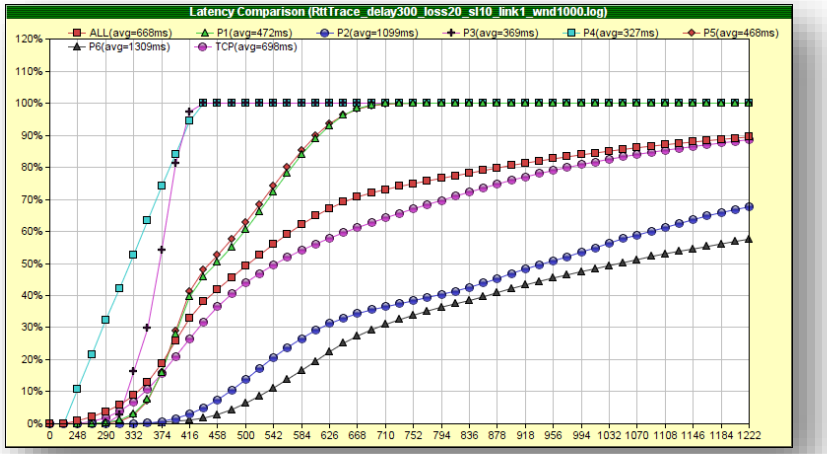
- MVC框架 *MVC Framework*
  - 双路状态混合
    - *Dual-input state mixing*
  - 利用多状态机做混合开关
    - *Multi-state Machine: mixer switch*

指令/状态	移动状态	前摇状态	后摇状态
移动	取消	中断	取消
前摇	取消	中断	取消
后摇	未定义	未定义	中断



•网络优化 *Network Optimization*

- •TCP的固有特性 *TCP's inherent features*
  - •拥塞控制 *Congestion control*
  - •完全可靠 *Completely reliable*
- •我们想要的 *What we need*
  - •重发频率可控 *Controllable retransmission*
  - •部分协议允许丢包 *Partial reliable protocol*
- •最终选择 *Our choice*
  - •基于UDP实现半可靠协议 *UDP-based semi-reliable protocol*
- •环回测试 *Loopback testing*



Type/PLR	5%	15%	25%
TCP	9.09s	64.61s	N/A
UDP	3.84s	13.47s	20.25s

- 总结 *Summary*
  - 拥抱横向架构
    - *More scale-out*
  - 用户体验为王，而非正确性
    - *Target to user experience, not correctness*
  - 永远冒险
    - *Always put yourself in risk*

谢谢~ *Thank you~*