

OSC原创会

年终盛典 2016

野狗与web实时化

--聊聊架构与思想



架构：前端 VS 后端

通信管道：WebSocket VS HTTP

接口设计：面向资源 VS 面向活动

同步：CAP的取舍

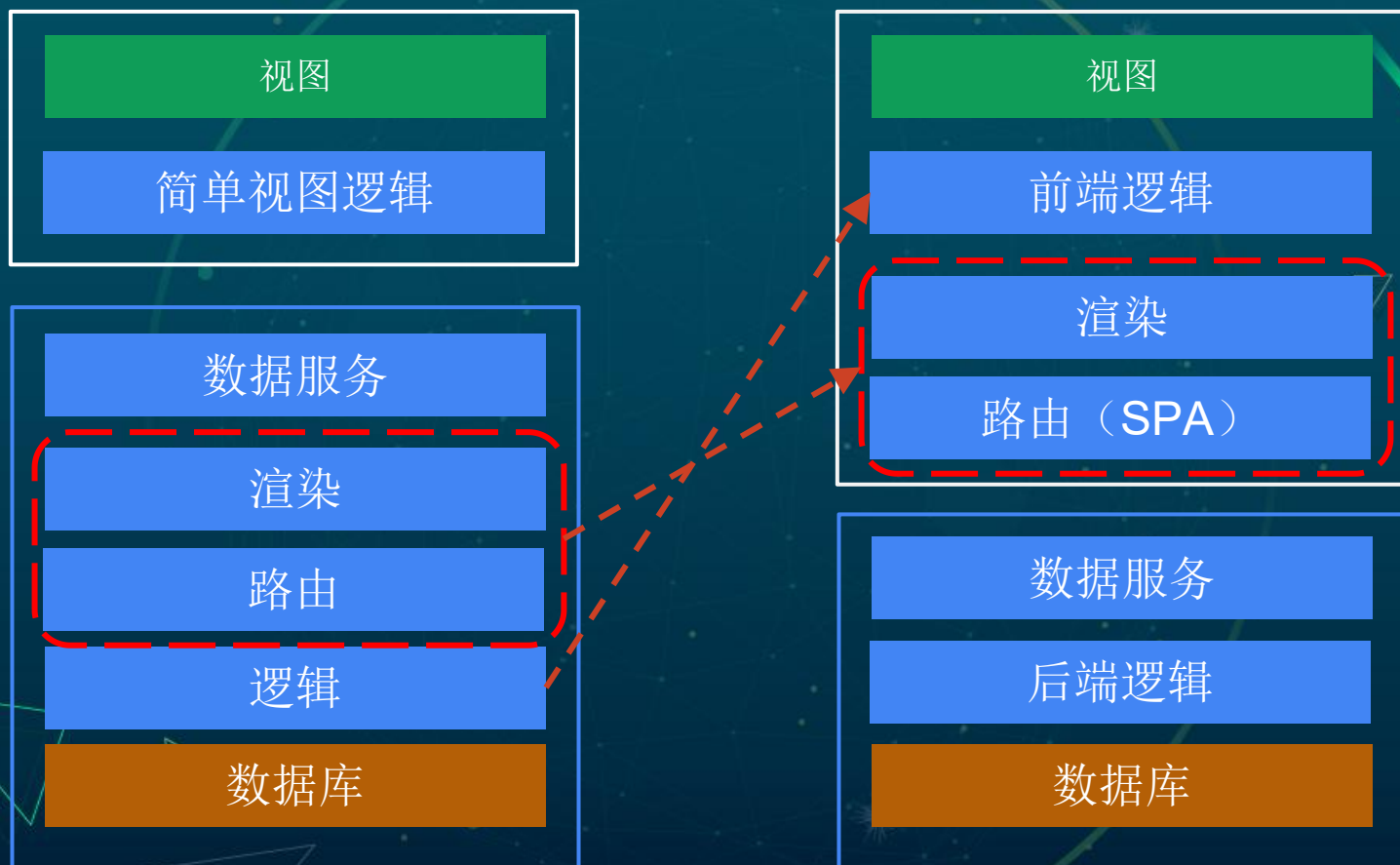
交互：乐观策略与悲观策略

集成：野狗与MV* 框架的集成

认证与授权



架构：前端 vs 后端



优势

OSC 原创会
年终盛典 2016

文档请求减少

资源的请求减少

数据与静态资源分离有利于cdn缓存

客户端渲染分担服务端压力

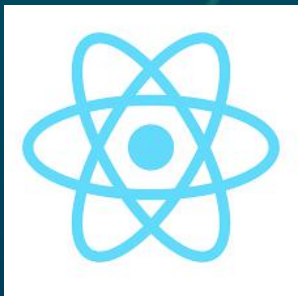
客户端有状态，减少交互等待

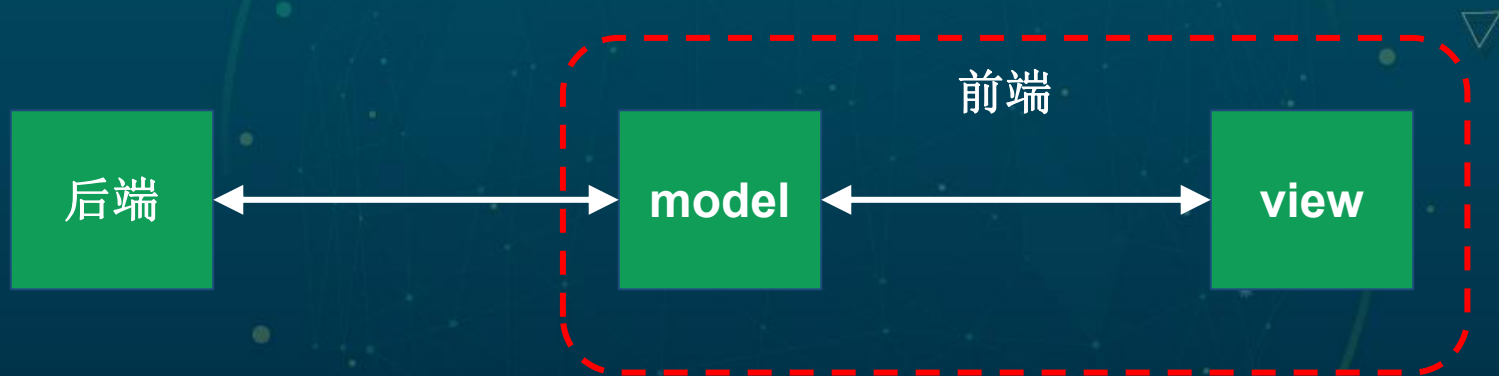
客户端有状态，容易实现复杂交互

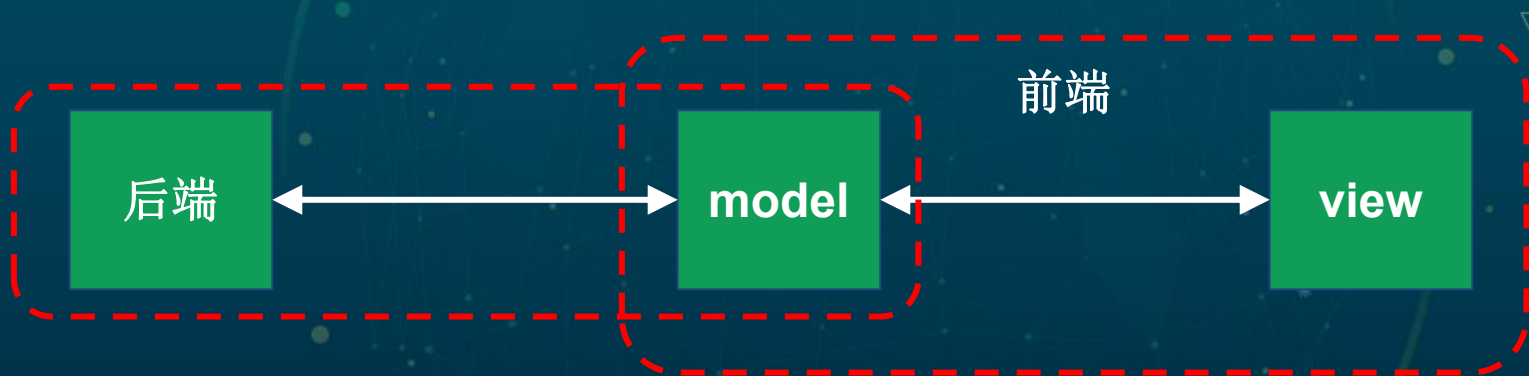


前端框架的繁荣

OSC 原创会
年终盛典 2016







对于这部分我们思考的并不
够
没有提升空间吗？
能否用类似数据绑定的思想
做网络交互？

非常多前端框架已经帮你搞
定了这部分
Model 与 **view** 之间是数据绑
定的



通信管道：HTTP vs WebSocket

OSC 原创会
年终盛典 2016

HTTP的劣势：

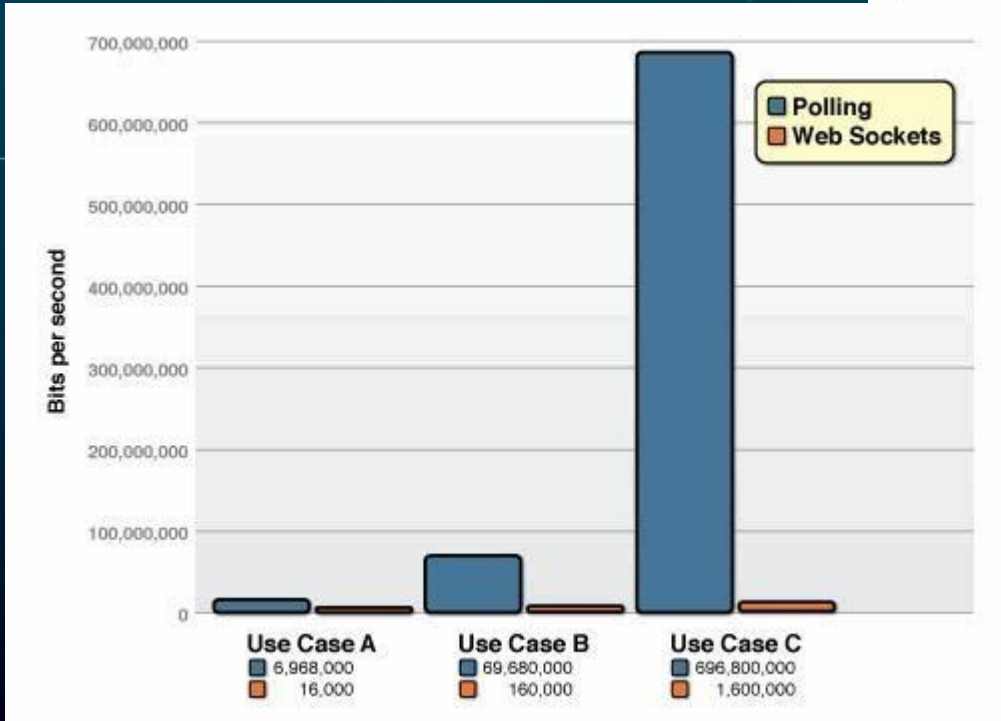
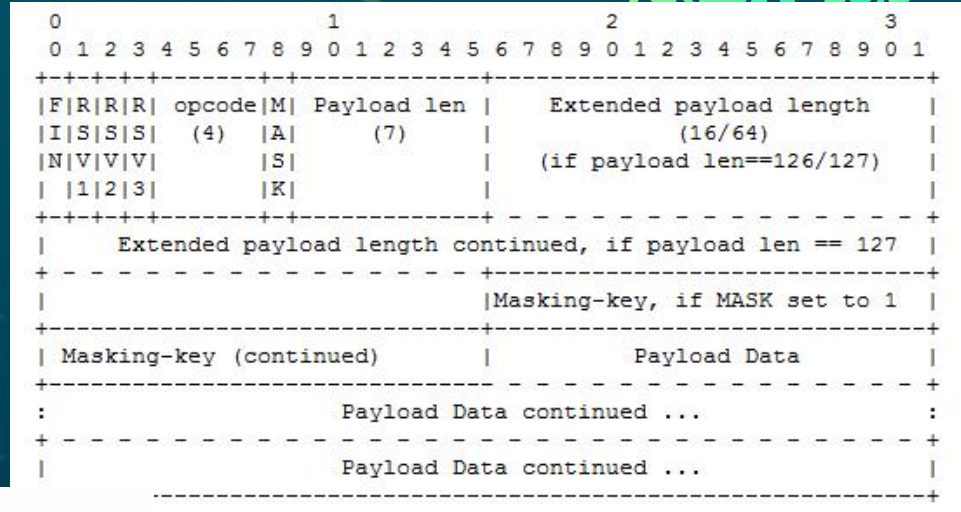
- 1) 单向连接，服务端无法主动推送
- 2) 无状态导致状态要重复传输
- 3) 文本协议，头巨大，有效载荷小

——以上HTTP的劣势正是WebSocket的优势



WebSocket

WebSocket frame



WebSocket 与 polling 性能对比：
1000/10000/100000 客户端每秒接收一条消息产生的网络开销

结论

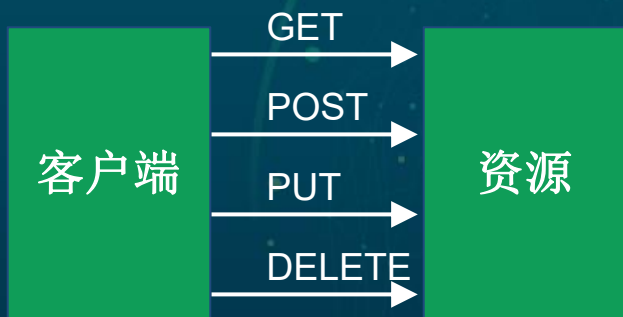
OSC 原创会
年终盛典 2016

WebSocket 更适合数据交互
Http 更适合资源下载



接口设计：面向资源 vs 面向活动

OSC 原创会
年终盛典 2016

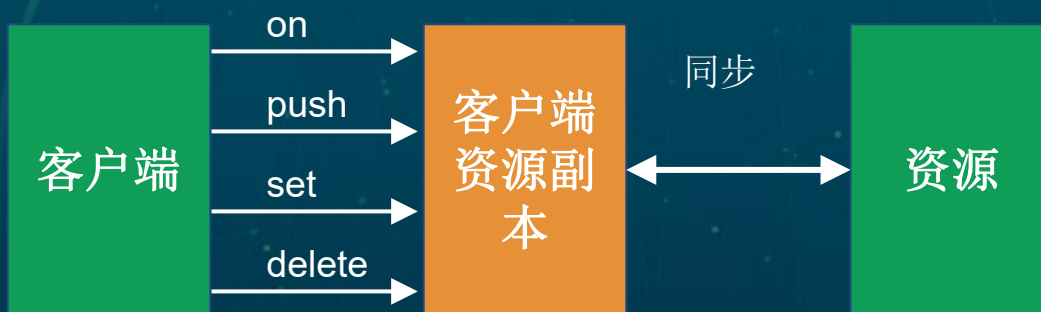


▶ 面向资源无需定义接口，只需定义资源路径和读写权限
适用多数场景，活动也可以使用资源的方式来描述



面向资源与同步

OSC 原创会
年终盛典 2016



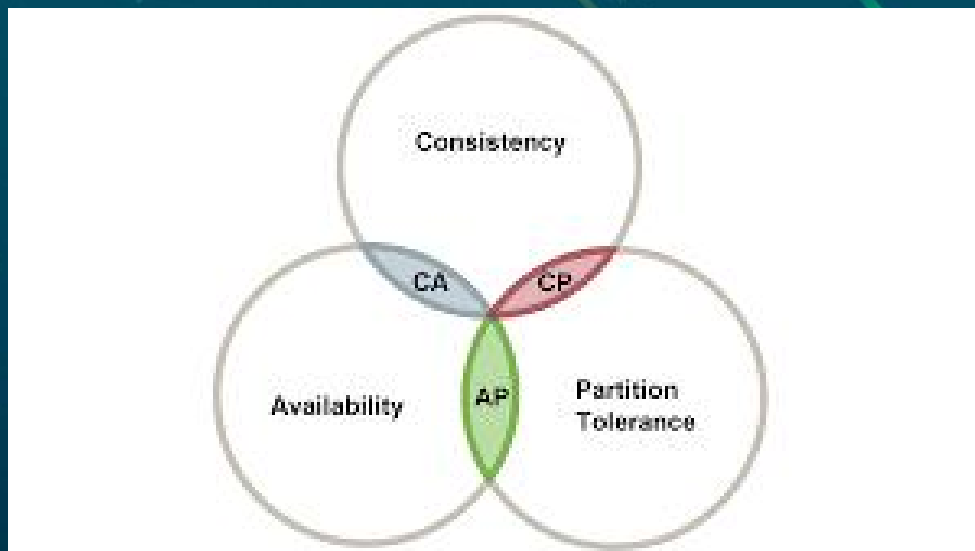
客户端不直接操作资源，而是操作客户端资源副本
客户端可以通过数据事件获得最新的数据

对可用性的追求-离线可用

OSC 原创会
年终盛典 2016

CAP:

野狗是 AP 系统



分区容忍：客户端离线

可用性：任何时候都可以操作客户端副本

一致性不可以放弃

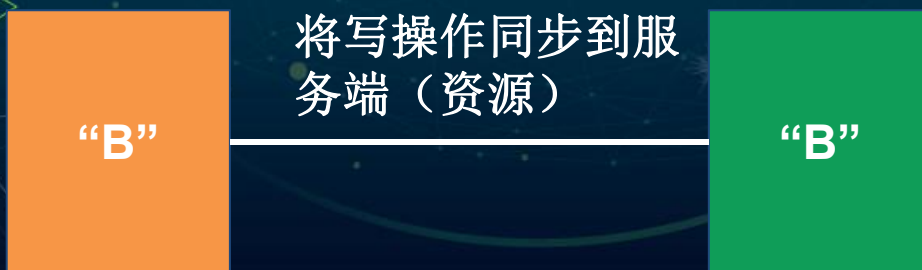
connected



disconnected



reconnected



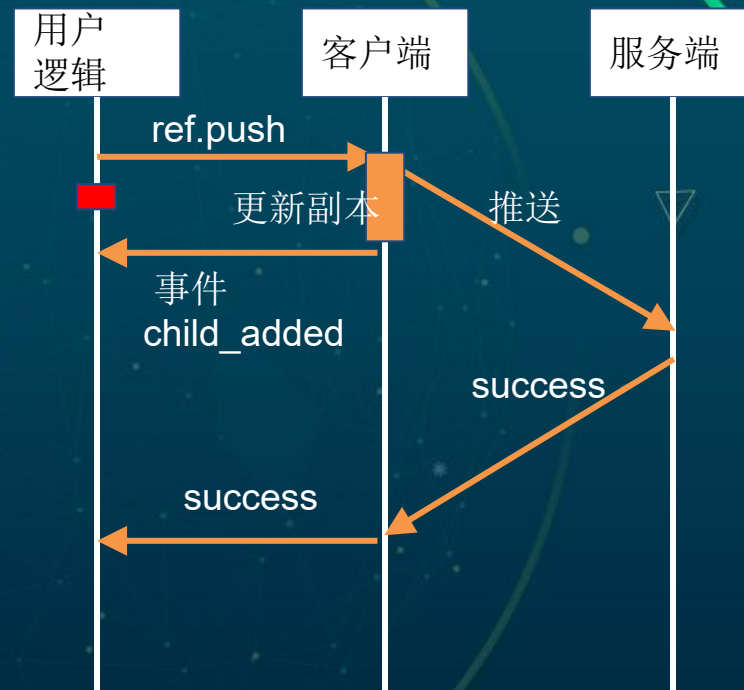
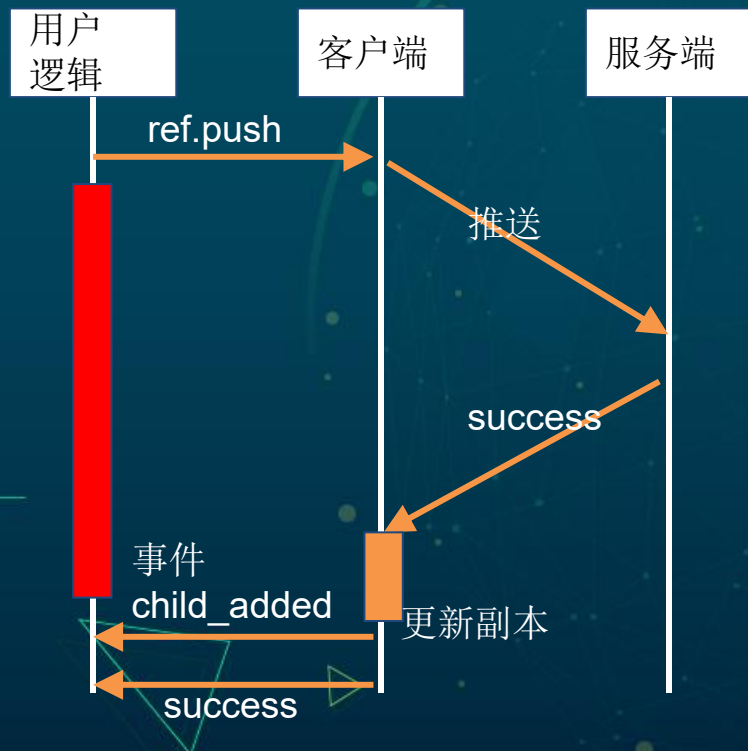
考虑场景:

```
//wilddog 已经初始化
ref.on('child_added', (snapshot) => {
    addToView(snapshot.key(), snapshot.val()) //如果有新的消息, 把这条消息添加到视图中
})
//某个按钮
button.on('click', () => { //点按钮的时候往ref所代表的资源位置新增一条数据
    ref.push('this is a new item')
})
```



乐观策略与悲观策略-正常

OSC 原创会
年终盛典 2016

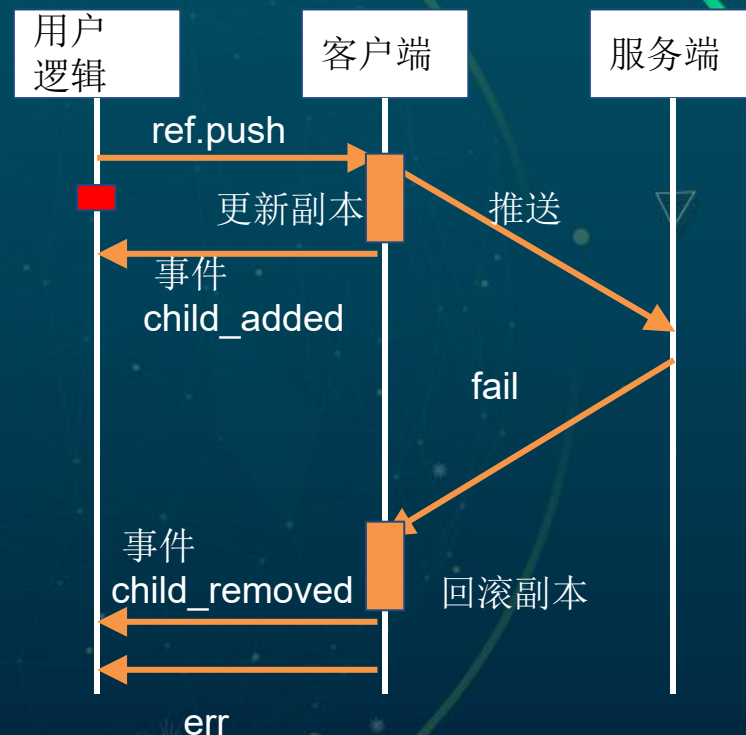
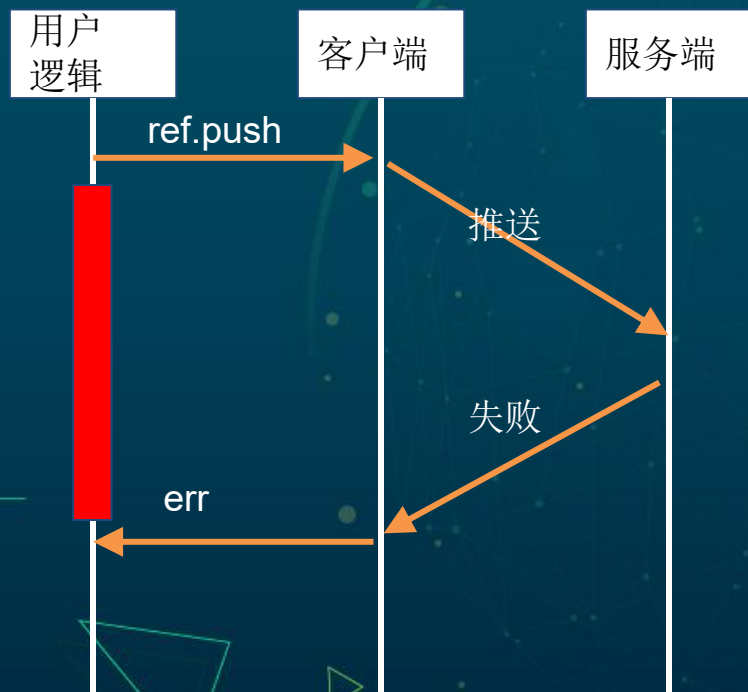


悲观策略：默认操作会不成功，直到收到服务端 `success` 的通知才给用户发送数据变化的通知

悲观策略：默认操作会成功，直接给应用发送数据改变的事件。

问题：如果操作不成功怎么办？

乐观策略与悲观策略-失败



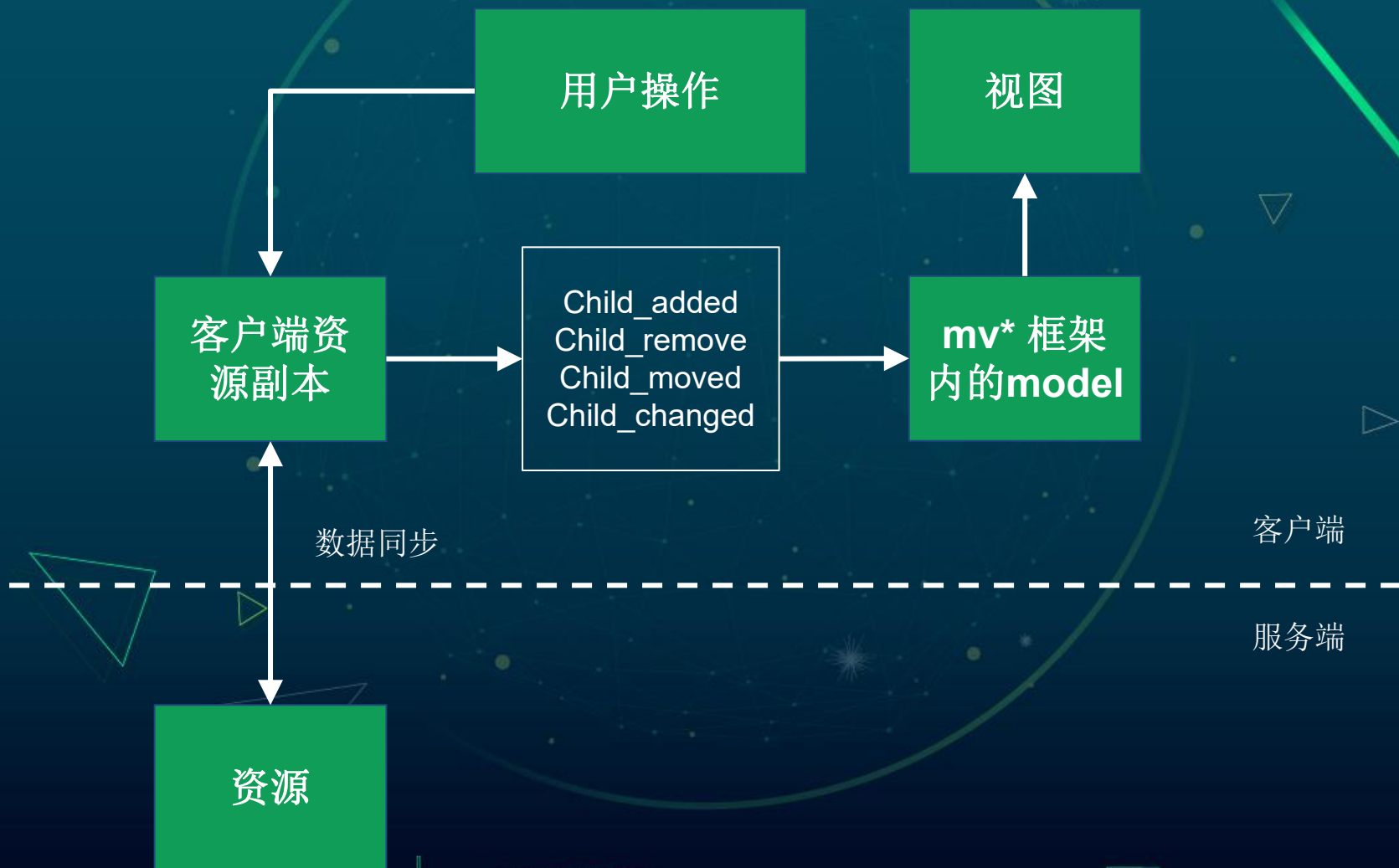
悲观策略：直接把错误结果给用户逻辑就好

悲观策略：要为乐观付出代价，本地副本需要回滚，而且需要通知用户逻辑回滚

```
//wilddog 已经初始化
ref.on('child_added',(snapshot) => {
    addToView(snapshot.key(), snapshot.val())//如果有新的消息, 把这条消息添加到视图中
})
ref.on('child_removed',(snapshot) => {
    removeFromView(snapshot.key())//如果某条数据被删除, 更新view
})
//某个按钮
button.on('click',() => { //点按钮的时候往ref所代表的资源位置新增一条数据
    ref.push('this is a new item',(err) => {
        if(err){/*处理出错信息*/}
    })
})
})
```

数据同步与MV* 的结合

OSC 原创会
年终盛典 2016



举个栗子-微信小程序

OSC 原创会
年终盛典 2016

```
//index.js
var app = getApp()

Page({
  ...
  onLoad: function () {
    wilddog.sync().ref('todo').bindAsArray(this, 'todo')
  }
  ...
})

<!--index.wxml-->
<view wx:for = "{{todo}}">
  {{item[".key"]}}-{{item[".value"]}}-{{item[".priority"]}}
</view>
```

认证：让服务端知道我是谁

JSON-WEB-TOKEN (JWT)

授权：根据用户信息和资源信息判断读写权限 ▽

```
{
  "rules":{
    ".read":true,
    "$post":{
      ".write": "data.val()===null||data.author === auth.uid",
      ".validate": "newData.val().hasChildren(['author','content','title'])"
    }
  }
}
```

<http://jack-x.engineer>



官网: <https://www.wilddog.com>

Github: <https://github.com/WildDogTeam>

Issues: <https://github.com/WildDogTeam/wilddog-issue/issues>

