

# 协同编辑系统设计

主讲人 南帝

过去

外企、国企、互联网

10+年后端研发架构经验

面试官100+

现在

阿里巴巴

研发、架构师

面试官30+

未来

To be continued

	<ul style="list-style-type: none"><li>• 【互动】走进系统设计 &amp; 新鲜事系统设计</li><li>• 【直播】秒杀系统与订票系统设计</li></ul>
Week1	<ul style="list-style-type: none"><li>• 【互动】从用户系统中学习数据库与缓存</li><li>• 【互动】网站系统、API设计与短网址</li><li>• 【直播】容器技术 (K8S, Docker)</li></ul>
Week2	<ul style="list-style-type: none"><li>• 【互动】数据库拆分 &amp; 限流器和实时数据系统设计</li><li>• 【互动】分布式数据库系统设计</li><li>• 【直播】在线文档协同编辑系统设计</li></ul>
Week3	<ul style="list-style-type: none"><li>• 【互动】聊天系统设计</li><li>• 【互动】分布式文件系统设计</li><li>• 【直播】视频流系统设计 (Youtube / Netflix)</li></ul>
Week4	<ul style="list-style-type: none"><li>• 【互动】基于地理位置的信息系统</li><li>• 【互动】分布式计算系统设计</li><li>• 【互动】爬虫系统与搜索建议系统</li><li>• 【直播】Google Search / Search Ads Ranking 系统设计</li></ul>



WebSocket  
FileSystem  
Redis  
Collaborative  
Mysql CRDT  
Domain OT  
Javascript  
Database

多人同时在线编辑同一份文档，比如Google Docs

- Google Docs
- Office 365
- Quip
- 石墨

- Open&Loading File
- Editing File
- Saving File



- Open&Loading File
- Editing File
- Saving File



- Collaboration
- Display Co-Authoring
- Who update
- Content lock

同时编辑，多个客户端之间如何同步数据？

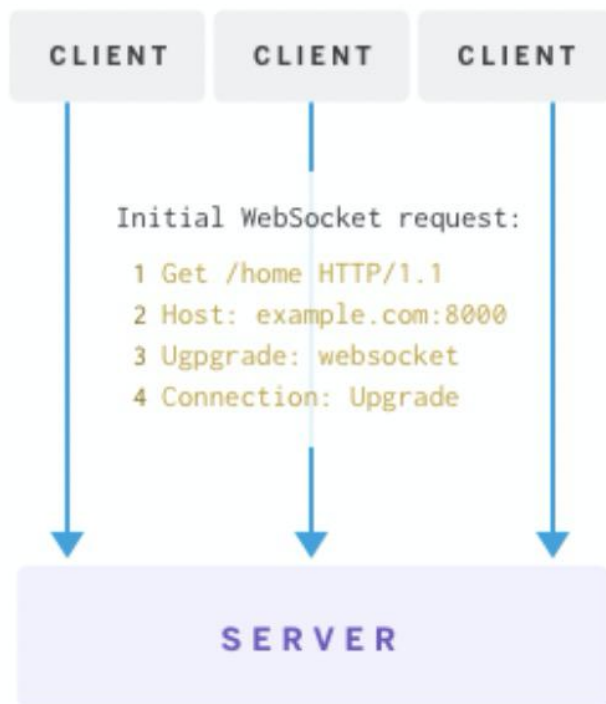
A: websocket

B: HTTP1.0

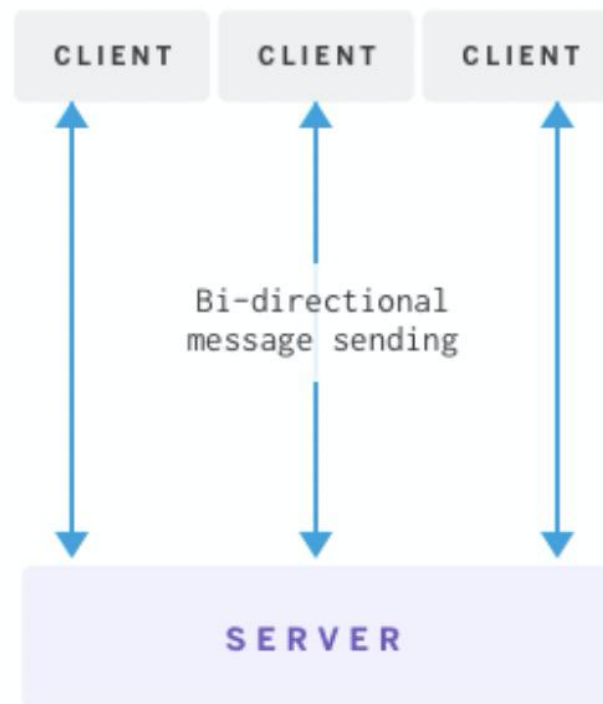
C: HTTP2.0

## WEBSOCKET PROTOCOL

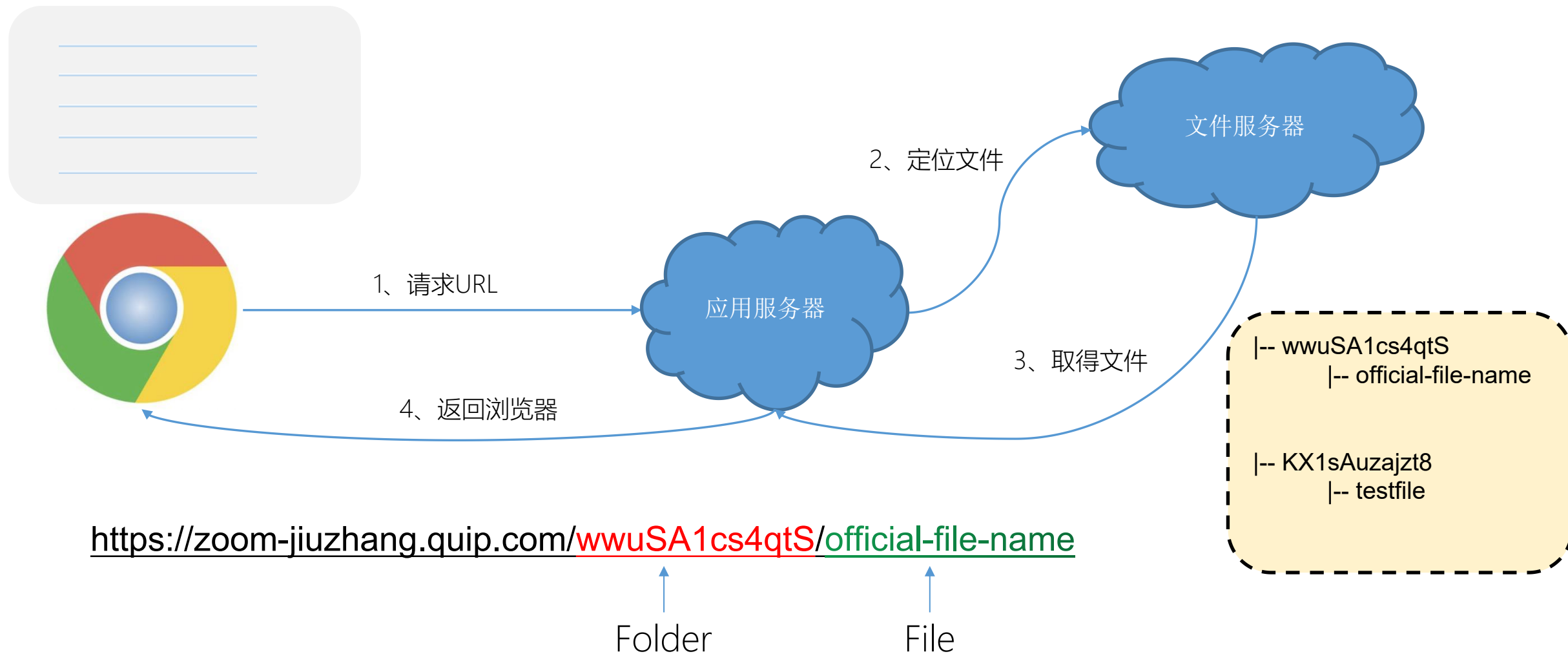
**Step 1:** A Request to initiate a WebSocket connection is sent to the server, from any number of clients



**Step 2:** Open and maintain WebSocket connections between multiple clients and a single server



# 文件是如何locate&load的?



<https://zoom-jiuzhang.quip.com/wwuSA1cs4qtS/official-file-name>

↑  
Folder

↑  
File

## 文件目录什么时候创建?

- Upload File/New File

## 文件内容在服务端以什么样的数据结构保存？

A: 数组(Array)

B: 双向链表+哈希表(LinkedHashMap)

<http://lintcode.com/problem/lru-cache>

## 为何要用双向链表存放数据呢？

- 链表的优势在任意位置插入或者删除，性能好
- 文件中每一行的数据匹配链表的一个节点
- 内容的变化相当于链表节点的变化



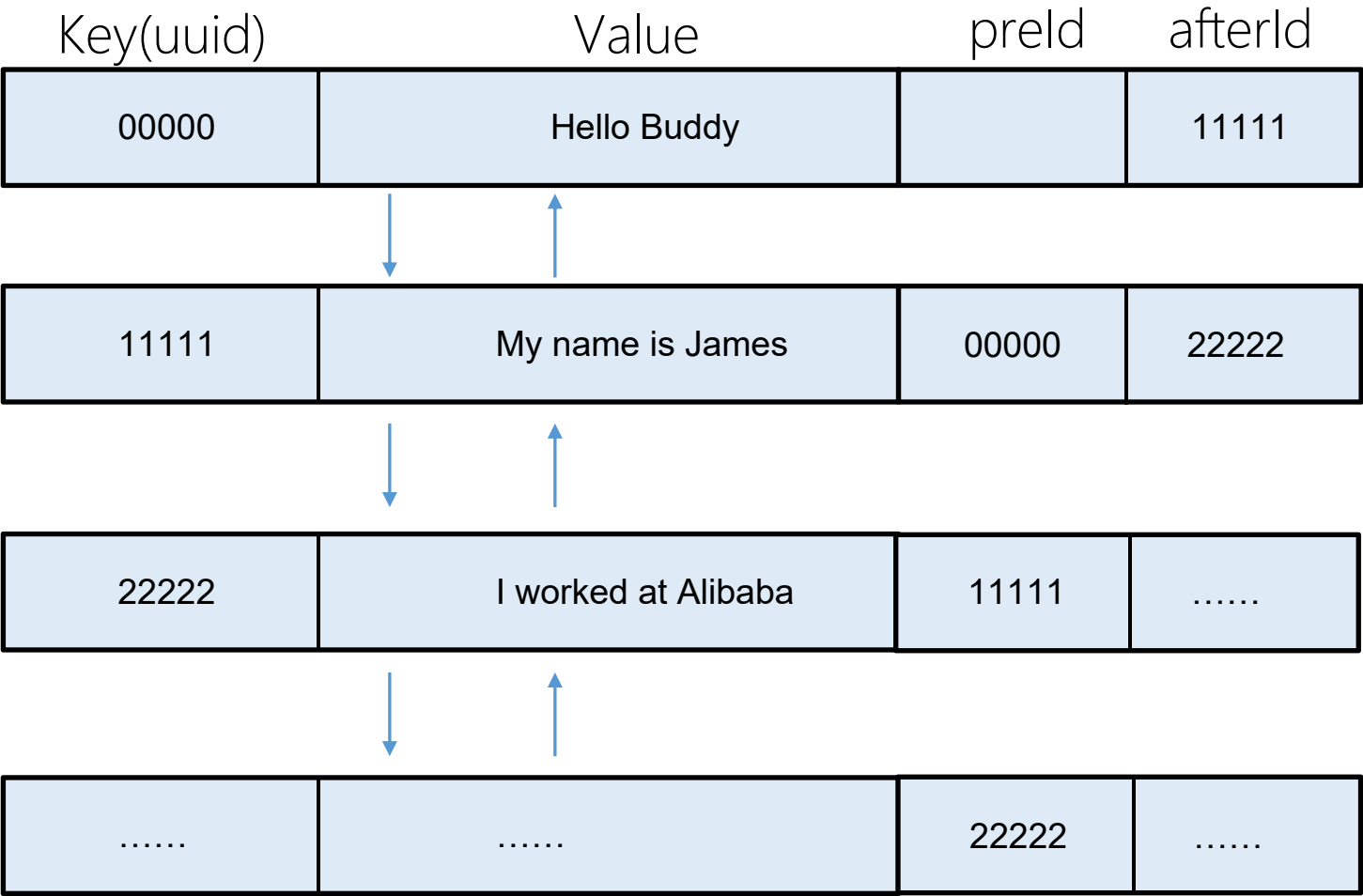
那文件内容怎么转换成链表的呢？

File: Introduce

Hello Buddy  
My name is James  
I worked at Alibaba  
...  
...  
...  
...  
Bye!

Key: 文件每一行的唯一标识  
Value: 每一行的内容

唯一标识由前端生成

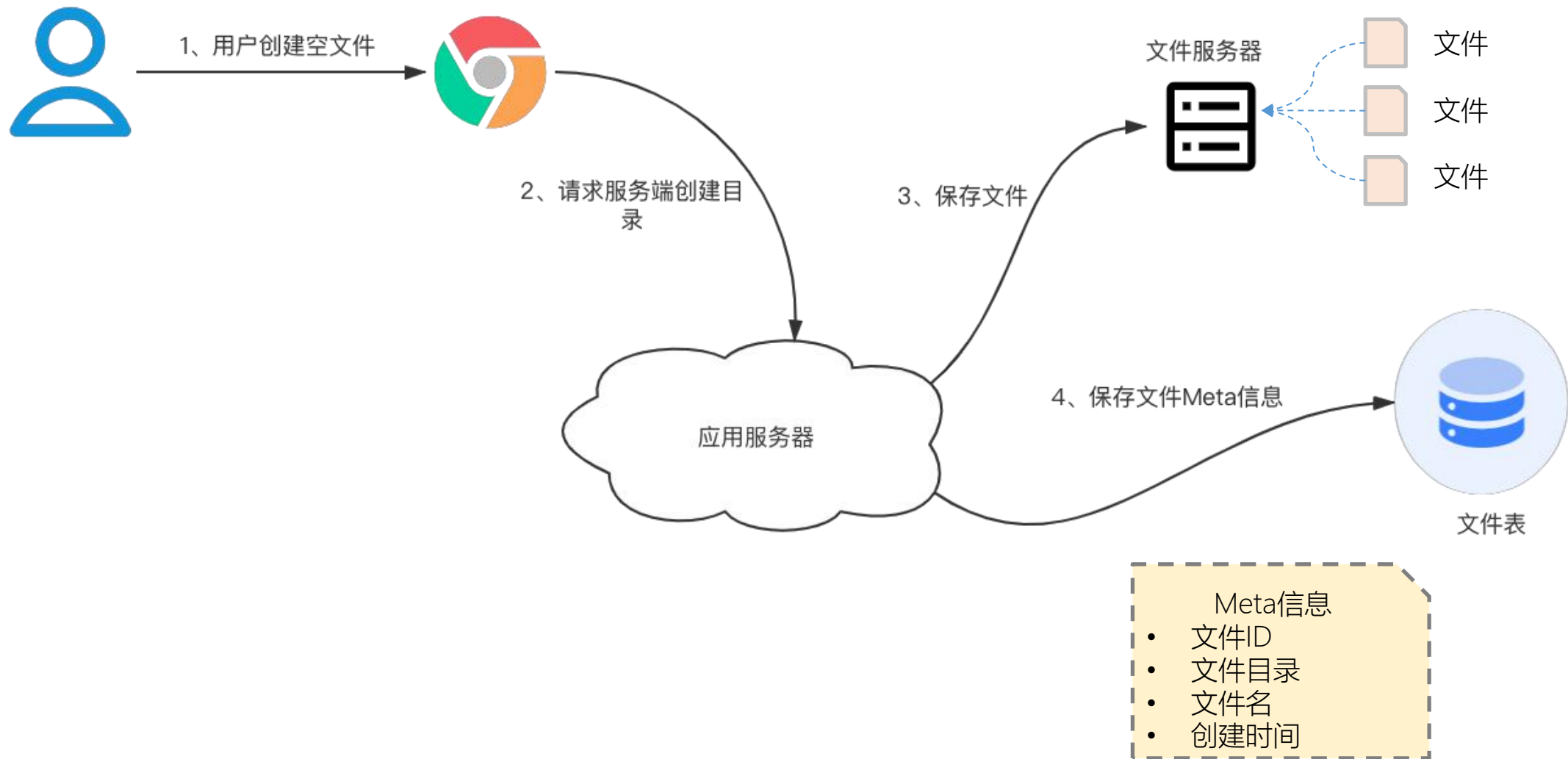


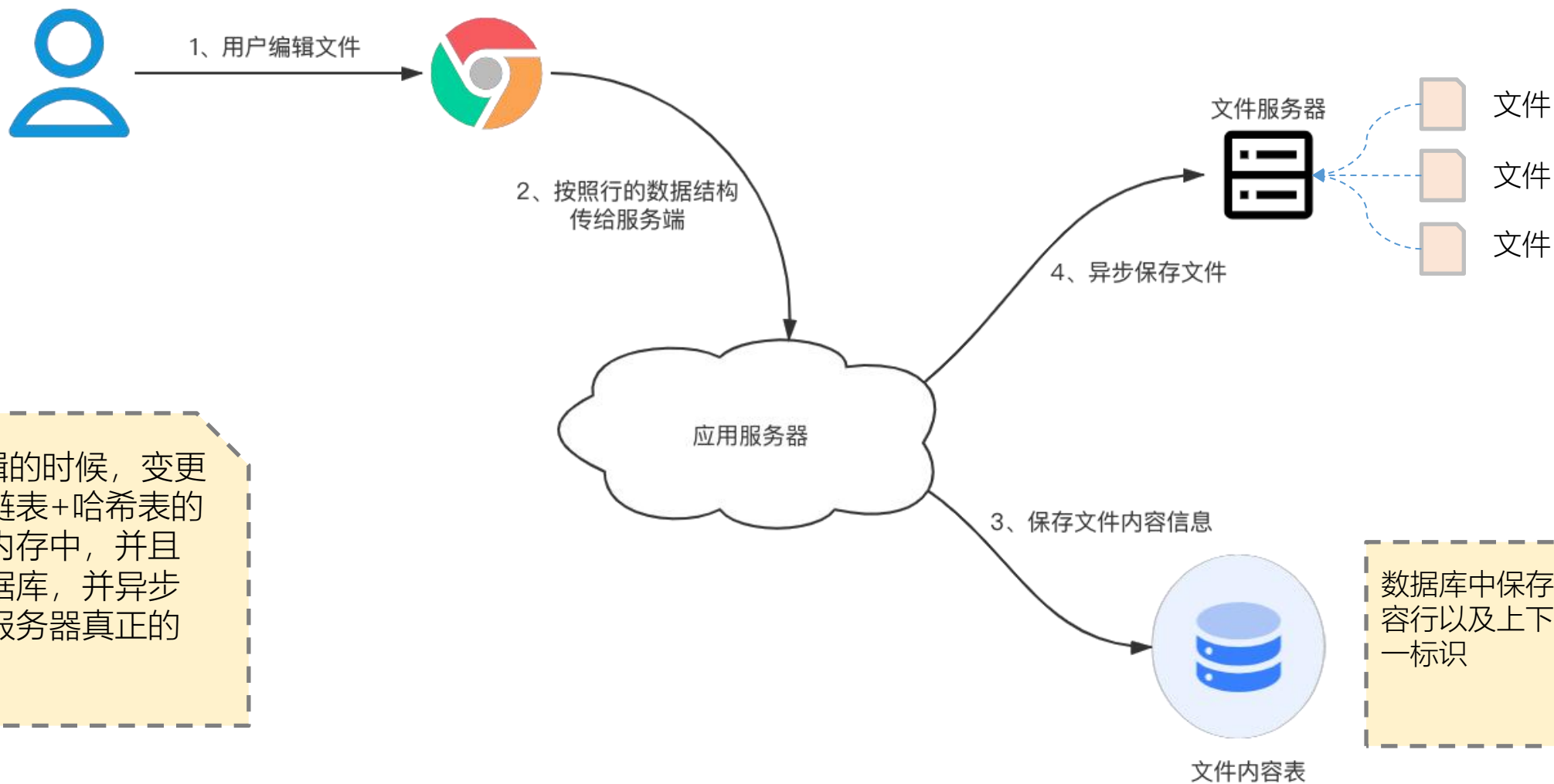
那前端新增或者变更内容，需要传哪些信息给服务端呢？

- 修改/新增了哪一行
- 代表前后行的唯一标识
- 操作的时间
- 操作类型，新增/删除/修改

```
{
  "changes": [
    {
      "uid": "33333",
      "content": "updated content",
      "date": 1596044453949220,
      "preId": "00000",
      "afterId": "55555",
      "opType": "update"
    },
    {
      "uid": "44444",
      "content": "new content",
      "date": 1596064880092835,
      "preId": "33333",
      "afterId": "55555",
      "opType": "insert"
    }
  ]
}
```

前端传到服务端之后做了哪些事情？





文件Table

文件表	文件ID	文件名称	文件Path	文件后缀名
	d23fer-a356	memo	/wwuSA1cs4qtS/	txt

文件内容Table

文件内容表	文件ID	行唯一UID	preId	content	afterId
	d23fer-a356	22222	00000	hello	33333



(00000) Hello Buddy  
(11111) My name is James  
(99999) **My favorite color is blue**  
(33333) I worked at Alibaba  
...  
...  
...  
...  
(12345)Bye!

Before

11111	My name is James	00000	33333
33333	I worked at Alibaba	11111	44444

After

11111	My name is James	00000	99999
99999	<b>My favorite color is blue</b>	11111	33333
33333	I worked at Alibaba	99999	44444

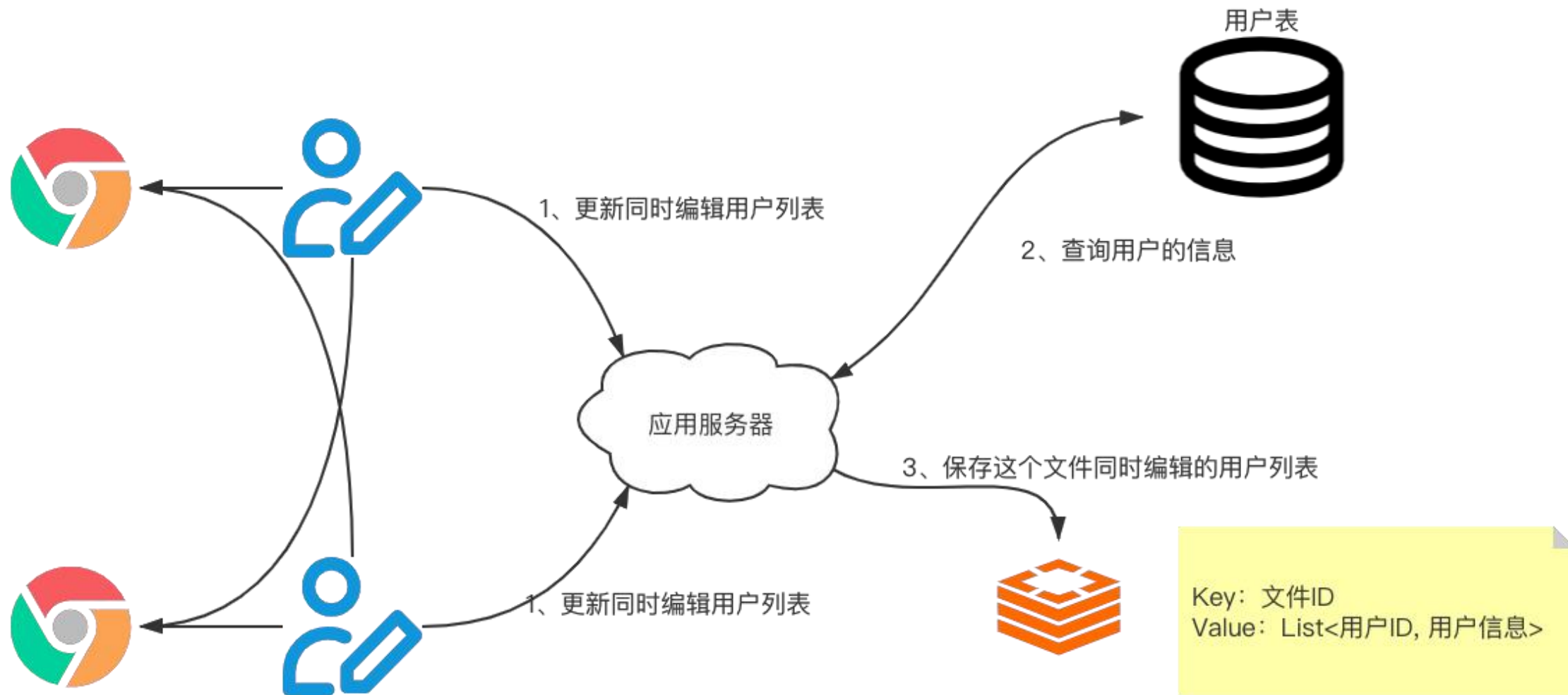
## 如何在网页中显示有多个人同时编辑的头像？



# 同时编辑一份文档的用户列表适合存在哪？

A: Nosql

B: Mysql



用户Table

用户表	用户ID	用户名	性别	头像URL
	12345	James	male	http://xxx/xx.png

# 如何记录某一行谁正在改?

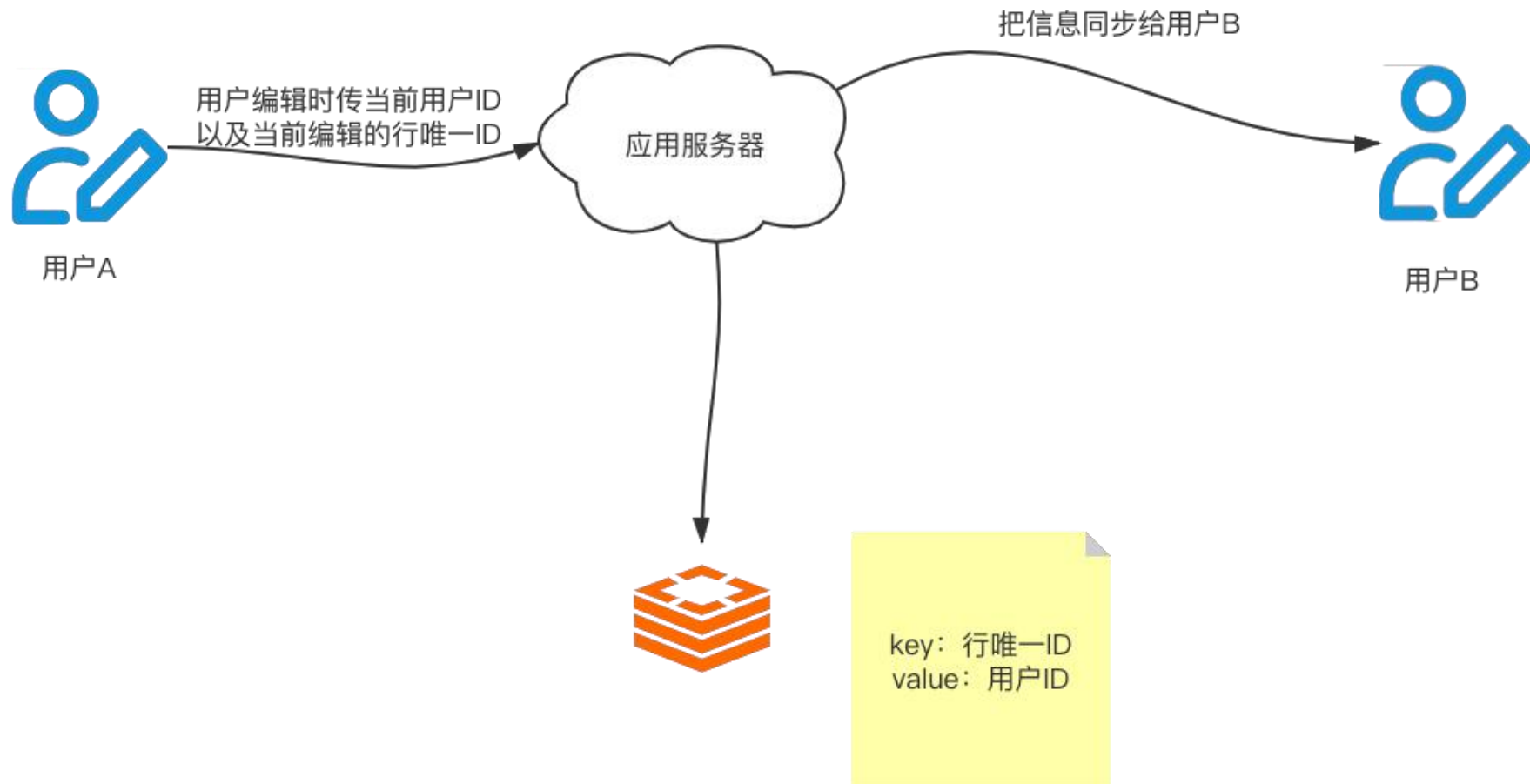
## test

hello world  
I am here  
where are you  
I am ok  
来了 hello

正在编辑



# 显示某一行内容正在被谁编辑



# 基于这个流程，行锁定如何实现？

**test**

hello world  
I am here  
where are you  
I am ok  
来了 hello

前端判断当前用户ID和  
正在编辑行的用户ID是否同一人

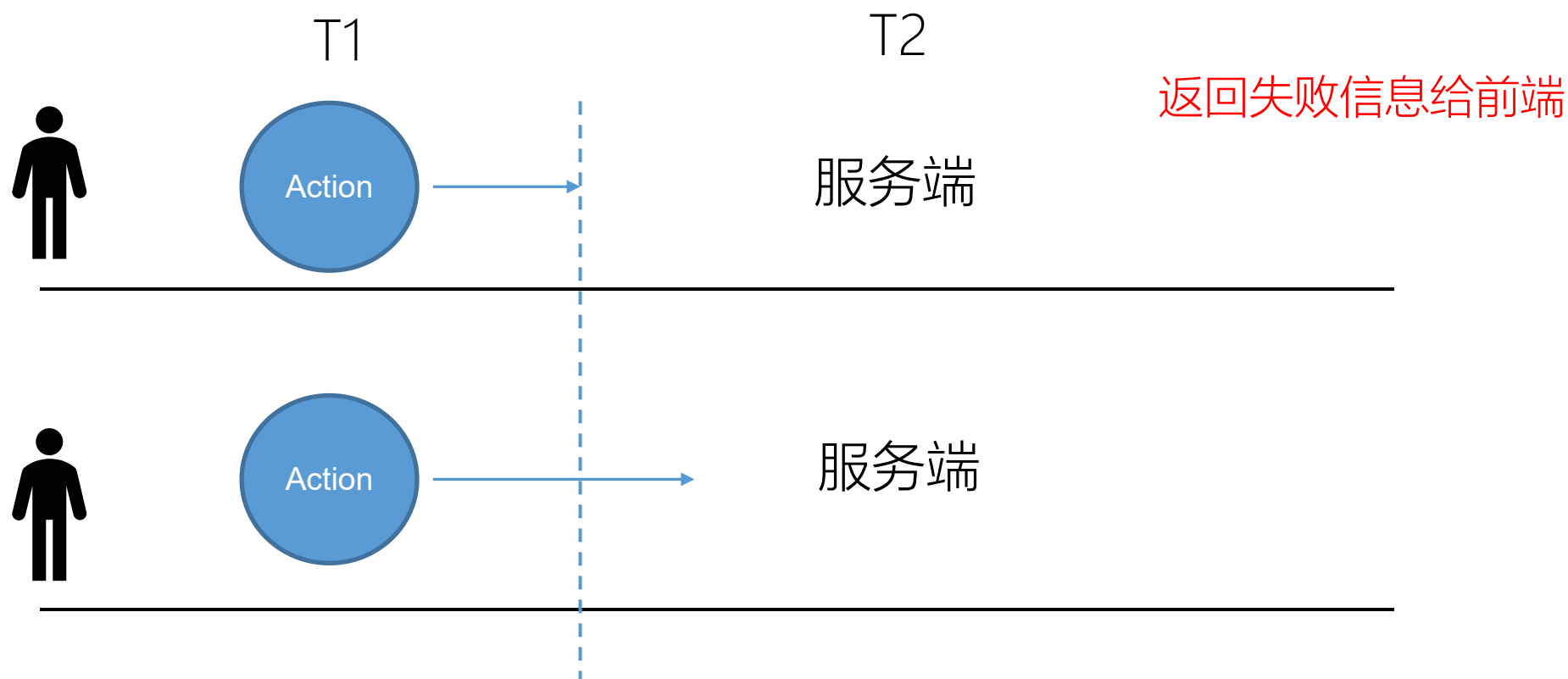




## 那多用户同时编辑，可能会遇到哪些Case

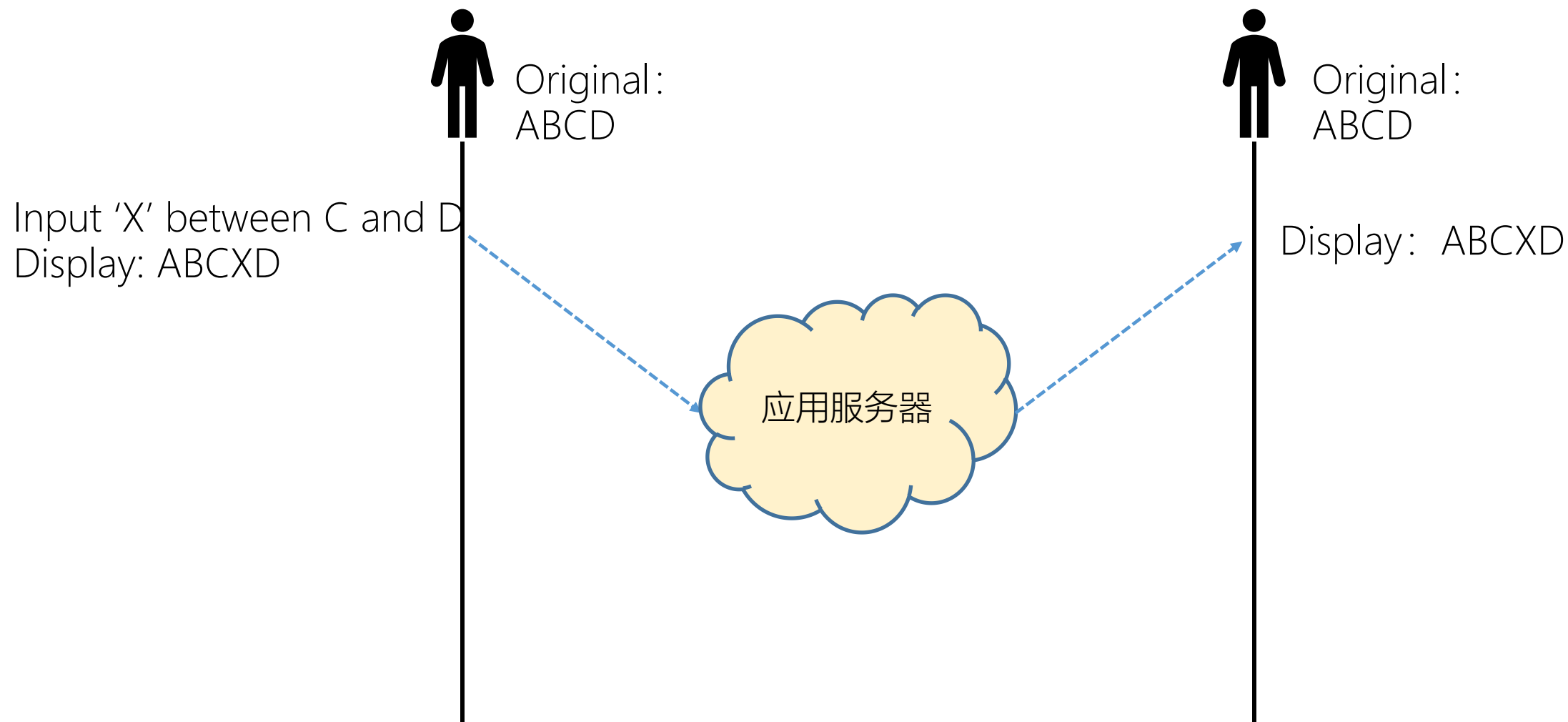
- 多人同时编辑某一行（同时锁定）
- 一个人正在删除某一行内容，一个人正在增加内容

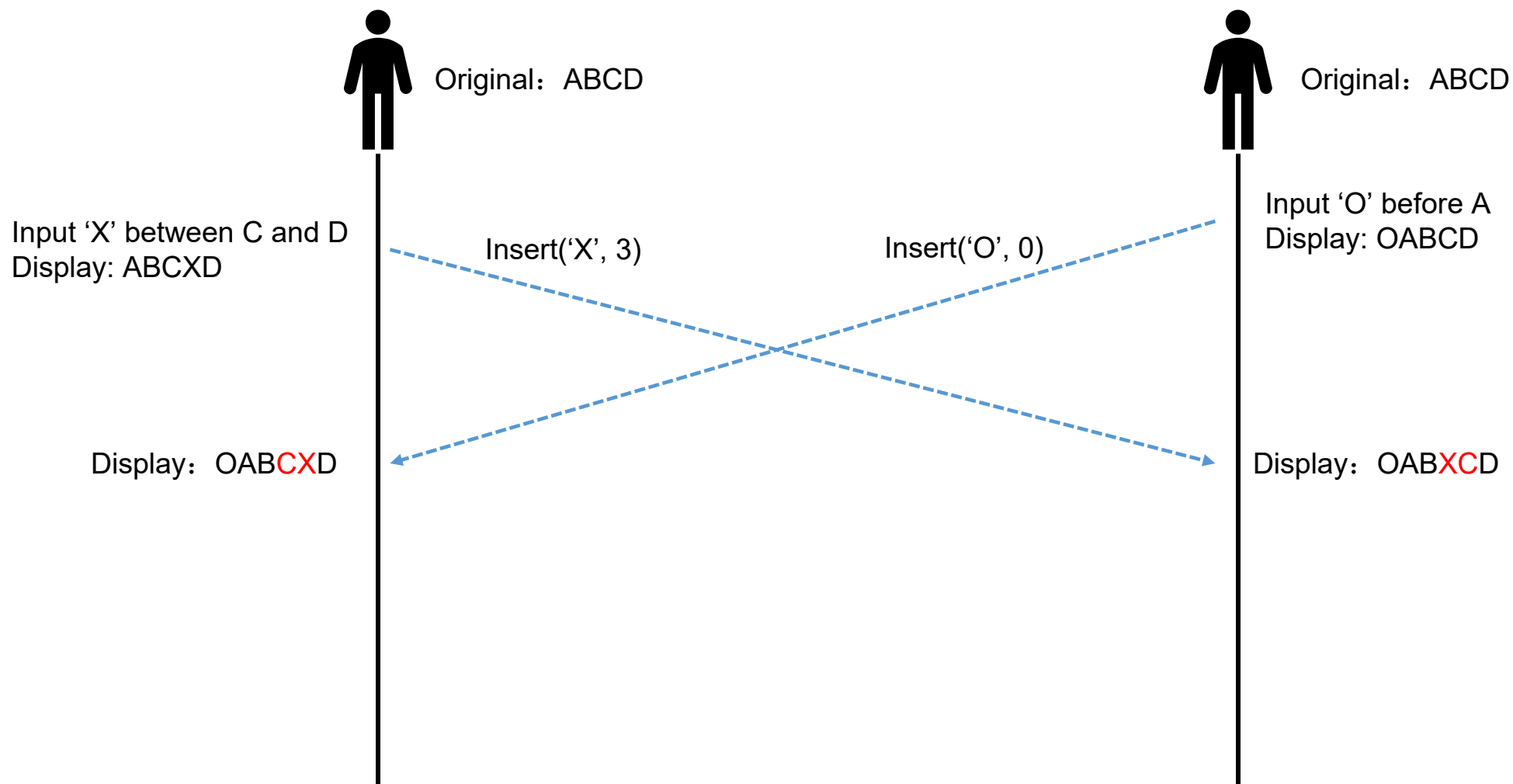
## 通过到达服务端的时间来决定采纳谁的

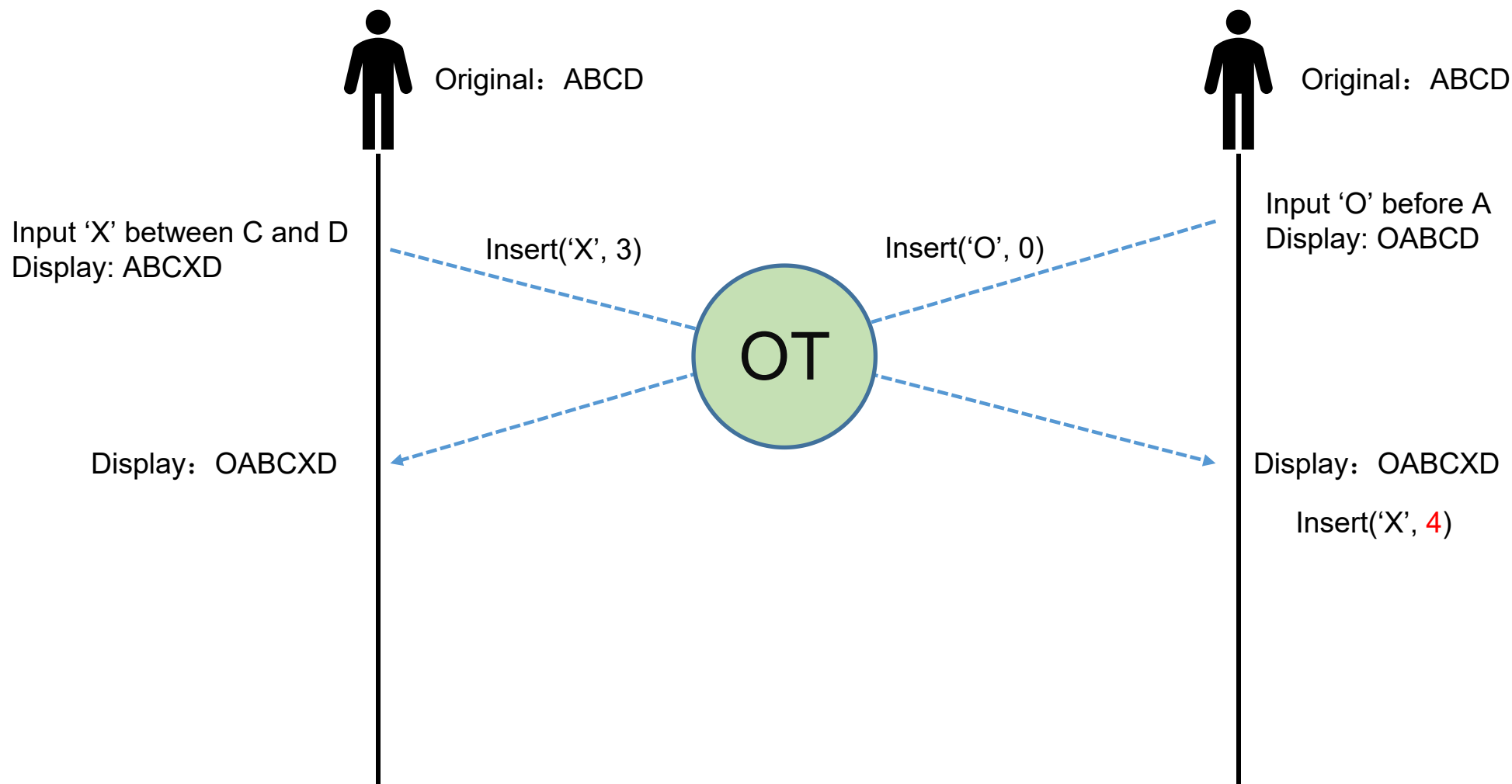


之前的设计都是基于行锁定的设计，有没有不需要锁定的？

- OT(Operational Transform)
- CRDT(Conflict-Free Replicated Data Type)







# 考虑到用户体验，限制同时编辑人数

- <https://svn.apache.org/repos/asf/incubator/wave/whitepapers/operational-transform/operational-transform.html>  
G-Suite协同引擎的协议白皮书
- <https://dl.acm.org/doi/10.1145/274444.274447>  
GOT算法及一维数据操作变换算法论文
- <https://www.tiny.cloud/blog/real-time-collaboration-ot-vs-crdt/>  
To OT or CRDT, that is the question
- <https://github.com/share/sharedb>  
real-time database backend based on OT of JSON documents



	<ul style="list-style-type: none"><li>• 【互动】走进系统设计 &amp; 新鲜事系统设计</li><li>• 【直播】秒杀系统与订票系统设计</li></ul>
Week1	<ul style="list-style-type: none"><li>• 【互动】从用户系统中学习数据库与缓存</li><li>• 【互动】网站系统、API设计与短网址</li><li>• 【直播】容器技术 (K8S, Docker)</li></ul>
Week2	<ul style="list-style-type: none"><li>• 【互动】数据库拆分 &amp; 限流器和实时数据系统设计</li><li>• 【互动】分布式数据库系统设计</li><li>• 【直播】在线文档协同编辑系统设计</li></ul>
Week3	<ul style="list-style-type: none"><li>• 【互动】聊天系统设计</li><li>• 【互动】分布式文件系统设计</li><li>• 【直播】视频流系统设计 (Youtube / Netflix)</li></ul>
Week4	<ul style="list-style-type: none"><li>• 【互动】基于地理位置的信息系统</li><li>• 【互动】分布式计算系统设计</li><li>• 【互动】爬虫系统与搜索建议系统</li><li>• 【直播】Google Search / Search Ads Ranking 系统设计</li></ul>

# Thanks