



Class 6

Gavin

gavin@blockstack.com



一块链刃



Blockstack

课前回顾



■ • Radiks

- Radiks 环境架构 (server)
- Model 结构
- 增删改查操作

■ • Demo设计

- 留言板
- Message Model 的设计与操作
- 信息解析





一块链刃



Blockstack

本课内容



- • Radiks 用户分组授权功能
- • Demo 开发演示分组授权功能



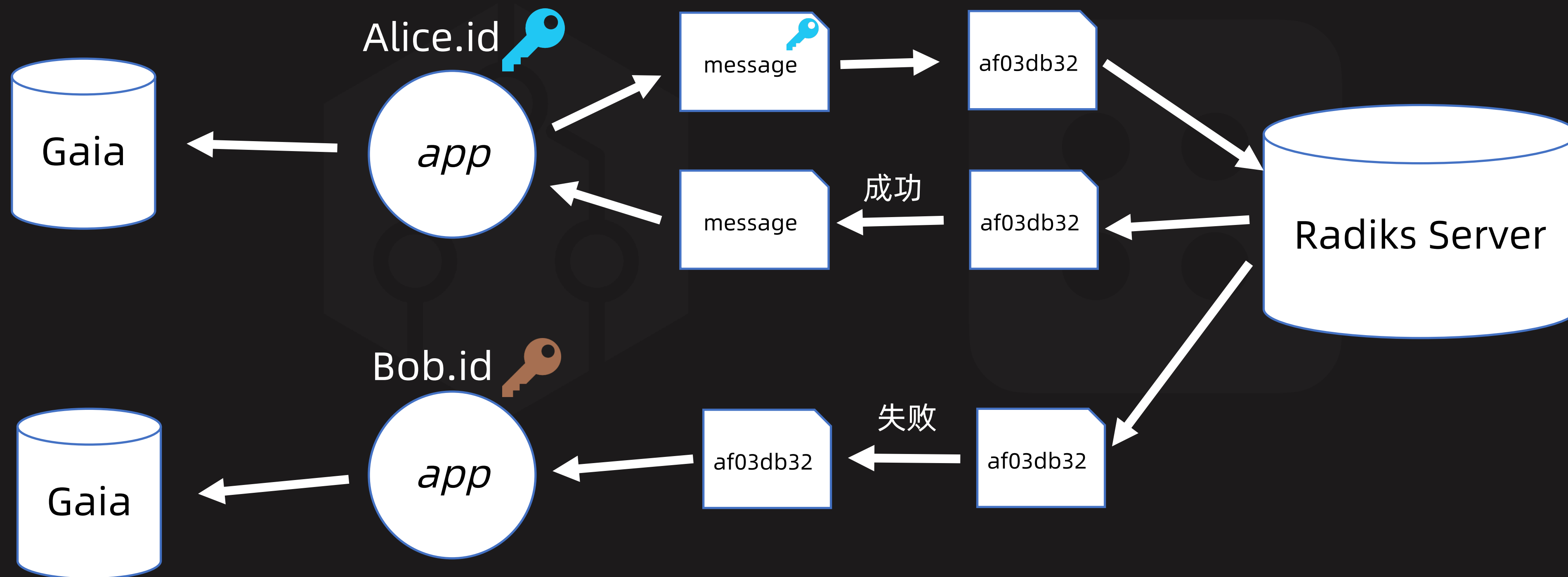


Radiks 用户分组授权



• Radiks 数据交互流程

如果Alice.id加密信息用的不是自己的密钥，而是一个公共的密钥，且Bob.id可以通过某些安全的方法获取该密钥



■ • Radiks 群组功能

- 引入新的 Class 进行数据加密的密钥管理
- 加入群组的用户可以非常方便的共享信息
- 群组外的用户，以及Radiks server的管理员无法获取群组内成员信息，以及群组内消息信息（除非消息信息字段设置为decrypted: true）

■ • Radiks 群组管理流程

1. 用户 A 创建群组 G
2. 用户 A 为用户 B 生成邀请码，并发送给 用户 B
3. 用户 B 激活邀请码
4. 用户 A、B 可以在群组 G 下共享信息



■ • Radiks 群组管理流程解析1

• 用户 A 创建群组 G

1. 用户 A 创建群组 G
2. 用户 A 为用户 B 生成邀请码，并发送给 用户 B
3. 用户 B 激活邀请码
4. 用户 A、B 可以在群组 G 下共享信息

```
import { UserGroup } from 'radiks'; 1

export async function createGroup(groupName){ 2
  const group = new UserGroup({ name: groupName });
  const newGroup = await group.create();
  console.log(newGroup._id) // 该_id为群组的ID，用于后续查找群组
```

1. 引入 UserGroup 类
2. 输入群组名创建群组

■ • Radiks 群组管理流程解析2

• 用户 A 为用户 B 生成邀请码，并发送给 用户 B

1. 用户 A 创建群组 G
2. 用户 A 为用户 B 生成邀请码，并发送给 用户 B
3. 用户 B 激活邀请码
4. 用户 A、B 可以在群组 G 下共享信息

```
const group = await UserGroup.findById(myGroupId); 1
const usernameToInvite = 'gavin.id';
const invitation = await group.makeGroupMembership(usernameToInvite); 2
console.log(invitation._id); // 该_id为邀请的邀请码, 用于后续 gavin.id 激活邀请 3
```

1. 获取想要 用户 B 进入的群组
2. 发出邀请
3. 获取邀请码

■ • Radiks 群组管理流程解析3

• 用户 B 激活邀请码

1. 用户 A 创建群组 G
2. 用户 A 为用户 B 生成邀请码，并发送给 用户 B
3. 用户 B 激活邀请码
4. 用户 A、B 可以在群组 G 下共享信息

```
import { GroupInvitation } from 'radiks'; 1

const invitation = await GroupInvitation.findById(myInvitationID); 2
await invitation.activate(); 3
```

1. 引入 GroupInvitation 类
2. 根据邀请码获取邀请详细信息
3. 激活邀请

■ • Radiks 群组管理流程解析4

• 用户 A、B 可以在群组 G 下共享信息

1. 用户 A 创建群组 G
2. 用户 A 为用户 B 生成邀请码，并发送给 用户 B
3. 用户 B 激活邀请码
4. 用户 A、B 可以在群组 G 下共享信息

```
import { Model } from 'radiks';

export default class Message extends Model {
  static className = 'Message';

  static schema = {
    from: String,
    content: String,
    userGroupId: {
      type: String,
      decrypted: true,
    },
  }
  static defaults = {
  }
}
```



• 用户 A 发送指定信息

- 必须有userGroupId字段，且该字段为非加密类型



一块链刃



Blockstack

Demo展示与讲解



<UserGroupList/>



已经选择的群组

<MessageList/>

用户群组选择

Blockstack教研组 ▼

邀请码生成：

登出

1. 添加了UserGroupList组件
2. 使用Redux传输已经选择的群组
3. radiks-gavin-test 版本

留言框

提交

gavin.id

大家好，我是Gavin

9498c481e5ab5e8d06e6c25a1f79a8e2

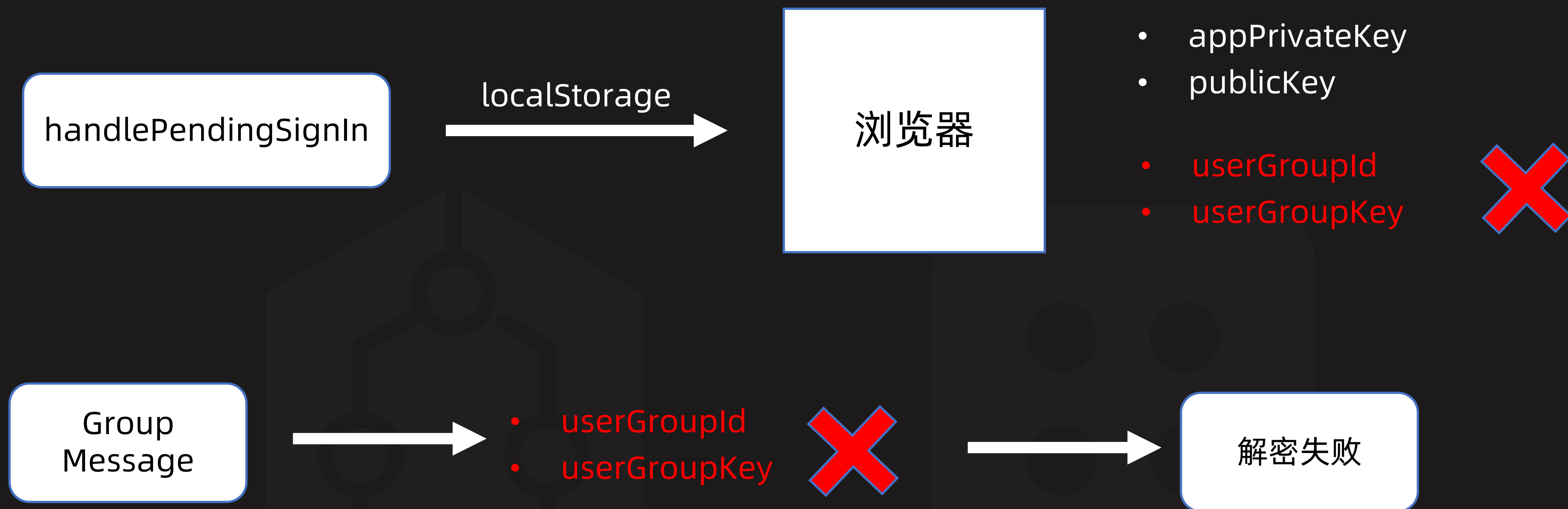
590482db836a95c11d32a2



细节讲解



• 细节讲解-UserGroup权限



- 在应用中完成UserGroup信息的存储 => UserGroupRecord
- 在应用开始之前完成localStorage的userGroup写入



一块链习



Blockstack

课后作业



■ • 课后作业

- 跑通 feature-radiks-userGroup 分支代码，将视频中出现的分组信息流程对应 Mongodb 客户端内的数据截图提交到指定仓库里
- 根据 demo 的 feature-radiks-userGroup 分支完成自己的产品技术流程图设计，并提交在自己开发的产品仓库中，将产品仓库链接放在 HomeWork6 中
- 根据 *demo 的 feature-radiks-UserGroup 分支的代码完成自己的产品 Radiks-UserGroup 相关的部分代码*