



# Class 5

---

Gavin

[gavin@blockstack.com](mailto:gavin@blockstack.com)



一块链刃

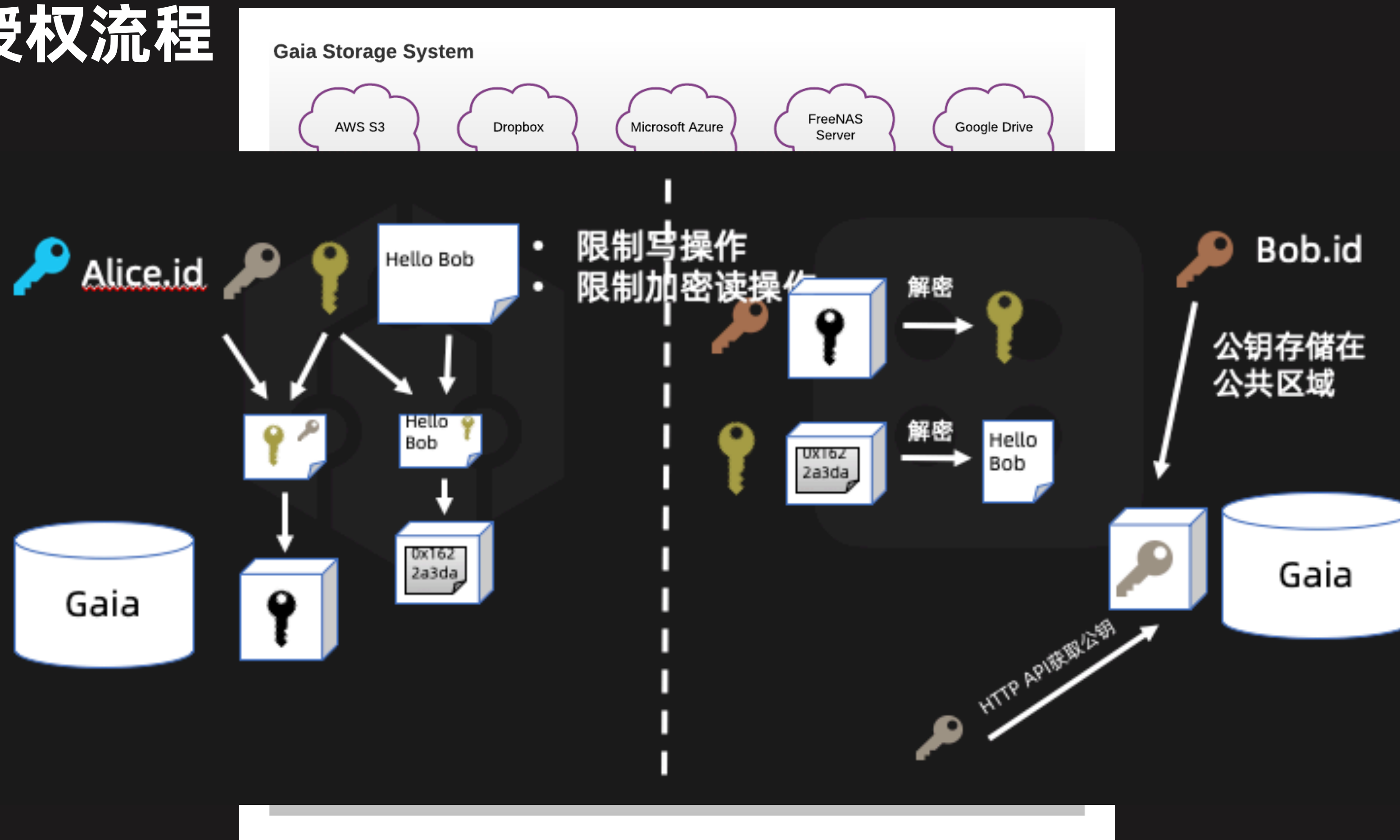


Blockstack

# 课前回顾



- 完成了Blockstack三个层级的讲解
- 定向授权流程





一块链刃



Blockstack

# 本课内容



- • 项目分析 Mumble
- • Radiks 介绍
- • 如何使用Radiks开发去中心化应用

# 项目分析 - Mumble



## ■ • 项目分析 Mumble

- • 数据读取不是从自己的 Gaiahub , 而是从 radiks 相关网址
- • 读取的信息的格式与Gaia存储相似
- • 聊天室分组、邀请等功能是原有Gaia不太容易实现的

# Radiks





## ■ • Gaia 特点分析

### • Gaia 特性

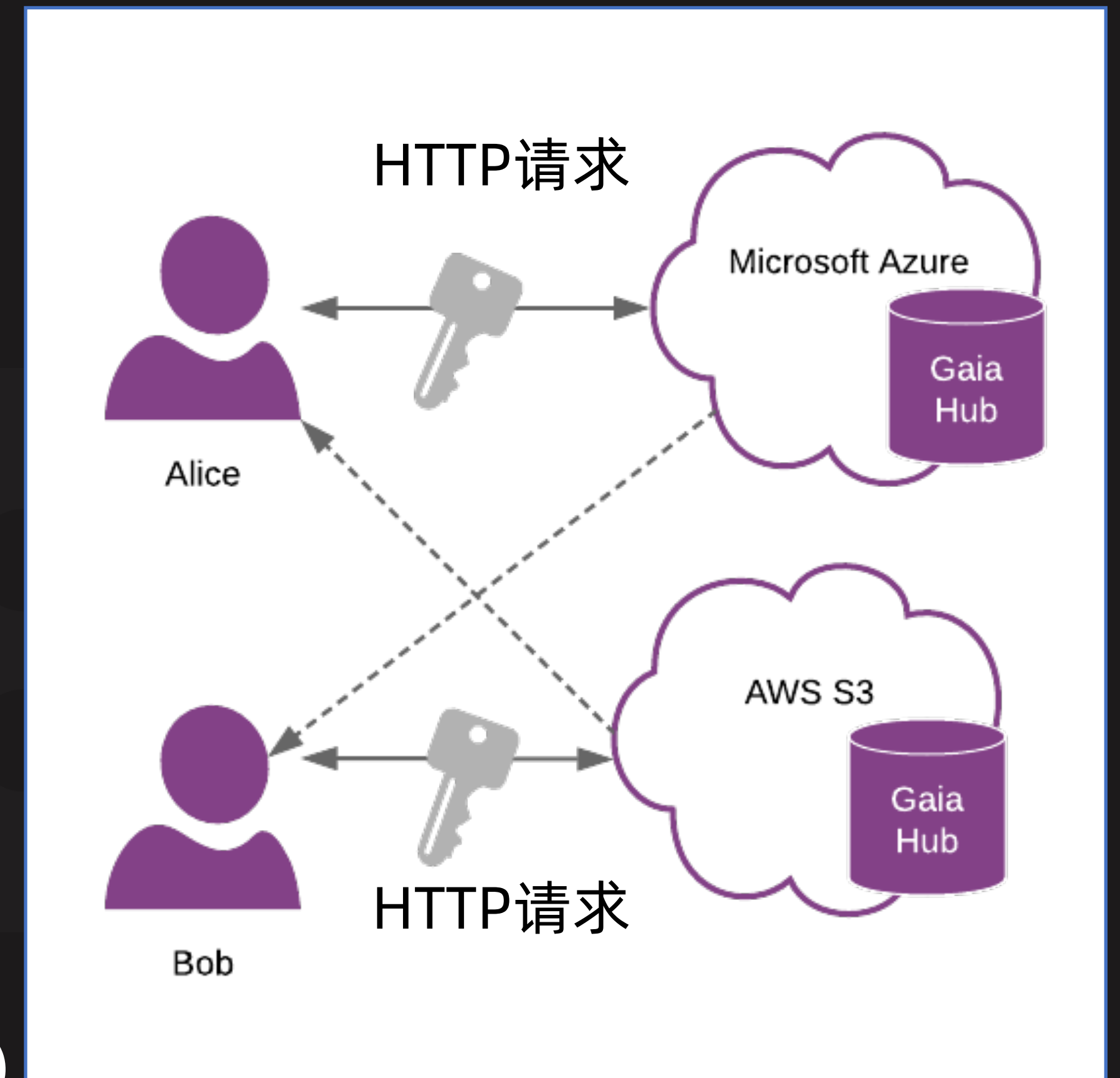
- 提供了基于HTTP的服务协议
- 可以寄托在任何云服务平台上
- 通过DID对应的私钥来授权用户数据

### • CRUD API

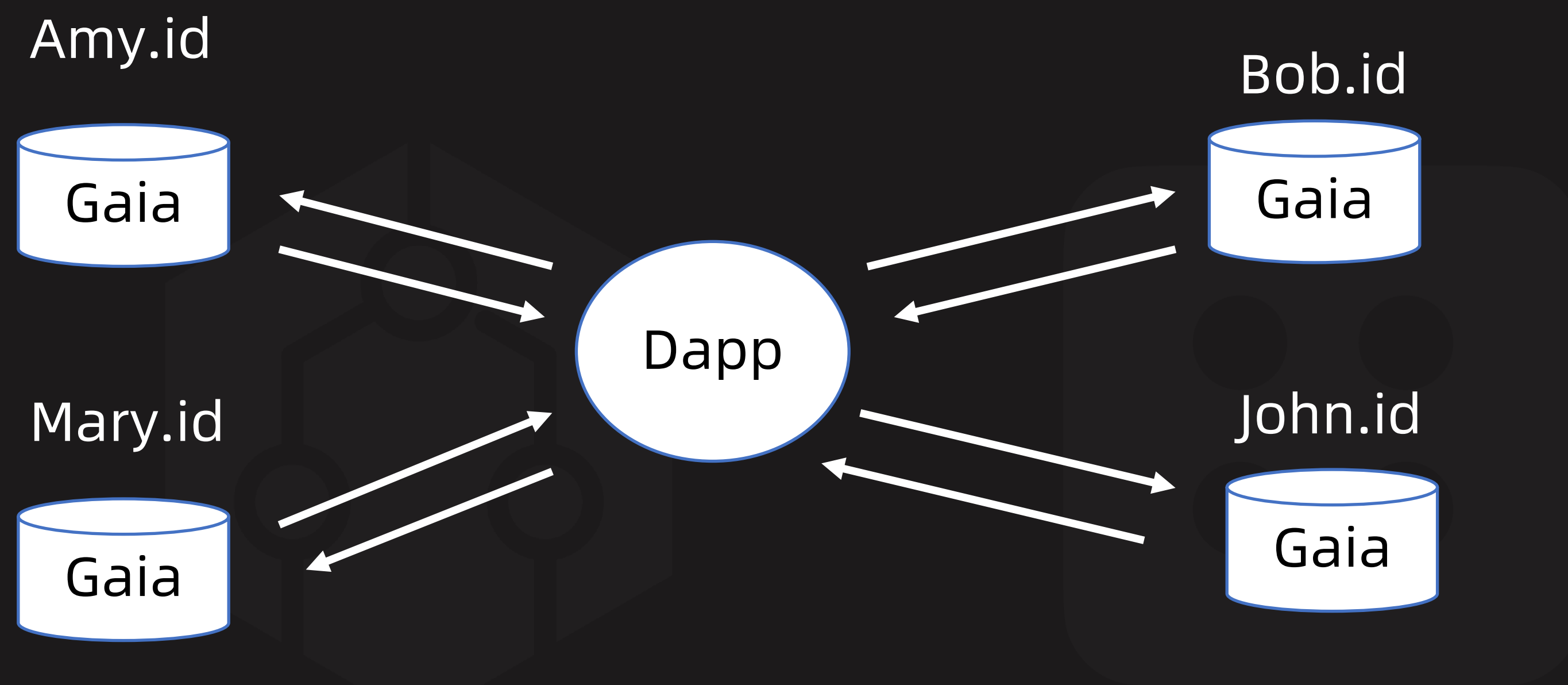
- 增：putfile    删：deletefile
- 改：putfile    查：getfile

### • Gaia 短板

- 应用方统计用户数据成本很大（无法做数据聚合）
- 限制了应用自身的逻辑



## • 原有产品结构图

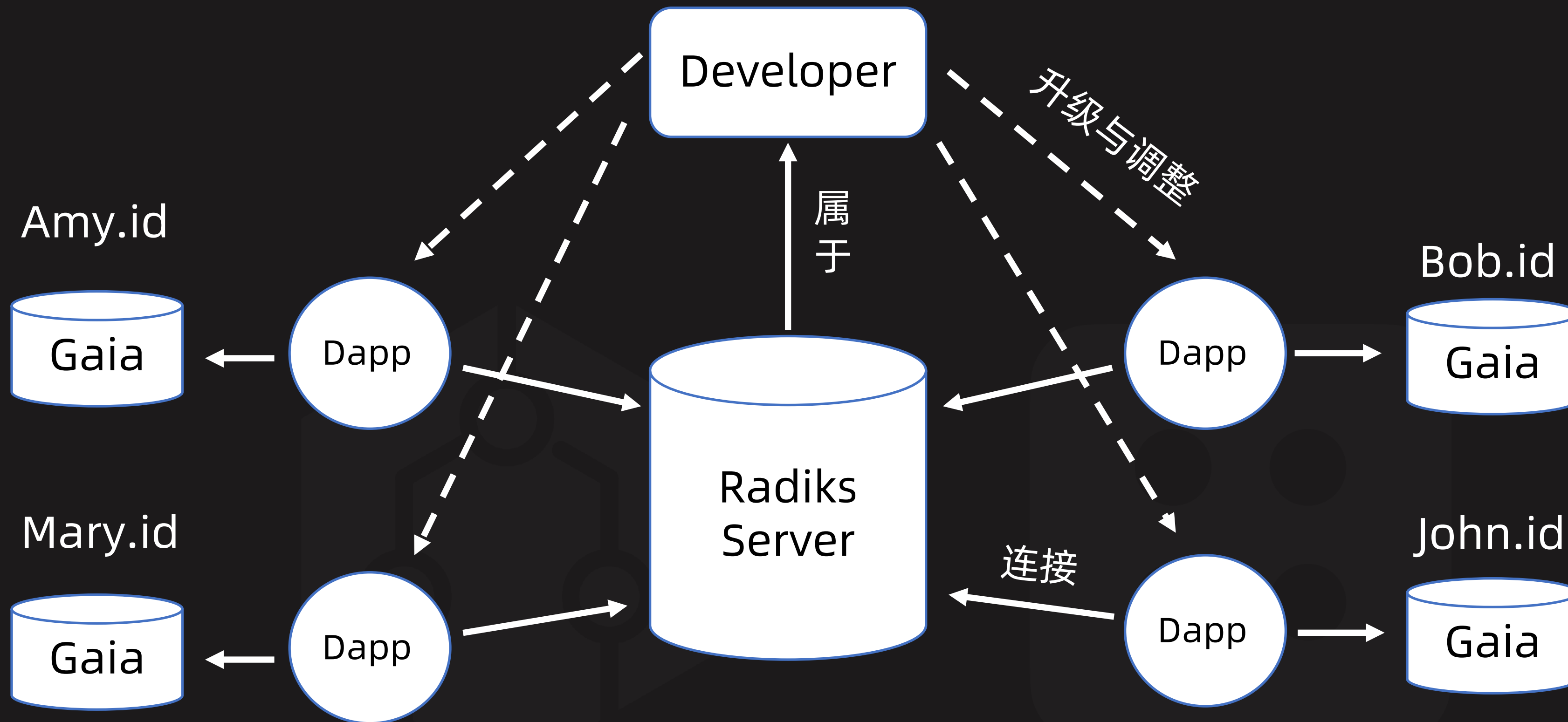


- Dapp相当于只做数据的转发
- 数据读和写操作与应用开发者无任何关系

## ■ • Radiks主要想解决的问题

- 存储与查询复杂的数据结构
- 轻松索引用户公开存储的数据
- 与其他用户的协作
- 数据实时更新

## • Radiks



- Dapp开发者可以部署该应用的Radiks Server
- 将属于该应用的数据存储到中心化的Server中
- Developer可以根据统计Radiks Server中的数据来进行Dapp的升级

## ■ • Radiks为什么是去中心化的

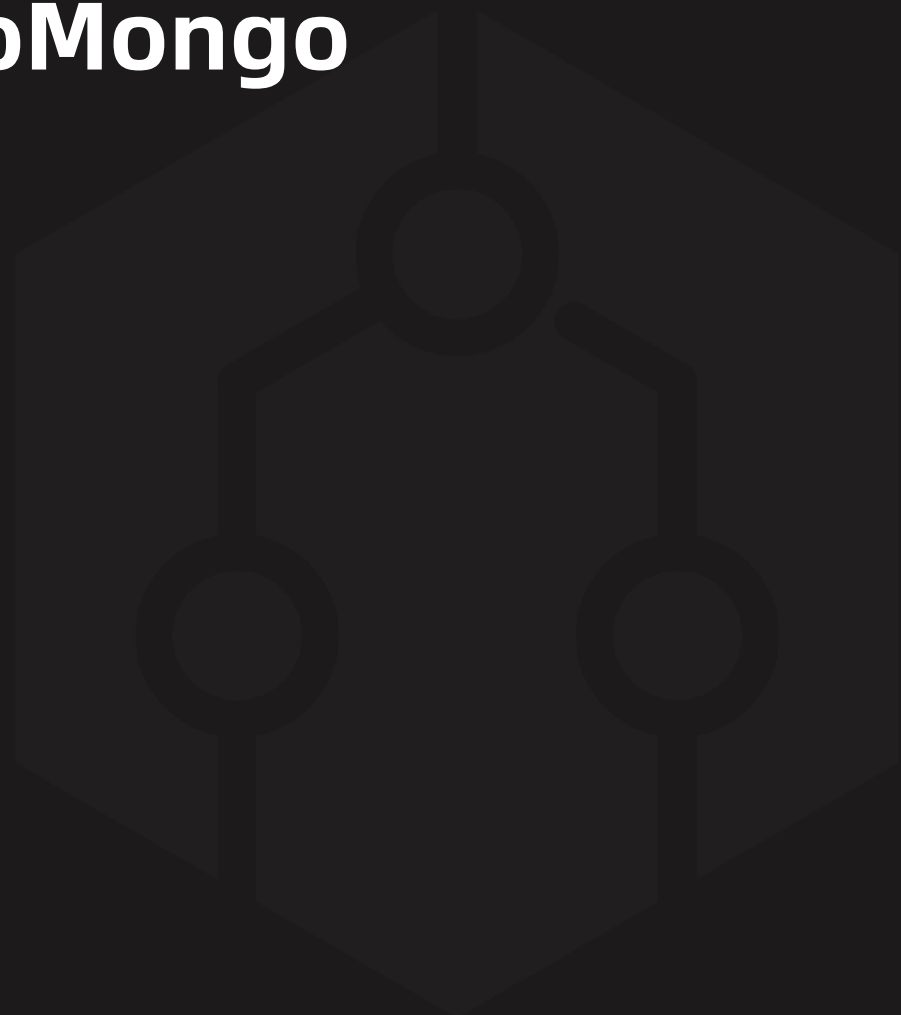
- 没有数据锁入  
(Gaia存储着原始数据, Radiks只存储数据备份)
- 权限限制  
(第三方无法剥夺数据权限)
- 隐私保护  
(数据在存储到Gaia以及Radiks之前在客户端已经加密过了)
- 基于去中心化授权方式  
(Radiks深度依赖Blockstack身份授权系统)

## ■ • Radiks 环境配置

- **Radiks-Server** (<https://github.com/blockstack/radiks-server>)
  - MongoDB > 3.6
  - npm install -g radiks-server 或 yarn global add radiks-server
- **Radiks** (<https://github.com/blockstack/radiks>)
  - 如果使用blockstack.js <=18, 使用radiks 0.1.\* 版本
  - 如果使用blockstack.js >=19, 使用radiks 0.2.\* 版本
  - npm install --save radiks 或 yarn add radiks

## ■ • Radiks 环境配置

- MongoDB 后台查看器
  - Studio3T
  - Navicat
  - RoboMongo



## ■ • 使用Radiks开发去中心化应用

- 留言板应用（已上传到Blockstack\_demo的feature-radiks分支上）
- Radiks 登入（身份登入，连接 Radiks-server） # App.js

```
import { configure } from 'radiks';  
import { User, getConfig } from 'radiks';
```

```
configure({  
  apiServer: 'http://localhost:1260',  
  userSession  
});
```

```
await User.createWithCurrentUser();
```

配置

在Radiks-server中写入新用户



## ■ • 使用Radiks开发去中心化应用

- Model 设计

- schema

# models/Message.js

- Model 的增删改查

#MessageList.js

```
const message = new Message({
  from : "gavin.id",
  content : "I love miumiu",
  flag: true
});
const handleQuery = async() =>{
  const fetchMessages = await Message.fetchList({flag : true});
  console.log(fetchMessages)
}
const handleSave = async () =>{
  await message.save()
}
const handleUpdate = async() =>{
  const newAttributes = {
    from: "gavin.id",
    content: "I love qiuqiu",
    flag: false
  }
  await message.update(newAttributes)
}
```

## ■ • 使用Radiks开发去中心化应用

- 完成留言板设计
  - 留言添加
  - 自动刷新
  - 明文/密文的展示





一块链习



Blockstack

# 课后作业



## ■ • 课后作业

- 完成 Radiks 环境配置
- 跑通 feature-radiks 分支代码，将网页对应的save、update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 分支完成自己的Radiks Model设计
- 根据 *demo 的 feature-radiks 分支的代码完成自己的产品Radiks Model 部分代码*