02

OPEN ORIENTED

凹 凸 实 验 室

登录那些事

1

如何登录

2

状态管理

1 账号密码登录

状态管理

02

- session
- JWT（Json Web Token）

状态管理 - session

```
{
    username: xx
    password: **
}
```

setCookie(sessionID)

session ID

session

{Object}

...

状态管理 - session

困难

1. 分布式 server
2. API server 与 APP server 不同域
3. 服务端引入了 state

RESTful api 的原则是 stateless

状态管理 - JWT

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```

Decoded EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) secret base64 encoded
```

状态管理 - JWT

```
{
    username: xx
    password: **
}
```

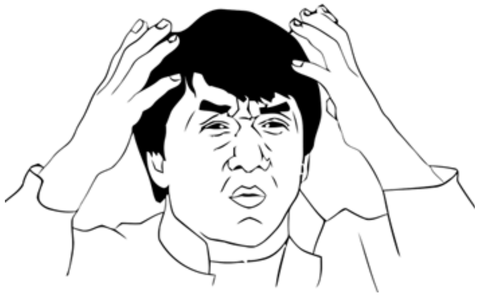
token

生成 token

2

OpenID

使用 OpenID 登录



URI

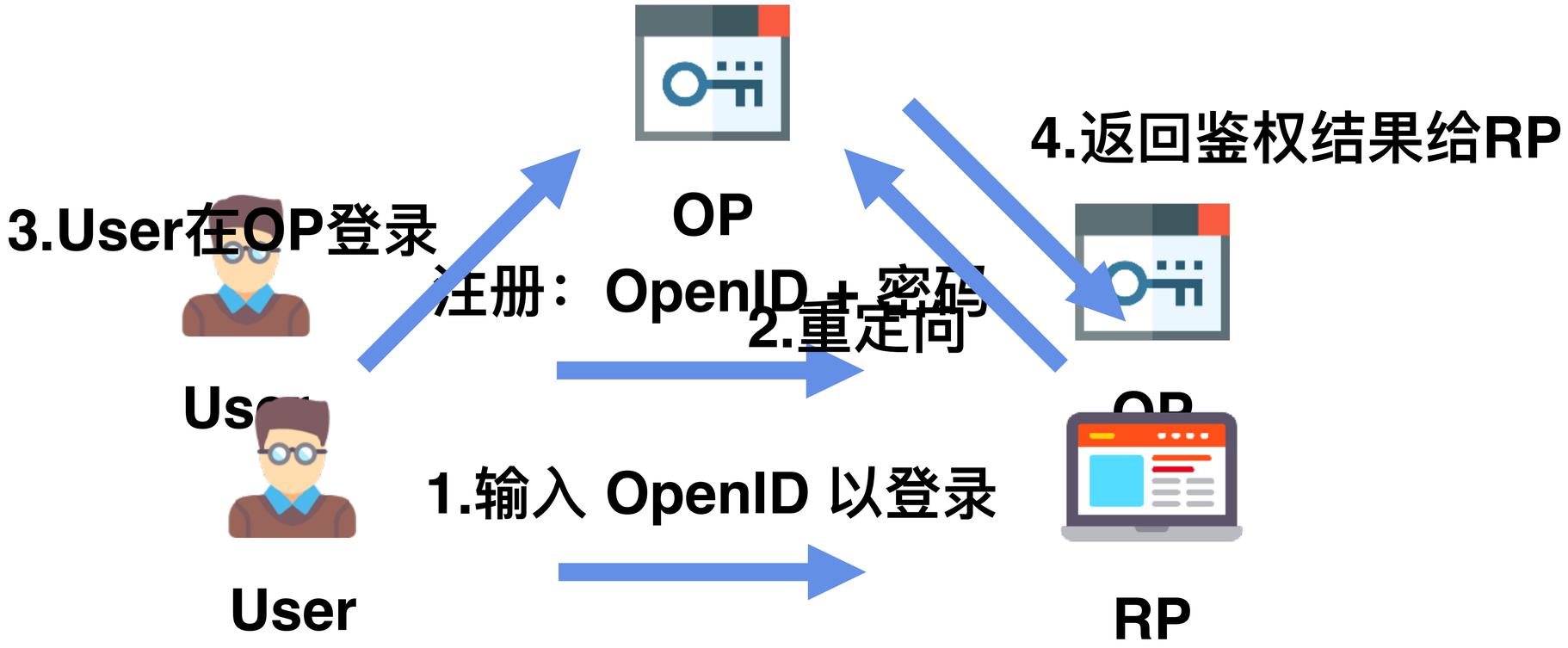
.....

使用 OpenID 登录

角色

1. User - 用户
2. RP（Relying Party）- 服务提供者
3. OP（OpenID Provider）- OpenID 提供者

使用 OpenID 登录


4.返回鉴权结果给RP
OP
3.User在OP登录
注册： OpenID + 密码
2.重定向
User
1.输入 OpenID 以登录
User
RP

Authentication

验证

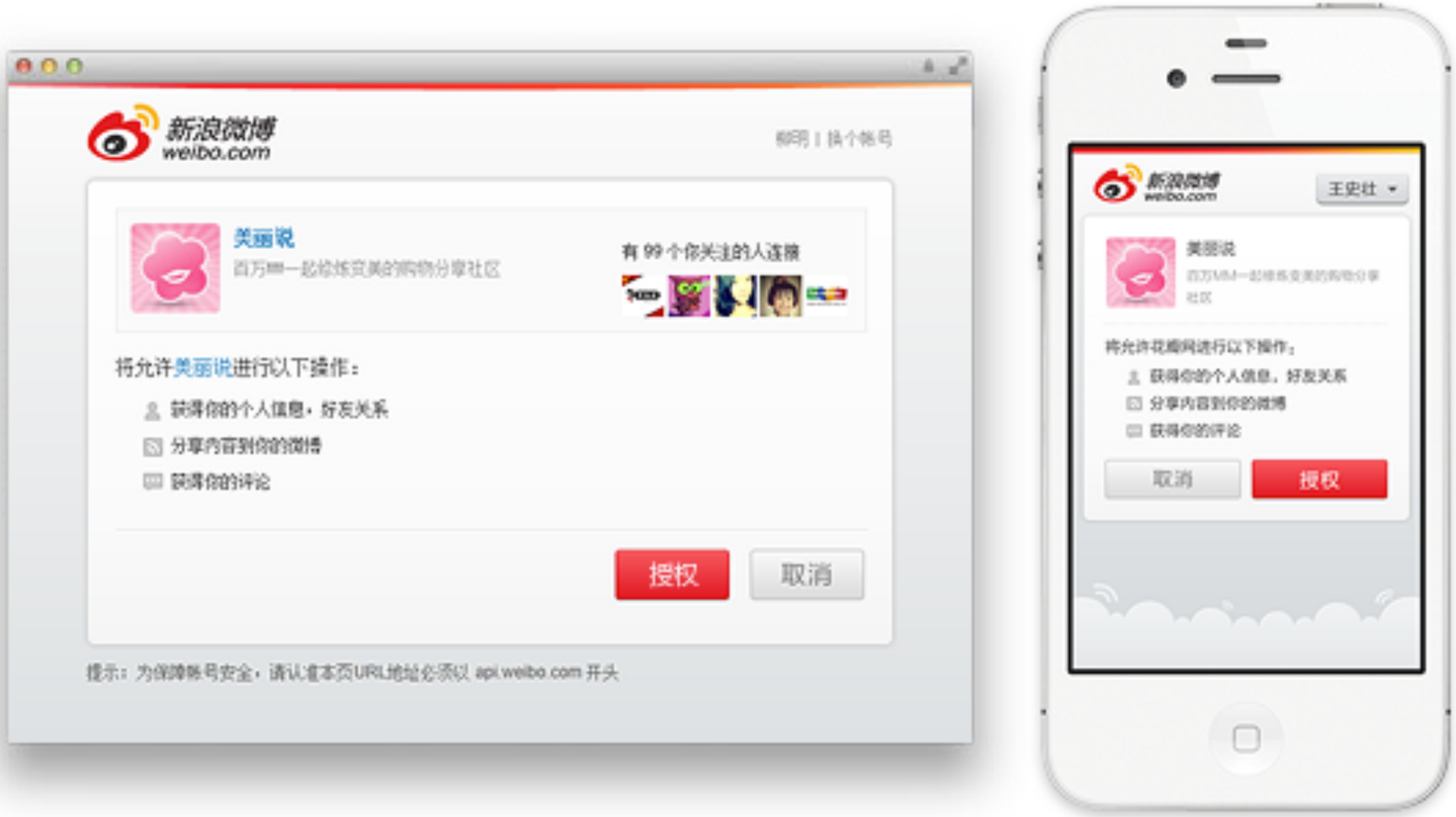
Authorization

授权

3

OAuth

使用 OAuth 获取授权

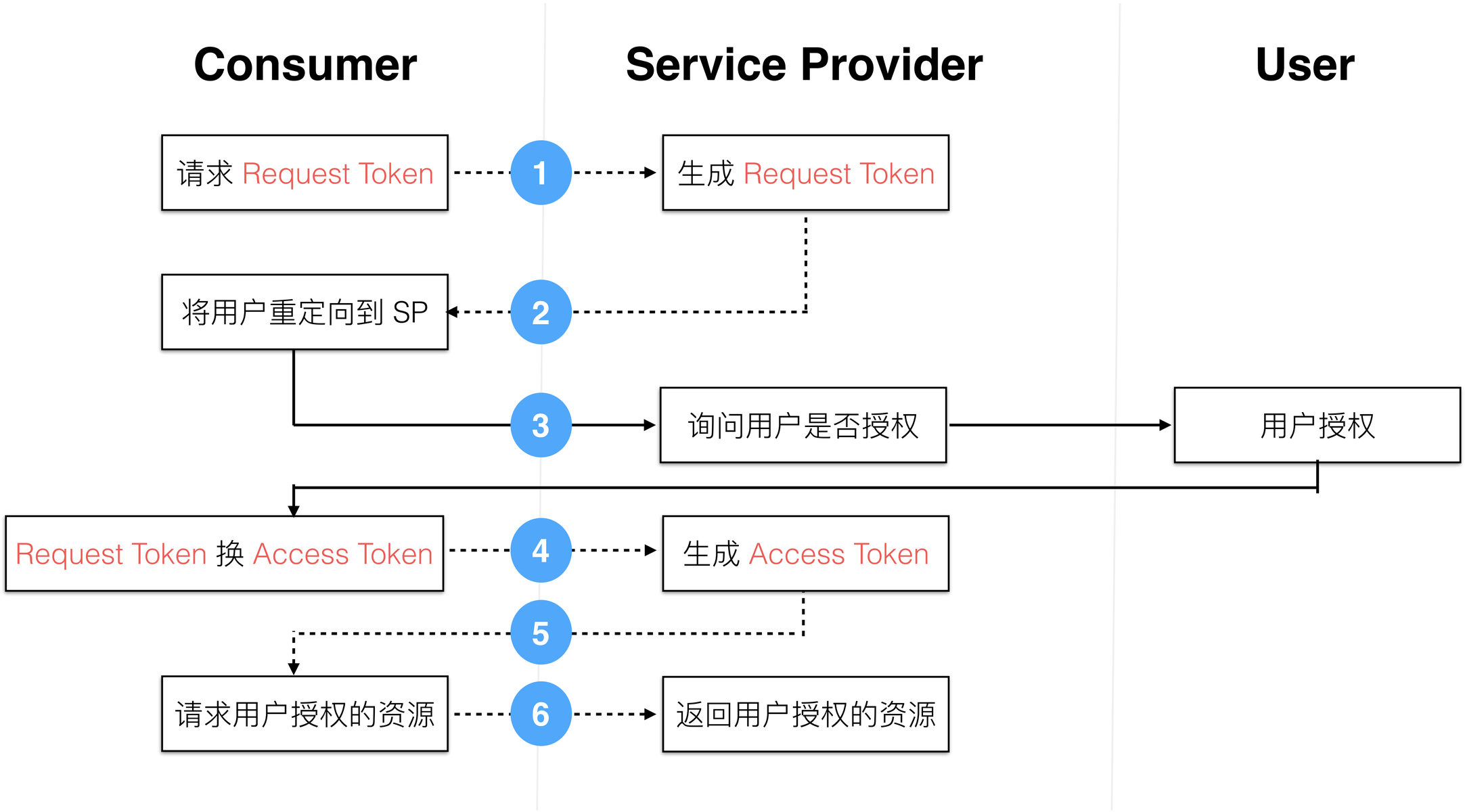


使用 OAuth 获取授权

角色

1. User - 用户
2. Service Provider - 服务提供者
3. Consumer - 消费方

使用 OAuth 获取授权



使用 OAuth 作第三方登录

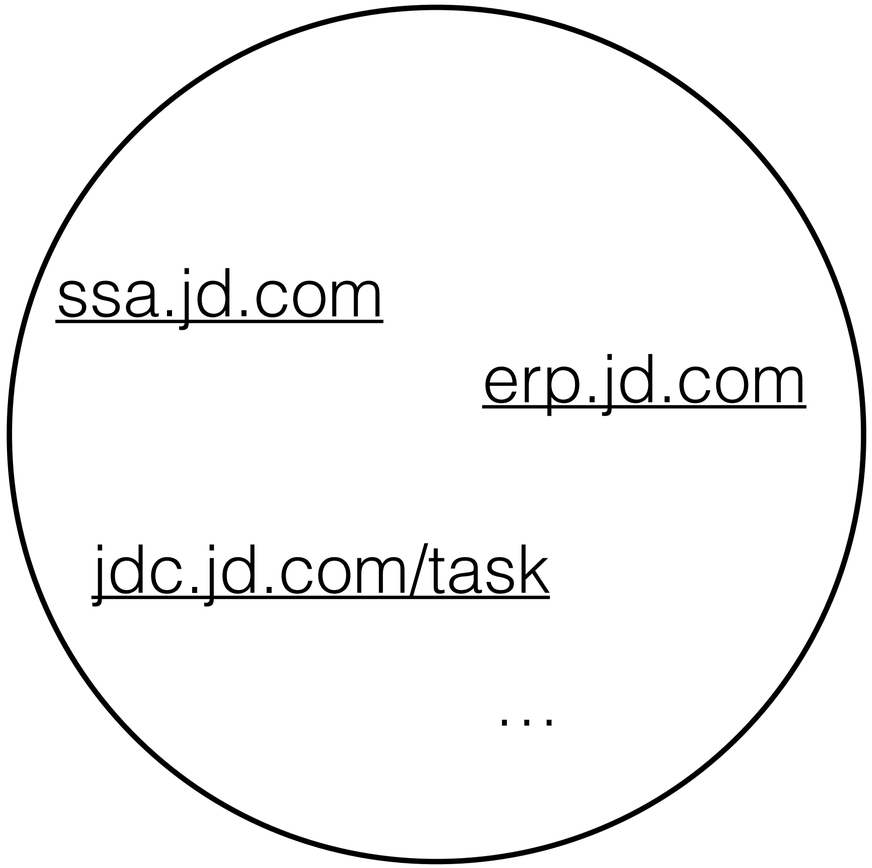
…

1. 服务端获取到 Access Token，并取得用户在第三方平台的信息
2. 对应该服务的账号体系
3. 派发 sessionID 或 JWT

4

单点登录

单点登录 SSO (single sign on)



sso.jd.com

单点登录

1. 用户在 sso.domain.com 登录
2. 单点登录服务写入 token 到 cookie，并设置一级域名可用（*.domain.com）
3. 用户访问某一系统
4. 该系统向单点登录服务询问 token 是否合法
5. 派发自己的 sessionID 或 JWT（可选）

http://passportjs.org

THANKS

FOR YOUR WATCHING