人人网 开放平台

验证与授权

实践

开 场 白

我叫戴洵

API

OAuthT/Z

人人Profile

http://renren.com/xundai

Email

xun.dai@me.com

目录

- 开放带来的问题
- OAuth 1.0的帅呆与鸭梨
- OAuth 2.0的弹性
- OAuth 2.0的Single Sign On
  - Web SSO、Mobile SSO
- OAuth 2.0实现的经验
  - Token的设计
  - HTTPS的问题

缩写

- Autht = Authentication（验证）
- Authz = Authorization（授权）
- 中间凭证 = Authorization Grant
- Password = Resource Owner Password Credentials

实践之路

- 07年11月：OAuth Core 1.0（社区）
- 07年12月：OpenID 2.0 正式发布
- 08年07月：人人（校内）网开放平台正式发布
- 09年05月：人人 Connect 发布
- 09年07月：OAuth 1.0a Draft 00
- 09年09月：人人OAuth 1.0
- 10年07月：OAuth 2.0 Draft 00
- 10年11月：人人 OAuth 2.0

场景 1

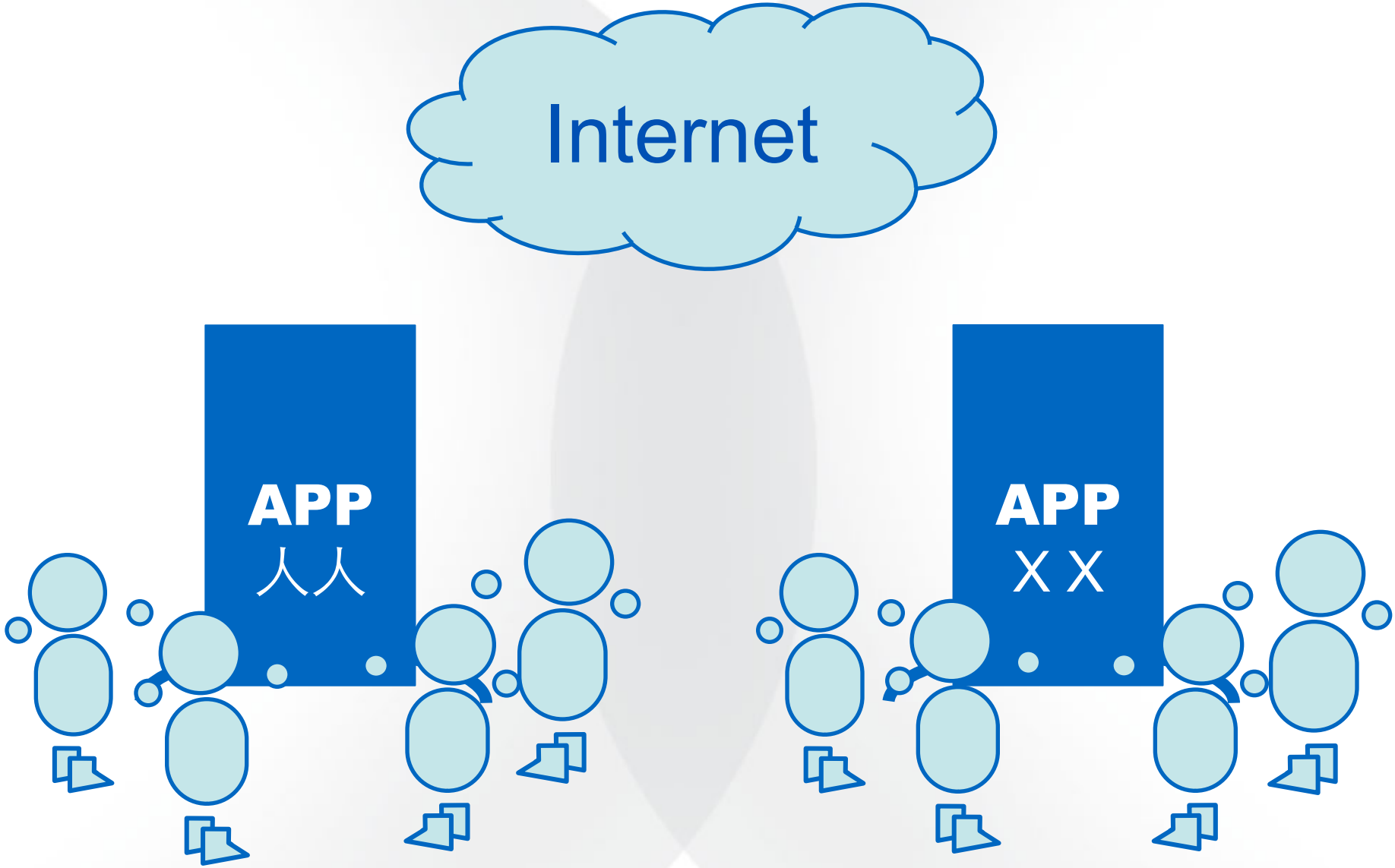
开放带来的问题

2008年

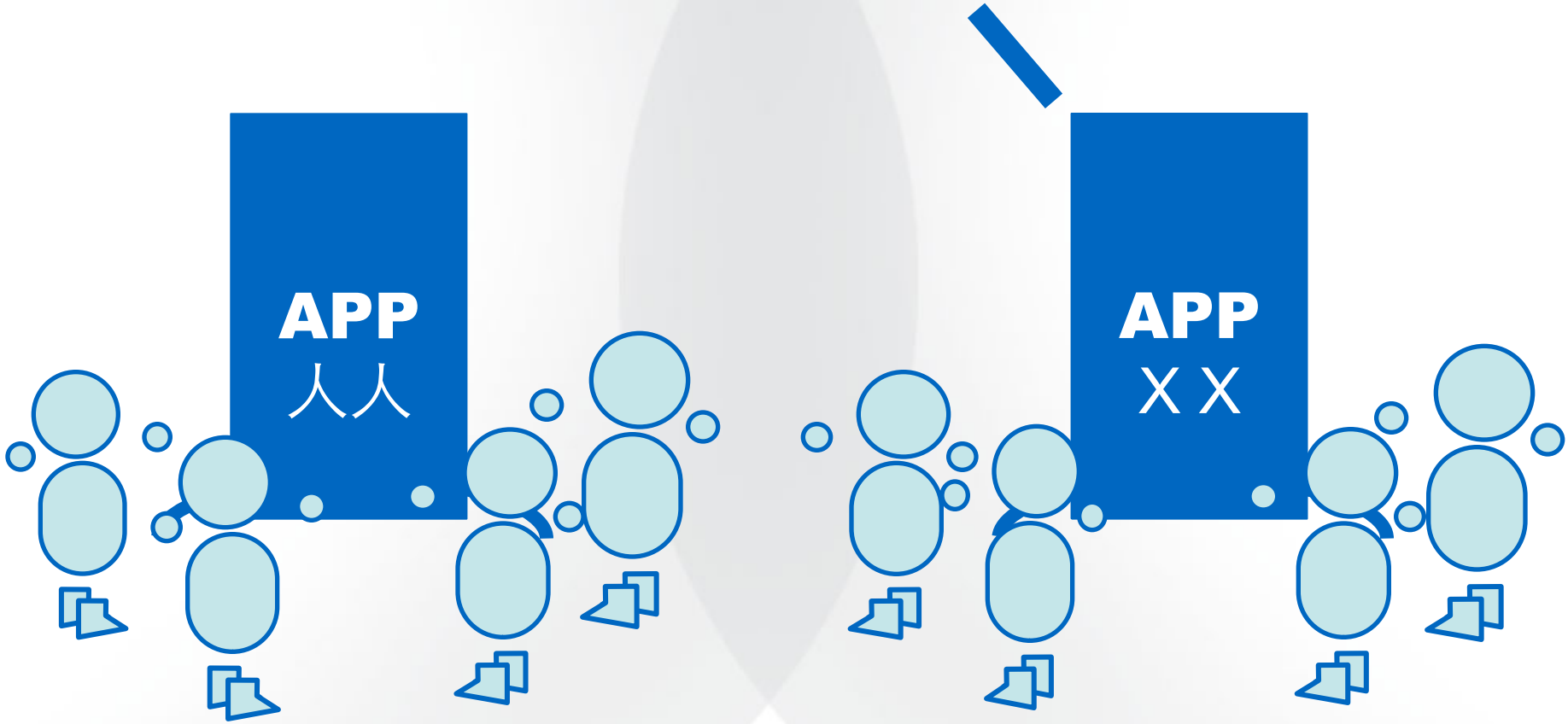
互联网上有很多APP


APP
人人


APP
人人


Internet
APP
人人
APP
X X

“嘿！ 我的用户想访问他们
存储在你那里的资源。”



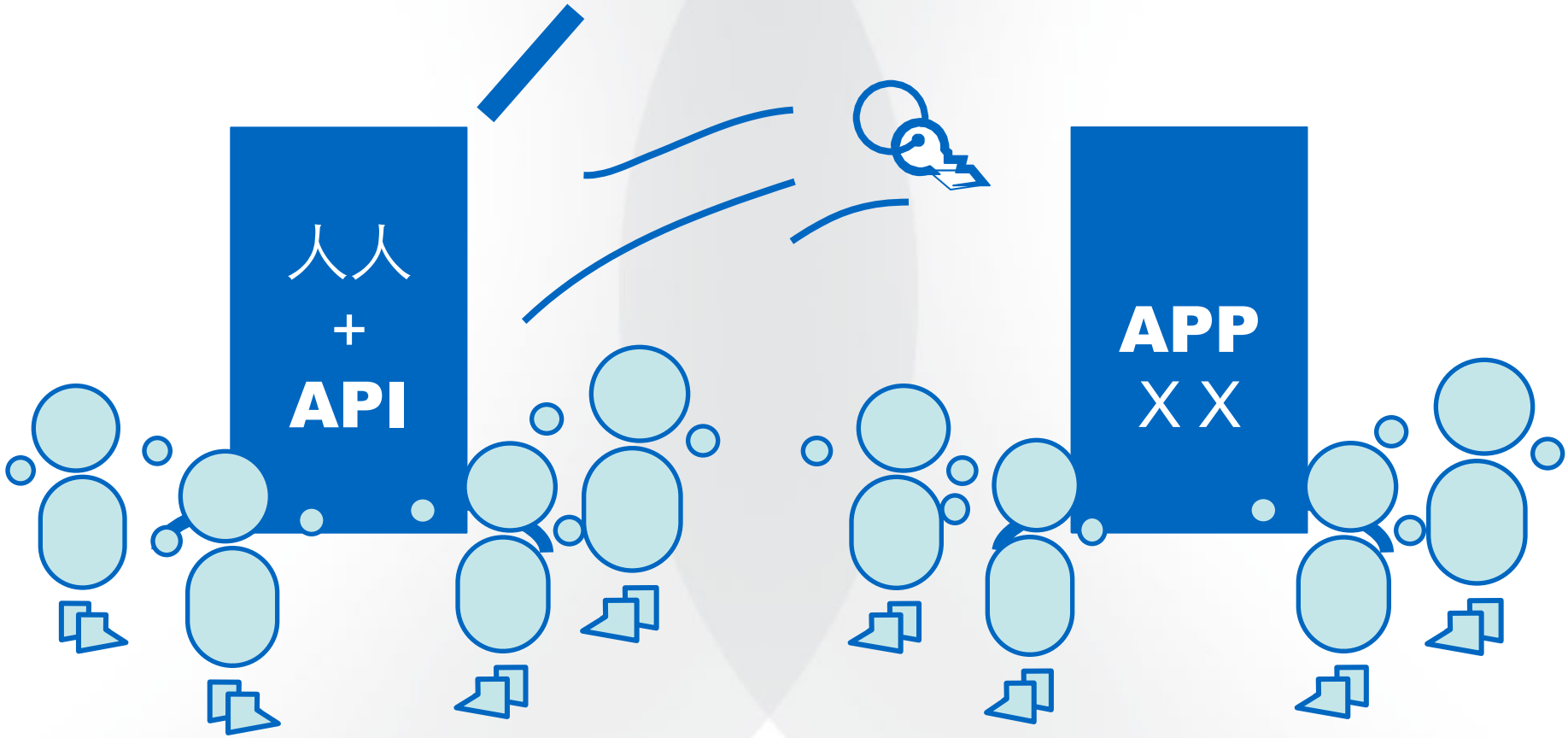
“出来混都不容易，我顶你！我开放API！”

HTTP BASIC

http://user:password@renren.com

Authorization: Basic dxSE2xOkl2Ua3Es098fwL==

“给你钥匙，要慎用哦！”



某哥哥：“妹妹，今晚7点村头玉米地见。”

某姐姐：“我银行密码是XXX。”

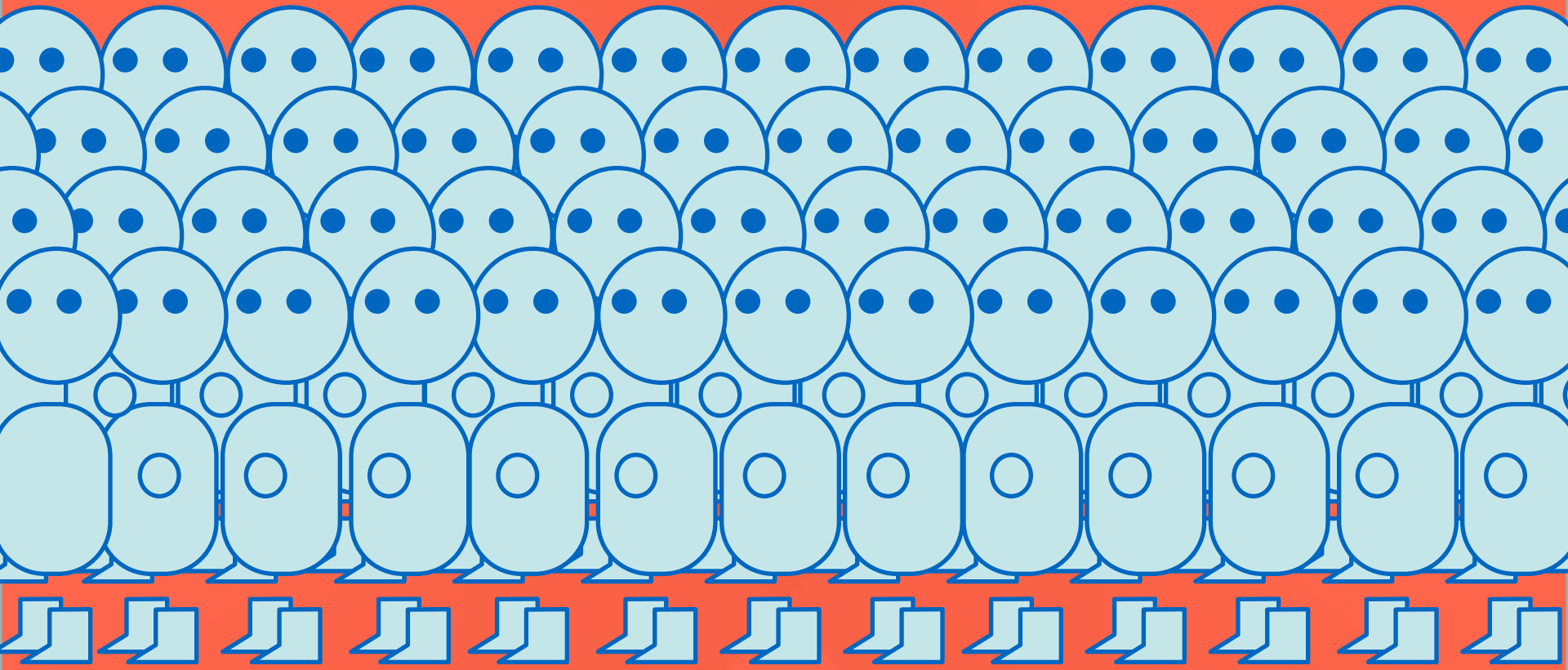
悲了个剧

用户再也不敢
随便输入密码

COUGH

给我个安全的验证与授权机制先！！

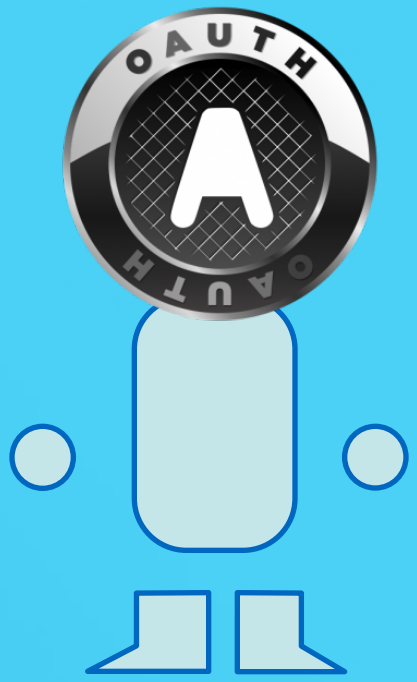
用户伤不起！！！



场景 2

OAuth1.0的
帅呆与鸭梨

2009年



“嘿！ 我们的用户想访问他们存储在你那里的YY资源。”

“把用户交给我的管家（OAuth）吧！”

人人
+
API

APP
X X

人人
+
API

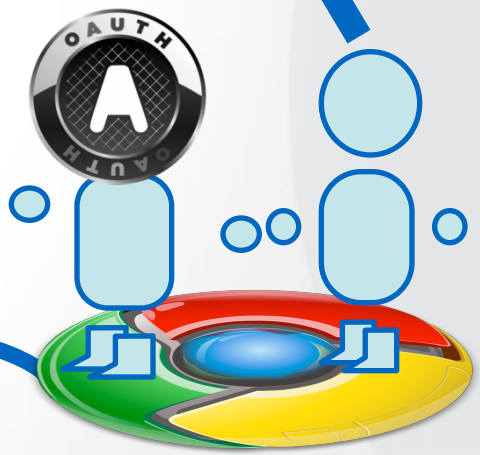
APP
X X

“密码是神马？”

他们在
干神马?

“password”

人人
+
API



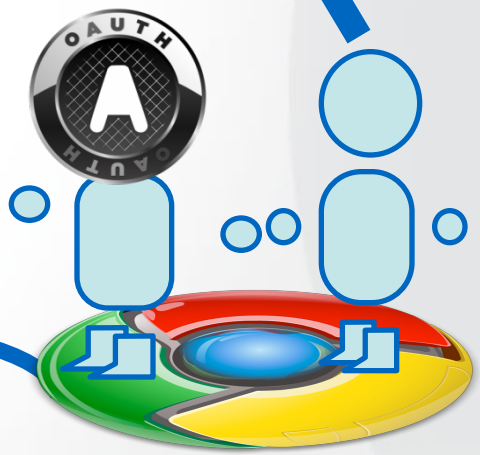
APP
X X

“XX要YY。给吗？”

YY...

“给！”

人人
+
API

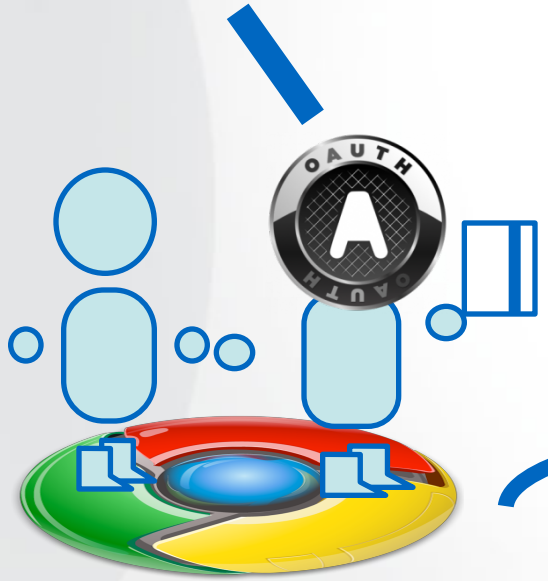


APP
X X


Access Token
(xx)

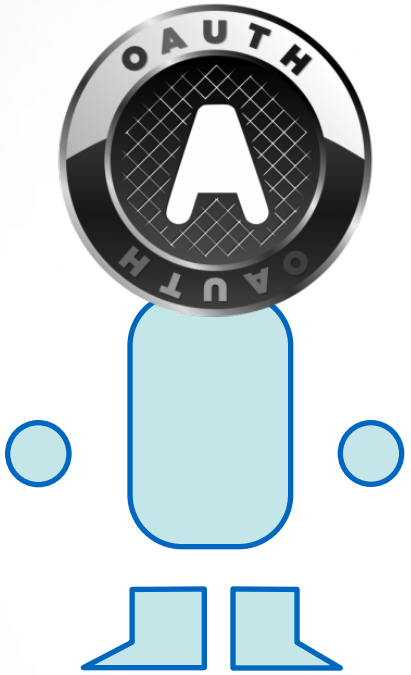
“要YY请刷卡！”

人人
+
API



APP
X X

帅呆



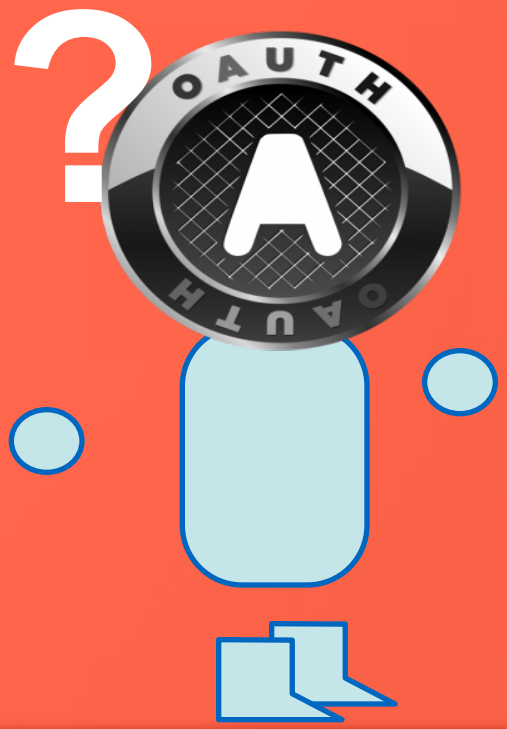
总是?

时间飞逝

智能手机

2009年　　2010年　　2011年

Native APP

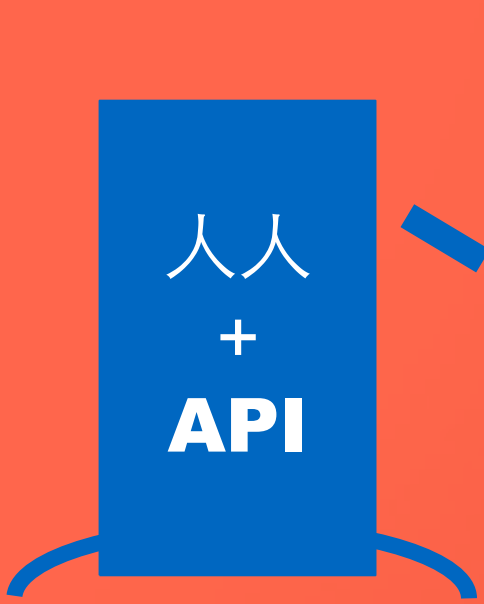

OAUTH
OAUTH


APPLE 3G
5:18
Wednesday, September 15
slide to unlock

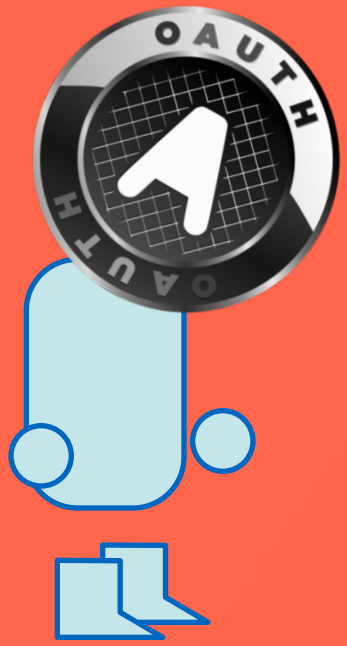
Native APP

复杂

“先获取YY,再换XX再,
换ZZ,再SIG,再…”



“不对！ 不对！
先XX,再…”



鸭梨

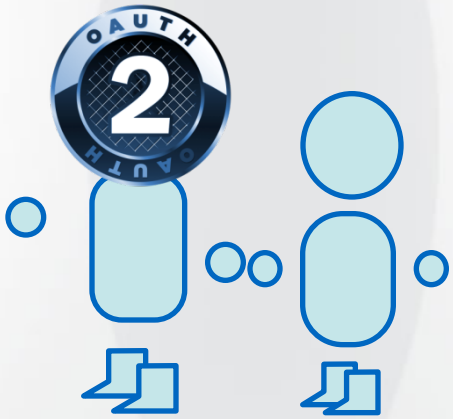
场景 3

OAuth2.0的弹性

2010年

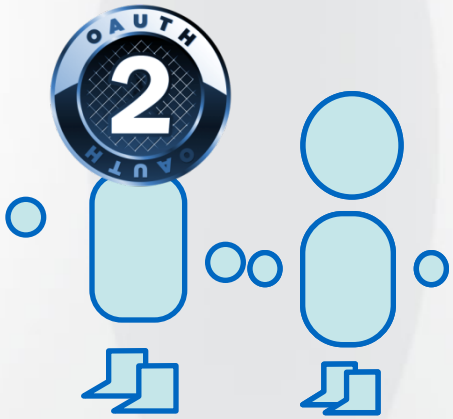
1、Authz Code

人人
+
API



2、Implicit

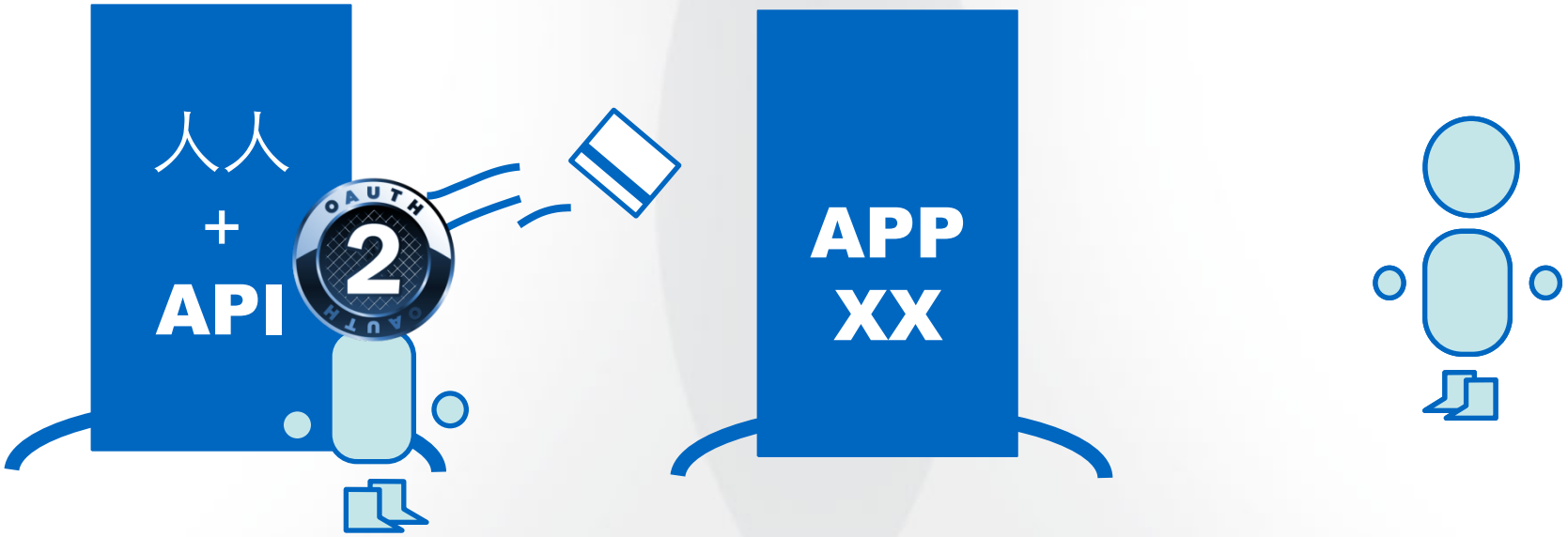
人人
+
API



3、Password

3、Password

3、Password



3、Password

4、Client Credentials

人人
+
API

APP
X X

4、Client Credentials

人人
+
API

APP
X X

简 单

弹性

UNIT 4

OAuth 2.0
SSO

2011年

安全可控的允许第三方访问用户数据

简单、快速、安全的验证用户

SSO

SSO

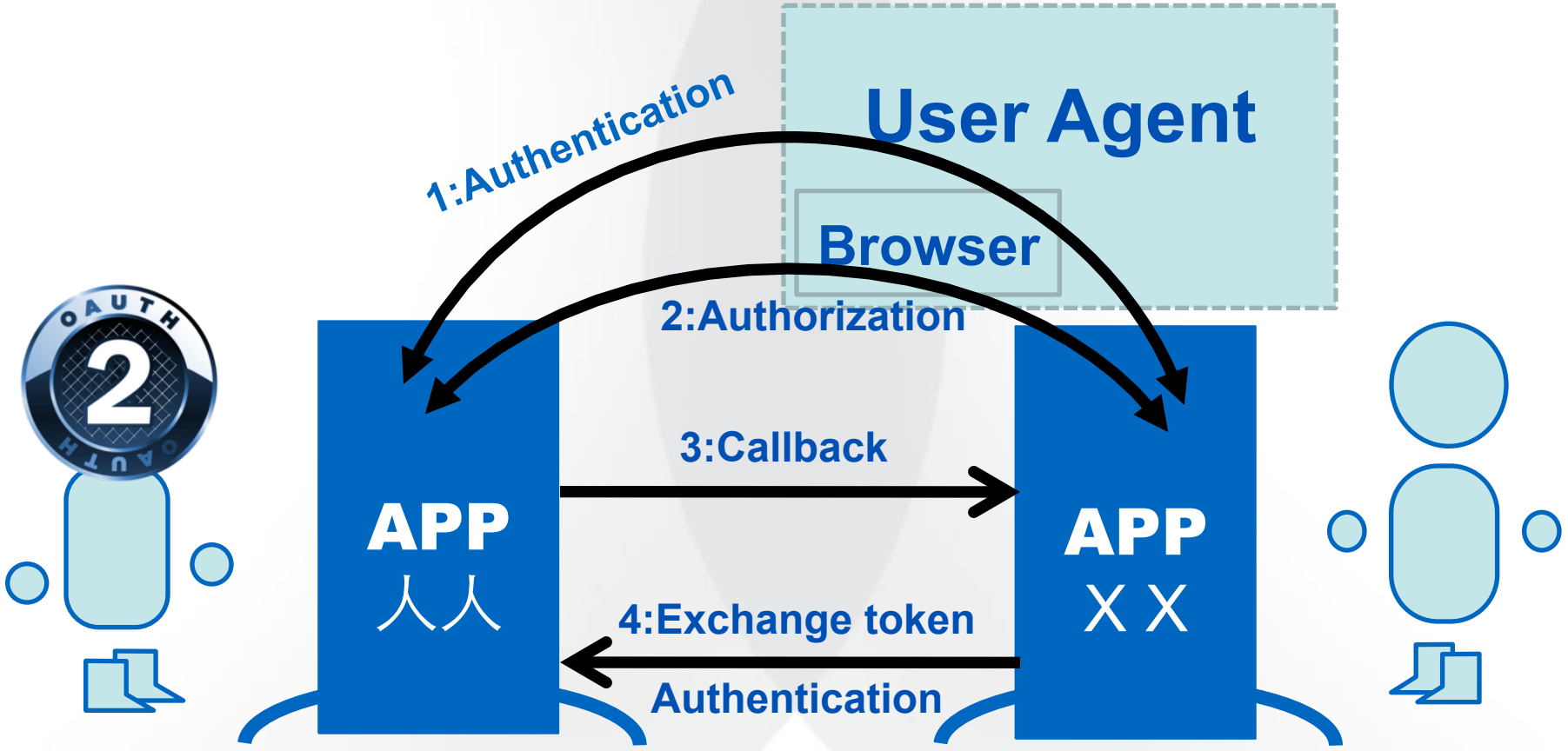
可被调用

登录界面

保持登录状态

安全的回调

抽象流程


User Agent
Browser
1:Authentication
2:Authorization
3:Callback
4:Exchange token
Authentication
APP
人人
APP
X X

WEB SSO

单点：Browser

状态：Cookie

回调：redirect_uri(URL)

回调安全：域名

Mobile SSO

单点：人人客户端/Browser

状态：本地存储的票

回调：Android(Activity) ; iOS(AppDelegate)

回调安全：App签名/标识

User Autht

x_sso_ticket in query

ticket in cookie

Authz Request

GET /oauth/authorize?client_id=aclient
&redirect_uri=http%3A//www.example.com/cb
&response_type=code
&scope=bread+milk+car HTTP/1.1
Host: graph.renren.com
Cookie: ticket=aticket(renren.com)

redirect_uri: http://www.example.com/cb

Authz Request

void android.app.Activity.startActivity(Intent intent)

GET /oauth/authorize?client_id=aclient
&redirect_uri=http%3a%2f%2frenren.com%2fcb%3fandroid_key%3dak
&response_type=code
&scope=bread+milk+car
&x_sso_ticket=xst HTTP/1.1
Host: graph.renren.com

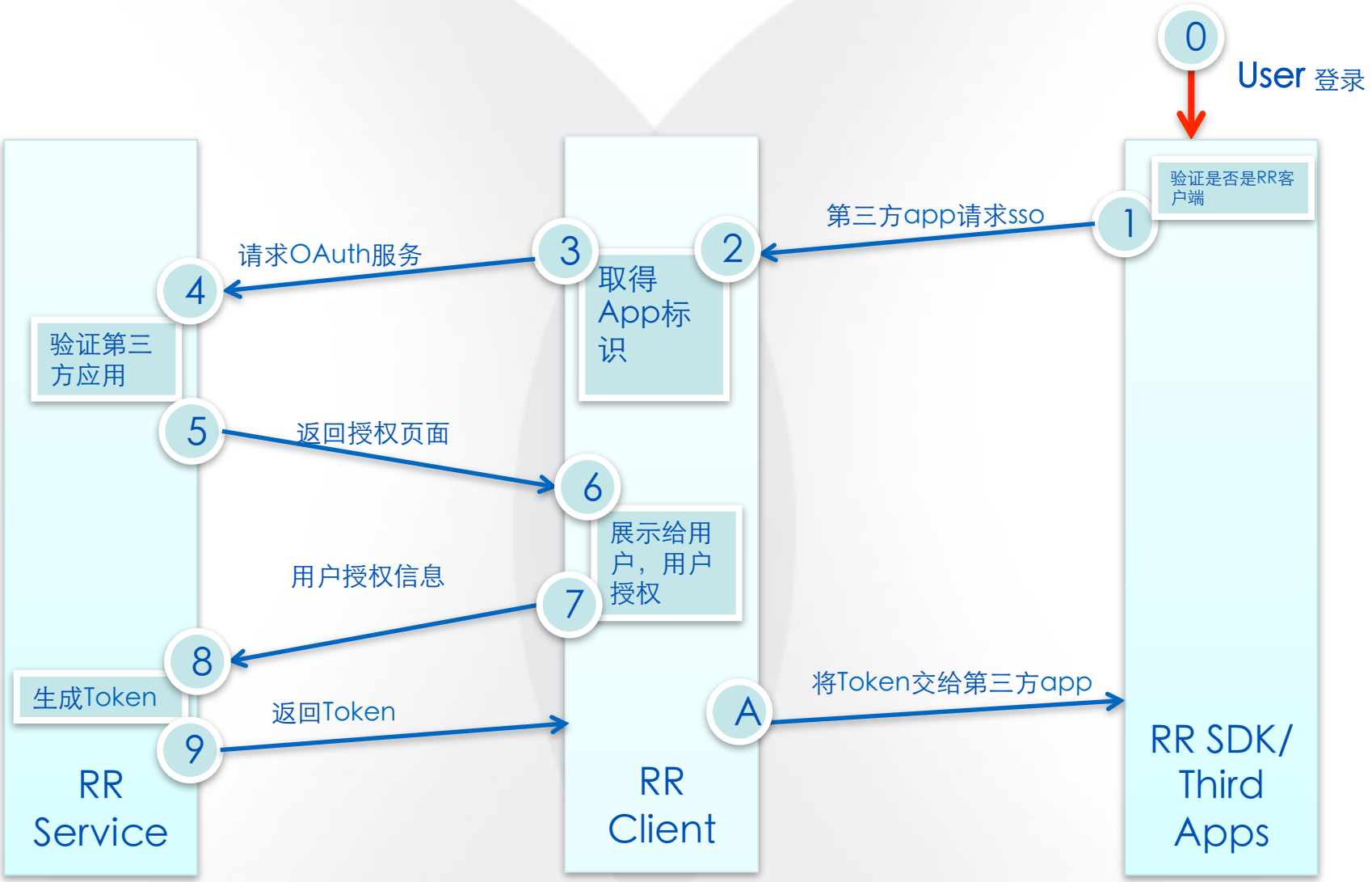
redirect_uri: http://renren.com/cb?android_key=ak

Authz Request

[[UIApplication sharedApplication] openURL:
[NSURL URLWithString:url]];

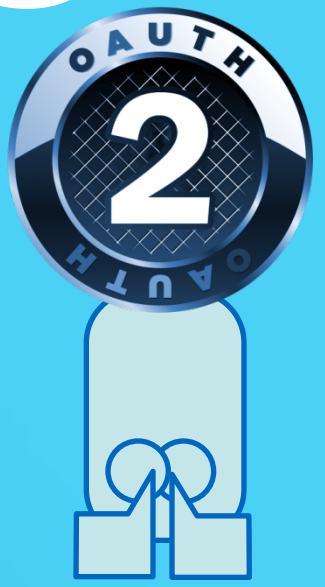
GET /oauth/authorize?client_id=aclient
&redirect_uri=http%3a%2f%2frenren.com%2fcb
%3fapp_store_id%3dsid%26ios_bundle_id%3dibid
&response_type=code
&scope=bread+milk+car
&x_sso_ticket=xst HTTP/1.1
Host: graph.renren.com

redirect_uri: **http://renren.com/cb?app_store_id=sid**
&ios_bundle_id=ibid


0
User 登录
验证是否是RR客户端
1
第三方app请求sso
2
3
请求OAuth服务
4
验证第三方应用
取得App标识
5
返回授权页面
6
展示给用户，用户授权
用户授权信息
7
8
生成Token
返回Token
9
A
将Token交给第三方app
RR Service
RR Client
RR SDK/ Third Apps

UNIT 5

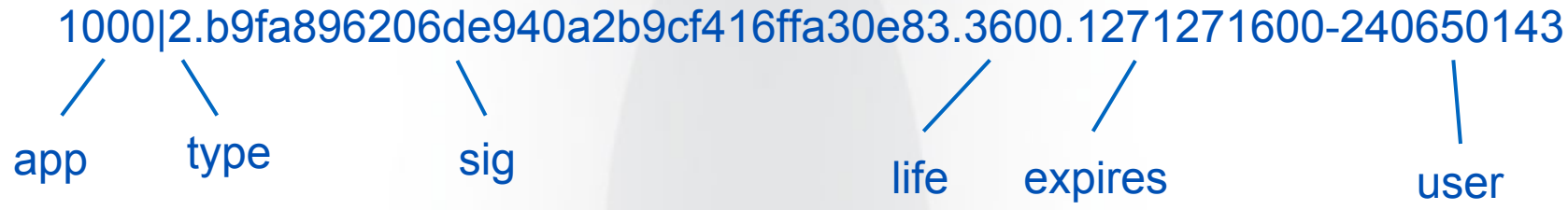
OAuth 2.0实现的经验




Access Token

Access Token的设计

- **Access Token的组成**



- **生成Access Token(sig & expires)**

```
expires = current + life;
//每天随机生成UUID：key
date = (Date) current
key = getEncryptionKey(date);
sig = md5sum(type + life + expires + user + app + key);
```

Access Token的设计

- **Access Token的检验**

```
if (current > expires) return false;
date = expires – life;
key = getEncryptionKey(date);
sig2 = md5sum(type + life + expires + user + app+ key);
if (sig == sig2) return true;
return false;
```

Access Token的设计

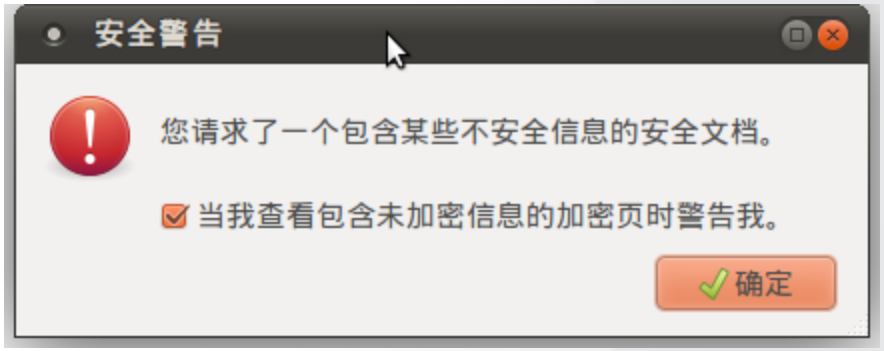
- 优点
  - 低耗验票
  - 全内存操作
  - 后端服务依赖少
- 缺点
  - 更改状态难

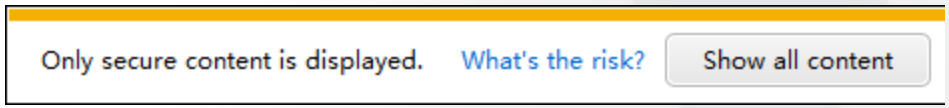
Access Token 生命周期

- OAuth2.0 Access Token的理念
  - Bearer
  - Short-life Access Token
  - Long-life Refresh Token

HTTPS页面包含非HTTPS静态文件警告问题

- End-point在HTTPS下
- 中间显示页面放到HTTP下





人人网
renren.com

情 系 人 人

谢 谢


QCon
北京站·2012年4月18~20日
www.qconbeijing.com（11月启动）
QCon杭州站官网和资料
www.qconhangzhou.com
全球企业开发大会
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE