

统一认证服务切换方案

第一版

修订记录

版本	修订人	时间	修订内容
V0.1	顺炆国	2015-06-26	起稿
V0.2	顺炆国	2015-07-07	增加开发任务列表章节

目录

- 1. 概述.....4
- 2. 整体方案.....4
 - 2.1. 现状及目标.....4
 - 2.2. 切换策略.....4
 - 2.3. 切换方案.....5
- 3. 接口迁移.....6
 - 3.1. 现有接口列表.....6
 - 3.2. 接口映射关系说明.....6
 - 3.3. Mysso 未提供的接口7
- 4. 数据迁移.....8
 - 4.1. 迁移计划.....8
 - 4.2. 数据映射关系.....9
- 5. 回滚方案.....10
- 6. 已知问题.....10
- 7. 开发任务列表.....10
 - 7.1. 数据迁移脚本.....10
 - 7.1.1. 功能点.....10
 - 7.1.2. 注意事项.....10
 - 7.2. 数据回滚脚本.....11
 - 7.2.1. 功能点.....11
 - 7.2.2. 注意事项.....11
 - 7.3. mysso 代理服务及消息转换.....11
 - 7.3.1. 功能点.....11
 - 7.3.2. 注意事项.....11
 - 7.4. uiac 客户端 SDK 库.....12
 - 7.4.1. 功能点.....12
 - 7.4.2. 注意事项.....12

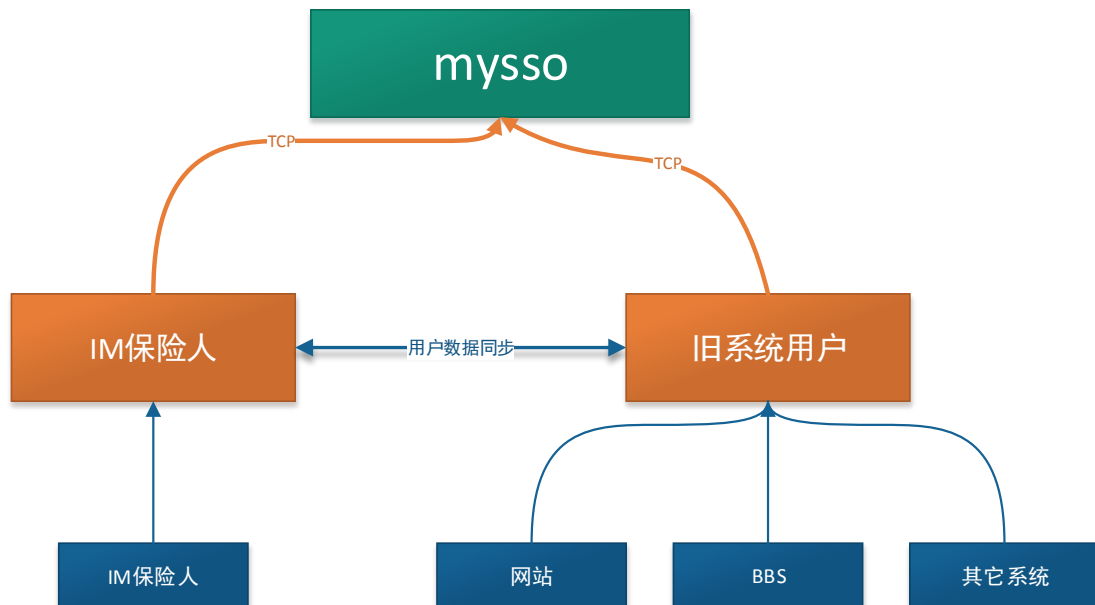
1. 概述

此方案主要描述将原有 myssso 的用户登录、认证及注册等功能迁移到新开发的统一用户认证服务（UIAC）。由 UIAC 来接管用户的注册、账号管理、登录、授权等业务。

2. 整体方案

2.1. 现状及目标

目前是由 SSO 负责所有的用户注册、基本信息修改、登录等功能，因业务扩展，需要使用 UIAC 替换掉 SSO 服务。



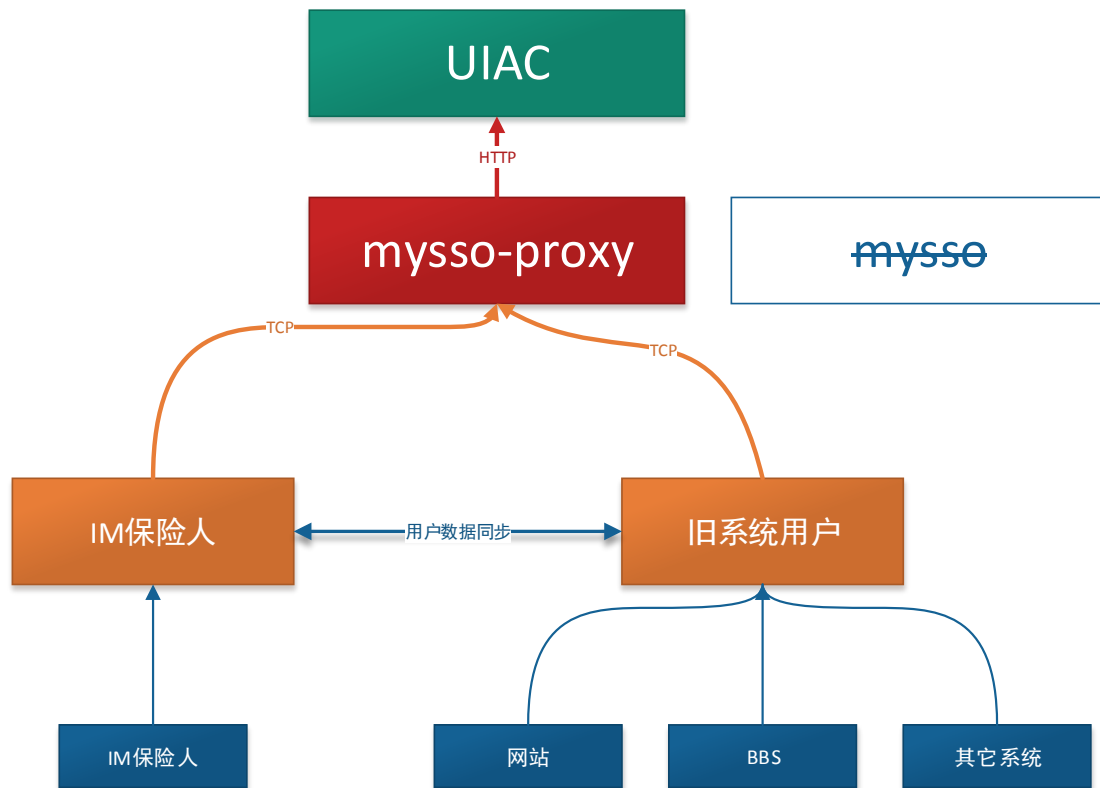
2.2. 切换策略

为了尽量减少对现有系统的影响，切换暂定为两步进行：

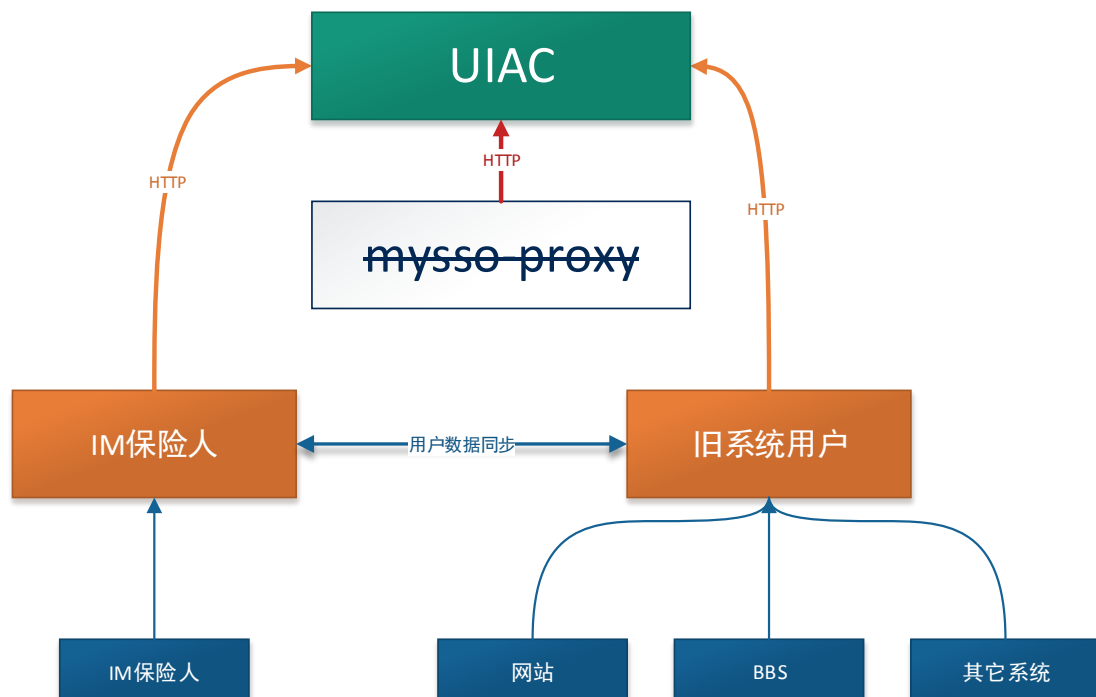
1. 切换 myssso 服务到 UIAC，对原有系统与 myssso 继续使用旧协议对接，提供一个代理服务将原有 myssso 的接口转换到 UIAC 服务上
切换时需要准备如下工作：
 - a. 接口转换时的功能差异及说明
 - b. 数据迁移方案及数据同步脚本
 - c. 切换服务失败时的数据恢复脚本（从 UIAC 库恢复到 myssso 库）
2. 新程序接入时，如果 UIAC 系统已稳定，则可以直接对接 UIAC。对于原有服务则可以分步使用新接口对接 UIAC 服务。

2.3. 切换方案

第一步：替换掉 mysso 服务



第二步：旧系统逐步迁移到 UIAC 新接口上，废弃掉 mysso 代理服务



3. 接口迁移

3.1. 现有接口列表

SSO 接口	说明
checkTicket(String oid, String ticket)	验证用户的登录 Ticket
getTicket(string oid)	获取用户的登录 Ticket
signup(String oid, String uid, String passwd, String account, String email, String mobile, String nick, String status)	注册用户
find(String oid)	检测用户 ID 是否被使用
find(String fieldname, String fieldvalue)	验证用户参数是否使用，filename 的可选值如下： 可以是 oid,uid,account,email,mobile，如果 fieldname=null, 全匹配
exists(String oid)	检测用户 ID 是否被使用
exists(String fieldname, String fieldvalue)	验证用户参数是否使用，filename 的可选值如下： 可以是 oid,uid,account,email,mobile，如果 fieldname=null, 全匹配
getAllUserinfo(String oid)	获取用户信息
signon(String id, String passwd)	用户登录
signon(String fieldname, String fieldvalue, String passwd)	指定参数登录
signoff	未实现
setUserinfo(String oid, String name, String value)	设置用户扩展信息
getUserinfo(String oid, String name)	获取用户扩展信息
getBaseUserinfo(String oid, String name)	获取用户基础信息
setBaseUserinfo(String oid, String name, String value)	设置用户基础信息
updatePasswd(String oid, String passwd)	更新用户密码
updatePasswd(String oid, String oldpasswd, String passwd)	更新用户密码
updateUser(String oid, String uid, String passwd, String account, String email, String mobile, String nick, String status)	更新用户信息

3.2. 接口映射关系说明

mysso	uiac	接口说明
find	account/user/{uid} [GET]	Mysso 支持 uid, account, email, mobile 的查找方式，而 UIAC 中，仅 account=mobile, email 登录已取消，所以仅支持 uid 和 mobile 两种查找方式，email 查找则默认返回空
signup	account/user [POST]	差异点： 1. 密码加密方式不同，需要作转换 2. mysso 未有 AppID，因此需要默认使用一个 AppID

signon	AccessToken [POST]	差异点: 1. uiac 支持 account 的密码和验证码登录, 而 mysso 支持 uid、account、email 登录, 其中 uid 和 email 登录都会失败 2. 返回值仅返回了 uid 或空值, UIAC 返回内容较多 3. mysso 未有 AppID, 因此需要默认使用一个 AppID
signoff	AccessToken [DELETE]	注销登录, mysso 未实现, 可以直接中转到 UIAC 执行注销
checkTicket	AccessToken [GET]	需要转换 UIAC 的查询结果
getTicket	AccessToken/{uid} [GET]	需要转换 UIAC 的查询结果
getAllUserinfo	account/user/{uid} [GET]	需要转换用户信息
setmultiuserinfo	account/user/{uid} [PUT]	
getUserinfo	account/user/{uid} [GET]	需要转换用户信息
setUserinfo	account/user/{uid} [PUT]	
getBaseUserinfo	account/user/{uid} [GET]	需要转换用户信息
setBaseUserinfo	account/user/{uid} [PUT]	
exists	account/parameter account/user/{uid} [GET]	验证用户参数
updatePasswd	account/password [PUT][POST]	需要对接口进行转换操作
updateUser	account/user/{uid} [PUT]	

注:

1. 因 mysso 服务中 UserInfo 表的信息未启用, 因此代理服务可不实现此接口。
2. Uiac 中查询用户信息是同时将基础信息和扩展信息一并查出, 因此只有一个接口, 在实现 mysso 接口时需要拆分数据。
3. 实现中对于缺少的字段, 对应数据置为空

3.3. Mysso 未提供的接口

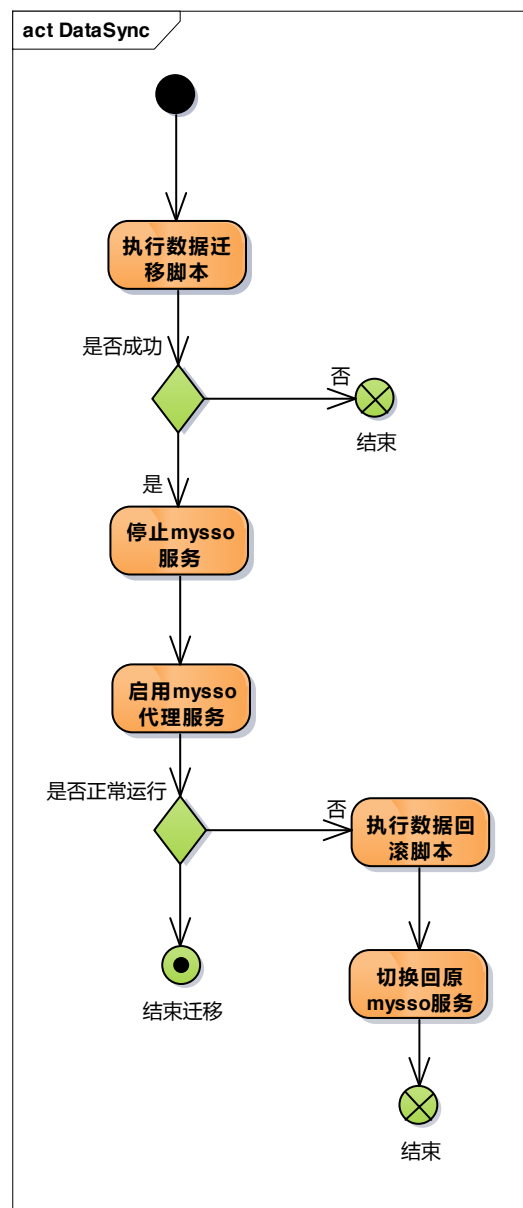
- 短信验证码接口
- 绑定子帐号接口
- 账号状态变更接口
- 授权码更新接口
- 短信验证码登录接口

- 授权信息通知服务

4. 数据迁移

4.1. 迁移计划

在接口上线前，预先执行数据迁移脚本，数据迁移脚本执行完成后，执行 mysso 服务替换操作。



4.2. 数据映射关系

Mysso 数据库与 UIAC 数据库的迁移，是在切割前执行，通过代码将 mysso 的数据内容同步到 UIAC 数据库中。

字段及表映射关系如下：

BaseUser

字段名	对应表-字段	说明
Oid	User.uid	用户唯一 ID
Uuid	User.uid	
Account	User.account	用户帐号
Nick	User_info.user_name	用户名
Email	User_info.email	邮箱
Passwd	User.password	需要作密码转换
Crttime	User.add_date	添加时间
Moditime	User_info.edit_date	修改时间
Lastsignon	user_stat.last_login_date	最后登录时间
Signontimes	User_stat.login_num	登录总次数
Status	User.status	用户状态

UserInfo

字段名	对应表-字段	说明
Oid	user_extend_info.uid	用户唯一 ID
UserOid	user_extend_info.uid	
Name	user_extend_info.ext_key	扩展字段名
Value	user_extend_info.ext_value	扩展字段值

baseuserext

字段名	对应表-字段	说明
Oid	user_info.uid	用户唯一 ID
F1	无	用户所属城市编号
F2	无	用户形象照
F3	无	用户类型：A 经纪，U 市民 B 博主
F4	User_info.sex	用户性别：G 男，M 女
F5	User_info.email_is_verify	邮箱是否验证
F6	无	省份 ID
F7	无	地区 ID
F8	无	是否聚才通用户
F9	无	是否聚才通用户

未知字段：

用户注册 AppID

需要确定的数据：

AppInfo 表中的初始数据，即当前已有系统的信息，以及对应的编号。

5. 回滚方案

当切换后发生异常时，停止代理服务，启动 mysso 服务，可以继续提供账号服务。对于运行中注册的用户数据，可以使用回迁脚本将数据从 uiac 库同步到 mysso 库中。映射关系同数据迁移方案。

6. 已知问题

- 因接口定义差异问题，使用代理服务的方案会导致 uiac 的部份功能无法使用。
- 因数据库定义的差异，会丢失某些数据字段
- 回滚时的密码问题，uiac 采用了不可逆的方式存储密码，因此回滚数据时，密码无法恢复为原始密码。
- 需要约定当前正在运行系统以及其对应的 AppID
- Mysso 中没有注册时用户所使用的 appid，因此需要约定一个值，以确定用户是从哪个 App 注册

7. 开发任务列表

7.1. 数据迁移脚本

7.1.1. 功能点

- ✧ 根据“4.2 数据映射关系”实现同步脚本编写
- ✧ 密码转换处理逻辑
- ✧ 数据迁移测试

7.1.2. 注意事项

- ✧ 导入时需要注意数据表的主从关系
- ✧ UIAC 密码为不可逆加密回滚时需要注意
- ✧ 在导入前，如果 UID 是自增加，则需要暂停 UID 增加，导入完成后将其初始值设置为 10000000
- ✧ 对于没有的数据默认置空，时间设置为当前时间，APPID 预先约定好

7.2. 数据回滚脚本

7.2.1. 功能点

- ✧ 回滚脚本编写
- ✧ 密码重置处理
- ✧ 回滚脚本测试

7.2.2. 注意事项

- ✧ 需要记录最后同步数据的时间，回滚时根据此时间查找新增、修改、删除的用户信息
- ✧ 回滚时字段映射关系同“4.2 数据映射关系”中描述的
- ✧ 密码默认不可逆，回滚时只能将其设置为一个默认值，此时需要记录受影响的用户 ID

7.3. mysso 代理服务及消息转换

7.3.1. 功能点

- ✧ Mysso 代理服务框架搭建
- ✧ Mysso 服务协议层代码实现
- ✧ 日志记录功能
- ✧ Mysso 接口协议功能适配功能开发（参数转换、返回内容分解、缓存处理）
 - .1. 各个接口实现，详见“3.1 现有接口列表”
 - .2. 接口参数转换（JSON 解析、参数格式处理）
 - .3. 返回内容分解（因接口返回内容不一致，需要组合成 mysso 协议支持的内容）
 - .4. 增加缓存处理（如：登录 Token 的验证、获取等，减少数据交互）
 - .5. UIAC 客户端 SDK 对接开发
- ✧ 功能测试

7.3.2. 注意事项

- ✧ 对于缺少的参数，需要约定默认值
- ✧ 返回结果中缺少的数据，默认返回空值
- ✧ 对于不支持的登录，如：邮箱、普通帐号等，直接拒绝登录

7.4. uiac 客户端 SDK 库

7.4.1. 功能点

- ✧ HTTP 客户端库实现（用于提交请求，需要支持 POST、PUT、GET、DELETE、PATCH 等请求方法）
- ✧ HTTP 服务端监听库实现（用于接收服务端推送消息）
- ✧ SDK 框架编写
- ✧ 接口实体对象编写
- ✧ 接口功能实现（可支持同步、异步调用两种方式）
- ✧ 回调通知接口编写
- ✧ 功能测试

7.4.2. 注意事项

- ✧ 接口详细内容请参见“统一认证服务接口文档”
- ✧ 本期优先实现 mysso 相关的接口，不实现 HTTP 监听服务、验证码等“3.3 mysso 未提供的接口”
- ✧ 接口调用优先实现同步调用接口，预留异步调用接口的（本期可不实现）