**PaaS平台框架与统一身份认证系统**

**软件需求规格说明书**

Software Requirements Specification

**编号**：**TMP-SRS**

**版本 1.0**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 总页数 |  | 正文 |  | 生效日期 |  | | |
| 制作单位 |  | | | 责任者 |  | 批准者 |  |

**文档变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修改编号** | **版本** | **修改内容** | **修改人** | **修改日期** |
| 1 | V1.0 | 文档建立 | 邓清林 | 2015-11-5 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

[1 引言 4](#_Toc435368417)

[1.1 编写目的 4](#_Toc435368418)

[1.2 背景 4](#_Toc435368419)

[1.3 文档编写约定 4](#_Toc435368420)

[1.3.1 优先级定义 4](#_Toc435368421)

[1.4 参考资料 5](#_Toc435368422)

[2 任务概述 5](#_Toc435368423)

[2.1 目标 5](#_Toc435368424)

[2.2 范围 5](#_Toc435368425)

[2.3 用户特点 5](#_Toc435368426)

[2.4 假设和约束 6](#_Toc435368427)

[3 详细需求 6](#_Toc435368428)

[3.1 统一身份认证系统 6](#_Toc435368429)

[3.1.1 登录 7](#_Toc435368430)

[3.1.2 登出 7](#_Toc435368431)

[3.1.3 统一认证 7](#_Toc435368432)

[3.1.4 与已有认证系统对接 8](#_Toc435368433)

[3.1.5 多租户支持 8](#_Toc435368434)

[3.1.6 认证API 9](#_Toc435368435)

[3.1.7 策略管理 9](#_Toc435368436)

[3.1.8 权限管理 9](#_Toc435368437)

[3.1.9 系统管理与监控 9](#_Toc435368438)

[3.2 cfWeb管理系统 10](#_Toc435368439)

[3.2.1 管理员系统 12](#_Toc435368440)

[3.2.2 用户系统 16](#_Toc435368441)

[3.3 ESB系统 18](#_Toc435368442)

[3.3.1 部门管理员 20](#_Toc435368443)

[3.3.2 服务提供者 20](#_Toc435368444)

[3.3.3 服务使用者 23](#_Toc435368445)

[3.3.4 系统管理员 25](#_Toc435368446)

# 引言

## 编写目的

编写本文档的目的是为了便于宁波市政务云计算中心和PaaS项目组开发人员间进行理解和交流，反映出功能需求的结构，使之成为软件开发工作的基础和依据，并作为确认测试和验收的依据。

## 背景

宁波市政务云计算中心是全市智慧城市和电子政务应用的支撑平台。宁波市政务云计算中心以同构化信息系统建设，谋划平台即服务模式，引导部门间业务协同发展，建立一体化公共基础应用支撑体系为突破。在公共基础应用支撑的优化配置和集中管理的基础上，实现应用系统与中间件的松耦合布局。利用总线设计的思想，为智慧城市和电子政务应用提供业务互动基础生态环境，提高跨部门业务之间的协作能力。

PaaS 平台基于网络提供对应用完整生命周期（包括设计、开发、测试和部署等阶段）的支持，减少了用户在购置和管理应用生命周期内所必须的软硬件以及部署应用和 IT 基础设施的成本，同时简化了以上工作的复杂度。

## 文档编写约定

### 优先级定义

必须且优先 5

必须但一般 4

可选且优先 3

可选且一般 2

本期不考虑 1

## 参考资料

术语参见《PaaS平台框架及统一身份认证系统-术语定义》文档。

用例规约参见《PaaS平台框架与统一身份认证系统用例规约描述》文档。

# 任务概述

## 目标

通过建设支撑智慧城市各领域应用运行及基础设施资源的PaaS平台框架，为智慧城市应用创建发布及管理提供统一的运行环境及服务资源支撑，实现资源的统一规划管理、应用的快速部署及时、资源的弹性扩展。

通过建设基于PaaS平台框架的统一身份认证系统，为各应用及系统提供统一的、标准的用户身份认证和权限管理，通过统一身份认证系统实现不同应用系统的用户、角色和组织机构统一化管理，实现各种应用系统间跨域的单点登录和单点退出和统一的身份认证功能。

## 范围

## 用户特点

PaaS平台框架及统一身份认证系统的用户主要为政府内部人员。PaaS平台框架整合宁波市政务云计算中的基础设施服务（IaaS）、各种中间件及应用，提供统一的平台服务；统一身份认证系统为政府资源区应用、公众区应用提供完善的统一身份认证和单点登录。

## 假设和约束

列出本项目的假设和依赖关系，如准备条件、开发周期限制等。

|  |  |  |
| --- | --- | --- |
| 假设 | 序号 | 假设 |
| 假设一 |  |
| 假设二 |  |

|  |  |  |
| --- | --- | --- |
| 依赖 | 序号 | 内容 |
|  |  |

# 详细需求

## 统一身份认证系统

应用通过接入统一身份认证系统可实现应用的单点登录和用户认证的统一化管理；同时统一身份认证系统支持与已有的认证系统对接，通过对应用和已有的认证系统的少量修改即可实现已有认证系统的用户可登录统一身份认证系统的应用，同样，统一身份认证系统的用户可登录已有认证系统的应用。

功能模块图如下：



**图1 统一身份认证系统功能模块图**

### 登录

统一身份认证系统对应用提供SSO（Single Sign On，单点登录）功能，即在多个应用系统中，用户只需登录一次就可访问所有相互信任的应用。

### 登出

统一身份认证系统为应用提供SSO（Single Sign Out，单点登出）功能，即在多个应用系统中，用户只需完成一次登出操作就能登出所有相关联的应用。

### 统一认证

应用接入统一身份认证系统后，便可把用户认证模块从应用中解耦出来，相关联的应用之间不再需要交差访问用户，只需通过统一身份认证系统即可完成认证操作，以此实现用户认证的统一化管理。

同时，统一身份认证系统支持多协议认证方式（如SAML、OpenID、OpenID Connect、Passive STS），以满足多元化应用多样化认证方式的需求。

### 与已有认证系统对接

在特定应用场景下，一些部门应用或门户网站提供了独立的认证系统供自己的应用认证，当用户访问权限开放时，第三方应用可直接访问用户接口以实现认证操作。然而，一些安全部门应用或门户网站对诸如用户信息之类的敏感数据把控较严，不对外开放用户访问权限，第三方应用想接入到这些认证系统，则必须修改认证系统以支持安全认证协议，且需要对第三方应用支持的多样化认证方式做匹配，大大提高了认证系统的复杂性和维护难度。

通过统一身份认证系统与已有认证系统的对接，有效的解决了上述问题。第三方应用接入已有认证系统，首先，第三方应用向统一身份认证系统发起认证请求，统一身份认证系统在收到请求后，解析请求信息并寻找对应的IdP（Identity Proveider，即身份提供商），已有的认证系统作为统一身份认证系统的IdP向统一身份认证系统提供认证信息，接着统一身份认证系统把认证信息返回给第三方应用以实现认证操作。同样，已有认证系统的应用接入统一身份认证系统，首先，应用向已有认证系统发起认证请求，已有认证系统解析请求信息并寻找对于的IdP，统一身份认证系统作为已有认证系统的IdP为已有认证系统提供认证信息，接着已有认证系统把认证信息返回给应用以实现认证操作。在统一身份认证系统和已有认证系统对接的整个过程中，只需要确定两个认证系统间的安全认证协议，避免了用户数据的交差访问，不仅保证了用户信息的安全，还大大降低了认证系统的复杂性，实现了应用认证的统一化管理。

### 多租户支持

统一身份认证系统为企业应用提供多租户支持，不同部门可申请成为统一身份认证系统的一个租户，每个租户下的应用享有一套完整的统一认证体系，从而实现多个租户间应用和数据的完全隔离，在节省计算资源的同时为企业应用认证的统一管理带来便利。

### 认证API

API（Application Programming Interface,应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。在统一身份认证系统范畴内，由于需要认证的系统在语言、功能、形式上各有不同，导致接入方式及接入程度也不尽相同。由认证系统提供基本的操作API，如登陆、登出、用户查询等60余种，被认证方根据自己需求合理整合后接入认证系统。

### 策略管理

XACML是一种用于决定请求/响应的通用访问控制策略语言和执行授权策略的框架，它在传统的分布式环境中被广泛用于访问控制策略的执行。在典型的访问控制框架中，有策略执行点PEP(Policy Enforcement Point)和策略决定点PDP(Policy Decision Point)。PEP用于表达请求和执行访问控制决定。PDP从PEP处接受请求，评估适用于该请求的策略，并将授权决定返回给PEP。在同一身份认证系统中，主要完成对XACML做相应的支持。

### 权限管理

权限管理，一般指根据系统设置的安全规则或者安全策略，用户可以访问而且只能访问自己被授权的资源，不多不少，按用户身份及其所归属的某项定义组来限制用户对某些信息项的访问。在统一身份认证系统中，需要能够对RESTful或者SOAP API进行权限管理，支持高级权限审计和管理等。在单点登录后能够对访问资源进行一定程度上的管理。

### 系统管理与监控

系统管理和监控，需要对统一身份认证系统进行统一的管理与健康，比如支持内置标准的访问和性能统计信息的收集和监控，提供对一些关键度量进行监控和管理，支持企业级日志系统的集成等。

## cfWeb管理系统

建设应用基础设施的PaaS平台，为智慧城市应用创建发布及管理提供统一的运行环境及服务资源支撑，实现资源的统一规划管理、应用的快速部署、资源的弹性扩展。对于Cloud Foundry平台，提供Web访问界面，以便用户可以摆脱使用命令行部署应用等，方便快捷进行应用部署，资源管理等。

PaaS平台框架应广泛地支持流行的应用开发语言、开发框架和后端服务，让开发者专注于应用开发和创新，而无需考虑应用开发所需的环境搭建和维护。应具有开放、智能、敏捷和安全的特点，为用户提供满足不同需求的计算资源，实现应用生命周期智能化管理，包括应用软件开发管理，连续集成，部署运维，监控管理，弹性扩容等；提高应用的开发和运维效率，降低IT基础设施支出，减少部署和运维的成本。特别是提供用户所需的组织结构、计算资源隔离和权限管理等功能。应基于目前领先的PaaS技术Cloud Foundry。PaaS平台可根据用户和角色使用平台的不同需求满足其对计算平台资源需求的不同抽象，提供不同的计算资源组合单元。应用可以通过不同的计算资源和服务组件来实现动态整合并安全交付。特别是作为一个开放、敏捷、智能和安全的云计算平台，PaaS平台应可以使得用户实现协同开发、连续集成、快捷部署、可视监控、高效管理和弹性扩展应用服务；应能够极大减少应用部署、升级、扩展和运维时间和成本。应让部门用户专注于应用本身和创新以推动新的业务增长点。

PaaS平台应能支持计算能力的横向扩展（Scale-out），并提供可定制标准规范，引导智能应用支撑服务的新增及部署。

cfWeb系统将分为两个子系统，管理员和用户所使用的系统将不同。其所具有的功能也不同。管理员使用的cfWeb管理员系统将以监控和对CF集群设置为主，而用户使用cfWeb用户系统将以应用管理和应用监控为主。

cfWeb功能模块图如下，

图1是管理员模块的功能模块图：

图2是用户模块的功能模块图：



图1：管理模块功能模块图



### 管理员系统

#### 登录

功能说明：

1. ：系统登录需要与WSO2平台进行整合。当用户登录时，需将用户名和密码发送给WSO2系统，在WSO2的系统中进行验证，返回验证的结果和token。后续操作将通过用户、密码和token来进行验证。
2. ：cfWeb系统的用户注册需要将用户名和密码信息提交到WSO2系统中，调用WSO2的接口将用户信息添加。

#### 仪表盘

功能说明：

1. ：管理员通过登录页面验证进入cfWeb管理员系统的主页面，在主页面中对应CF集群的仪表盘，其中包括：所有机器的状态、所有组件状态以及所有服务状态。同时，应显示整个CF系统的资源使用情况，包括内存、硬盘、CPU、服务的使用量。同时，还应该显示出CF集群的应用运行情况，运行中的应用、停止的应用等信息。

#### 应用管理

应用管理主要分为应用模板和应用列表两个模块。在cfWeb管理员系统中，管理员应注重的为应用的状态。

**模块1：应用管理**

功能说明：

通过此功能上传应用代码并开启应用；平台将自动根据用户申请的资源环境从资源池获取资源，通过容器的方式部署应用运行环境及服务，将用户上传的部署包部署到容器中并启动应用。

在该块内容中，管理员将有权上传应用，但不作为重点。应用状态的监控是本模块内容的核心。将提供更详细的仪表盘进行监控应用的状态。总体来说，这块内容的核心是从真正的用户的使用角度出发，提供最方便最全面的应用监控。

a)：新建应用：根据应用进行名称配置，系统会根据名称+域名自动生成应用访问的URL。

b)：应用详情：显示当前应用所占用的资源，包括内存、CPU等等

c)：删除应用

d)：启动应用

c)：重启应用

e)：终止应用

**模块2：应用模板**

功能说明：

应用模板模块主要作用是用于创建后续部署应用时需要的基础环境，应用编写语言、使用的数据存储服务等。管理员可适时的根据需求，创建不同的应用模板，提供给开发者使用。例如，可提供针对Java的不同规格的应用模板。

a)：新建模板：建立应用部署标准模板，模板建立好之后可供用户根据不同应用选择合适的模板进行应用部署。

b)：编辑模板

a)：删除模板

#### CF设置

在该模块，可对Cloudfoundry平台以及cfWeb系统进行管理，主要应包含域名管理、配额管理和用户管理。

**模块1：域名管理**

功能说明：

该模块包含基础域名管理和应用域名管理。

基础域名管理可以对Cloudfoundry平台的基础域名进行设置。在修改过程中，首先应选择相对应的组织。可进行添加域名、删除域名、修改域名等操作。

应用域名管理可以对Cloudfoundry平台的应用域名进行设置。在修改过程中，首先应选择相对应的组织和空间。针对不同的应用，可相应的设置访问域名，同时进行添加域名、删除域名、修改域名等操作。

**模块2：配额管理**

功能说明：

该模块可以对Cloudfoundry平台不同的组织空间的配额进行设置。默认的配额为default，在空间管理中可选择。配额指的是该空间可用的内存、服务、应用域名等数量。在该模块，可进行配额的添加、删除、修改等操作。要根据不同的服务会有不同的配额设置方式。

**模块3：用户管理**

功能说明：

该模块可以对Cloudfoundry平台的用户进行管理。在该模块中，可对用户进行权限的分配，以及用户的查询操作。

**模块4：集群扩展**

CF集群支持横向扩展。

#### 服务管理

功能说明：

服务管理模块会提供给普通用户cfWeb系统中预置好的服务功能，用户在部署应用时，可以根据需要选择使用哪些服务，cfWeb系统中包括的服务如下：

a)：数据库服务

b)：缓存服务

c)：中间件服务

d)：文件服务

e)：web服务

f)：NoSql服务

注：此模块强调的是普通用户的功能，管理员有服务管理的权限，但一般情况下并不会使用

### 用户系统

#### 服务展示

功能说明：

展示当前CF平台可用的服务有哪些，注意，服务展示仅提供展示的功能，当前用户还没有登录，除了查看不可以进行其它操作。

#### 登录

功能说明：

1. ：系统登录需要与WSO2平台进行整合。当用户登录时，需将用户名和密码发送给WSO2系统，在WSO2的系统中进行验证，返回验证的结果和token。后续操作将通过用户、密码和token来进行验证。
2. ：cfWeb系统的用户注册需要将用户名和密码信息提交到WSO2系统中，调用WSO2的接口将用户信息添加。

#### 仪表盘

功能说明：

用户通过登录页面验证进入cfWeb用户系统的主页面，在主页面中对应CF集群的仪表盘，其中包括：所属space下的所有机器的状态、所有组件状态以及所有服务状态。同时，应显示所属space下的配额展示（资源使用情况，包括内存、硬盘、CPU、服务的使用量）。同时，还应该显示出所属space下的应用运行情况，运行中的应用、停止的应用等信息。

#### 应用管理

应用管理主要分为应用模板和应用列表两个模块。在cfWeb用户系统中，用户应注重的为应用的部署。

**模块1：应用管理**

功能说明：

通过此功能上传应用代码并开启应用；平台将自动根据用户申请的资源环境从资源池获取资源，通过容器的方式部署应用运行环境及服务，将用户上传的部署包部署到容器中并启动应用。

a)：新建应用：根据应用进行名称配置，系统会根据名称+域名自动生成应用访问的URL。

b)：应用详情：显示当前应用所占用的资源，包括内存、CPU等等

c)：删除应用

d)：启动应用

c)：重启应用

e)：终止应用

**模块2：应用模板**

功能说明：

应用模板模块主要作用是用于创建后续部署应用时需要的基础环境，应用编写语言、使用的数据存储服务等。管理员可适时的根据需求，创建不同的应用模板。

注意：此模块不是普通用户的重点，cfWeb会提供给用户创建应用模块的功能，但一般情况下应用模板都是由管理员创建，用户在部署自己的应用时只需要按需选择应用模板即可。

a)：新建模板：建立应用部署标准模板，模板建立好之后可供用户根据不同应用选择合适的模板进行应用部署。

b)：编辑模板

a)：删除模板

#### 服务管理

功能说明：

服务管理模块会提供给普通用户cfWeb系统中预置好的服务功能，用户在部署应用时，可以根据需要选择使用哪些服务，cfWeb系统中包括的服务如下：

a)：数据库服务

b)：缓存服务

c)：中间件服务

d)：文件服务

e)：web服务

f)：NoSql服务

同时，在服务管理模块提供服务扩容接口。当服务的使用量不够的情况下，用户可提供该接口进行扩容。

## ESB系统

ESB全称为Enterprise Service Bus，即企业服务总线。它是传统中间件技术与XML、Web服务等技术结合的产物。它将企业内部开发的服务进行集成，提供了细粒度、标准化的服务统一管理，并包括服务的授权认证管理系统，以及服务监控系统。

ESB系统分为服务注册中心和服务申请中心以及监控中心，这三个平台都会接入到统一身份认证系统进行统一认证管理。

* 服务注册中心

服务注册的平台，由服务提供者使用。服务提供者可以在此进行服务注册，服务发布，数据元、组件、消息管理，服务管理。

* 服务申请中心

服务申请的平台，由服务使用者和部门管理员使用。服务使用者在此查看、搜索服务，进行服务申请，查看我的申请，查看我的访问token；部门管理员在此进行审批服务，用户管理。

* 监控中心

整个ESB系统的可视化监控中心，此监控中心分为两种权限，一种是系统管理员，可以监控所有部门的服务情况；另一种是各部门的管理员和服务提供者，他们可以在此查看自己部门的服务监控。

功能模块图如下：

**ESB功能模块图**

按ESB系统的使用逻辑，将用户分为部门管理员，服务提供者，服务使用者，系统管理员这四种，这四种用户的角色关系如下：



**ESB四种用户角色关系**

所有角色的用户信息都在统一身份认证系统中进行统一管理。起初，由系统的运维人员给每个部门发放一个部门管理员账号。部门内的用户由部门管理员使用部门管理员帐号登录到统一身份认证的管理平台进行创建和管理。所有被创建的普通用户都具有服务使用者的身份，而服务提供者的身份则需要部门管理员在统一身份认证的管理平台给用户分配。部门管理员也可以给普通用户分配部门管理员的身份，从而让普通用户也拥有部门管理员的权限。

以下分别以这四种用户阐述需求。

### 部门管理员

#### 审批权限

部门管理员登录到服务申请中心，就可以对服务使用者发起的服务申请进行审批，审批后申请者即可拥有访问相应服务的权限。

#### 用户管理

部门管理员可以使用其帐号登录到统一身份认证系统的管理平台，对本部门内的用户进行管理操作。这里包括添加、删除、修改用户信息，给用户分配服务提供者和部门管理员的权限。

#### 监控管理

部门管理员登录到监控中心，可以查看本部门的所有服务信息，可以看到的信息有服务访问量，访问延时，成功率，调用记录等各种统计信息。

### 服务提供者

#### 数据元、组件、消息管理

服务提供者在登录到服务注册中心后，可以进行部门内部的数据元、组件、消息的管理，包括添加、删除、修改、查看。

数据元（Data element）：在又称数据类型，通过定义、标识、表示以及允许值等一系列属性描述的数据单元。标准化定义中是最小的标准化数据单元。数据元包含对象类、特性和表示三个方面，在构建标准化规范中，开发人员对数据元的域、取值范围、定义、适用环境等进行标准化定义，避免了冗余、杂乱、无意义的数据元存在，最终输出标准化消息，实现标准化策略。

数据元、组件、消息属于逐层嵌套结构，组件由基础的数据元组合形成，组件组合嵌套，形成一条消息。服务治理平台支持细粒度的数据元管理，满足服务治理平台对更精细级别管理的需求。如下所示为三个管理列表的内容：

* 数据元管理列表显示使用该数据元的组件和消息列表。
* 组件管理列表中包含数据元列表、子组件列表、组件的消息列表等。
* 消息管理列表显示用户消息和组件消息，消息中包含消息详情和JSON格式预览。

#### 服务注册

服务提供者登录到服务注册中心进行服务注册，并及时的更新服务，建立服务目录（分组）并进行管理。在这里注册的服务是以部门为单位的，部门内的每个服务提供者都可以管理该部门注册的所有服务，而不能管理其他部门的服务。

这里服务提供者可以新建Restful服务，新建SOAP服务，新建服务分组，对已经建立的服务信息进行更新修改。其中新建SOAP服务也可以直接导入本地或远程的WSDL文件。

在新建Restful服务时，需要填写的信息有：服务名称（必填）、服务分组（必填）、标签（选填）、服务编码（必填，同步服务后无法修改）、传输协议（即HTTP方法，必填）、消息交换方式（必填，默认为请求响应）、服务地址（必填）、描述（选填）、服务提供商（必填）、服务版本号（选填）、服务的原始URL地址（必填，可有多个）。另外还需填入参数，可选的参数包括Head参数、Path参数、URL参数、Body参数、返回参数。

在新建SOAP服务时，需要填写的信息有：服务名称（必填）、服务分组（必填）、标签（选填）、服务编码（必填，同步服务后无法修改）、消息交换方式（必填，默认为请求响应）、服务地址（必填）、描述（选填）、服务提供商（必填）、服务版本号（选填）、服务的原始URL地址（必填，可有多个）。另外还需填入一个以上的方法。每个方法包括方法名称（必填）、输入参数类型（必填）、输出参数类型（必填），方法描述（选填），其中输入和输出参数的类型都要从消息列表中选择。

新建服务的流程图如下：



**新建服务流程图**

#### 服务发布

服务提供者在注册或修改了服务之后，需要将服务发布出去服务才能生效。在进行服务发布的时候，会有一个所有部门的列表，需要先勾选该服务默认可以由哪些部门进行访问。同时列表中有一个“所有”选项，如果选择了所有，则所有人都可以访问此服务（即“匿名服务”）。如果之后要对该默认发布权限进行修改，则需要重新发布。

#### 监控管理

部门的服务提供者登录到监控中心，可以查看本部门的所有服务信息，可以看到的信息有服务访问量，访问延时，成功率，调用记录等各种统计信息。

### 服务使用者

#### 查看服务列表

普通用户登录到服务申请中心，可以查看所有部门提供的服务的列表（未登录时也可以查看）。如果用户登录了，则可以在列表中看到该用户是否有权限访问每个服务。如果没有权限，则用户需申请。

#### 服务分类搜索

服务申请平台为用户提供友好的服务分类搜索功能。在查看服务列表时，会以服务分组、服务标签、是否有权限访问（如果用户登录）、服务提供方（部门）进行过滤搜索，并可以根据服务名称和描述字段进行模糊搜索，服务使用者可根据搜索结果选择查看的服务。

#### 查看服务详细信息

可以查看服务的详细参数，包括：服务名称，访问量，服务分组，标签，最近更新时间，服务编码，传输协议（HTTP的GET或POST等）,消息交换方式（单向或请求响应），用户是否有权限访问（如果登录），服务地址（访问的路径），服务描述，参数列表等。

#### 申请服务

服务使用者第一次登录到服务申请中心时就会为用户创建一套token，该token与用户一一对应，访问服务时用户需要用此token来标记自己的身份。

服务使用者如果默认不具有某个服务的访问权限，可以选择服务发起申请。发起的申请将由该服务提供方的部门管理员进行审批，审批通过后用户使用自己的token就可以访问服务了。

申请服务的流程图如下：



**申请服务流程图**

#### 下载开发包

在服务治理平台，用户可选择生成Java API的开发包并下载，为开发者提供功能丰富、使用高效、文档规范、管理可控的接口。

### 系统管理员

#### 监控管理

系统管理员登录到监控中心，可以查看所有部门的所有服务信息，可以看到的信息有服务访问量，访问延时，成功率，调用记录等各种统计信息。