OpenStack认证管理

前言

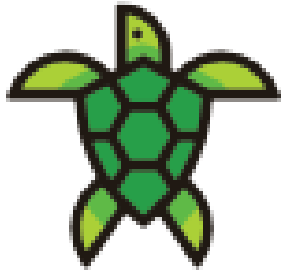
- Keystone为OpenStack提供共用的认证与鉴权机制，在整个OpenStack中占有举足轻重的地位。
- 本章节分为两个部分：理论和实验
  - 理论部分主要讲解Keystone作用，架构，原理和流程。
  - 实验部分重点锻炼学员Keystone日常运维操作，帮助学员理论联系实际，真正掌握Keystone。

目标

- 学完本课程后，您将能够：
  - 描述Keystone作用
  - 描述Keystone架构
  - 描述Keystone工作原理和流程
  - 具备Keystone日常运维能力

目录

OpenStack认证服务是什么?



KEYSTONE

认证服务
首次出现在OpenStack的“Essex”版本中。

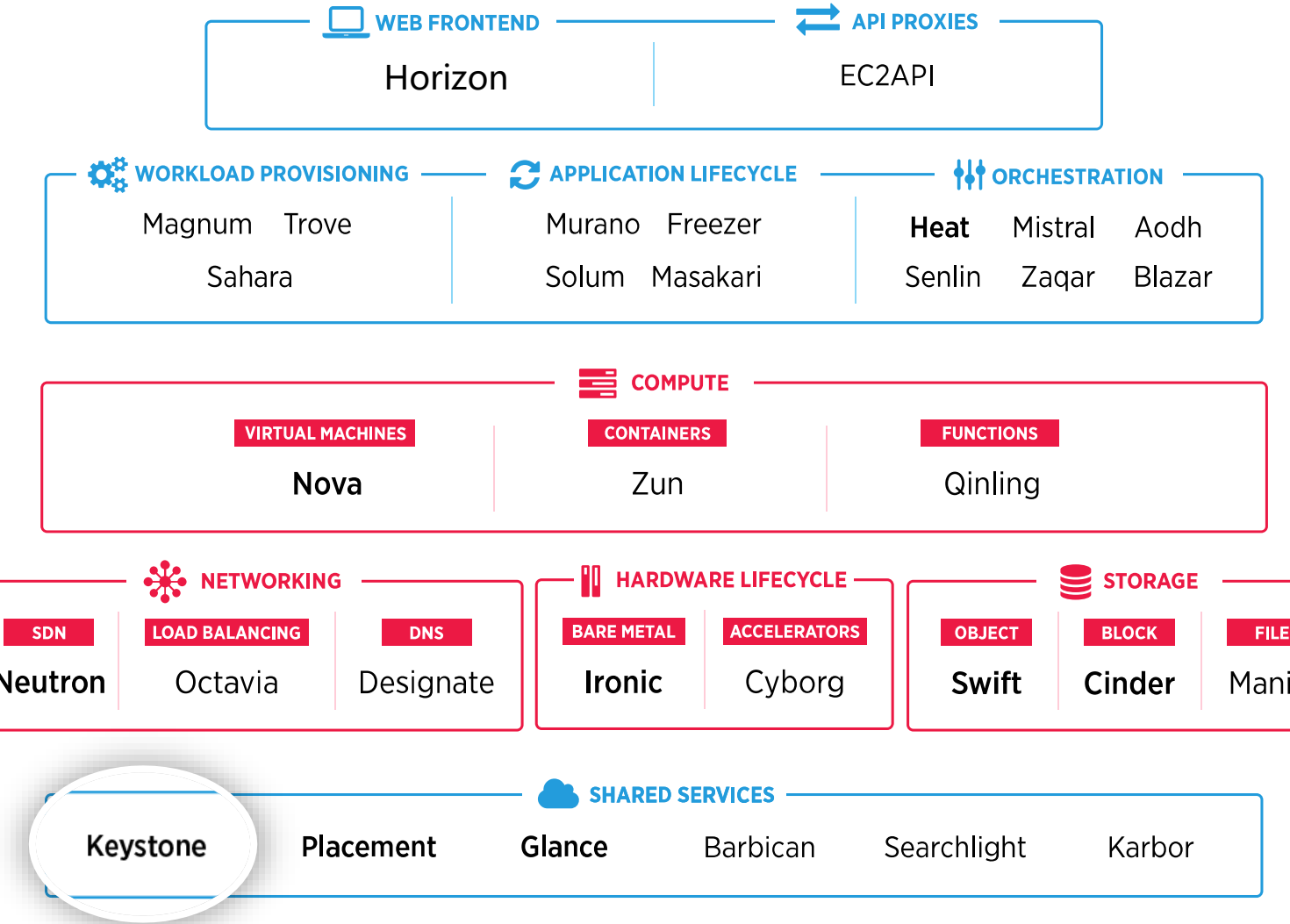
简介

Keystone提供身份验证，服务发现和分布式多租户授权。
Keystone支持LDAP，OAuth，OpenID Connect，SAML和SQL。

依赖的OpenStack服务

Keystone为其他项目提供认证。
外部请求调用OpenStack内部的服务时，需要先从Keystone获取到相应的Token。
类似的，OpenStack内部不同项目间的调用也需要先从Keystone获取到认证后才能进行。

Keystone在OpenStack中的位置

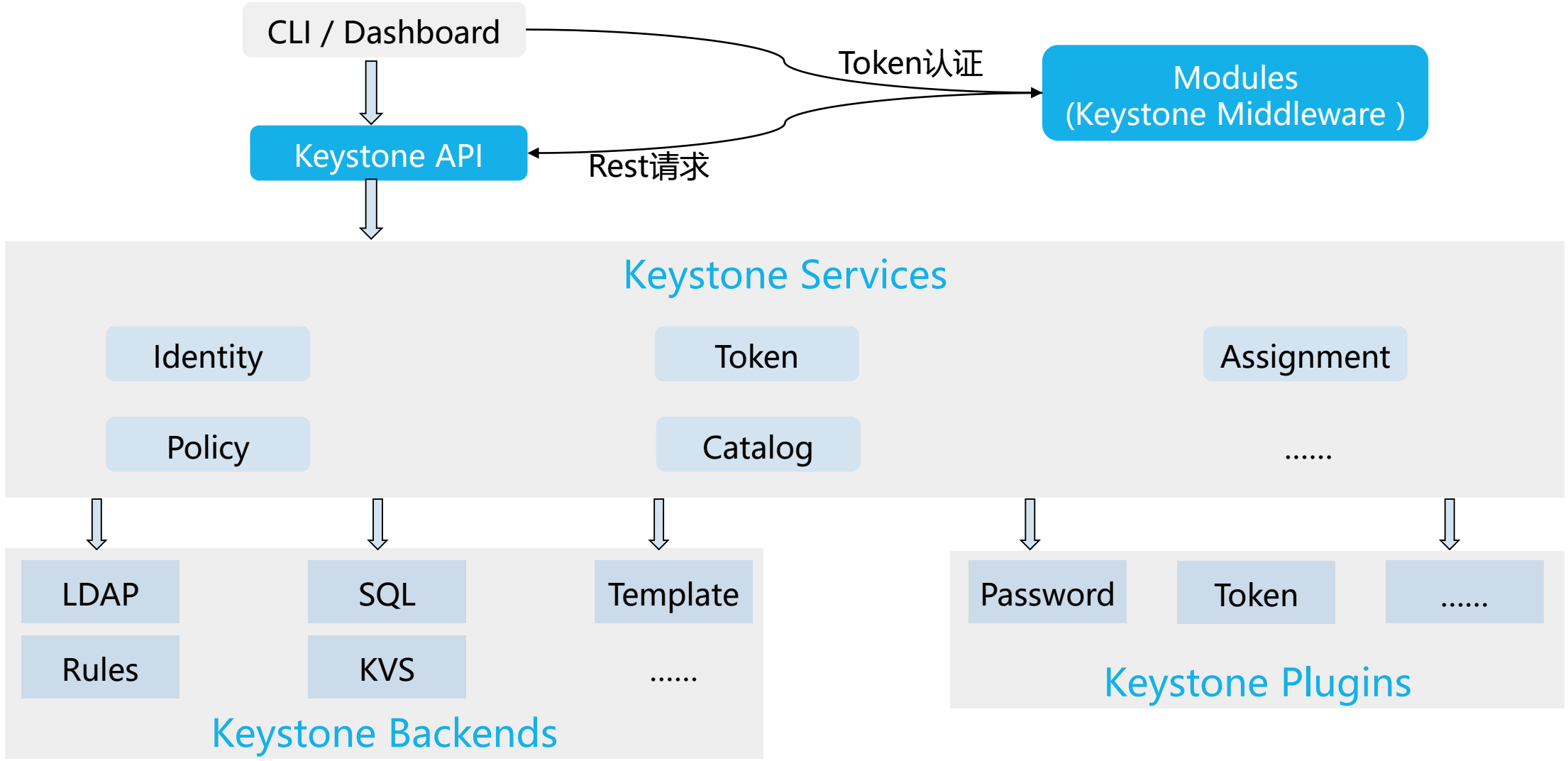


source: openstack.org

Keystone在OpenStack中的作用

目录

Keystone架构图

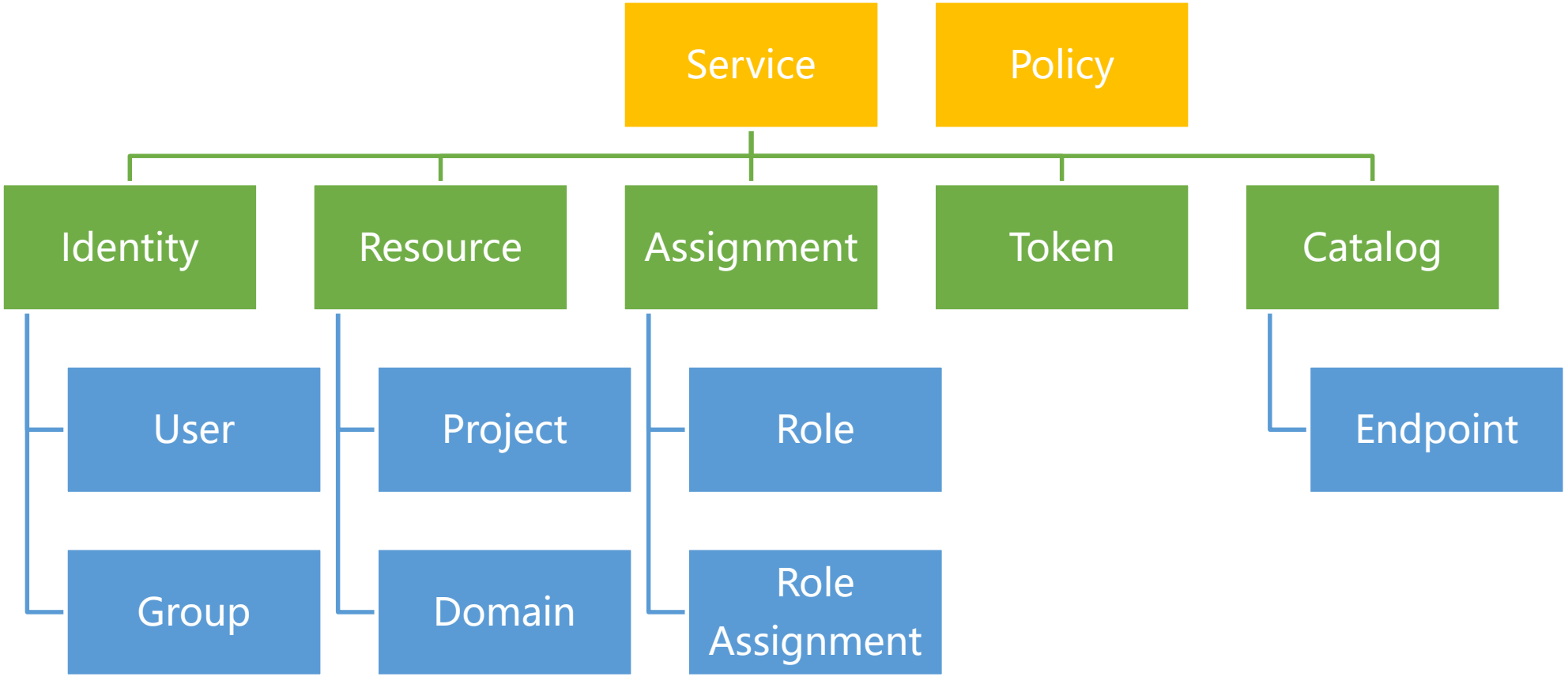


Keystone各组件作用

Keystone API	• 接收外部请求
Keystone Middleware	• 缓存Token等，减轻Keystone Services压力
Keystone Services	• 不同的Service提供不同的认证或鉴权服务
Keystone Backends	• 实现Keystone服务，不同的Service由不同的Backend提供
Keystone Plugins	• 提供密码、Token等认证方式

目录

Keystone对象模型



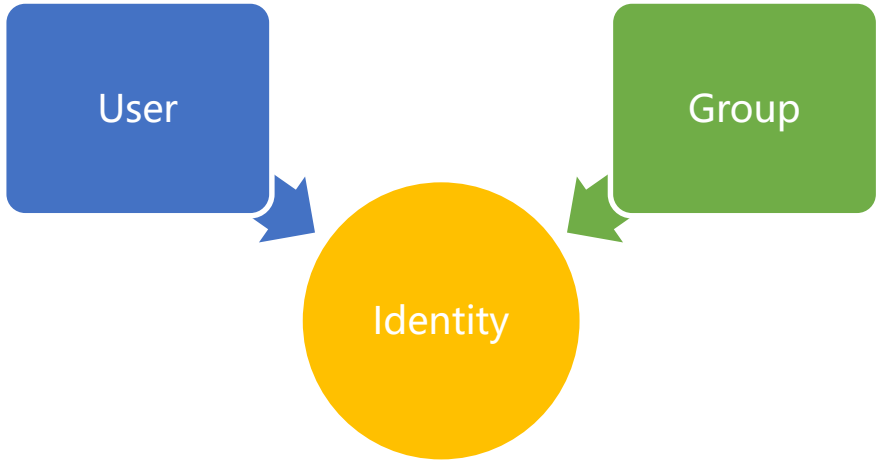
Keystone对象模型 - Service

- Keystone是在一个或多个端点（Endpoint）上公开的一组内部服务（Service）。
- Keystone内部服务包括Identity、Resource、Assignment、Token、Catalog等。
- Keystone许多内部服务以组合方式使用。
  - 例如，身份验证时将使用认证服务（ Identity ）验证用户或项目凭据，并在成功时创建并返回带有令牌服务（Token）的令牌。

- 除内部服务外，Keystone还负责与OpenStack其他服务（ Service ）进行交互，例如计算，存储或镜像，提供一个或多个端点，用户可以通过这些端点访问资源并执行操作。

Keystone对象模型 - Identity

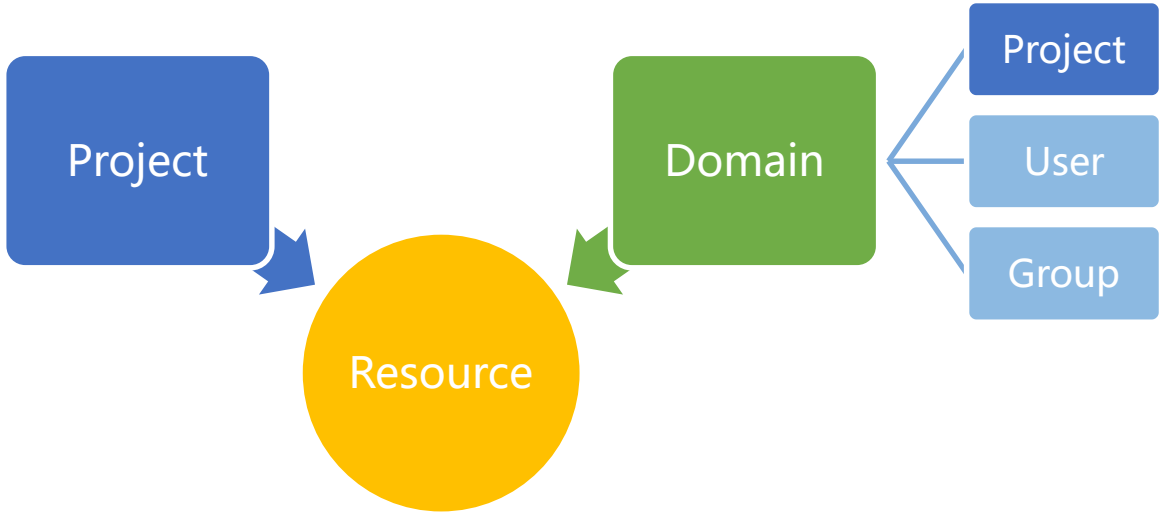
- Identity服务提供身份凭据验证以及用户（User）和用户组（Group）的数据。



- User是单个OpenStack服务使用者，用户本身必须属于某个特定域。所有用户名不是OpenStack全局唯一的，仅在其所属域唯一。
- Groups把多个用户作为一个整体进行管理。组本身必须属于某个特定域。所有组名不是OpenStack全局唯一的，仅在其所属域唯一。

Keystone对象模型 - Resource

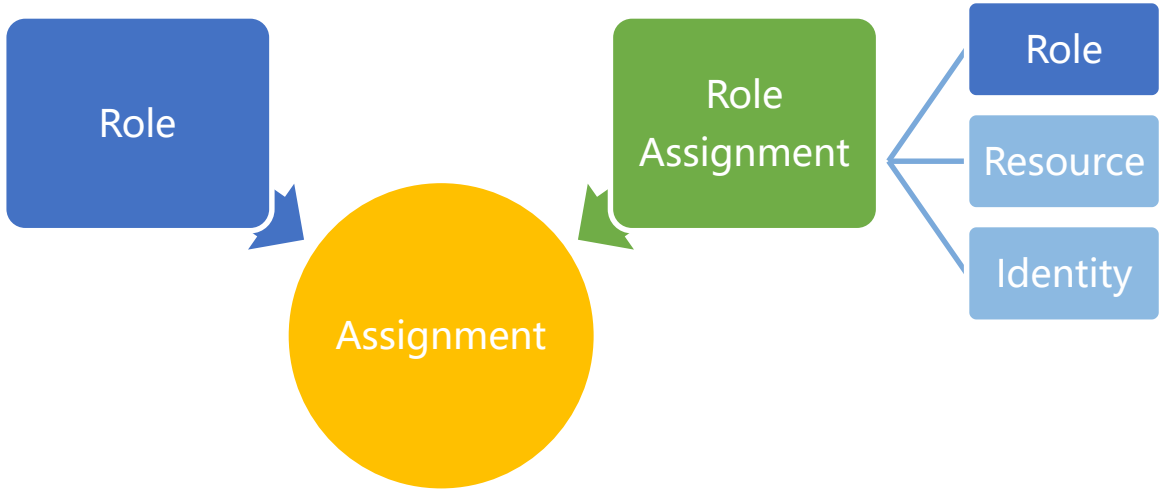
- Resource服务提供有关项目（Project）和域（Domain）的数据。



- Project是OpenStack资源拥有者的基本单元，OpenStack中所有资源都属于特定项目。
- Domain把项目、用户和组作为一个整体管理，每种资源都属于某个特定域。Keystone默认域名为”Default”。

Keystone对象模型 - Assignment

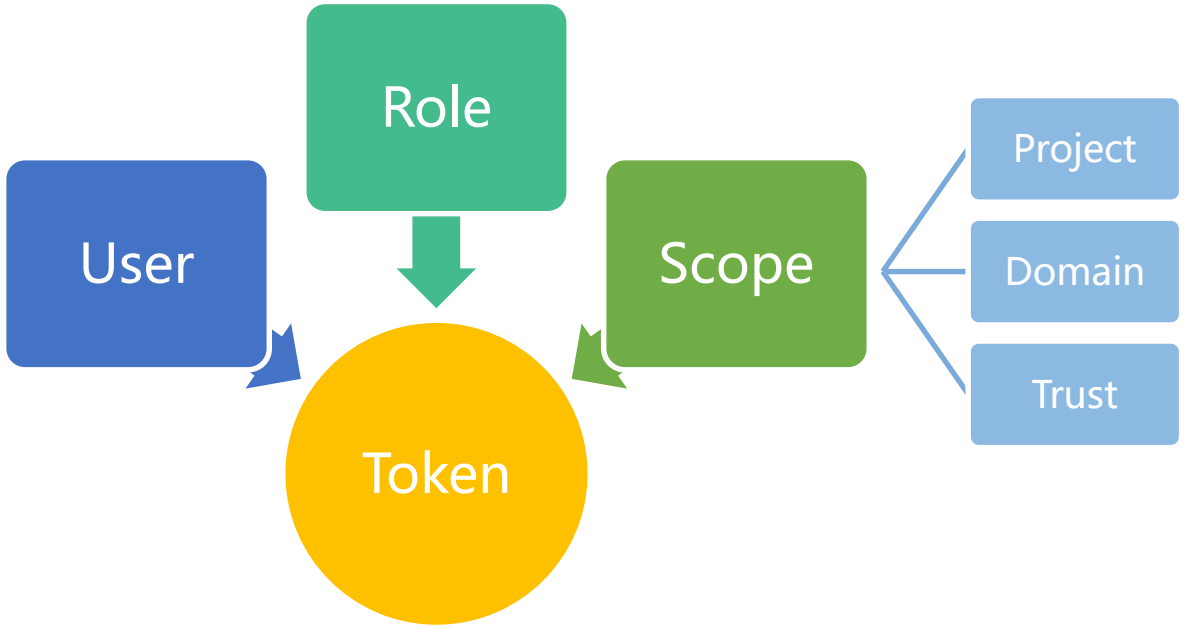
- Assignment服务提供有关角色（Role）和角色分配（ Role Assignment）的数据。



- Role规定最终用户可以获得的授权级别。角色可以在域或项目级别授予。可以在单个用户或组级别分配角色。角色名称在拥有该角色的域中是唯一的。
- Role Assignment是一个3元组，有一个Role，一个Resource和一个Identity。

Keystone对象模型 - Token

- Token服务提供用户访问服务的凭证，代表着用户的账户信息。
- Token一般包含User信息、Scope信息（Project、Domain或者Trust）、Role信息。



Keystone对象模型 - Catalog

- Catalog服务提供用于查询端点（Endpoint）的端点注册表，以便外部访问OpenStack服务。

- Endpoint本质上是一个URL，提供服务的入口，有如下几种：
  - Public：最终用户或其他服务用户使用，通常在公共网络接口上使用。
  - Internal：供最终用户使用，通常在未计量的内部网络接口上。
  - Admin：供管理服务的用户使用，通常是在安全的网络接口上。

Keystone对象模型 - Policy

- 每个OpenStack服务都在相关的策略文件中定义其资源的访问策略（Policy）。
- 访问策略类似于Linux中的权限管理，不同角色的用户或用户组将会拥有不同的操作权限。

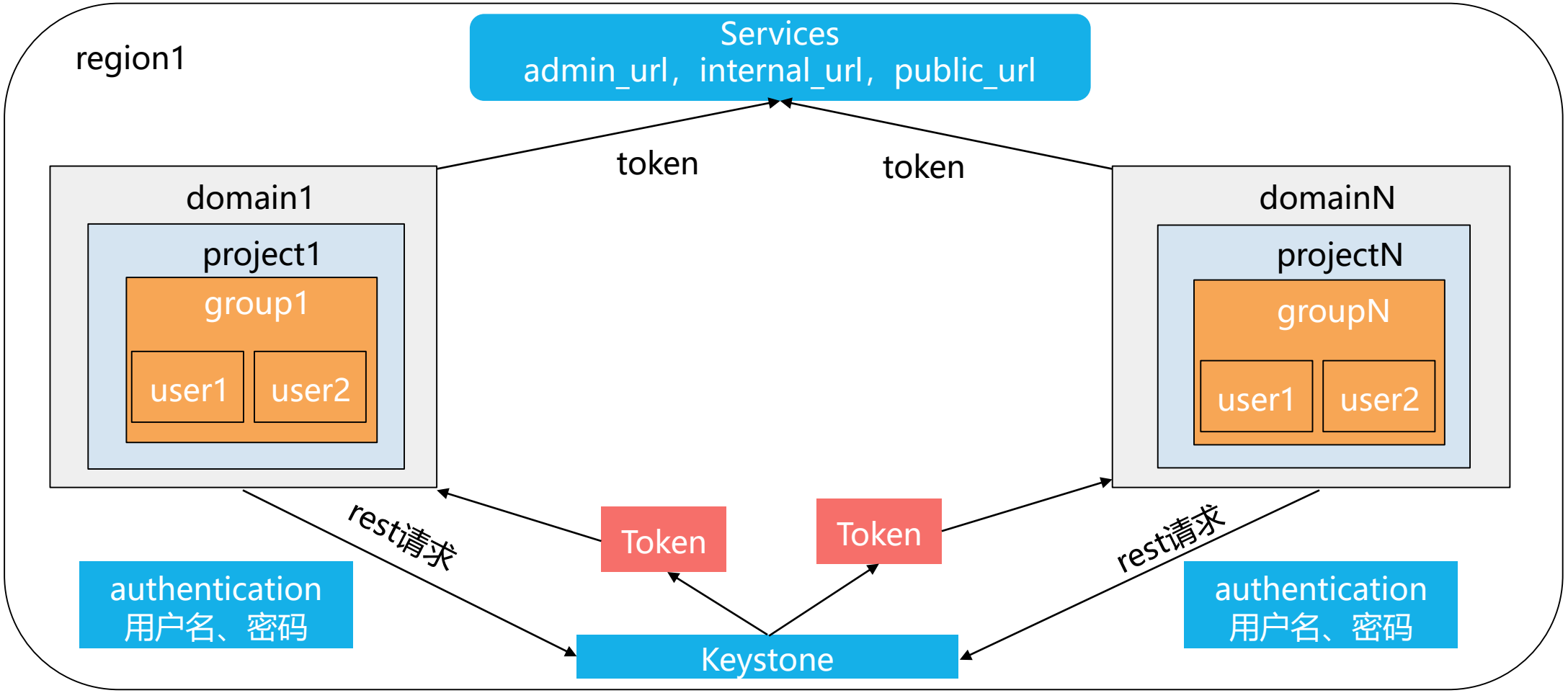
```
{
    "admin_required": "role:admin",
    "cloud_admin": "rule:admin_required and domain_id:admin_domain_id",


    "default": "rule:admin_required",

    "identity:get_service": "rule:admin_or_cloud_admin",
    "identity:list_services": "rule:admin_or_cloud_admin",
    "identity:create_service": "rule:cloud_admin"
}
```

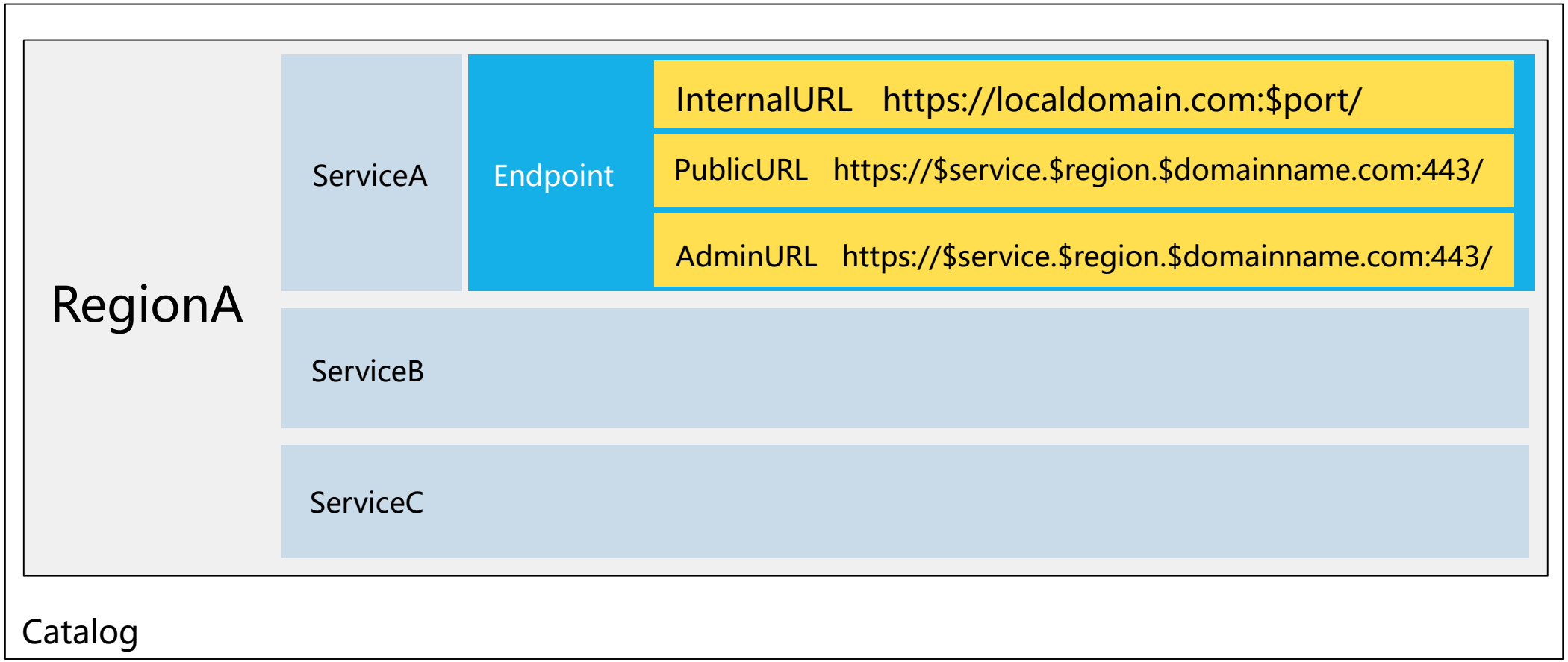
- 访问策略规则以JSON格式指定，文件名为policy.json。
  - 策略文件的路径是/etc/SERVICE_NAME/policy.json，例如/etc/keystone/policy.json。

Keystone对象模型分配关系示例



Keystone对象模型分配关系示例

- Region, Service, Endpoint:



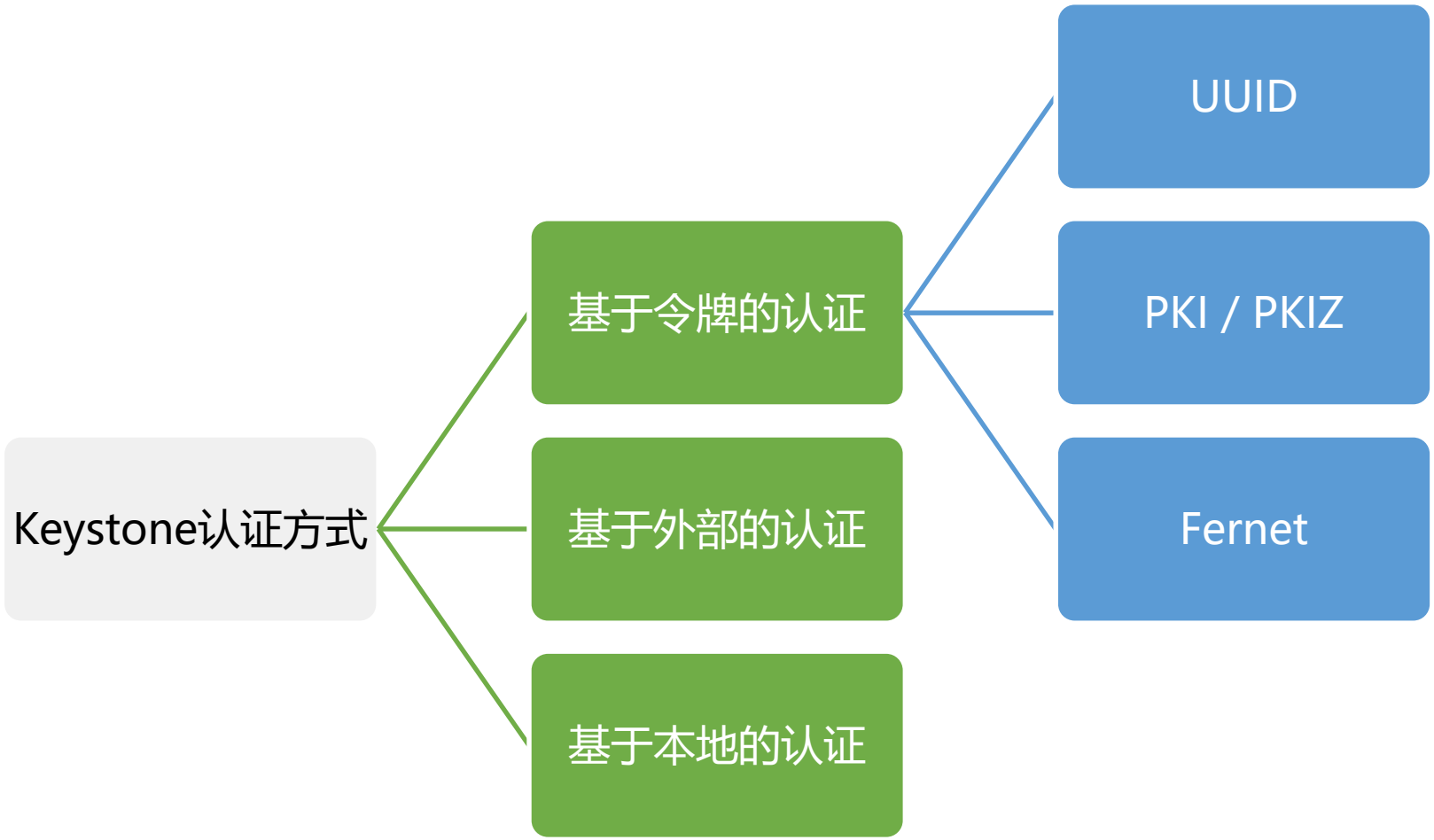
Keystone对象模型使用示例

- User：
  - 获取Token
  - 获取Service Catalog
- Admin User：
  - 管理Users，Projects，Roles
  - 管理特定Project中Users的Roles
  - 管理Services，Services的Endpoints
- Service:
  - 验证Token
  - 定位其他Service的位置
  - 调用其他Service

目录

Keystone认证方式概览

- Keystone最重要的工作是认证，Keystone支持多种认证方式。

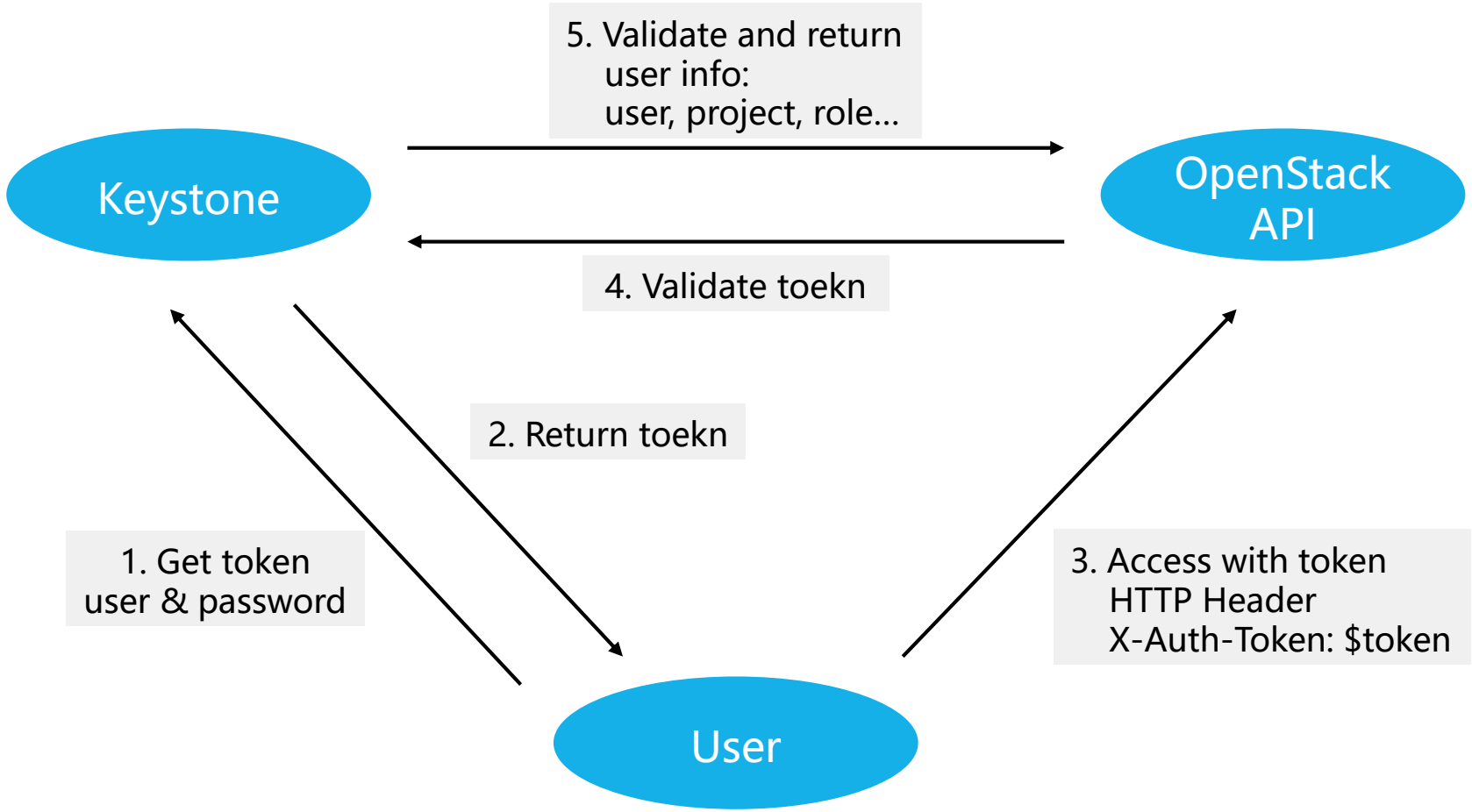


Keystone三种认证方式对比

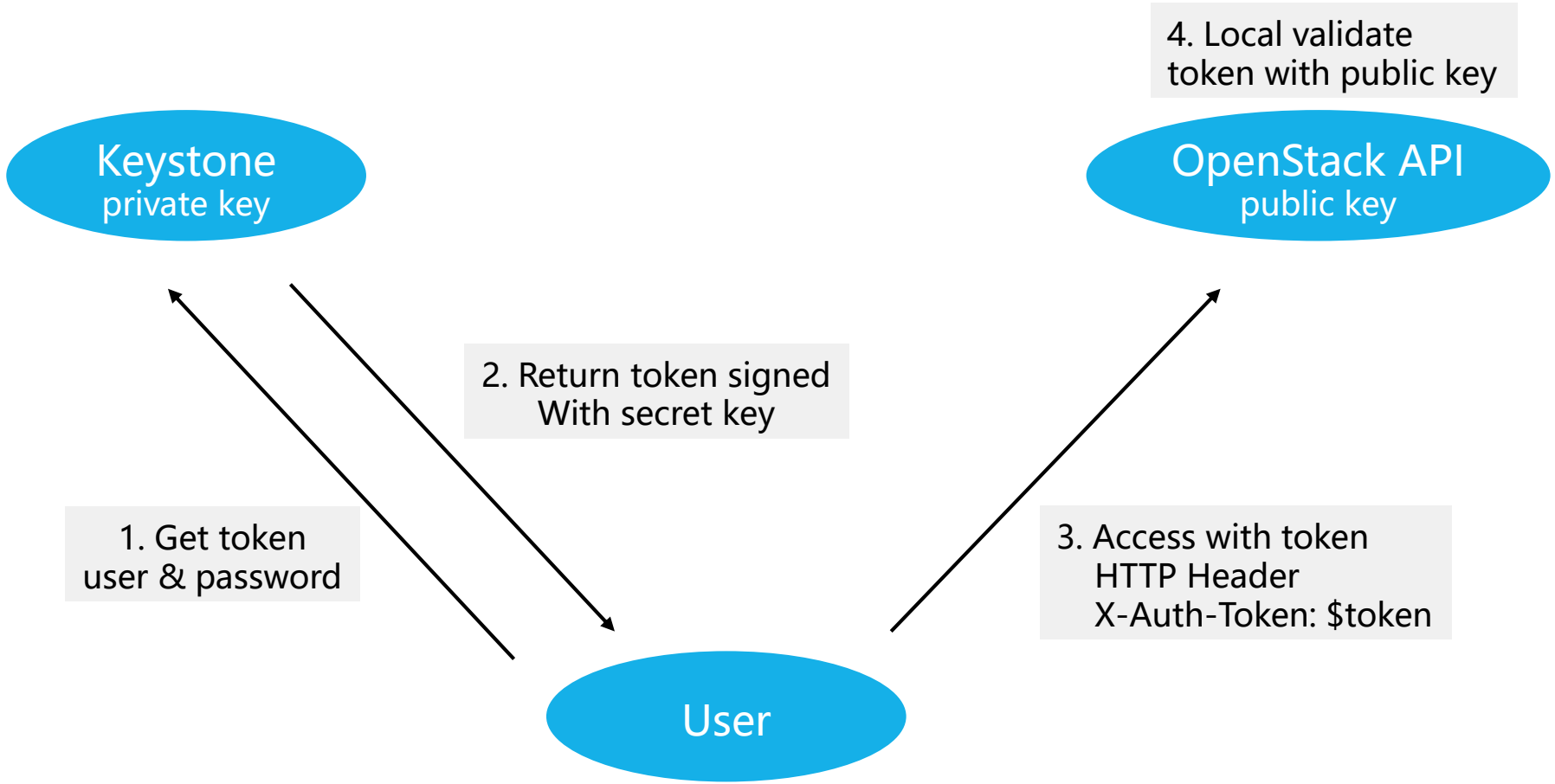
基于令牌的认证方式	基于外部的认证方式	基于本地的认证方式
• 最常用的Keystone认证方式，使用方式简单。 • 认证请求发送时添加一个“X-Auth-Token”的HTTP头，Keystone检查该HTTP头中的Token值，并与数据库中的令牌值进行比对验证。	• 集成使用第三方认证系统，在认证请求中添加“REMOTE_USER”信息。	• 默认认证方式，即用户名和密码认证。

生产环境中常用的是基于令牌的认证方式，需要重点学习。

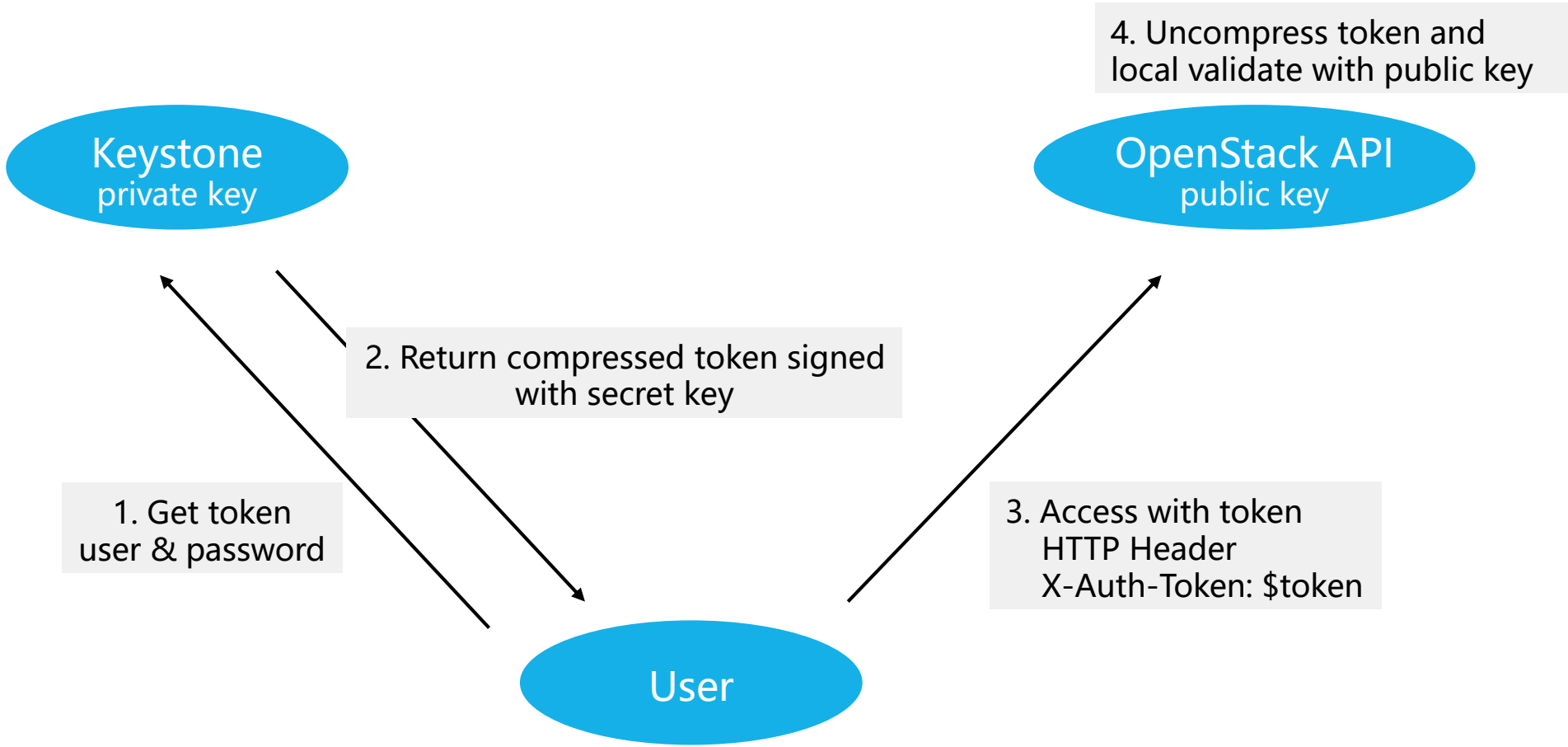
Keystone基于令牌的认证 - UUID



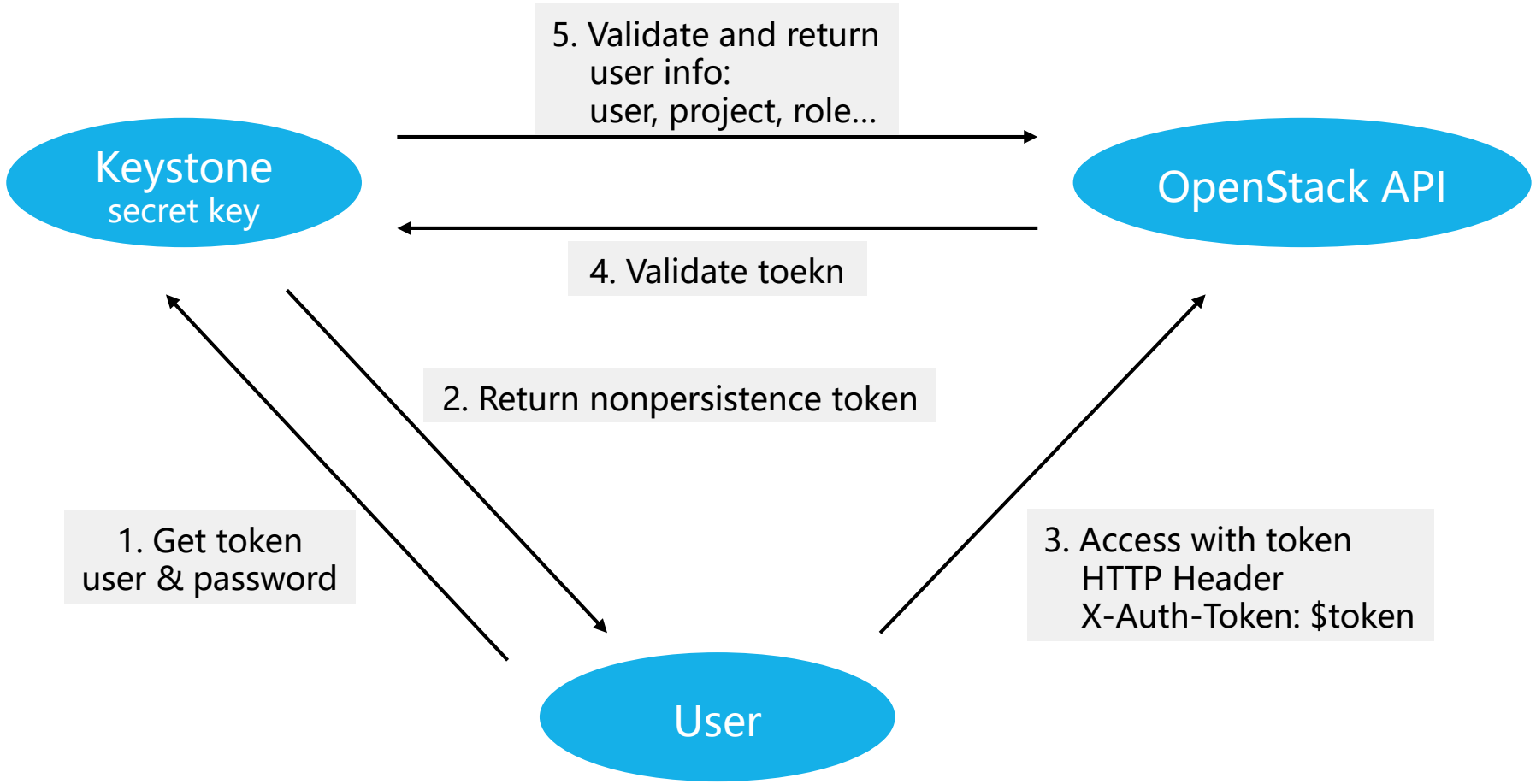
Keystone基于令牌的认证 - PKI



Keystone基于令牌的认证 - PKIZ



Keystone基于令牌的认证 - Fernet

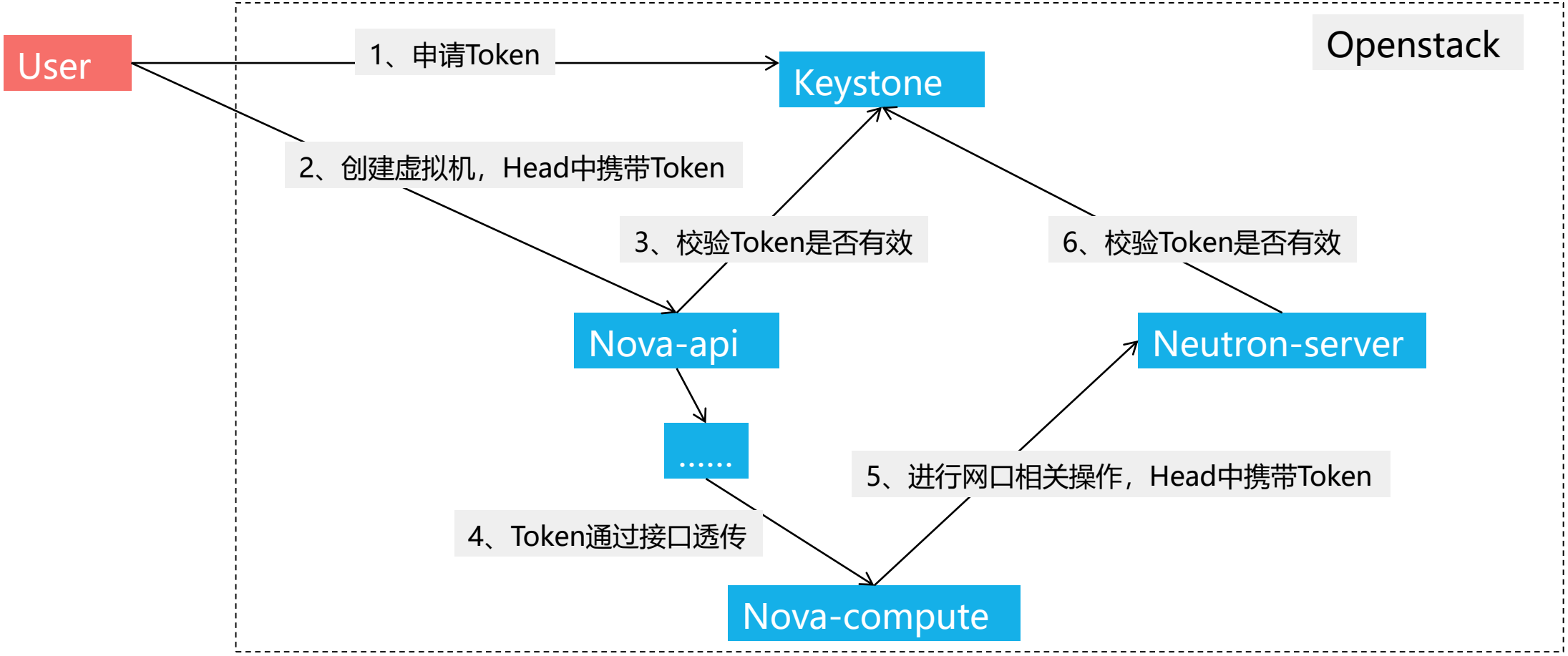


如何选择Keystone基于令牌的认证方式?

Token 类型	UUID	PKI	PKIZ	Fernet
大小	32 Byte	KB 级别	KB 级别	约 255 Byte
支持本地认证	不支持	支持	支持	不支持
Keystone 负载	大	小	小	大
存储于数据库	是	是	是	否
携带信息	无	user, catalog 等	user, catalog 等	user 等
涉及加密方式	无	非对称加密	非对称加密	对称加密(AES)
是否压缩	否	否	是	否

目前OpenStack新发布版本默认采用Fernet令牌。

OpenStack认证流程 - 以创建VM为例



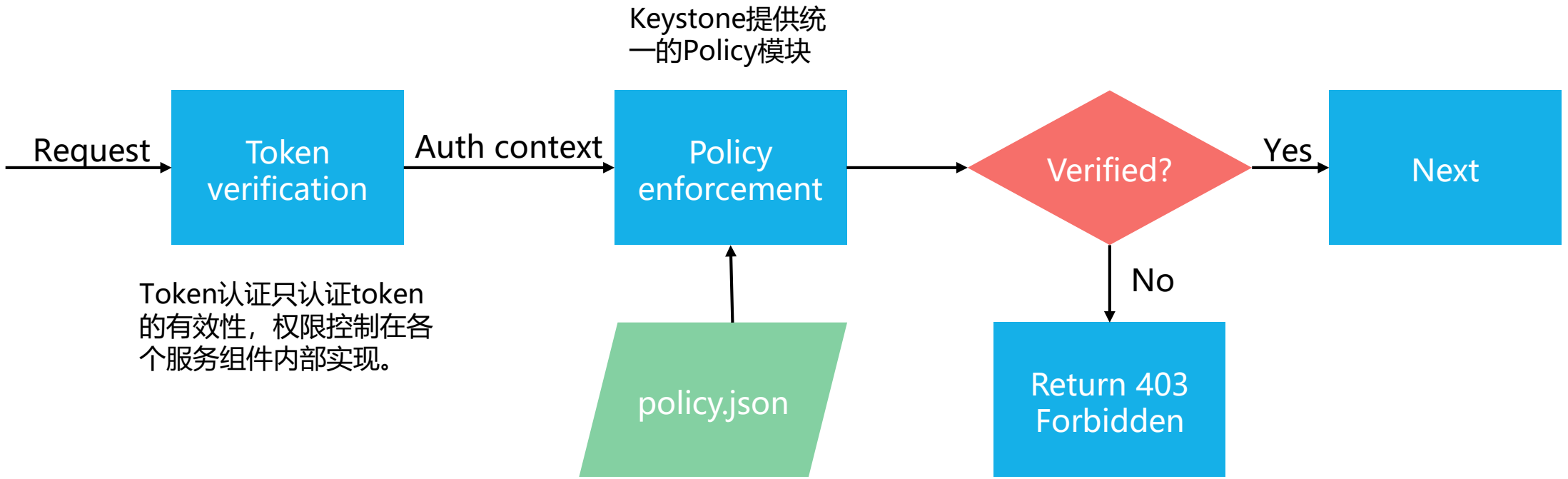
Keystone只检验Token是否有效，那每个服务的操作权限控制是怎么实现的?

讨论： RBAC，基于角色的访问控制如何实现?

- 创建VM时，不同OpenStack服务需要交互，Keystone会发放和校验Token有效性，但每个服务如何检验用户的操作权限呢?
  - 例如用户是否有创建VM权限，是否有更改VM规格权限?

- 请大家花5分钟，思考或讨论OpenStack中基于角色的访问控制如何实现?
  - 日常生活中有哪些基于角色的访问控制?
  - 本章节哪个地方有提到Keystone访问控制相关知识?

RBAC：基于角色的访问控制 - 流程



RBAC：基于角色的访问控制 - 原理

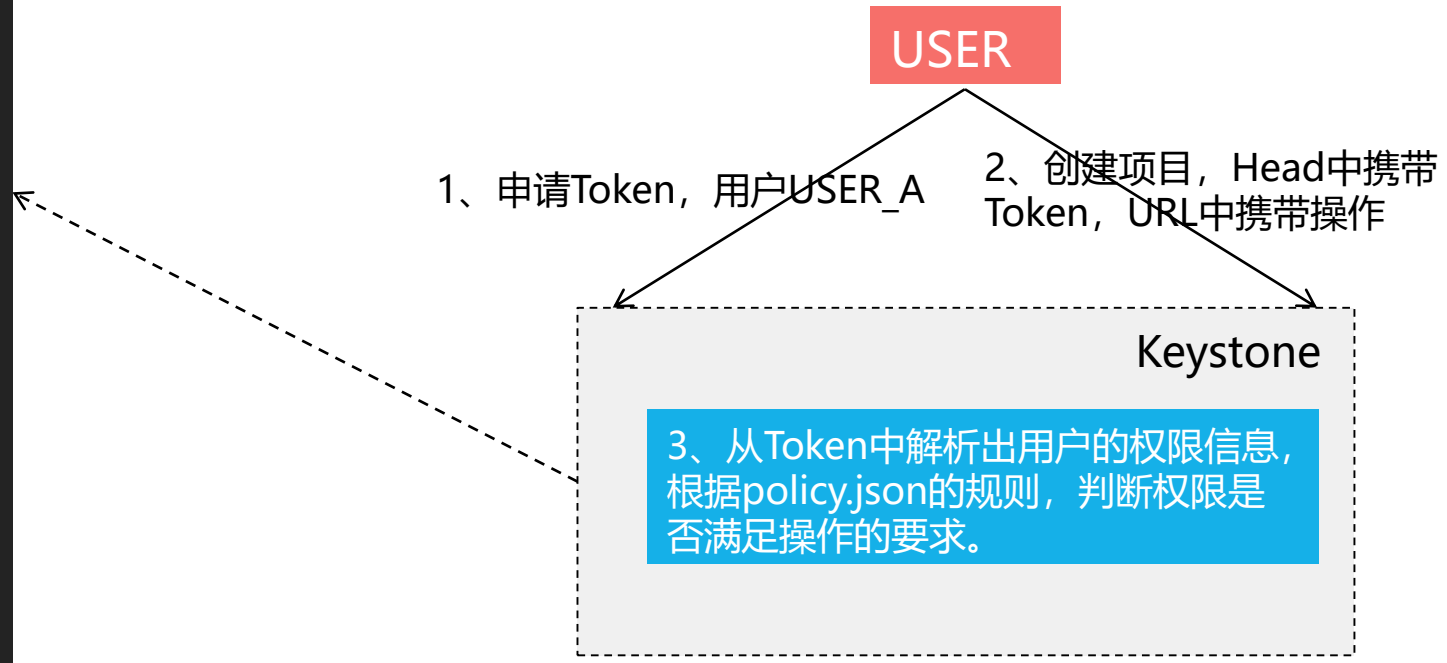
```
{
 ...
    "all_admins": [
        [
            "role:admin"
        ],
        [
            "role:internal_admin"
        ]
    ],
 ...
    "identity:list_projects": [
        [
            "rule:all_admins"
        ]
    ],
    "identity:create_projects": [
        [
            "role:admin"
        ]
    ]
 ...
}
```

Policy模块在检测时需要三方面的数据：

1、policy.json策略配置文件；

2、auth_token添加到http头部的token数据；

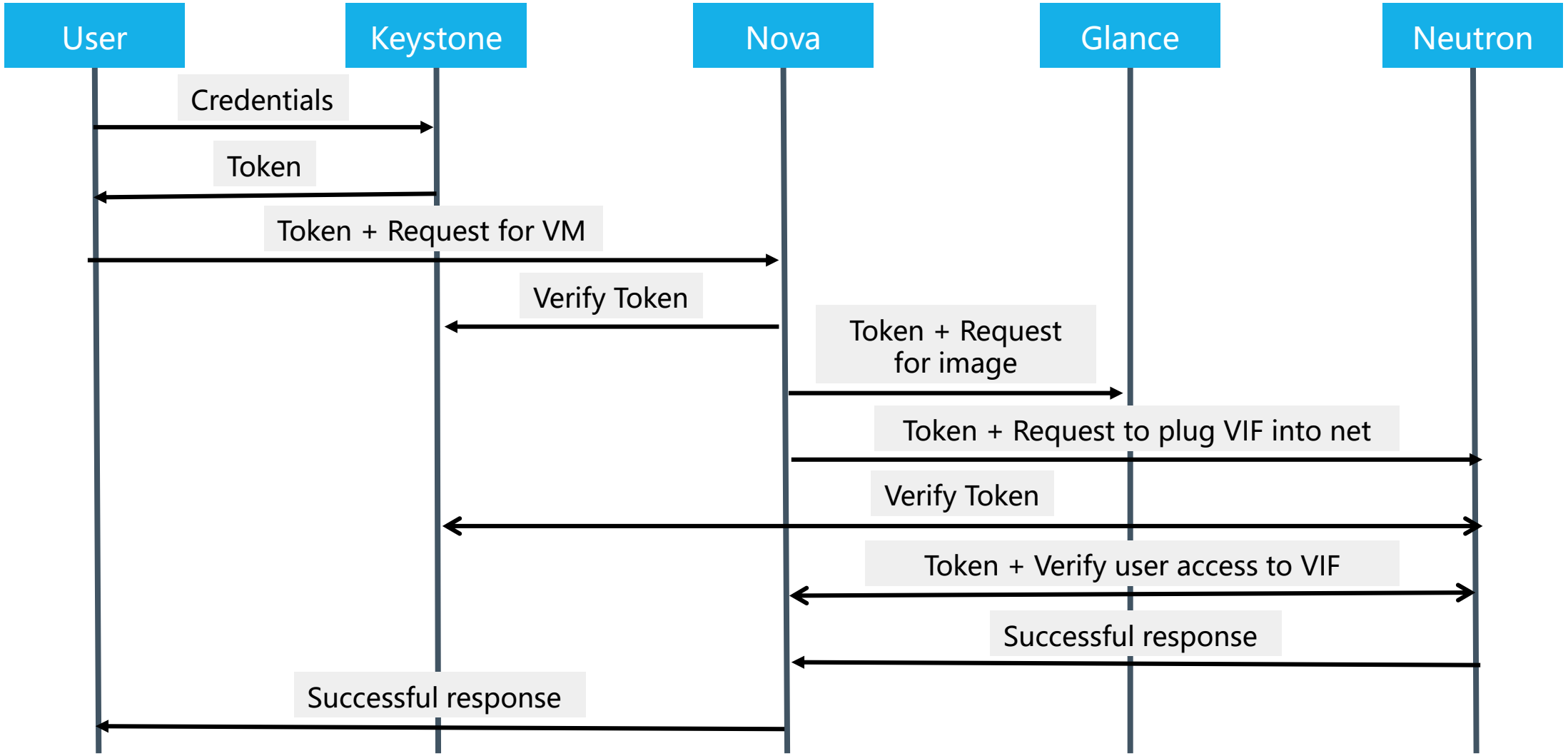
3、用户的请求数据。



提问： Keystone如何实现认证和权限控制?

- 1、用户在OpenStack操作界面上创建一个VM，Keystone如何认证该用户，如何验证该用户具有创建VM的权限?
- 2、用户在OpenStack CLI上创建一个VM，Keystone如何认证该用户，如何验证该用户具有创建VM的权限?
- 3、两种方式对Keystone有区别吗?

总结： Keystone如何实现认证和权限控制?



目录

OpenStack动手实验：Keystone操作

- 命令help
- 角色管理
- 用户与用户组管理
- 域如何与项目，角色，用户和组一起使用
- 项目管理
- 配额管理
- 服务管理

思考题

1. Keystone对象模型有哪些?
2. 请举例说明Keystone认证流程。

本章总结

- OpenStack认证服务Keystone简介
- Keystone架构
- Keystone对象模型
- Keystone认证工作原理和流程

学习推荐

- OpenStack社区
  - https://www.openstack.org/


谢 谢
www.huawei.com