ICS 35.040
L80

中华人民共和国国家标准

GB/T XXXXX—XXXX

信息安全技术 鉴别与授权 访问控制中间件框架与接口

Information security technology —Authentication and authorization —

Access control middleware framework and interface

点击此处添加与国际标准一致性程度的标识

（征求意见稿）

（本稿完成日期：2017 年 6 月）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

XXXX－XX－XX 发布　　XXXX－XX－XX

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目 次

前　言

本标准依据GB/T 1.1-2009《标准化工作导则 第1部分：标准的结构和编写》给出的规则起草。

本标准由全国信息安全标准化技术委员会（TC260）提出并归口。

本标准起草单位：中国科学院软件研究所、中国科学院数据与通信保护研究教育中心、北京数字证书认证中心有限公司、中国电子技术标准化研究所、国家射频识别产品质量监督检验中心。

本标准主要起草人：张严、张立武、高志刚、王鹏翩、冯登国、荆继武、吴槟、李强、陈星、高能、阎实、罗艳。

信息安全技术 鉴别与授权
访问控制中间件框架与接口

1 范围

本标准规定了访问控制中间件的框架结构与内部组件关系，定义了该框架内各组成部分的功能、操作流程及接口定义。

本标准适用于访问控制中间件的设计与实现，并可指导对该类中间件系统的检测及相关应用的开发，对该类中间件产品的采购亦可参照使用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 9387.2- 1995 信息技术 开放系统互连 基本参考模型 第2部分 安全体系结构

GB/T 18793-2002 信息技术 可扩展置标语言（XML）1.0

GB/T 18794.3-2003 信息技术 开放系统互连 开放系统安全框架 第3部分 访问控制框架

GB/T 25069-2010 信息安全技术 术语

GB/T 29242-2012 信息安全技术 鉴别与授权 安全断言置标语言规范

GB/T 30281-2013 信息安全技术 鉴别与授权 可扩展访问控制标记语言规范

GB/T 31501-2015 信息安全技术 鉴别与授权 授权应用程序判定接口规范

3 术语与定义

GB/T 25069-2010、GB/T 18794.3-2003 界定的以及下列术语和定义适用于本文件。

3.1

属性 attribute

主体、资源、动作和环境的某个特征，该特征可以在策略中被引用。

3.2

决策 decision

依据访问控制策略做出的评估结果。

3.3

环境 environment

一组与决策相关的属性集合，独立于特定的主体，资源或者动作。

3.4

资源 resource

数据，服务或者系统组件。

3.5

主体 subject

访问控制行为的发起者。

3.6

用户 user

使用系统和系统资源的自然人。

4 缩略语

下列缩略语适用于本文件：

IF-AQ：属性查询接口（Interface-Attribute Query）

IF-CDAQ：跨域属性查询接口（Interface-Cross Domain Attribute Query）

IF-DM：决策管理接口（Interface-Decision Management）

IF-PD：策略决策接口（Interface-Policy Decision）

IF-PQ：策略查询接口（Interface-Policy Query）

LDAP：轻量级目录访问协议（Lightweight Directory Access Protocol）

RPC：远程过程调用（Remote Procedure Call）

SAML：安全断言标记语言（Security Assertion Markup Language）

SOAP：简单对象访问协议（Simple Object Access Protocol）

SSL：安全套接层（Secure Sockets Layer）

TLS：传输层安全（Transport Layer Security）

XACML：可扩展访问控制标记语言（eXtensible Access Control Markup Language）

XML：可扩展标记语言（eXtensible Markup Language）

5 访问控制中间件体系框架

5.1 概述

访问控制过程包含三个参与方：发起者、访问控制中间件和访问目标。发起者是试图访问目标的实体；访问控制中间件是通过对适用的访问控制信息进行评估以决定发起者是否可以对目标进行特定类型访问的一系列组件及组件间接口的集合；访问目标是被试图访问的实体。

访问控制中间件体系框架如图 1 所示。该体系框架对于不同的设备、拓扑结构与应用配置的访问控制需求进行了综合考虑，并且对此类需求是通用的。访问控制中间件包括了访问控制实施组件、访问控制决策组件、访问控制策略应答组件和访问控制属性应答组件。组件的功能见 5.2 节，组件的接口定义见 5.3 节。附录 A 给出了访问控制中间件的典型应用场景。

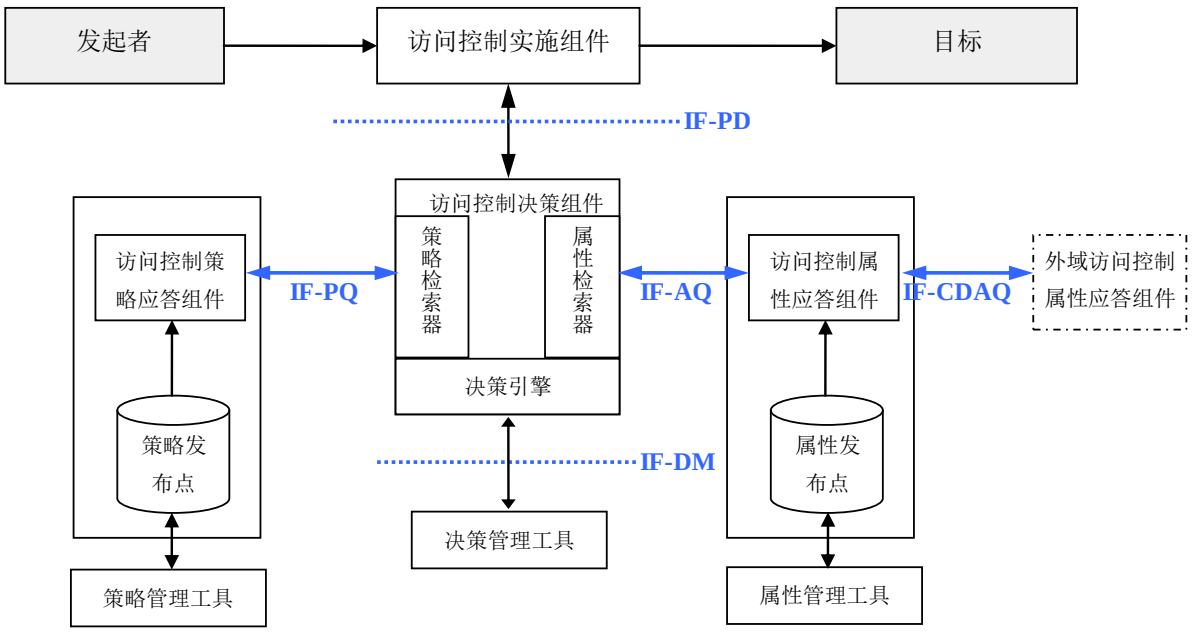


图1　访问控制中间件体系框架

5.2　组件定义

5.2.1　访问控制实施组件

5.2.1.1　概述

访问控制实施组件处于发起者与目标之间，拦截发起者的访问请求，协助收集访问决策的辅助性信息，传递决策请求至访问控制决策组件并根据返回的判定结果决定访问是否可以执行。

访问控制实施组件是直接面向访问请求的应答模块，将发起者请求和具体的授权决策过程等复杂的业务逻辑进行剥离，是决定访问控制中间件和具体业务应用系统能否实现插拔组装的基础性模块。

访问控制实施组件应实现5.2.1.2至5.2.1.5节中规定的功能。

5.2.1.2　接收访问请求

访问控制实施组件应能够接收来自发起者的访问请求。发起访问请求的发起者（用户）可能来自不同物理位置并通过不同的访问代理方式，例如“浏览器/服务器”结构或者“客户端/服务器”结构，访问控制实施组件应该根据固定交互模式对来自用户代理的请求进行一致化处理，对原始请求规定一致的逻辑实体，例如用户标识、访问动作、资源标识、属性信息等，访问请求的格式可依据GB/T 30281-2013等标准。

访问请求信息的传递不限制具体的传输协议，满足访问控制实施组件要求的应用协议都可以负责处理信息传递。

5.2.1.3　收集访问决策相关辅助性信息

访问控制实施组件应能够收集其他对访问决策提供帮助的辅助性信息。用户原始请求中包含的访问信息可能并不足以使访问控制决策组件给出确定的决策结果，访问控制实施组件在将访问请求转发之

前，应根据应用需求对访问请求附加若干有利于加速决策过程的辅助信息，例如用户的属性信息、组件本身可以感知的若干系统信息。强制性规定添加哪些辅助信息并不合适，不同的业务系统会有不同的专注点，可能来自于主体、资源以及环境，应允许管理者通过配置的方式指定优先附加的辅助信息。

辅助信息的添加并不一定限制在访问控制实施组件中，其他组件根据需要也可以具备该功能，例如访问控制决策组件中。根据GB/T 18794.3-2003中的说明，辅助信息的获取方式可分为“推”模式和“拉”模式，采用哪种模式取决于系统的决策逻辑和某些强制性选择，本标准在可以添加辅助信息的模块进行显式说明，采用何种方案最终由用户选择。

5.2.1.4 请求格式转换

访问控制实施组件应能够对对请求格式进行标准形式的转换。将用户发送的访问请求和系统收集的辅助信息用统一的方式描述可显著提高组件的运行效率并降低开发成本，本标准建议采用对访问请求进行统一描述，基于属性的描述机制通过灵活丰富的表达能力对访问决策中涉及的主体、资源、动作、环境等信息进行定义。

5.2.1.5 传递决策请求及接收决策结果

访问控制实施组件应能够传递决策请求至访问控制决策组件并接收决策结果。组件将访问请求完成统一格式转换后，生成决策请求并将其转送至访问控制决策组件并等待其传回最终决策结果。此间的请求/应答交互应遵循相关标准中定义的格式。

访问请求的决策结果可能表示为某种抽象形式，访问实施组件应根据组件所在的应用场景和技术背景将其转换为具体的应用程序执行逻辑，保证高层抽象安全约束和底层程序逻辑的一致性。

5.2.2 访问控制决策组件

5.2.2.1 概述

访问控制决策组件负责从访问控制实施组件接受决策请求，通过查找适用策略和相对应的访问控制属性，依据访问判定逻辑产生一个决策结果，并将该决策结果返回给访问控制实施组件。

访问控制决策组件应实现5.2.2.2至5.2.2.6节中规定的功能。

5.2.2.2 接收决策请求

组件接收到来自访问控制实施组件的决策请求后，需要对请求内的信息进行解析分类，对不同类型的信息实体进行临时性存储。根据决策的具体执行逻辑，可能会针对某种类型信息优先和访问控制策略进行匹配，因此在做出实际的访问决策前，应通过信息分类机制提高后期的执行效率，例如可根据主体、资源、动作等进行分类存储，也可根据角色、组、安全等级等不同的属性类型进行分类。

5.2.2.3 执行访问决策逻辑

访问决策逻辑是整个组件的核心功能，其通用性和兼容性决定了整个中间件部署复杂度以及与应用场景的整合难度。决策逻辑应该考虑到已有的多种访问控制模型和访问控制机制，尽可能提高兼容性以减少重新部署安全策略的代价。GB/T 18794.3-2003对多种访问控制机制进行了说明，包括访问控制列表方案、权利方案、基于标签的方案、基于上下文的方案以及基于以上方案的变种等。上述方案间的区别在于如何将访问相关信息进行组合绑定和决策逻辑所关注的关键决策信息，另外各种方案的访问控制策略描述在实现过程中也有很大不同。

基于属性的访问控制模型提供了一种高层抽象的泛化描述能力，可以将以往出现的各种方案在一致的逻辑语义下进行转化表达。其访问决策逻辑主要依赖三部分输入参数：访问请求、属性信息、访问控

制策略，大体的决策流程为：首先以系统特别关注的敏感信息为关键字对访问控制策略进行检索，获取可能对决策结果产生影响的有效策略集合；然后根据决策请求对策略集合内的策略进一步详细匹配；策略匹配过程中可能需要决策请求之外更详细的属性信息，组件检索信息后完成对策略的精确评估，从而获得最后的决策结果。

该组件匹配的策略在基于属性的描述框架内进行定义，以便于对其他访问控制机制的兼容性转化。

该决策逻辑从多种访问控制机制和模型中概括出基于关键标识匹配的一致性逻辑，“匹配”和“检索”是决策逻辑的两类最基本操作，其他方案都可以在此基础上进行转换。例如：

访问控制列表方案的决策逻辑可以理解为：从请求中提取资源信息，以该信息为关键字检索策略，获取和该资源相关的策略集合，通过将请求中的主体信息和访问动作特征与集合内的策略逐一比较匹配，判断最终的权限拥有权。

该组件的实现框架支持基于角色的控制方案、基于历史的控制方案、基于时间约束的控制方案和基于任务的控制方案等新兴发展起来的访问授权机制。

大型的信息系统条目众多，导致其所涉及的安全策略数目繁杂、策略间关系不够清晰，为了避免出现不可预知的决策结果，需要从宏观角度制定最基本的资源安全策略，以提供最低限度的安全保障。访问控制决策组件通过开放式策略和保守式策略实现这种可预知的和最低限度的安全保障。开放式策略的决策逻辑为：如果没有提供显式策略明确禁止某访问行为，则认为允许该类访问进行；保守式策略的决策逻辑为：如果没有提供显式策略明确允许某访问行为，则认为禁止该类访问进行。采用何种策略取决于具体应用的资源对象敏感性和资源对象使用目的。

该组件应考虑如下情况，即多条策略同时给出明确决策结果，且决策结果存在冲突。此时组件需要指定冲突消解策略处理可能产生的决策结果不一致性，常用的消解策略包括：肯定判定优先、否定判定优先、首次判定优先等。组件也可以根据应用场景的具体要求或者是安全属性的自身特性，自定义冲突消解逻辑满足安全决策需要。

5.2.2.4 对所需访问控制策略进行检索收集

组件在执行决策逻辑的过程中需要以适用策略集合作为请求匹配的参照，因此其内部应该具有策略检索收集的功能。一种最简单的处理方式就是：组件在运行决策逻辑前，将所有策略一次性导入临时存储区，之后所有的匹配操作都针对存储区内的策略进行。当策略数目较大时，应提供针对性更强的策略检索功能，即根据某些属性特征或者策略标识从策略库中获取规模较小的策略子集，减少实际匹配的策略数量，提高匹配效率。例如以某个属性类型或者具体的属性值为关键字，或者以某种特定的策略类型为关键字对策略库进行检索。

该组件不限定策略检索源的具体实现，其存储方式可以为数据库、目录服务器或者文件系统等。建议组件内部设定临时性策略检索结果存储区，对检索后的适用策略实现本地存放，避免策略匹配过程涉及过多的远程交互。同时，应考虑本地策略存储的及时更新。

5.2.2.5 对所需属性信息进行检索收集

虽然决策请求中包含了若干决策需要的属性信息，但不能保证其充分满足策略匹配的全部需要，因此组件应具有相关属性信息的检索功能。策略匹配过程涉及多种类型的属性信息，其特征、来源及发布形式可能多有不同，属性检索过程应考虑能够兼容处理不同的属性格式，例如X509格式的属性证书、SAML格式的安全断言以及LDAP目录中的属性条目等。

组件内部的属性查询过程应由决策逻辑触发，即当决策逻辑无法完成策略匹配时，建立与访问控制属性应答组件的查询请求及应答。请求双方应该在属性描述格式、属性类型、请求/应答方式等方面达成一致的情况下，对查询所得信息进行安全传递。

组件在获取到所需属性后，可以在本地建立临时存储区，避免之后的属性查询涉及过多的远程交互过程，造成策略匹配延迟。同时，应考虑本地属性存储的及时更新。

5.2.2.6 决策历史记录的相关查询

考虑到和其他安全组件的集成性，访问控制中间件应提供相应的服务功能，例如为安全审计提供历史访问的相关数据等。访问控制决策组件在完成请求决策的同时，应对整个过程涉及的信息进行分类记录，例如某访问请求涉及的匹配策略标识、检索到的主体属性信息、资源特征信息、各种环境信息、最终决策结果、组件当时的配置状态等。以上信息应保证存储的安全性和与历史记录的一致性，并且可根据特殊的审计需要增加相应的记录信息类型。

所有历史记录信息对其他安全组件提供基本的输出功能，但不提供其他领域的业务安全分析功能，本标准也不具体约束数据存储的形式和方案。

5.2.3 访问控制策略应答组件

5.2.3.1 概述

访问控制策略应答组件负责响应访问控制决策组件的策略检索请求，对不同形式的策略表达进行一致性转化，完成对适用策略的检索并以安全的方式传输至访问控制决策组件。

访问控制策略应答组件的详细功能描述及细节如下。

访问控制策略应答组件负责响应访问控制决策组件的策略检索请求，负责整个中间件访问控制策略的底层处理。

访问控制策略应答组件应实现 5.2.3.2 至 5.2.3.5 节中规定的功能。

5.2.3.2 统一策略描述方式

不同的安全系统可能采用不同的描述方式定义安全策略，但从中间件的角度出发，其决策逻辑所依赖的策略集必须具有统一的格式和语义。策略应答组件需要确定系统应用的策略类型，并对不同形式的策略表达进行一致性转化。策略转化过程可能需要界定不同策略特征间的转换规则，但应保证策略转化不影响最终的安全目标，因此需要根据系统特征从不同的访问控制策略中抽象高层安全目标视图，并在转化过程中实施目标一致性检测。

5.2.3.3 策略检索

策略应答组件必须能够处理来自决策组件带有多种查询参数的策略检索请求，并获取满足要求的策略集合。最简单的策略请求处理方式是应答组件返回所有可用的安全策略，但在策略规模庞大的应用场景下，这种模式显然无法提供高效的策略匹配效率。决策组件可能会以某些策略特征为关键字对策略库进行精确查找，例如用户角色名称、资源标识、访问类型等，应答组件应该支持这些定制查询参数的检索请求。实际的策略存储点可能采用不同的实现方式，例如数据库、LDAP服务器、文件系统等，应答组件应该具备检索多种存储方式的能力，并对检索后的结果进行整合。应答组件可通过自身开发或借助外部工具的方式提高策略检索的速度、减少检索响应延迟，也可以通过内部缓存机制降低组件间的通讯交互。

5.2.3.4 策略传输

应答组件应与决策组件就策略传输的方式和格式进行统一制定，应答组件完成策略检索后，将响应策略集合以安全可靠的传输协议传输至决策组件。本标准不强制定义具体的策略传输协议，只对一些可选的传输方案进行说明。可以通过网络层的socket通讯协议直接对策略条目进行编码传输，另外针对XML

类型的策略格式也可以采用类似SOAP协议的XML RPC方式进行传输。访问控制中间件可根据技术集成难度、应用环境特点、通讯质量要求等自行选用具体的传输协议。

5.2.3.5 策略管理

实际应用中，访问控制策略管理本身可能涉及一个相对独立的技术领域，其实现技术和方案复杂多样，本标准不试图对策略管理给出完整详细的功能规范，只针对访问控制中间件的实际需求对策略管理应提供的功能提出具体的规范定义，其涵盖的功能模块是策略管理的功能子集。

策略管理服务（工具或模块）应提供对策略的一般性管理功能，例如策略的添加、修改、删除、更新等，以方便中间件对系统安全策略的控制和掌握。

在多条策略的安全约束之间，可能存在潜在的冲突威胁，策略管理服务应提供策略优先级机制，制定策略冲突消解规则，便于访问控制决策组件执行具体的决策逻辑。

策略管理服务还应提供策略一致性检测功能，在策略实体和高层安全目标间进行一致性验证和测试，保证策略实体符合系统的安全管理初衷。

本标准不对以上功能做具体的实现说明，也不强制访问控制中间件必须实现上述功能。

5.2.4 访问控制属性应答组件

5.2.4.1 概述

访问控制属性应答组件负责对访问判定过程中需要的各种类型属性信息进行收集，生成并发布属性断言，并将属性信息集合以安全的方式传输至访问控制决策组件。

访问控制策略应答组件的详细功能描述及细节如下。

该组件主要负责访问决策可能触发的属性信息收集，辅助访问控制决策组件完成最终的请求决策。

访问控制策略应答组件应实现5.2.4.2至5.2.4.6节中规定的功能。

5.2.4.2 用户属性信息收集

当决策请求中包含的用户属性信息不足以使决策逻辑给出决策结果时，决策组件需要向访问控制属性应答组件发送属性查询请求。用户的属性信息可能来自多个属性管理权威机构，属性的颁发格式可能不同，最终的属性存储点也可能分布在不同的物理地址，属性应答组件应该能根据用户标识对属性信息进行集成检索，形成统一的属性表达语义，便于进一步的属性断言发布。

组件处理的属性颁发格式可能为X.509格式的属性证书、SAML断言、LDAP属性条目等。在对检索后获取的用户属性进行确认前，应对这些属性信息的有效性进行验证。验证过程可能是针对属性实体的数字签名验证，也可能涉及对属性颁发实体的数字身份验证，验证能否通过取决于对验证信息的可信性。

针对来自外域的用户属性信息，组件根据外域用户属性检索适用的属性映射规则，推导出外域属性对应的本域属性信息，实现域间属性转译，以决策组件可理解的域内属性信息格式进行发布。转译过程涉及属性信息内容的转译和属性描述格式的转换。

应答组件可通过自身开发或借助外部工具的方式提高属性检索的速度、减少检索响应延迟，也可以通过内部缓存机制降低组件间的通讯交互。

5.2.4.3 其他类型属性信息收集

基于属性的访问控制机制决定了决策组件除了可能需要对用户属性进行检索外，还需要其他类型的信息辅助判断。这些信息可能来自信息系统自身状态、上下文环境、网络状况等一些可以描述访问进行时的外界信息感应点，应答组件应有能力接收或者主动查询来自这些感应点的属性信息。

本标准不试图定义感应点发布属性的方式和属性格式，属性应答组件应将这些属性信息转换为决策组件可理解的语义及格式并以属性断言的格式进行转发。

5.2.4.4 属性断言发布

属性应答组件是决策组件唯一信任的属性发布点，其他的属性存储点和感应点对决策组件来说都应该是透明的，因此应答组件在获取到查询的属性信息后，应该以决策组件可验证的属性断言方式发布属性信息。断言应包含属性的主体标识、属性类型或名称、具体的属性值、应答组件对属性信息摘要的签名等。

属性断言格式的定义宜采用GB/T 29242-2012标准。

5.2.4.5 属性信息传递

属性应答组件应与决策组件就属性传输的方式和格式进行统一制定，应答组件完成属性检索后，将属性信息集合以安全可靠的传输协议传输至决策组件。类似策略传输，本标准不限制定义具体的属性传输协议，可以通过网络层的套接字通讯协议直接对属性条目进行编码传输，另外针对XML类型的属性格式也可以采用类似SOAP协议的XML RPC方式进行传输。

5.2.4.6 属性管理

属性管理已经存在相关标准规范，其实现技术和方案复杂多样，本标准不试图对属性管理细节给出完整详细的功能规范，只针对访问控制中间件的实际需求对属性管理应提供的功能提出具体的规范定义，其涵盖的功能模块是属性管理的功能子集。

属性管理服务（工具或模块）应提供对属性信息的一般性管理功能，例如属性的颁发、撤销、更新等，以方便中间件对属性信息的控制和掌握。

为了支持跨域访问控制等多域应用场景，属性管理服务应提供域间属性映射功能，制定属性映射规则，可发布映射断言供外域的属性发布组件进行查询。

属性管理服务还应提供属性一致性检测功能，限制用户同时拥有违反安全约束的多个属性。

本标准不对以上功能做具体的实现说明，也不强制访问控制中间件必须实现上述功能。属性管理的具体实现应参照相应的数字身份管理标准与规范及其他相关标准。

5.3 组件间接口

5.3.1 策略决策接口（IF-PD）

IF-PD是访问控制实施组件和访问控制决策组件之间的接口。IF-PD接口主要用于传递决策请求消息至访问控制决策组件，并将访问控制决策组件产生的判定结果以决策应答消息的方式传递给访问控制实施组件。此接口的具体实现可见GB/T 31501-2015或者GB/T 30281-2013中的相关定义。

5.3.2 决策管理接口（IF-DM）

IF-DM是管理访问控制决策组件的接口。IF-DM接口主要用于向访问控制决策组件传递消息，控制组件功能的启动与停止，配置组件并控制组件的执行流程与执行环境。

IF-DM接口的定义细节见6.3节。

5.3.3 策略查询接口（IF-PQ）

IF-PQ是访问控制决策组件和访问控制策略应答组件之间的接口。IF-PQ接口主要用于策略的检索以及访问控制策略应答组件的配置。IF-PQ按照指定的检索模式将获取到的策略转换成指定类型的策略，并返回给访问控制决策组件。

IF-PQ接口的定义细节见6.4节。

5.3.4 属性查询接口（IF-AQ）

IF-AQ是访问控制决策组件和访问控制属性应答组件之间的接口。IF-AQ接口主要用于属性的检索以及访问控制属性应答组件的配置。IF-AQ获取主体的属性后，将查询到的属性转为指定格式，并返回给访问控制决策组件。

IF-AQ接口的定义细节见6.5节。

5.3.5 跨域属性查询接口（IF-CDAQ）

IF-CDAQ是不同域的访问控制属性应答组件之间的接口。IF-CDAQ接口主要用于外域访问控制属性应答组件查询本域某个主体的属性。本域访问控制属性应答组件获取主体的属性后，将查询到的属性转为指定格式，并返回给外域的访问控制属性应答组件。

IF-AQ接口的定义细节见6.6节

5.4 接口间消息流

通过上述定义的各不同接口，体系结构中的各组件进行消息交换。这些基本的消息流如图2所示。

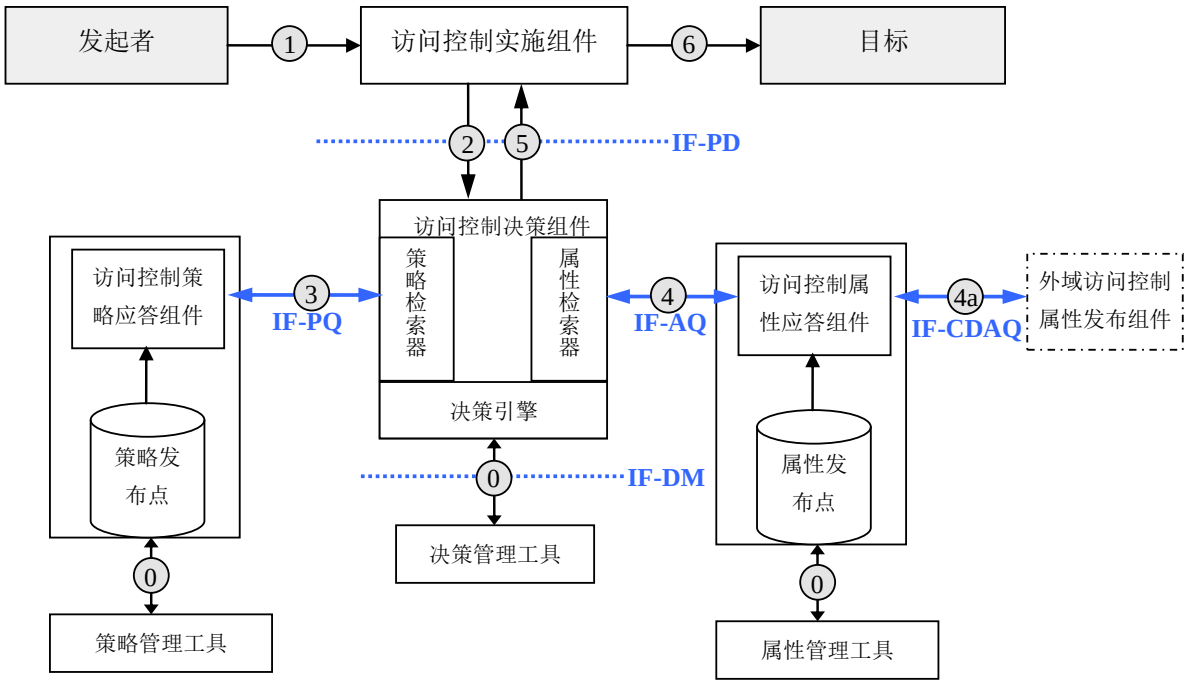


图2 访问控制中间件体系框架中的消息流

a) 在发起者开始访问目标之前，访问控制中间件需要通过管理工具进行初始化与配置管理，包括以下三种操作：通过策略管理工具对访问控制中间件的策略进行管理操作；通过属性管理工具进行属性颁发与撤销等管理操作；通过决策管理工具进行决策引擎的管理与配置。（图 2 步骤 0）
b) 当发起者试图对目标进行访问时，访问控制实施组件拦截发起者的访问请求。（图 2 步骤 1）
c) 访问控制实施组件拦截访问请求后，向访问控制决策组件发送决策请求。（图 2 步骤 2）
d) 访问控制决策组件以决策请求为参数调用策略检索器从访问控制策略应答组件检索适用策略，并对检索的适用策略进行评估。（图 2 步骤 3）
e) 如果访问控制决策组件在评估过程中发现缺乏相应的属性，则通过属性检索器向本安全域的访问控制属性应答组件发出属性查询请求；访问控制属性应答组件查询并验证属性发布点上存储的属性，生成属性应答返回至访问控制决策组件。（图 2 步骤 4）
f) 如果所查询的属性是其它安全域中的属性，则由本安全域的访问控制属性应答组件向外域的访问控制属性应答组件进行查询，以获得外域中的访问控制属性，并通过属性映射关系确定属性的可信性，生成属性应答消息。（图 2 步骤 4a）
g) 访问控制决策组件依据访问控制策略与访问控制属性完成决策评估，向访问控制实施组件发送最终决策结果。（图 2 步骤 5）
h) 访问控制实施组件根据返回的决策结果拒绝或允许发起者对目标的访问。（图 2 步骤 6）

6 访问控制中间件接口

6.1 概述

本章主要对访问控制中间件的接口进行说明和定义，对接口的输入参数，输出参数以及逻辑功能进行规范，但不强制定义接口的具体实现方案和形式。

接口的输入参数与输出参数以XML格式定义，消息的示例见附录B。

接口的实现应支持本节所定义的接口、以及接口所定义的输入参数与输出参数，但可根据应用环境进行扩展。

6.2 常量定义

访问控制中间件各接口返回的消息码定义如下。

表1 消息码定义

返回消息码	值	语义
IF_RESULT_SUCCESS	0	调用接口完成预定功能
IF_RESULT_FAIL	1	调用接口未完成预定功能
IF_RESULT_ILLEGAL_ACTION	2	非法调用接口
IF_RESULT_INVALID_PARAM	3	参数错误
IF_RESULT_NOT_INIT	4	未初始化
IF_RESULT_SELFTEST_ERROR	5	自检错误

6.3 策略决策接口(IF-PD)

IF-PD 是访问控制实施组件和访问控制决策组件之间的接口。IF-PD 接口主要用于传递决策请求消

息至访问控制决策组件，并将访问控制决策组件产生的判定结果以决策应答消息的方式传递给访问控制实施组件。

此接口的具体实现可见 GB/T 31501-2015 中的相关定义。

IF-PD 的调用需要建立在安全信道的基础上，安全信道应保证通信数据的完整性。

6.4 决策管理接口(IF-DM)

IF-DM 是管理访问控制决策组件的接口。IF-DM 接口主要用于向访问控制决策组件传递消息，控制组件功能的启动与停止，配置组件并控制组件的执行流程与执行环境。

IF-DM 的调用需要建立在安全信道的基础上，安全信道应保证通信数据的机密性、完整性。安全信道的宜依据 SSL/TLS 建立。

6.4.1 决策管理登录接口(IF-DM-Login)

6.4.1.1 功能

决策管理的实施需要对决策管理员的身份进行认证，并在认证通过后建立会话。决策管理员的所有操作需要基于建立的会话完成。在调用决策管理其它的接口之前，决策管理登录接口必须首先被调用。

6.4.1.2 输入参数

a) 输入参数类型定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="login">
<xs:complexType>
<xs:sequence>
<xs:element name="uerId" type="xs:string"/>
<xs:element name="credential" type="xs:base64Binary"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明

1) 决策管理员的身份标识；

2) 调用决策管理登录接口应提供调用者的认证信息。认证信息由决策管理组件支持的认证方式决定。例如，若采用证书认证方式，认证信息应该包含决策管理员身份证书。

6.4.1.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
```

```
<xs:element name="sessionId" type="xs:string"minOccurs="0" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用决策管理登录接口应能够得到一个预定义的消息码；

2) 若决策管理员身份通过认证，还需返回所建立的会话的标识。

表2 IF-DM-Login 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	认证决策管理员身份成功
IF_RESULT_FAIL	认证决策管理员身份失败

6.4.2 决策管理登出接口(IF-DM-Logout)

6.4.2.1 功能

决策管理完成后，需要关闭为完成此次决策管理而创建的会话。此接口提供注销所建立的会话的功能。决策管理员完成所有操作后应调用此接口。

6.4.2.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="logout">
<xs:complexType>
<xs:sequence>
<xs:element name="sessionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明

所注销的本次会话的标识。

6.4.2.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
```

```
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用决策管理登出接口应能够得到一个预定义的消息码。

表3 IF-DM-Logout 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	注销会话成功
IF_RESULT_FAIL	注销会话失败

6.4.3 决策管理配置接口(IF-DM-Config)

6.4.3.1 功能

访问控制策略决策组件应是可配置的。决策管理员应能够通过配置访问控制策略决策组件，灵活控制访问控制策略决策组件的执行流程及执行环境。决策管理配置接口可以但不是必须提供对配置的检测功能。调用决策配置管理接口后，访问控制策略决策组件可以即时对配置响应，也可通过重新启动对配置进行响应。

6.4.3.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="config">
<xs:complexType>
<xs:sequence>
<xs:element name="plicyStoragePoint" type="xs:anyURI"/>
<xs:element name="attributeIssuePoint" type="xs:anyURI"/>
<xs:element name="combiningAlg" type="xs:string"/>
<xs:element name="supprotPolicy" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="supportPolicyType" type="xs:string"/>
<xs:element name="policySchema" type="xs:base64Binary"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="sessionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
```

b) 输入参数说明

调用决策管理配置接口应提供但不局限于提供以下配置信息。

1) 策略存储点：访问控制策略决策组件策略查找点；
2) 属性发布点：访问控制策略决策组件属性查找点；
3) 合并方法：访问控制策略决策组件对多个策略评估时的组合逻辑。例如，采用拒绝优先，只要有一个策略的评估结果为拒绝，则最终的决策结果也为拒绝。访问控制策略决策组件只有在对多个策略评估时使用合并方法；
4) 支持的策略：访问控制策略决策组件支持的策略类型以及相应的策略类型模式。访问控制策略决策组件可以根据策略模式，在对查询的策略解析之前，首先判断其是否为自己支持的策略类型。例如，访问控制策略决策组件可以但不限于支持 XACML 格式策略的解析；
5) 本次调用的会话标识。

6.4.3.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用决策管理配置接口应能够得到一个预定义的消息码。

表4 IF-DM-Config 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	配置访问控制策略决策组件成功
IF_RESULT_FAIL	配置访问控制策略决策组件失败
IF_RESULT_INVALID_PARAM	配置参数不符合规定的格式

6.4.4 决策启动接口(IF-DM-Start)

6.4.4.1 功能

启动访问控制策略决策组件提供的服务。访问控制策略决策组件启动时，应首先检查配置信息是否完备。决策管理启动接口可以但不是必须提供访问控制策略决策组件检测功能，以确定系统的状态。调用该接口前应先调用决策管理配置接口。

6.4.4.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="start">
<xs:complexType>
<xs:sequence>
<xs:element name="selfTest" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="sessionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明

1) 调用决策管理启动接口可以但不是必须指定访问控制策略决策组件自检项。

2) 本次调用的会话标识。

6.4.4.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用决策管理启动接口应能够得到一个预定义的消息码，详见表 5：

表5 IF-DM-Start 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	访问控制策略决策组件启动成功
IF_RESULT_FAIL	访问控制策略决策组件启动失败
IF_RESULT_NOT_INIT	访问控制策略决策组件未配置
IF_RESULT_SELFTEST_ERROR	访问控制策略决策组件启动自检失败

6.4.5 决策停止接口(IF-DM-Stop)

6.4.5.1 功能

停止访问控制策略决策组件提供的服务。访问控制策略决策组件停止时，可以采用如下两种模式：

可停止正在提供的服务，同时拒绝新的服务请求；继续完成正在提供的服务，但是拒绝新的服务请求。访问控制策略决策组件停止模式由组件开发者自行选择，或同时支持但由调用者选择。

6.4.5.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="stop">
<xs:complexType>
<xs:sequence>
<xs:element name="stopPattern" type="xs:string"/>
<xs:element name="sessionId" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明

1) 调用决策管理停止接口必须提供采用的停止模式的标识。停止模式的标识由访问控制策略决策组件自行规定。
2) 本次调用的会话标识。

6.4.5.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用决策管理停止接口应能够得到一个预定义的消息码。

表6 IF-DM-Stop 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	访问控制策略决策组件停止成功
IF_RESULT_FAIL	访问控制策略决策组件停止失败
IF_RESULT_INVALID_PARAM	停止模式的标识不被识别

6.5 策略查询接口(IF-PQ)

IF-PQ 是访问控制决策组件和访问控制策略应答组件之间的接口。IF-PQ 接口主要用于策略的检索以及访问控制策略应答组件的配置。IF-PQ 按照指定的检索模式将获取到的策略转换成指定类型的策略，然后返回给访问控制决策组件。

IF-PQ 的调用需要建立在安全信道的基础上，安全信道应保证通信数据的完整性。

6.5.1 策略查询支持类型接口(IF-PQ-SupportPT)

6.5.1.1 功能

访问控制策略应答组件应返回支持的策略类型。例如，只支持 XACML 策略。

6.5.1.2 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SupportPT">
<xs:complexType>
<xs:sequence>
<xs:element name="policyTypeId" type="xs:ID" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用策略查询支持类型接口应可以获得访问控制策略应答组件支持的策略类型信息。返回值的数据结构，以及策略类型的标识由访问控制策略应答组件自行规定。
2) 调用策略查询支持类型接口应能够得到一个预定义的消息码。

表7 IF-PQ-SupportPT 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	查询支持策略类型成功
IF_RESULT_FAIL	查询支持策略类型失败

6.5.2 策略查询返回类型接口(IF-PQ-ReturnPT)

6.5.2.1 功能

访问控制决策组件可能只支持某种类型的组件。访问控制策略应答组件应能够指定返回的策略的类型。调用该接口前应先调用策略查询支持类型接口。

6.5.2.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="setRetPolicyType" type="xs:string"/>
</xs:schema>
```

b) 输入参数说明

调用策略查询返回类型接口应提供指定的返回的策略的类型。策略类型的标识由访问控制策略应答组件自行规定。

6.5.2.3 输出参数：、

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用策略查询返回类型接口应能够得到一个预定义的消息码。

表8 IF-PQ-ReturnPT 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	设置返回的策略的类型成功
IF_RESULT_FAIL	设置返回的策略的类型失败
IF_RESULT_INVALID_PARAM	设定的策略类型不被支持

6.5.3 策略查询查找模式接口(IF-PQ-SearchSchema)

6.5.3.1 功能

访问控制策略应答组件应能够指定策略查找的模式，即只查询第一条适用的策略，或查询所有适用的策略。

6.5.3.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="setSearchPattern" type="xs:string"/>
</xs:schema>
```

b) 输入参数说明：

调用策略查询查找模式接口应提供指定的查找模式标识。策略查找模式标识由访问控制策略应答组件自行规定。

6.5.3.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

a) 输出参数说明

调用策略查询查找模式接口应能够得到一个预定义的消息码。

表9 IF-PQ-SearchSchema 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	设置策略查找模式成功
IF_RESULT_FAIL	设置策略查找模式失败
IF_RESULT_INVALID_PARAM	设定的策略查找模式标识不被识别

6.5.4 策略查询返回模式接口(IF-PQ-ReturnSchema)

6.5.4.1 功能

访问控制策略应答组件应能够指定策略查询结果返回的模式，即若查询到多个适用策略时，或将这些策略直接返回，或将这些策略合并为一个策略返回。

6.5.4.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="setReturnPattern" type="xs:string"/>
</xs:schema>
```

b) 输入参数说明

调用策略查询返回模式接口应提供指定的返回模式标识。返回模式标识由访问控制策略应答组件自行规定。

6.5.4.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
```

```
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用策略查询返回模式接口应能够得到一个预定义的消息码。

表10 IF-PQ-ReturnSchema 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	设置策略查询返回模式成功
IF_RESULT_FAIL	设置策略查询返回模式失败
IF_RESULT_INVALID_PARAM	设定的策略查询返回模式标识不被识别

6.5.5 策略查询策略合并模式接口(IF-PQ-PolicyCombine)

6.5.5.1 功能

访问控制策略应答组件应指定策略合并的模式。即若访问控制策略应答组件的策略查询返回模式设定为将查询到的多个策略合并为一个策略返回时，应按照通过调用该接口指定的策略合并模式对查询到的策略进行合并。

6.5.5.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="setCombiningAlg" type="xs:string"/>
</xs:schema>
```

b) 输入参数说明

1) 调用策略查询策略合并模式接口应提供指定的策略合并模式。策略合并模式由访问控制策略应答组件自行规定

6.5.5.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
```

```
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用策略查询策略合并模式接口应能够得到一个预定义的消息码。

表11 IF-PQ-PolicyCombine 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	设置策略合并模式成功
IF_RESULT_FAIL	设置策略合并模式失败
IF_RESULT_INVALID_PARAM	设定的策略合并模式解析错误

6.5.6 策略查询获取策略接口(IF-PQ-GetPolicy)

6.5.6.1 功能

访问控制策略应答组件应能够返回适用于某一次访问控制请求的策略。调用此接口前，应先设定策略查询返回类型、策略查询查找模式、策略查询返回模式、策略查询策略合并模式。

6.5.6.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getPolicyRequest">
<xs:complexType>
<xs:sequence>
<xs:element name="subject" type="xs:string"/>
<xs:element name="resource" type="xs:string"/>
<xs:element name="action" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明：

用策略查询获取策略接口应提供访问控制请求信息。

6.5.6.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getPolicyResponse">
<xs:complexType>
<xs:sequence>
<xs:choice>
```

```
<xs:element name="policy" type="xs:base64Binary" maxOccurs="unbounded"/>
<xs:element name="policySet" type="xs:base64Binary"/>
</xs:choice>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用策略查询获取策略接口应能得到适用于某一个访问控制请求的策略集,或某一个单独的策略。

2) 调用策略查询获取策略接口应能够得到一个预定义的消息码。

表12 IF-PQ-GetPolicy 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	获取适用的策略成功
IF_RESULT_FAIL	获取适用的策略失败
IF_RESULT_NOT_INIT	访问控制策略应答组件未配置
IF_RESULT_INVALID_PARAM	访问控制请求信息解析错误

6.6 属性查询接口(IF-AQ)

IF-AQ 是访问控制决策组件和访问控制属性应答组件之间的接口。IF-PQ 接口主要用于属性的检索以及访问控制属性应答组件的配置。IF-PQ 获取主体的属性后，将查询到的属性转为指定格式，然后返回给访问控制决策组件。

IF-AQ 的调用需要建立在安全信道的基础上，安全信道应保证通信数据的完整性。

6.6.1 属性查询支持类型接口(IF-AQ-SupportAT)

6.6.1.1 功能

访问控制属性应答组件应提供支持的属性类型。访问控制属性应答组件对于本域可以支持多种类型的属性，也可以仅支持一种类型的属性。例如，只支持“角色”属性。

6.6.1.2 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SupportAT">
<xs:complexType>
<xs:sequence>
<xs:element name="attributeId" type="xs:ID" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用属性查询支持类型接口应可以获得访问控制属性应答组件支持的属性类型信息。返回值的数据结构，以及属性类型的标识由访问控制属性应答组件自行规定。

2) 调用属性查询支持类型接口应能够得到一个预定义的消息码。

表13 IF-AQ-SupportAT 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	查询支持属性类型成功
IF_RESULT_FAIL	查询支持属性类型失败

6.6.2 属性查询支持格式接口(IF-AQ-SupportSchema)

6.6.2.1 功能

访问控制属性应答组件应提供属性查询支持的返回格式。例如，或者以 SAML 断言的形式返回属性查询的结果，或者直接返回属性证书。

6.6.2.2 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SupportSchema">
<xs:complexType>
<xs:sequence>
<xs:element name="schemaName" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用属性查询支持格式接口应可以获得访问控制属性应答组件支持的返回格式。返回格式的标识由访问控制属性应答组件自行规定。

2) 调用属性查询支持格式接口应能够得到一个预定义的消息码。

表14 IF-AQ-SupportSchema 接口可以返回的消息码

返回消息码	**条件**
IF_RESULT_SUCCESS	查询支持的返回格式成功
IF_RESULT_FAIL	查询支持的返回格式失败

6.6.3 属性查询返回格式接口(IF-AQ-ReturnSchema)

6.6.3.1 功能

访问控制属性应答组件应能够设定属性查询的返回格式。例如，可以设定直接返回属性证书。调用该接口前应先调用属性查询支持格式接口。

6.6.3.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="setReturnSchema" type="xs:string"/>
</xs:schema>
```

b) 输入参数说明：

调用属性查询返回格式接口应提供指定的返回格式。属性查询返回格式的标识由访问控制属性应答组件自行规定。

6.6.3.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="message">
<xs:complexType>
<xs:sequence>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

调用属性查询返回格式接口应能够得到一个预定义的消息码。

表15 IF-AQ-ReturnSchema 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	设置属性查询返回格式成功
IF_RESULT_FAIL	设置属性查询返回格式失败
IF_RESULT_INVALID_PARAM	设定的属性查询返回格式标识未识别

6.6.4 属性查询获取属性接口(IF-AQ-GetAttribute)

6.6.4.1 功能

访问控制属性应答组件应能够返回某一主体所具有的属性。调用此接口前，应先设定属性查询返回格式。调用该接口前应先调用属性查询支持类型接口、属性查询返回格式接口。

6.6.4.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getAttributeRequest">
<xs:complexType>
<xs:sequence>
<xs:element name="userId" type="xs:ID"/>
<xs:element name="attributeId" type="xs:ID" minOccurs="0"/>
<xs:element name="Issuer" type="xs:QName" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数定义

1) 调用属性查询获取属性接口应提供所查询主体的唯一标识
2) 调用属性查询获取属性接口应提供所要查询的属性类型标识
3) 调用属性查询获取属性接口可以但不是必须提供所查询属性的颁发者

6.6.4.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getAttributeResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="attribute" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="attributeId" type="xs:ID"/>
<xs:element name="attributeValue" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用属性查询获取属性接口应获得某一主体拥有的属性信息。属性信息的格式通过调用属性查询返回格式指定。
2) 调用属性查询获取接口应能够得到一个预定义的消息码。

表16 IF-AQ-GetAttribute 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	获取某一个主体的属性成功
IF_RESULT_FAIL	获取某一个主体的属性失败
IF_RESULT_NOT_INIT	访问控制属性应答组件未配置
IF_RESULT_INVALID_PARAM	属性查询类型标识未识别

6.7 跨域属性查询接口(IF-CDAQ)

IF-CDAQ 是不同域的访问控制属性应答组件之间的接口。IF-CDAQ 接口主要用于外域访问控制属性应答组件查询本域某个主体的属性。本域访问控制属性应答组件获取主体的属性后，将查询到的属性转为指定格式，然后返回给外域的访问控制属性应答组件。

IF-CDAQ 的调用需要建立在安全信道的基础上，安全信道应保证通信数据的完整性。安全信道的可以但不是必须依据 SSL/TLS 建立。

6.7.1 跨域属性查询支持类型接口(IF-CDAQ-SupportAT)

6.7.1.1 功能

访问控制属性应答组件应提供外域可查询的属性类型。一般情况下，外域可查询的属性类型要少于本域可查询的属性类型。

6.7.1.2 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SupportAT">
<xs:complexType>
<xs:sequence>
<xs:element name="attributeId" type="xs:ID" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用跨域属性查询支持类型接口应可以获得访问控制属性应答组件外域可查询的属性类型信息。返回值的数据结构，以及属性类型的标识由访问控制属性应答组件自行规定。
2) 调用跨域属性查询支持类型接口应能够得到一个预定义的消息码。

表17 IF-CDAQ-SupportAT 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	获取跨域属性查询支持属性类型成功

IF_RESULT_FAIL	获取跨域属性查询支持属性类型失败

6.7.2 跨域属性查询返回格式接口(IF-CDAQ-SupportSchema)

6.7.2.1 功能

访问控制属性应答组件应提供跨域属性查询结果返回格式。例如以 SAML 断言格式返回跨域属性查询结果。

6.7.2.2 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SupportSchema">
<xs:complexType>
<xs:sequence>
<xs:element name="schemaName" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用跨域属性查询返回格式接口应可以获得访问控制属性应答组件跨域属性查询结果返回格式。返回格式的标识由访问控制属性应答组件自行规定。
2) 调用属性查询返回格式接口应能够得到一个预定义的消息码。

表18 IF-CDAQ-SupportSchema 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	获取跨域属性查询返回格式成功
IF_RESULT_FAIL	获取跨域属性查询返回格式失败

6.7.3 属性查询获取属性接口(IF-CDAQ-GetAttribute)

6.7.3.1 功能

访问控制属性应答组件应能够返回外域查询的本域某一主体所具有的属性。调用此接口前一般应先调用跨域属性查询支持类型接口和跨域属性查询返回格式接口,以确定外域可查询的属性类型以及属性查询结果的返回格式。

6.7.3.2 输入参数

a) 输入参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getAttributeRequest">
```

```
<xs:complexType>
<xs:sequence>
<xs:element name="userId" type="xs:ID"/>
<xs:element name="attributeId" type="xs:ID" minOccurs="0"/>
<xs:element name="Issuer" type="xs:QName" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输入参数说明

1) 调用跨域属性查询获取属性接口应提供所查询主体的唯一标识
2) 调用跨域属性查询获取属性接口应提供所要查询的属性类型标识
3) 调用跨域属性查询获取属性接口可以但不是必须提供所查询属性的颁发者

6.7.3.3 输出参数

a) 输出参数定义

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="getAttributeResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="attribute" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="attributeId" type="xs:ID"/>
<xs:element name="attributeValue" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="messageCode" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

b) 输出参数说明

1) 调用属性查询获取属性接口应获得某一主体拥有的属性信息。属性信息的格式通过调用属性查询返回格式指定。
2) 调用属性查询获取接口应能够得到一个预定义的消息码。

表19 IF-CDAQ-GetAttribute 接口可以返回的消息码

返回消息码	条件
IF_RESULT_SUCCESS	获取某一个主体的属性成功

IF_RESULT_FAIL	获取某一个主体的属性失败
IF_RESULT_INVALID_PARAM	属性查询类型标识不支持

附 录 A
（资料性附录）
应用场景

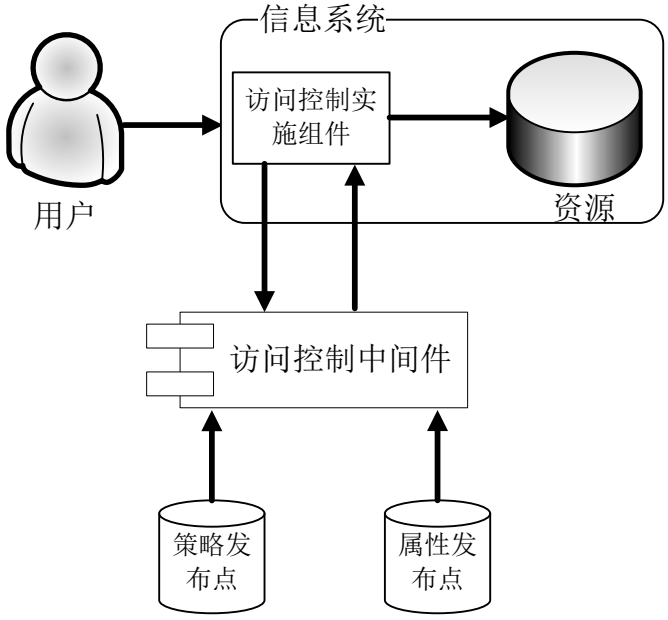
A.1 介绍

本附录描述了访问控制中间件的两种应用场景，部署访问控制中间件可参考但不局限于此两种应用场景。

A.2 访问控制中间件应用于单域

一般情况下，访问控制中间件应用于单个域。对域内用户的访问请求进行响应，并将判定结果返回给应用系统。在这种情况下，访问控制中间件在判定过程中若需要查询用户属性，只需在域内的属性查询点查询。

用户请求对需要进行访问控制的资源的访问时，请求由应用系统进行处理，应用系统需要与访问控制中间件交互，中间件此时被调用，如果判断过程需要查询详细信息，则可以访问策略发布点和属性发布点进行查询，借助获取的信息进行判定，并将结果返回给应用系统。应用系统根据判定结果来决定是否允许用户的访问请求。如图 A.1 所示。

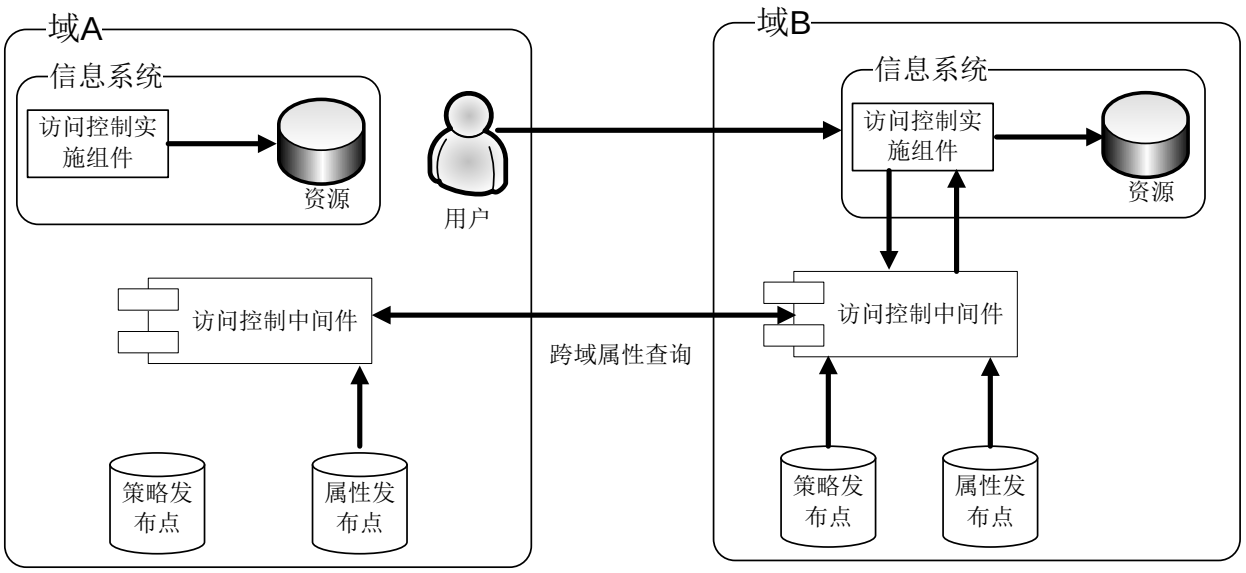


图A.1 单域应用场景

A.3 访问控制中间件应用于跨域

某些情况下，访问控制中间件可能应用于跨域，即需要对外域用户访问本域资源进行判定。在这种情况下，本域访问控制中间件可能需要与外域访问控制中间件交互，查询外域用户拥有的外域属性。

如图 A.2 所示，位于域 A 中的用户在对域 B 中的某些资源发出访问请求时，域 B 中的访问控制实施组件会调用访问控制中间件进行判定。此时域 B 中没有该用户的属性信息，因此域 B 中的访问控制中间件要跨域访问域 A 中的访问控制中间件，以获取该用户的属性信息。域 A 中的访问控制中间件进行查询并返回结果。域 B 中的访问控制中间件根据在本地策略发布点查询得到的信息和跨域交互返回的信息来进行判定，将结果返回给访问控制实施组件，访问控制实施组件根据判定结果作出决策。



图A.2　跨域应用场景

附 录 B
（资料性附录）
接口消息示例

B.1 概述

本章对本标准中的访问控制接口的消息给出了采用XML描述的具体示例。

B.2 接口消息示例

B.2.1 决策管理登录接口(IF-DM-Login)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<login>
<userId>lois</userId>
<credential>"凭证信息的Base64编码"</credential>
</login>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
<sessionId>0ED41D3E6BA1125A4FF0990128A511FE</sessionId>
</message>
```

B.2.2 决策管理登出接口(IF-DM-Logout)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<logout>
<sessionId>0ED41D3E6BA1125A4FF0990128A511FE</sessionId>
</logout>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.3 决策管理配置接口(IF-DM-Config)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<config>
```

```
<policyStoragePoint>http://192.168.0.1/policies</policyStoragePoint>
<attributeIssuePoint>http://192.168.1.2/attributes</attributeIssuePoint>
<combiningAlg>DenyOveride</combiningAlg>
<supportPolicy>
<supprotPolicyType>XACML</supprotPolicyType>
<policySchema>"策略模式文件Base64编码"</policySchema>
</supportPolicy>
<sessionId>0ED41D3E6BA1125A4FF0990128A511FE</sessionId>
</config>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.4 决策启动接口(IF-DM-Start)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<start>
<selfTest>policyStoragePoint</selfTest>
<selfTest>attributeIssuePoint</selfTest>
<selfTest>CombiningAlg</selfTest>
<sessionId>0ED41D3E6BA1125A4FF0990128A511FE</sessionId>
</start>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.5 决策停止接口(IF-DM-Stop)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<stop>
<stopPattern>StopAllServices</stopPattern>
<sessionId>0ED41D3E6BA1125A4FF0990128A511FE</sessionId>
</stop>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.6 策略查询支持类型接口(IF-PQ-SupportPT)

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<SupportPT>
<policyTypeID>XACML</ policyTypeID>
<policyTypeID >Ponder</policyTypeID>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</SupportPT>
```

B.2.7 策略查询返回类型接口(IF-PQ-ReturnPT)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<SetRetPolicyType>XACML</SetRetPolicyType>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.8 策略查询查找模式接口(IF-PQ-SearchSchema)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<SetSearchPattern>AllPolicies</SetSearchPattern>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.9 策略查询返回模式接口(IF-PQ-ReturnSchema)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<SetReturnPattern>SinglePolicy</ SetReturnPattern>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.10 策略查询策略合并模式接口(IF-PQ-PolicyCombine)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<SetCombiningAlg>DenyOverride</SetCombiningAlg>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.11 策略查询获取策略接口(IF-PQ-GetPolicy)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<getPolicyRequest>
<subject>user</subject>
<resource>test.txt</resource>
<action>read</action>
</getPolicyRequest>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<getPolicyResponse>
<policy>"访问控制策略的BASE64编码"</policy>
<policy>"访问控制策略的BASE64编码"</policy>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</getPolicyResponse>
```

B.2.12 属性查询支持类型接口(IF-AQ-SupportAT)

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<SupportAT>
<attributeID>group</attributeID>
<attributeID>role</attributeID>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</SupportAT>
```

B.2.13 属性查询支持格式接口(IF-AQ-SupportSchema)

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<supportSchema>
<schemaName>SAML</schemaName>
<schemaName>Certifcate</schemaName>
<messageCode>IF_RESULT_SUCCESS</messageCode>
```

```
</supportSchema>
```

B.2.14 属性查询返回格式接口(IF-AQ-ReturnSchema)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<SetReturnSchema>SAML</SetReturnSchema >
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<message>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</message>
```

B.2.15 属性查询获取属性接口(IF-AQ-GetAttribute)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<getAttributeRequest>
<userID>lois</userID>
<attributeID>role</attributeID>
<Issuer>iscas</Issuer>
</getAttributeRequest>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<getAttributeResponse>
<attribute>
<attributeID>role</attributeID>
<atributeValue>manager</atributeValue>
</attribute>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</getAttributeResponse>
```

B.2.16 跨域属性查询支持类型接口(IF-CDAQ-SupportAT)

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<SupportAT>
<attributeID>group</attributeID>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</SupportAT>
```

B.2.17 跨域属性查询返回格式接口(IF-CDAQ-SupportSchema)

——输出

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<supportSchema>
<schemaName>SAML</schemaName>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</supportSchema>
```

B.2.18 属性查询获取属性接口(IF-CDAQ-GetAttribute)

——输入

```
<?xml version="1.0" encoding="utf-8"?>
<getAttributeRequest>
<userID>lois</userID>
<attributeID>group</attributeID>
<Issuer>iscas</Issuer>
</getAttributeRequest>
```

——输出

```
<?xml version="1.0" encoding="utf-8"?>
<getAttributeResponse>
<attribute>
<attributeID>group</attributeID>
<atributeValue>Technology Department</atributeValue>
</attribute>
<messageCode>IF_RESULT_SUCCESS</messageCode>
</getAttributeResponse>
```