1. 服务依赖的提取：细粒度的获取微服务间的依赖关系

侵入式工作：手动注释，代码注入

优点：可以保证依赖的完全覆盖，但需要修改应用程序的源代码

非侵入式工作：

通过网络数据包/流推断或日志挖掘获得服务依赖性

这些方法不会侵入源代码，但是不能保证完全覆盖所有API

1. 分布式系统自动生成安全策略

* 基于文档的方法：NLP 虽然文档可以更好地表达开发人员的意图，但它们并不总是存在。同时，由于NLP的限制，这些方法通常是粗粒度和不完整的
* 基于历史方法：利用收集的痕迹或历史数据从流量中推断规则标准和策略结构。它们的有效性取决于跟踪的粒度和完整性，这很难保证。此外，它们要求应用程序提前运行以收集数据，这可能会导致攻击窗口。
* 基于模型的方法（例如，[7,25]），它手动构建模型以了解系统的安全需求，然后相应地生成安全策略。建模过程耗时且容易出错，因此不适合灵活的微服务应用程序

方法使用静态分析生成策略

许多研究[10,24,32]试图通过静态分析获得程序所需的特权行为，如文件访问或系统调用。然后，它们相应地生成不同类型的访问控制策略，以降低安全风险。最近的一项研究[16]将这一想法应用于集装箱。它利用代码分析提取容器化应用程序所需的系统调用，并生成Seccomp策略以缩小攻击面。

1. 策略更新：

更新策略的一般过程从分析和验证意图的变化开始，然后构建参考模型以获得最小更新

1. 基于图的策略管理

将策略抽象为图形是自动策略管理的最新技术。可以根据不同的意图和策略定义构造各种图结构

微服务+访问控制

Towards multi-party policy-based access control in federations of cloud and edge microservices

云和边缘微服务联合体中基于多方策略的访问控制

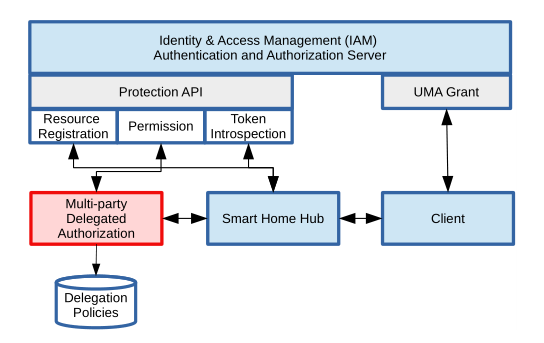
为了确保访问控制在如此复杂的服务交付模型中保持可持续性，我们提供了一个灵活、可扩展和上下文相关的动态粒度访问控制解决方案

这项研究的主要贡献扩展了开源构建块，可以概括如下：1）将授权委托给多个利益相关者（服务提供商和消费者、内容所有者等）2）授权策略和支持运行时框架的独立部署性和可管理性3）基于策略的访问控制面向服务，支持联邦中跨域的互操作性

服务的软件平台中访问控制的几种实践状态和最先进的解决方案。

我们的目标是一个多方访问控制解决方案，其中来自不同运营领域的最终用户或利益相关者可以共同拥有并授予对内容和资源的访问权，从而可以在单个微服务和联合级别管理细粒度授权

基于策略的微服务API访问控制



为了确保以用户为中心的访问控制在如此复杂的服务交付模型中保持可持续性，我们在RedHat的KeyClope 5上提出并评估了一个动态粒度访问控制解决方案，身份和访问管理（IAM）解决方案和用户管理访问（UMA）20规范以及其他基于策略的授权框架。特别是，我们的基于多方策略的访问控制解决方案可以处理每个资源的多个所有者，并委托策略授权，而且相对性能开销有限。

改进现有的安全协议框架

Towards Continuous Access Control Validation and Forensics

实现连续访问控制验证和取证

解决：系统平台 管理员在做出一些修改后，访问控制不会更改带来的不安全的行为

为了适应上面列出的更改，系统管理员可能会引入错误配置。适应这些更改可能并不容易，因为更改访问控制策略可能涉及修改多个组件，例如web服务器、应用程序服务器、数据库和文件系统等。同时，系统管理员在被请求进行更改时可能会有时间压力。一种常见的情况是关于处理拒绝访问问题：当用户抱怨被拒绝访问她应该拥有访问权限的内容时，系统管理员需要为她快速解决该问题。由于时间压力，系统管理员可能会在不仔细检查更改是否授予请求最少的权限或授予额外的意外权限的情况下执行一些快速更改作为变通方法。在这种情况下，此类更改很容易导致访问控制配置错误。最近对现实世界拒绝访问问题解决方案的分析表明，38。1%的更改引入了错误配置，过度授予权限并造成漏洞[68]。

DIFF，一个从访问日志推断访问控制行为和行为变化的实用工具。正如我们将在§2中所示，大多数软件系统生成的现有访问日志包含足够的信息来推断更改。访问控制策略更新问题

实现：基于日志，通过决策树+建模来实现 系统行为的验证，使用基于决策树的模型 访问控制本身就是一个决策分类器 机器学习算法

相关工作

针对：

访问控制的错误配置：

检验访问控制策略间的不一致性，然而，不一致性只反映了一组非常小的、特定的访问控制错误配置。正如在这些著作中所承认的那样，错误的配置可能是完全一致的，这通常会导致更严重的后果。此外，这些工作需要领域知识来解释不同软件的特定访问控制策略

虽然测试和验证已经证明了有希望的结果，但由于编写测试用例或验证规范的大量工作，它们尚未在实践中得到广泛应用。特别是，现有的测试和验证方法需要一个统一和集中的模型（如XACML）；然而，今天的系统访问控制策略以各种形式保存，包括各种配置文件格式、文件权限或数据库特权表。很难涵盖访问控制配置的所有组合。

Confine: Automated System Call Policy Generation for Container Attack Surface Reduction

限制：容器攻击面缩减的自动系统调用策略生成

为了为保护任意容器提供一种实用的解决方案，本文提出了一种自动生成Docker容器限制性系统调用策略的通用方法。我们的系统名为CONMICT，它使用静态代码分析来检查容器化应用程序及其所有依赖项，确定容器正确操作所需的系统调用超集，并生成相应的Seccomp系统调用策略，该策略在加载容器时可以轻松实施。

我们的目标是提供一种更通用、更实用的解决方案，该解决方案可以在无需培训的情况下轻松应用于任何集装箱的保护。为此，我们提出了一种自动技术，用于为任意容器生成限制性系统调用策略，并限制可能被滥用的底层内核的公开接口。通过依赖静态代码分析，我们的方法检查容器化应用程序的执行路径及其所有依赖项，并确定容器正确操作所需的系统调用超集。

Graph-Based IoT Microservice Security

基于图的物联网微服务安全

我们展示了一个基于图形的访问控制，它作为模块在物联网节点或网络中运行。我们的解决方案拦截和防火墙服务间通信。它会自动创建合法通信关系的模型。模型通过一个简单易懂的界面进行交互更新

我们遵循一种互补的、在网络中自我学习的方法

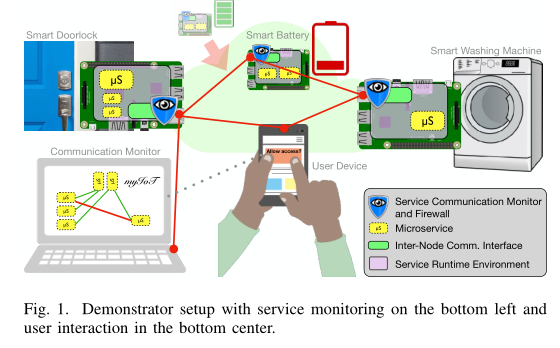
* 监控物联网µSs的通信
* 自动为每个µS创建通信模型
* 根据模型将内部服务通信流量分类为正常或异常。

流量监视+DPI解析->物联网通信拓扑图，通过模型来自更新

为了引导新服务的通信模型，我们在监控和数据采集（SCADA）领域引入了类似于[9]方法的学习阶段。在这个阶段，即服务第一次启动时，我们假设服务表现良好，并将所有通信流量列为白名单。典型的物联网站点缺少专业管理员。此外，物联网是高度动态的，频繁的拓扑变化导致管理复杂性高。为了降低这种复杂性，我们的安全管理是最自动化的。

对于µS通信图的交互式更新，我们向站点经理提供了一个列表，其中包含已检测到但尚未成为模型一部分的通信关系。这样的站点管理员可以是物联网智能空间的常住者。所呈现的信息处于高语义抽象级别，以使非专业人员能够做出有意义的决策。批准或拒绝提示的通信关系将更新我们的通信模型

我们正在拦截服务间通信以进行分析。在这一步中，我们还实现了一个防火墙，用于丢弃不允许的通信数据包。如前所述，我们的自学习异常检测独立于µSs中的安全实现运行。因此，它可以与多供应商物联网系统和复杂拓扑相结合。唯一的要求是了解DPI使用的通信协议。结合主动流量过滤，我们的解决方案成为一个自学习物联网防火墙。其默认密度策略通过设计为现有物联网系统增加了安全性



A Deep Learning Approach for Extracting Attributes of ABAC Policies

一种提取ABAC策略属性的深度学习方法

实现这一自动化的关键步骤是从NLACP（自然语言访问控制政策）自动提取ABAC属性、

们设计了一个实用的框架来自动化属性提取任务，它本身就是ABAC策略编写生命周期的整个阶段

我们构建了一个真实的合成数据集，用访问控制策略的主体和对象元素的属性进行注释。

A Secure Microservice Framework for IoT

API网关是在两个或多个微服务之间中继调用或请求的接口。API网关可以将多个微服务的API聚合到一个客户端接口中，还可以将调用或请求从一个入口点分发或路由到多个目标微服务。因此，不允许微服务之间进行直接通信，而是可以在它们之间放置一个API网关，该网关可以通过网关的聚合和分发功能调整互连性。然后通过注册实现微服务连接；当微服务将自身注册到API网关时，它会创建一个或多个具有唯一标识的端点，该端点可以通过静态配置或与API网关的动态协商进一步绑定到事件通道或方法。除了微服务之外，任何用户或机器客户端也可能需要将其身份作为标识符的名称呈现，以便与API网关交互以访问其关联的微服务。身份管理和身份验证服务可能利用管理帐户的现有系统，如单点登录（SSO）或OAuth[23]。

从API网关的角度来看，每个端点都是唯一可标识的，其中标识可以表示为唯一的名称或仅表示为标识符。此类端点名称或标识符仅存在于模型中，API网关实现需要将其映射到实际协议中

我们的API网关的角色类似于web服务的负载平衡代理或MQTT协议的消息代理。虚拟对象可以使用RESTful URL映射到资源，方法可以映射到该URL上的HTTP方法和JSON负载的组合。

在许多情况下，设备上的微服务不具有可公开访问的IP地址，这一特点使它们无法通过域intranet之外的Internet访问。多协议API网关可以弥补这一差距。在我们的模型中，我们将由API网关及其相关微服务组成的子系统称为物联网站点（类似于传统互联网中的网站）。API网关作为一个枢纽，将消息从一个协议转换为另一个协议，使不同域中的微服务可以相互访问。同时，网关还接受来自站点外部的用户或机器客户端的API调用，并将它们转换为相应的消息以进行intranet API调用。

我们借用了软件定义网络（SDN）的策略实施方法，在我们的模型中，有三种类型的策略：允许或拒绝创建或销毁特定命名服务API的规则、允许或拒绝服务API绑定到端点的规则以及绑定约束。前两个类似于操作系统中的文件系统权限，最后一个类似于SDN中的流策略，其中流是一个绑定。我们解释了三个基于使用的策略示例，它们是名称解析策略和QoS策略（在本小节中）以及访问控制策略（在后面的小节中）。

版本更新：

名称解析策略是服务API和端点之间的静态绑定。这种策略对于集成不同版本的微服务或负载平衡非常有用。再次考虑我们的体重管理IOT系统：开发人员可以构建一个新版本的推荐微服务并将其部署到云。作为响应，系统管理员只需更改一些策略即可将API网关上的方法绑定到新版本的microservice的端点。如果新版本在语法和语义上向后兼容，那么替换就很容易了。如果不兼容，系统管理员可以创建新的虚拟对象和方法，将它们绑定到新版本的微服务，并并行运行这两个版本，直到使用旧微服务的所有客户端都切换到新版本。对于负载平衡，管理员可以选择运行同一微服务的多个实例，并将它们绑定到同一组虚拟对象和方法。类似地，还可以通过将多个事件使用者绑定到同一事件通道来促进平衡，以便在它们之间调度事件，而不是将事件复制到所有它们。

QoS策略主要关注绑定上的流量限制（如果我们将绑定视为流量流）。限制可能包括限制每秒的消息数或每秒的通信量（以字节为单位）。此类策略有助于保护微服务免受DoS攻击，或防止从某些设备或外部客户端启动DoS

QoS策略的另一个用途是实现所谓的“断路器”[24]，其中，如果微服务的运行状态显示其正在耗尽其能力，则可以动态生成QoS策略以限制进一步的请求以防止级联故障。

我们建议每个微服务应提供一个具有微服务名称的虚拟对象，该虚拟对象应具有一个名为“内省”的方法，该方法返回一个清单，详细说明微服务的所有虚拟对象和服务接口（以及到各种协议的映射）。这样的清单可能只包含供人类开发人员使用的纯文本文档，或者要在浏览器上呈现的带注释的XML/HTML，甚至是OWL/RDF中的带注释的文档，以便人类开发人员和机器程序都可以阅读。此外，由于微服务可以动态添加到物联网系统，因此本体可能会发生变化。提供一个全局服务注册表微服务（可能与API网关共存）是很有帮助的，它管理本体和清单，以允许在多个站点之间注册和发现服务。每个单独的微服务都可以与注册表对话，以便发布其结构和接口，以便其他微服务可以通过SPARQL查询来定位它。

访问控制策略确定是否允许服务API和端点之间的动态绑定，从而有效地控制端点是否可以发布/订阅事件通道，或者是否可以进行或提供方法调用。策略本身可以动态生成，此类策略的实现通常依赖于附加的身份管理和身份验证微服务以及策略控制器微服务。

我们提出了一种基于属性的加密（ABE）[33]方法，作为构建物联网ABAC模型的重要构建块之一

Blockchain-Based Access Control Model to Preserve Privacy for Personal Health Record Systems

保护个人健康记录系统隐私的基于区块链的访问控制模型

尽管区块链技术具有许多良好的特性，如不变性和不可逆性[1,2]，但区块链技术在PHR系统开发中也存在一些潜在的缺点。区块链的不可逆性成为许可撤销功能的障碍，允许用户取消对指定个人数据的特定操作的许可。区块链的透明特性允许网络上的所有参与者查看所有数据，这可能导致机密性问题。区块链的有限存储成为各种医疗相关数据爆炸性增长的可用性问题。

相关工作：

例如，基于区块链的访问控制层被添加到现有数据库中，以保护系统的隐私[10–13]。这些系统可以支持不可变的访问日志和以用户为中心的访问控制。但是，不支持撤销同意和数据保密。一些系统[14–23]使用基于属性的加密方案来提供访问控制功能。然而，基于属性的加密导致操作时间随着未撤销用户的数量线性增长，区块链的不可逆性成为撤销同意的障碍。[24–27]中的系统试图将数据存储在区块链上。但是，医疗数据可能很大，区块链没有针对存储大规模数据进行优化

在这项工作中，区块链技术将用于支持不可否认性、可问责性和抗篡改性；代理重加密技术将用于提出一种访问控制机制，该机制可支持细粒度访问控制和同意撤销属性；云存储将用于支持可用性属性。提出了基于区块链的PHR系统访问控制的详细模型，以使用AFGH代理重加密算法显示工作流[29]。PHR数据将使用代理加密技术进行加密，并存储在云存储上。相关元数据将存储在私有区块链上。特别是，PHR数据的特征将永远存储在区块链中。因此，将检测并验证所有数据篡改。密码认证技术和访问控制列表用于验证用户，以支持可问责性和可撤销性功能。

（超级账本）Hyperledger区块链作为基础系统实施

提出的系统的医疗数据被加密并存储在云存储上以获得可用性，元数据存储在区块链上以防篡改。然而，在[14]中使用的基于属性的加密的访问控制机制有一些缺点，因为访问策略的更改或修改是困难的，因为访问策略的修改需要额外的计算成本来对数据执行属性撤销和重新加密过程。区块链的仅附加存储也不允许修改基于属性的加密数据以更新访问控制。

使用了基于区块链的数据共享系统，其中基于区块链的访问控制层被添加到提供商的现有数据库中。这些系统仅存储元数据来描述真实数据及其在区块链上的权限。这些系统还可以保持不变的访问日志，并支持以用户为中心的访问控制。然而，为了支持PHR系统的目的，可能需要撤销同意和对用户数据进行保密。此外，其中一些系统涉及交易费用，用户需要参与采矿活动。与这些系统类似，我们模型中的区块链不用于存储整个医疗数据。

基于属性的加密方案和半可信服务器用于将PHR数据存储在[15–23]中。在这些系统下，通过直接加密真实PHR数据来创建访问控制。真实的PHR数据可能包含多个大文件，基于属性的加密会导致计算成本随着未恢复用户的数量线性增加。

在[24]中使用区块链作为存储技术，并通过使用多个权限提出了基于属性的签名方案。

医疗数据可能包括图像等大数据，而区块链可能无法优化以存储海量数据。即使数据可以分解成许多较小的片段以存储在区块链上，这样的操作在加密数据时可能会导致一些性能问题。

区块链还用于[31–33]中交易处理的访问控制。这些系统不用于数据存储目的。区块链和其他机制仅限于授权某些人进行敏感交易，并实现此类交易的不可否认性。在这项工作中，区块链将被用于提出一个隐私保护个人健康记录系统，以支持不可否认性、问责制和防篡改属性。

A Permissioned Blockchain based Access Control System for IOT

一种基于许可区块链的物联网访问控制系统

基于许可区块链的物联网访问控制系统，其中访问控制的不同阶段，如创建访问策略和根据所有利益相关者的共识做出访问控制决策。更具体地说，我们在名为Hyperledger Fabric的许可区块链中设计和实施基于属性的访问控制（ABAC），并利用其智能合约和分布式共识实现物联网的分布式访问控制。这比基于角色的访问控制或基于身份的访问控制等其他访问控制机制更适合物联网。这是因为物联网生态系统由大量不同功能、特征和能力的物联网设备组成。在这样一个多样化的环境中，只有ABAC能够提供表现力强的细粒度访问控制

我们是第一个提出基于属性访问控制的基于私有区块链的物联网访问控制系统的人。由于我们的方案基于私有区块链，访问请求的解决速度比公共区块链快得多

相关工作：

除了加密货币，区块链的采用在物联网领域也很引人注目。例如，在[4]，[7]中提出了一种基于比特币的访问控制机制。由于比特币不支持智能合约，因此提议的访问控制机制非常基本，不为异构物联网设备提供细粒度访问控制。[8] 提出了一种基于以太坊智能合约的物联网设备可扩展访问管理架构，其中区块链在物联网设备中运行。它是在小型本地以太坊测试网络中实现的。然而，它在最初的公共以太坊网络中如何工作还不清楚。Dorri等人在[9]-[11]中提出了一种解决方案，其中覆盖网络由物联网节点构成。多个覆盖节点形成一个集群，每个集群都有一个选定的头儿（CH）。集群头儿维持一个新提议的公共区块链。它有一种新的事务格式，并且没有smartcontract，这使得复杂的访问控制策略的实现非常困难。[6]中提出了一种基于私有区块链的访问控制管理系统，尽管该系统并非直接用于物联网。此外，本文还考虑了基于角色的访问控制，这不适合于涉及大规模异构设备且缺乏标准化的物联网访问控制