威胁情报平台设计文档

#山石网科/云瞻

总体架构设计

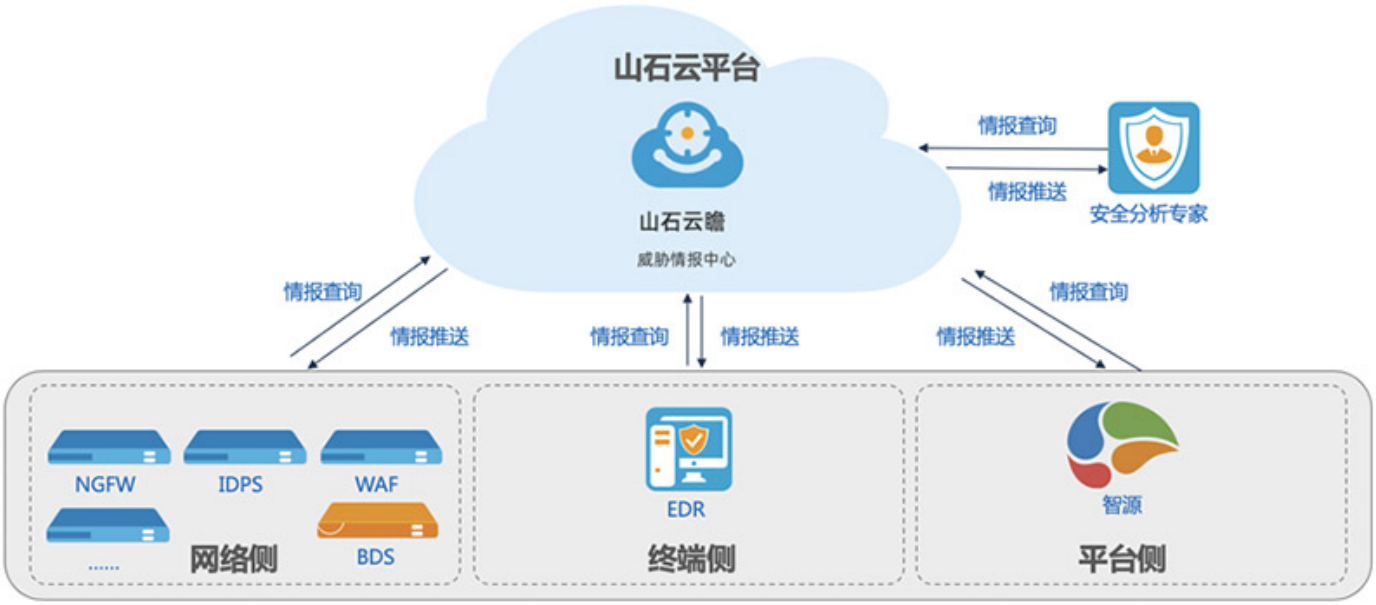
产品概述

随着信息化和网络的高速发展，针对我国关键基础设施及政府网站等的攻击事件高发，特别是实时性更强、传播规模更广的挖矿病毒、勒索软件等网络攻击行为。传统的安全防御方式难以应对高级持续性威胁（APT）、0day等新型网络威胁。威胁情报的出现改变了以往被动检测为主的防护模式，利用大数据等技术手段以更加智能的方式掌握网络安全事件、重大漏洞、攻击手段等信息，并在第一时间采取预警和应急响应等工作,威胁情报的应用和不断落地，使得安全防御更加智能。山石云瞻威胁情报服务，旨在帮助用户提升网络威胁防御水平，更好的应对威胁。

客户价值

山石云瞻威胁情报中心作为能力中心，通过云端威胁情报的收集、处理和分析，可为客户提供及时准确的威胁情报数据。通过云端与本地安全产品协同联动，客户能够及时了解当前最需关注的热点威胁，协助客户深入全面的掌握威胁信息，实现威胁追踪和攻击溯源，帮助客户实施积极主动的威胁防御和快速响应策略，提升客户网络威胁防御水平。

应用场景



威胁情报中心的应用场景如下：

- **本地安全设备协同联动提升威胁检测能力**

  山石云瞻威胁情报中心针对威胁情报数据进行深度整合，将威胁情报数据，比如IP、域名、AV等机读情报自动化集成到山石网科安全网关设备中，通过云端与本地的智能协同方式快速提升企业威胁检测能力。

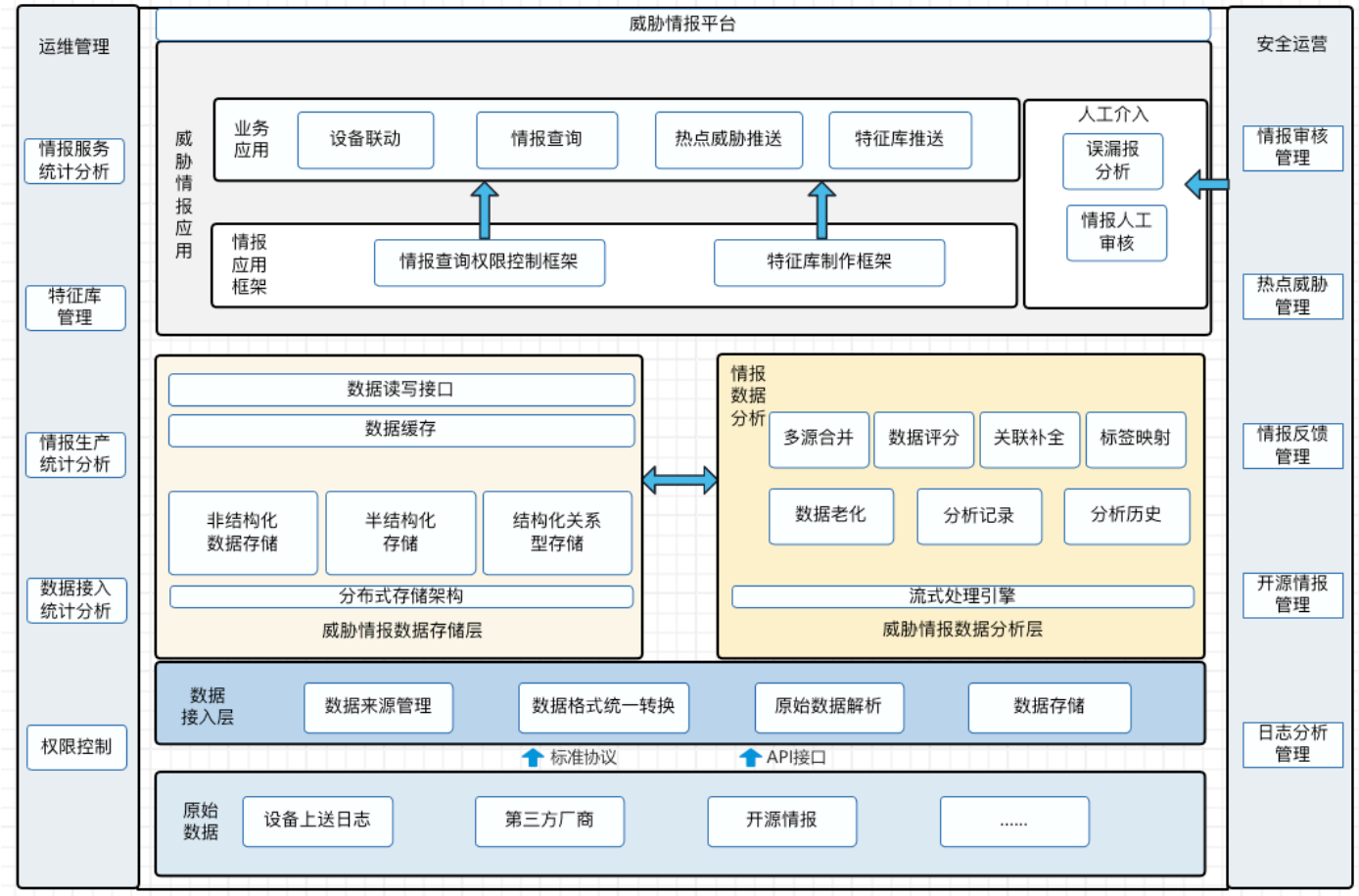
- **威胁事件溯源分析高位视角洞察威胁**

  山石云瞻威胁情报服务帮助管理员从海量告警信息中自动聚焦重点威胁，优先处理重点威胁，提升运维效率；同时通过与云端联动丰富威胁上下文信息，实现威胁事件溯源查询，及时应对关键威胁。

- **热点威胁情报主动推送提前一步防御威胁**

  山石云瞻威胁情报中心结合全球情报和热点威胁，第一时间将业界最高危的威胁事件情报主动推送到设备端，帮助用户聚焦高危重点威胁；提供安全事件详情，帮助用户了解设备的防护状态，提供针对性防护措施或者防护建议，实现快速响应；同时帮助用户了解企业资产是否已存在热点高危威胁，提供持续地威胁检测分析。

整体架构



威胁情报平台是一个集数据采集、整理、分析存储，并基于情报内容对外提供安全服务能力的平台。上图为威胁情报平台的整体架构图，其中主要包含威胁情报应用、威胁情报分析、威胁情报存储、威胁情报接入、安全运营以及运维管理六个部分。每个部分主要能力如下：

威胁情报应用：将情报内容进行分类整理，以设备联动、威胁推送、特征库推送等形式提供安全服务。

威胁情报分析：对收集到的情报进行深入分析，以识别复杂的威胁和关联。

威胁情报存储：对情报转换各个过程中的原始数据、中间格式数据、记录数据、情报数据等进行存储。

威胁情报接入：对内外部情报源、第三方情报源等进行数据收集、数据解析、格式转换等操作将情报来源的内容进行统一化。

安全运营：对情报来源进行管理和评估，对情报内容进行分析，对情报反馈进行处理，对情报更新进行控制。

运维管理：对情报提供服务、情报生产、数据接入等能力进行数据监控，以可视化数据面板来提供运维管理。

下面的章节会对上述的部分分别进行详细的功能设计介绍。

威胁情报应用

情报业务应用

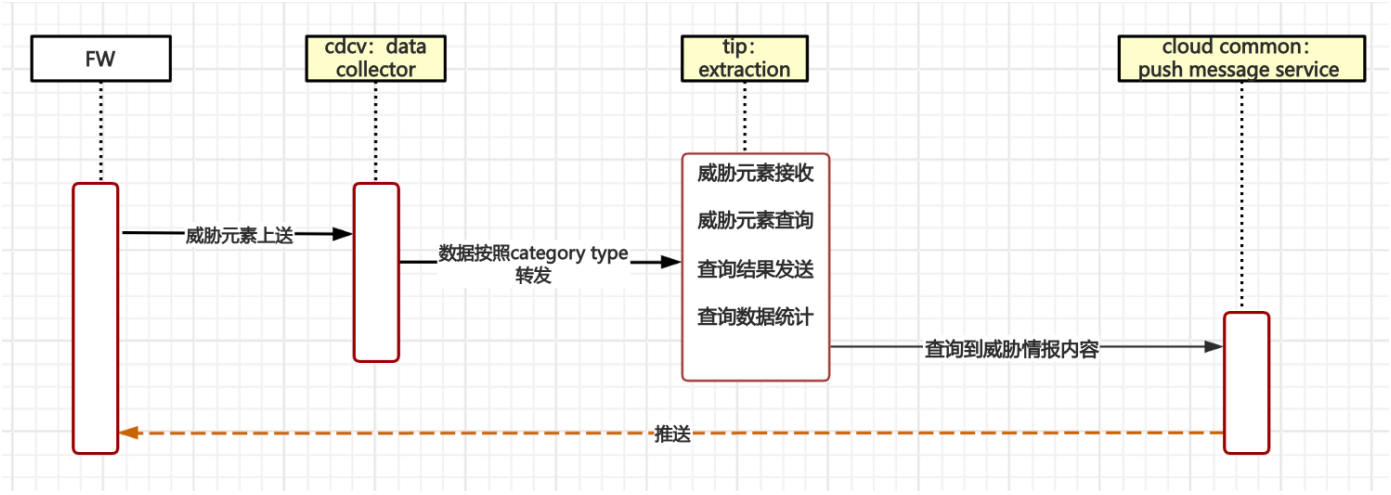
设备联动

设备联动是指设备与云平台之间建立通道，通过数据交换获取情报内容。

设备威胁中心接入

概要描述

设备威胁中心接入威胁威胁情报平台，通过上送威胁元素，查询威胁元素的情报内容，再由威胁情报平台将情报内容推送给设备，整体的数据流向图如下，其中data collector模块为设备对接模块，push message模块为中台公共模块，在此不做过多的介绍，tip:extraction为该功能在威胁情报平台上的实现模块。



由上图能够看出主要的流程有以下几个关键步骤

1. 威胁元素的接收
2. 威胁元素的查询和结果发送
3. 数据统计

下面对上述步骤进行详细介绍

威胁元素的接收

设备采用数据通道进行上送，数据分为定时上送数据和页面触发数据两种类型。使用schema中的from字段区分，from 为 1 表述定时触发，0 表示页面触发。

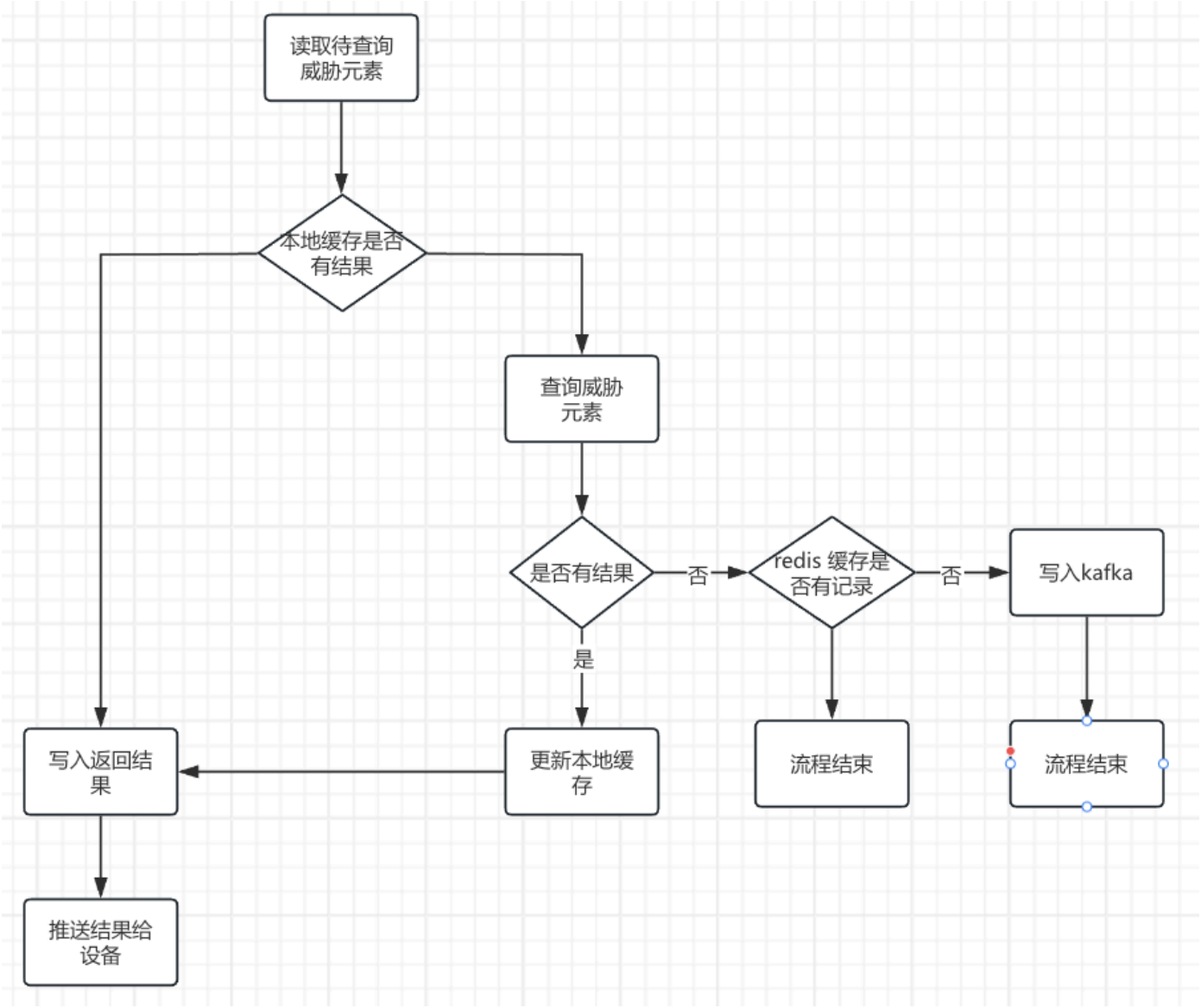
设备通过数据通道上送(http 1.1)的数据会被data collector模块接收，data collector模块接收后会将数据分发到 kafka 中，其topic的定义为 device_avro-{category}-{type}。

威胁元素的数据格式如下：

字段	值
category	tif
type	ip/domaon/url/md5
url	/1.0/data/avro/{category}/{type}/**

schema	{
	"type": "record",
	"namespace": "tifd_icloud",
	"name": "{type}*query",*
	"fields": [
	{
	"type": {
	"items": {
	"fields": [
	{
	"type": "string",
	"name": "element"
	},
	{
	"type": "int",
	"name": "from"
	}
	],
	"type": "record",
	"name": "tifd_icloud{type}*query"*
	},
	"type": "array"
	},
	*"name": "{type}*query"
	}
	],
	"doc": "Converted from 'tifd_icloud.xml' by xml2json.py"
	}

威胁元素的查询和结果推送



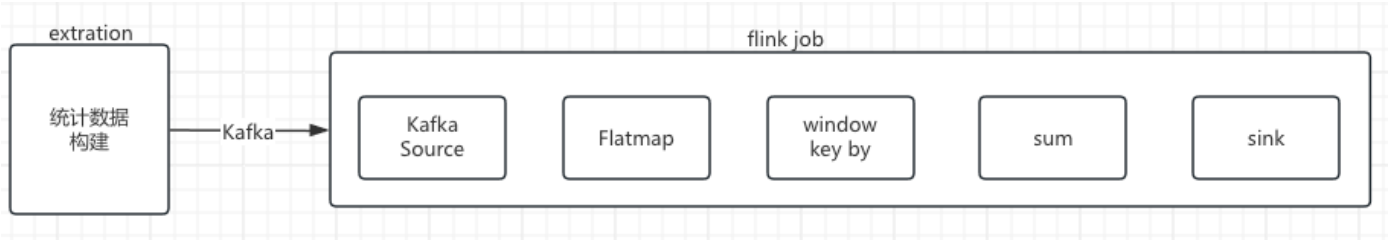
1. 本地缓存：为了尽量减少短时间内的重复查询，降低查询压力，需要设置元素状态缓存，分类缓存，每种类型10000 条，命中本地缓存的数据直接把情报内容写入返回结果。
2. 元素查询：没有命中缓存的数据，会从hbase表ti_ip/domain/file_md5中获取数据，查询过程为批量查询。如果能查询到结果，则将ioc和对应的结果更新进入本地缓存，如果没有结果的ioc则写入redis缓存中，待redis缓存触发淘汰机制时将没有结果的ioc发送入kafka去触发第三方查询。
3. 推送结果给设备：使用设备与云平台的控制通道将威胁元素的查询结果推送给设备，其中推送结果的形式时xml，格式如下：

```
<msg category="tif" type="element_result">
   <ip>
       <item>
           <element>1.1.1.1</element>
           <result>2</result>
           <tag_en>tag1,tag2,tag3</tag_en>
           <tag_cn>tag1,tag2,tag3</tag_cn>
           <tag_severity>1,2,3</tag_severity>
           <scenario_en>scenario1,scenario2</scenario_en>
           <scenario_cn>scenario1,scenario2</scenario_cn>
       </item>
   </ip>
</msg>
```

其中的tag为场景标签和分类标签的中英文，tag_severity为标签对应危险级别： 1(低风险) 2 (高风险) 3 (高风险) 4 (严重），result为 结果 0 (无结果) 1 (白名单) 2 (可疑) 3 (恶意)。

推送方式使用pushmessage 模块提供的推送能力，调用pushmessage的方式是调用 "/manage/push-message/device.message"接口。

数据统计



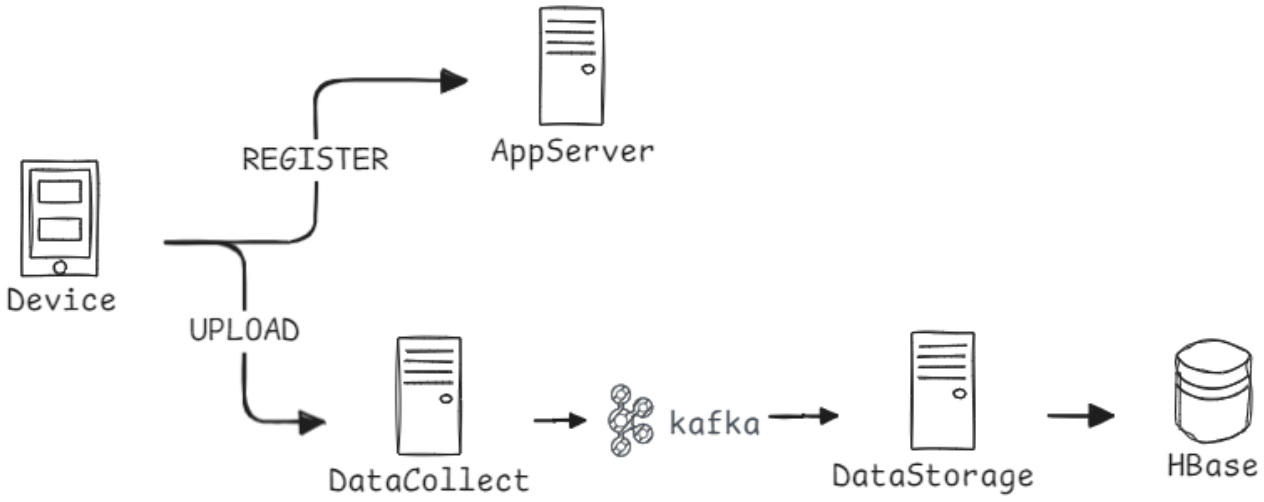
数据统计整理逻辑如上图所示，其中由extration模块构建统计数据并发送到kafka的topic中，由负责统计数据的flink job进行数据消费，转换、聚合、处理、入库这一系列的操作。此处只介绍统计数据的构建，统计数据的flink job的处理逻辑由运维管理-情报服务统计处进行详细说明。

统计数据共有两种：

1. 设备定时上送类型的威胁元素统计
   kafka topic：ti-query-timer ，统计数据的形式为<String sn,String type,String datas>
   其中datas 是多个威胁元素之前用逗号分隔的字符串，例如：1.1.1.1, 2.2.2.2, ……
   type类型： ip/domain/md5
2. 用户触发类型的威胁元素统计
   kafka topic：ti-query-ui，统计数据的形式为<String sn,String type,String datas>
   其中datas 是多个威胁元素之前用逗号分隔的字符串，例如：1.1.1.1, 2.2.2.2, ……
   type类型： ip/domain/md5

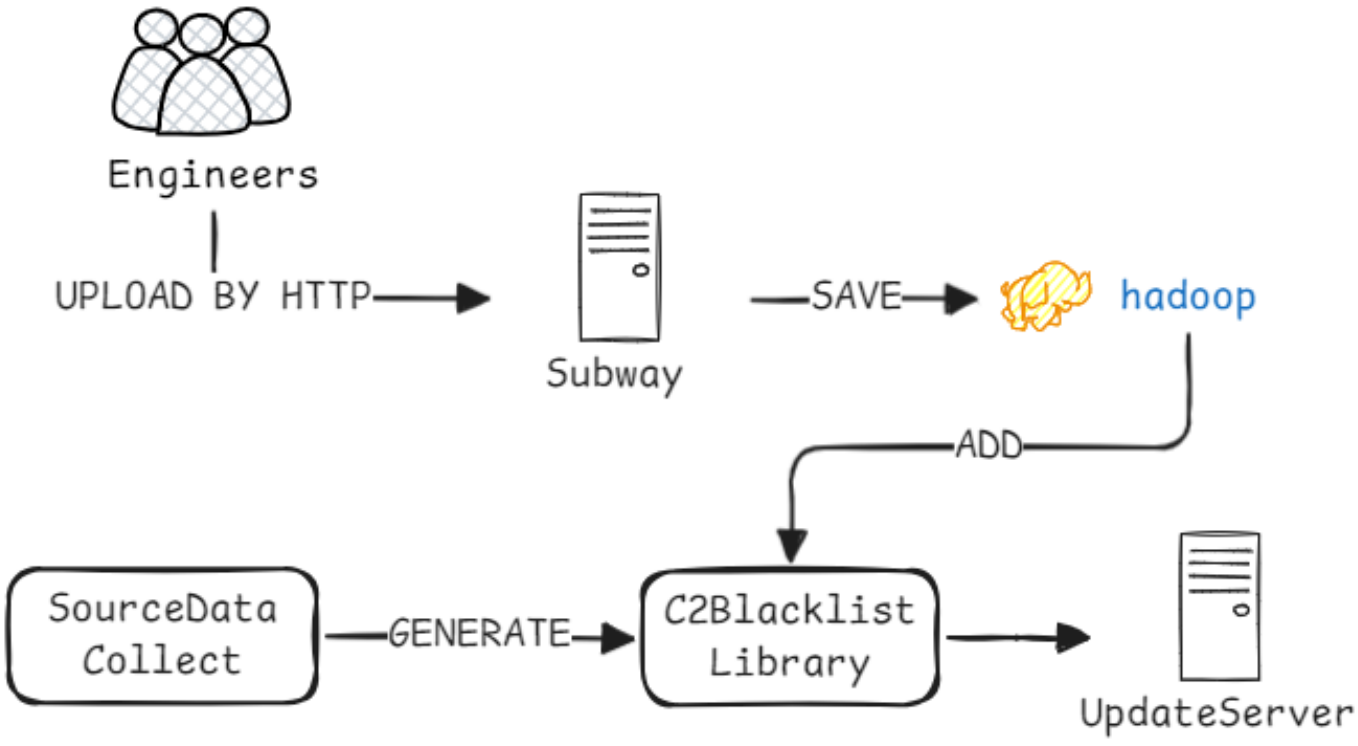
设备 DNS 安全接入

数据收集



1. 功能
   数据湖支持存储 NXdomain 元数据、DGA 检测结果
2. 实现
   设备注册至 AppServer 后通过数据通道上送 NXdomain 元数据、DGA 检测结果，直接存储至数据湖中供算法工程师获取

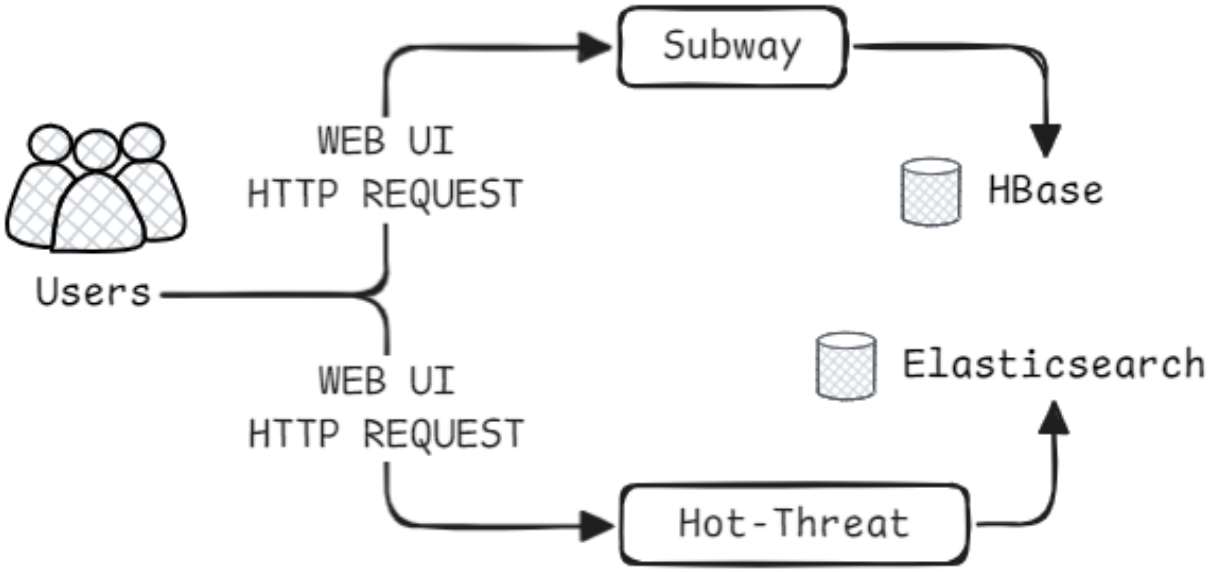
智能模型下发



1. 存储算法工程师上传的模型文件并提供记录，同时对文件 MD 5 进行校验
2. 将模型文件打包至 C 2 黑名单库中, 提供给 UpdateServer 进行加密下发

情报查询

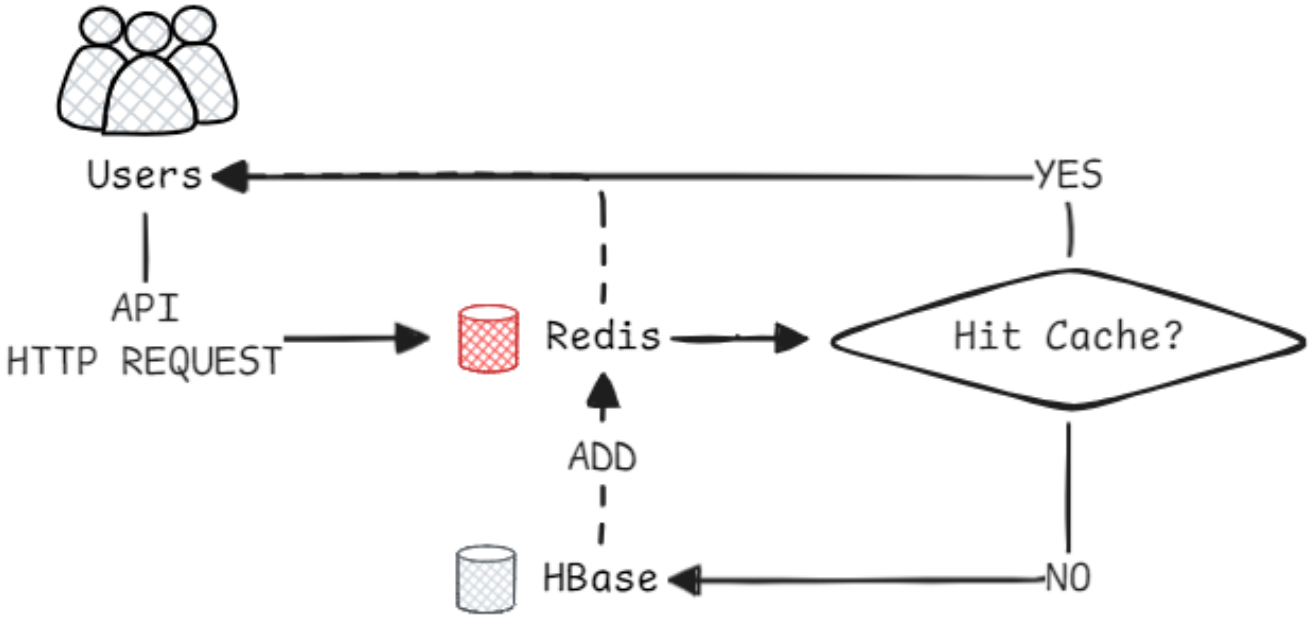
WEB 查询服务



接口	功能
GET,/ti/report/query-report-direct	威胁情报报告基础及其关联信息
GET,/ti/report/cite/source	页面获取聚合情报（情报源数据）
GET,/ti/report/link/source	设备跳转到目标源网页
GET,/ti/report/osintelligence	查询开源情报结果
GET,/ti/report/tag-info	获取恶意家族和攻击团伙标签的详细信息
GET,/ti/report/field-info	获取部分字段解释
GET,/ti/report/popularity-ranking	查询域名的流行性排名
GET,/ti/report/thintelligence	获取 IOC 整合历史
GET,/ti/report/cclass-ip	获取 IP 关联到的 C 段 IP

1. 功能（位于云瞻首页）
    1. 直接提供各类型 IOC（IP、DOMAIN、FILE、URL）的威胁情报查询功能，用户可直接输入查询，匿名用户也可
    2. 点击查阅热点威胁事件的 TOP 10，需登录云瞻账号
2. 实现
    1. 情报查询相关接口符合/ti/report/{}的格式，通过 Subway 模块查询 Hbase 的高性能查询层数据
    2. 从 ES 中获取 TOP 10 的热点威胁事件及其详细报告

API 查询服务



API 服务	功能	接口	说明
信誉接口	IP 信誉	/api/ip/reputation	此接口可提供一些关于此 IP 的基础信息，可用于识别此 IP 的威胁级别。
	Domain 信誉	/api/domain/reputation	此接口可提供一些关于此域名的基础信息，可用于识别此域名的威胁级别。
	File 信誉	/api/file/reputation	此接口可提供一些关于此文件的基础信息，可用于识别此文件的威胁级别。
	Url 信誉	/api/url/reputation	此接口可提供一些关于此 Url 的基础信息，可用于识别此 Url 的威胁级别。
高级接口	IP 高级	/api/ip/detail	此接口可提供一些关于此 IP 的高级信息，可用于识别此 IP 的威胁级别, 关联关系, 相关域名信息等。
	Domain 高级	/api/domain/detail	此接口可提供一些关于此域名的高级信息，可用于识别此域名的威胁级别, 关联关系, 相关域名信息等。
	File 高级	/api/file/detail	此接口可提供一些关于此文件的高级信息，可用于识别此文件的威胁级别, 关联关系, 相关域名信息等。
	Url 高级	/api/url/detail	此接口可提供一些关于此 Url 的高级信息，可用于识别此 Url 的威胁级别, 关联关系等。

1. 功能
   TIP 平台为设备和用户提供情报查询的 API 服务，方便不同的引擎对接。
   根据情报利用的需求，提供必要的 API 服务，其中 API 提供的能力与 TIP 平台现有的能力相同。
   所有注册用户免费使用，您只需要注册山石云平台并获取 API 密钥即可，其中有一些功能仅限 API 正式版客户才能使用。
   可通过查看 API 帮助文档页面查看相关信息。

2. 实现
    1. 数据来源于 TIP 高性能查询层（HBase）中
    2. 在高性能查询层之前添加 Redis 缓存，对于频繁查询的 IOC 数据能提供更快的相应速度

威胁推送

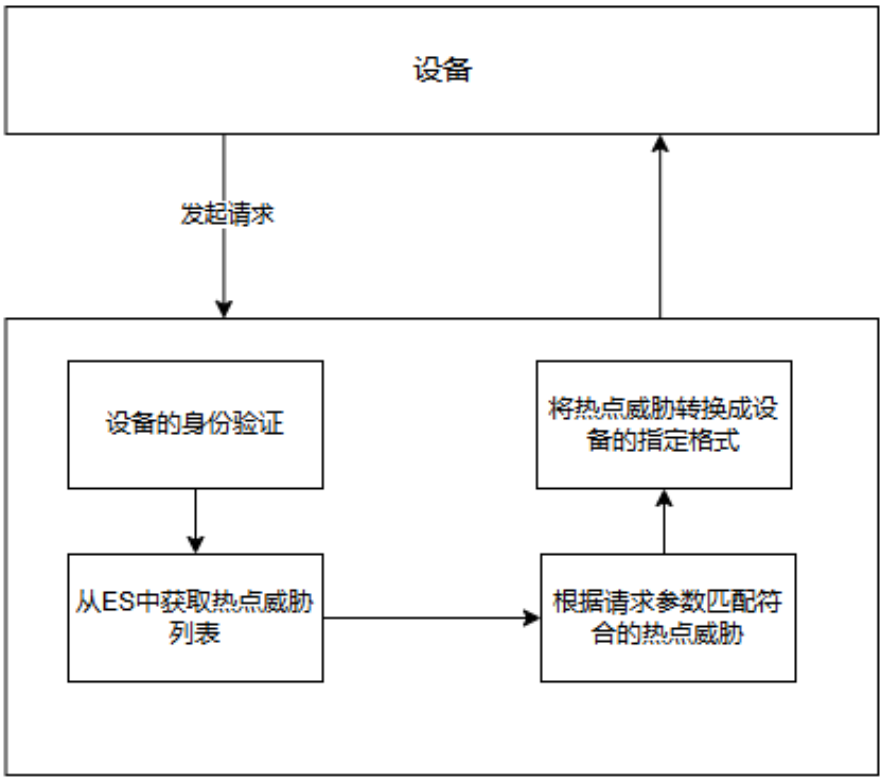
威胁推送是指IPS、WAF以及智源等设备向TIP发起请求时，TIP根据其产品系列、设备型号以及镜像版本返回对应支持的热点威胁列表。TIP的热点威胁情报中心维护了影响范围大，后果严重的安全漏洞、病毒以及威胁事件等，并附带了解决建议，用户可以通过请求TIP来获取最新的威胁热点。同时，云端可以临时向设备推送监管数据、重大威胁数据。TIP的热点威胁情报数据存储在ES中，每周都会有运维人员获取近期的热点威胁并进行更新。

设备型号的动态添加与存储

对于热点威胁的的匹配规则而言，设备的型号由产品系列、平台型号、设备系统版本三部分组成，每个型号都有对应的av以及ips特征库版本，对应数据存储在Zookeeper中。TIP后台提供页面接口供运维人员来管理设备型号的对应逻辑。

热点威胁事件推送

设备会主动向云端发起请求，要求推送当前最新的热点威胁，云端收到请求后根据设备的型号、系统版本以及授权信息来推送符合要求的热点威胁。威胁的推送过程如下：

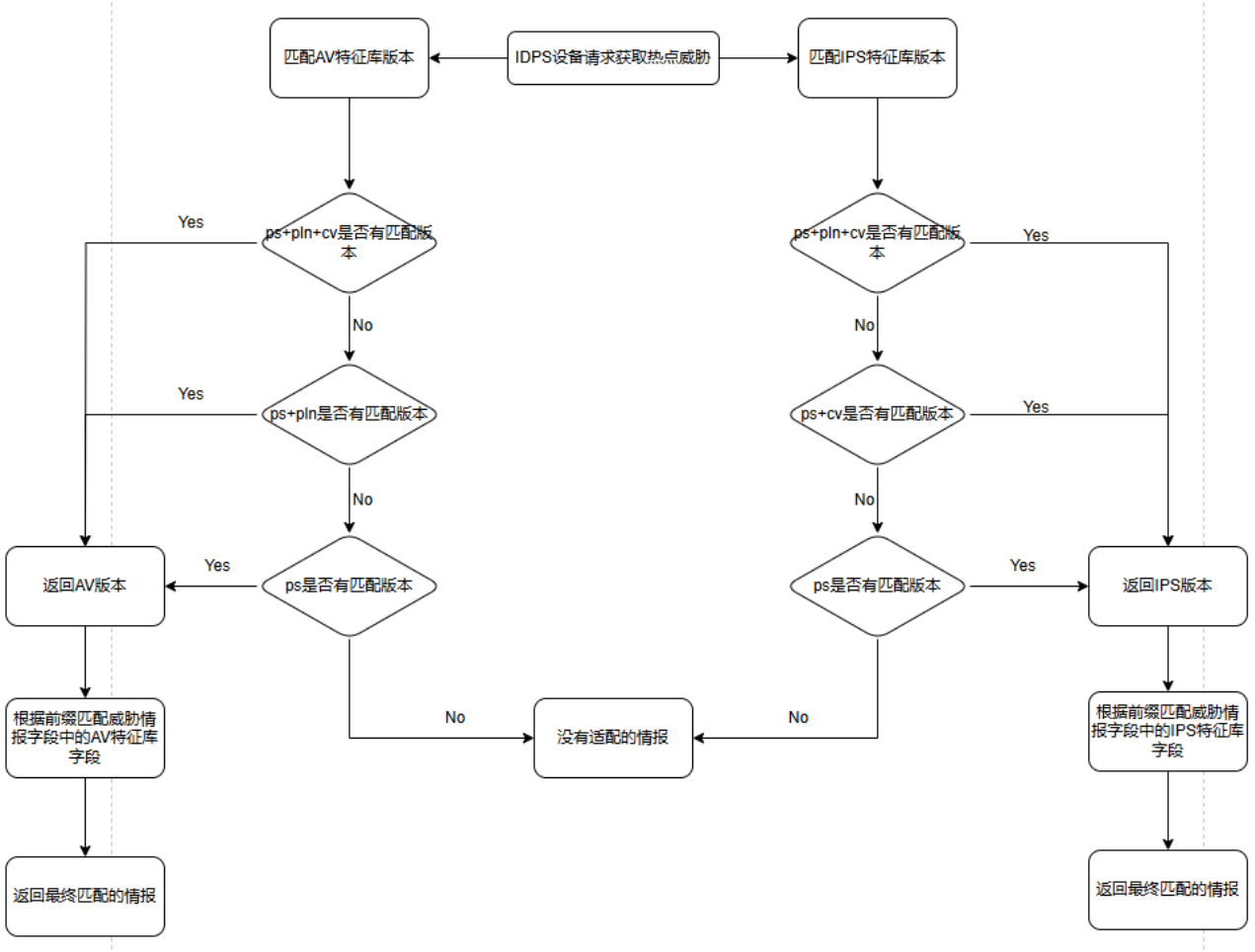


威胁匹配逻辑

目前云瞻接受三种设备的热点威胁请求，三种设备的获取威胁的逻辑不同：

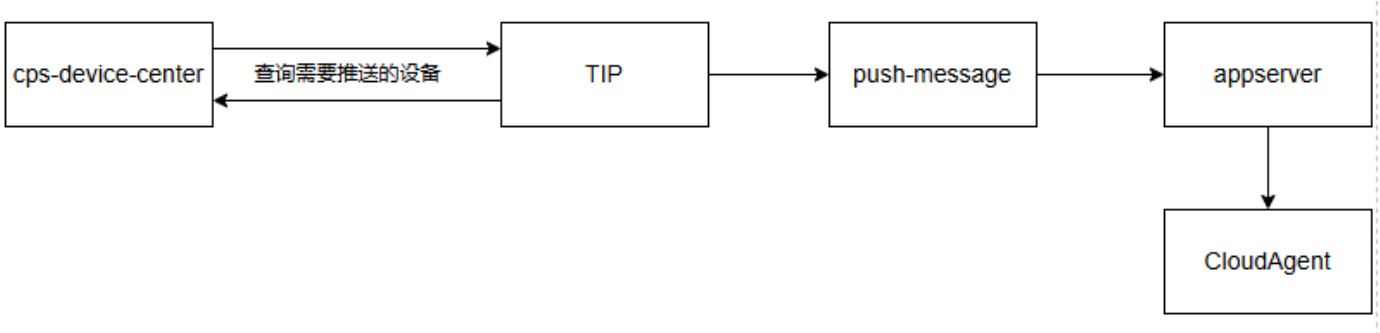
- 智源：直接返回ES索引中适用于智源的威胁
- WAF：直接返回ES索引中适用于WAF中使用智源的威胁

- IDPS：根据请求中的附带的产品系列、设备型号、镜像版本来获取对应的特征库版本，根据特征库版本来匹配对应的热点威胁
  其中IDPS设备请求时获取对应特征库版本的具体匹配逻辑如下：



监管数据推送

由云端按需主动进行推送，主要应用于一些特殊状况，例如监管单位通报了一些高风险站点，云端迅速将该数据下发。由此方式推送的数据，云端不会考虑防火墙设备的Capacity，由设备本身的能力来按需加载。推送功能的发起在后台页面进行，运维人员在页面上传文件，后台程序在处理后将文件内容推送给在线且具有c2功能授权的设备，版本限制为StoneOS5.5R9F1.10之后。推送的流程如下：



上传文件为txt，内容格式示例为：

```
baidu.com 0
google.com 1
bilibili.com -1
```

其中0代表白，1代表黑，-1代表未知。

特征库推送

概要描述

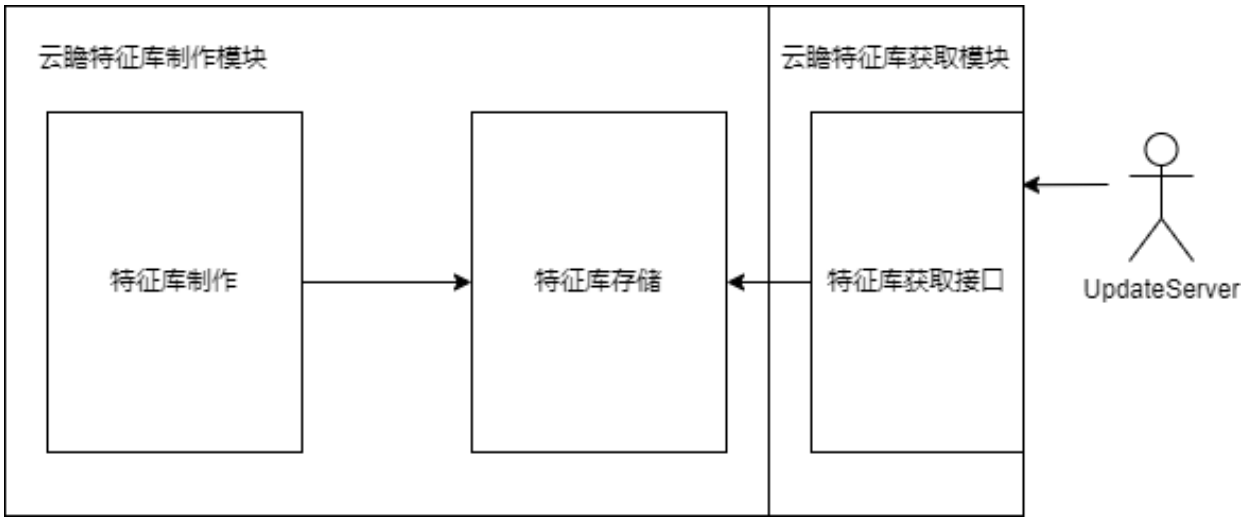
云瞻制作完成的特征库，如智源特征库、智影特征库、C2特征库等，需要以明确的形式给updateserver方提供获取支持。

云瞻以提供API接口的形式，支持updateserver方获取特征库文件。

功能设计

总体架构

总体逻辑示意图如下：



在“云瞻特征库制作模块”中，按照对应逻辑制作好的特征库文件，会进行存储。“云瞻特征库获取模块”对不同类型的特征库分别提供API接口，允许外部updateserver方进行访问下载。

不通特征库获取的API接口有所不同，分别由不同的团队分开调用。接口访问需要云瞻进行授权。

updateserver 获取到云瞻提供的原始特征库后，会基于原始的特征库采取版本切分、数据量截取等操作，最终产出设备可直接加载的特征库，由设备进行获取更新。

特征库获取接口

- 普通特征库

本章节列出普通特征库的获取接口信息，普通特征库是后续需要特殊说明的特征库之外的所有特征库：

特征库类型	接口方法	接口名	
智源特征库	GET	/api/library/isource	
C2白名单	GET	/api/library/c2/whitelist	
C2黑名单	GET	/api/library/c2/blacklist	
AV特征库	GET	/api/library/av	
安天定制AV特征库	GET	/api/library/customized_av	
IP信誉库（IPR）	GET	/api/library/ipr	

- ATT&CK离线库

ATT&CK离线库是云瞻维护的ATT&CK知识库的业务派生。为了设备端使用方便，将相关的知识内容以文件的形式进行产出，设备获取后即可加载对应版本的ATT&CK知识信息。

其获取方式仍为接口获取，但是对不同的使用团队做出区分：

使用方	接口方法	接口名	
StoneOS团队	GET	/api/library/attck	
智源团队	GET	/api/library/attck/isource	

需要特别说明的是，对于沙箱设备，云瞻直接提供可用的ATT&CK离线库包，即云瞻为沙箱设备获取ATT&CK离线库的updateserver。此时由于特征库的制作和updateserver处对原始库处理的逻辑均在云瞻，两处直接不再直接提供API接口交互，云瞻直接提供设备的获取接口。相关接口如下：

使用方	接口方法	接口名	
沙箱设备	GET	/signature/attck/download	

- 智影特征库

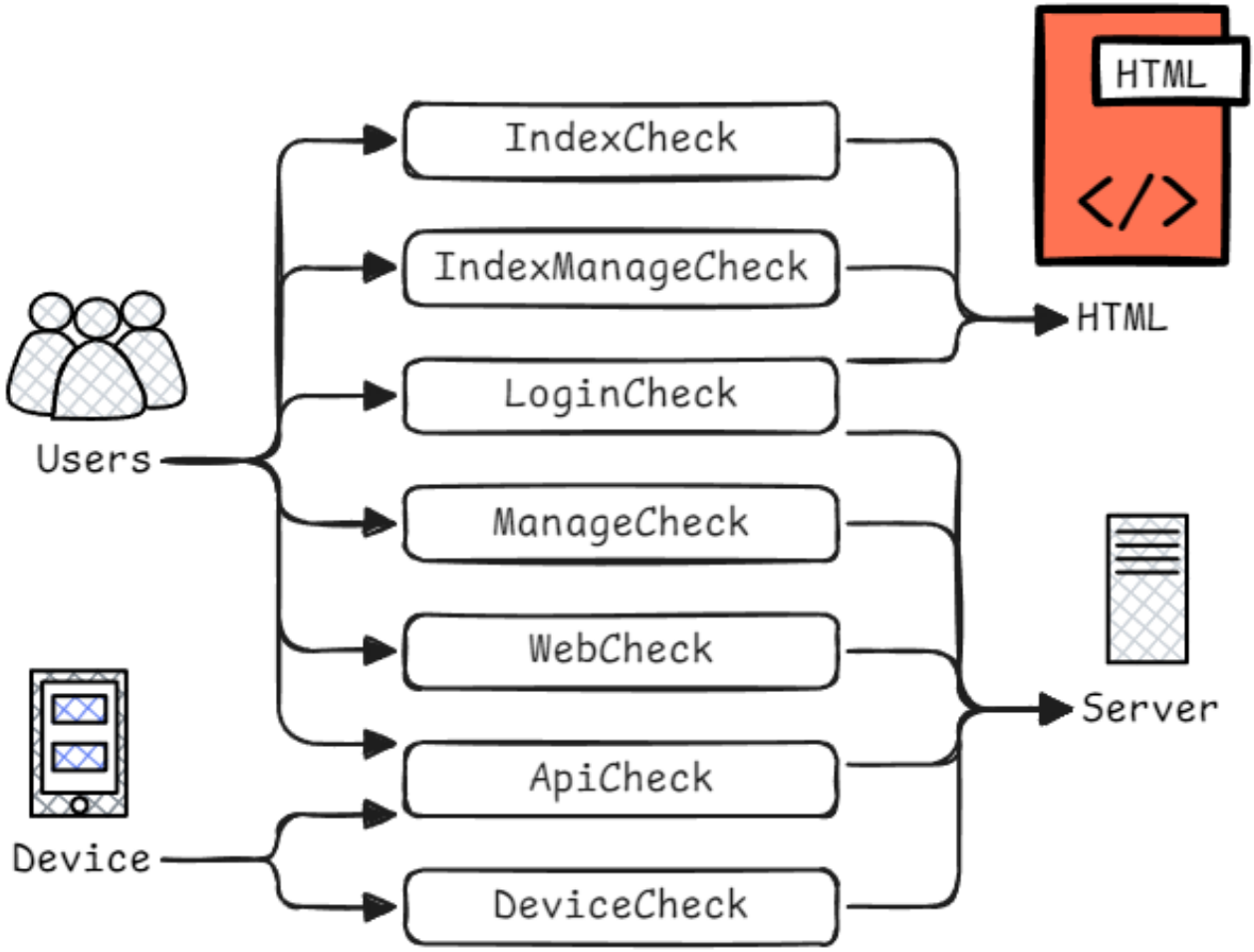
对于智影特征库而言，云瞻既为智影特征库的制作方，制作原始的智影特征库；同时也负责面向沙箱设备，充当智影特征库的updateserver角色，对原始智影特征库进行处理，提供给设备下载。

此时由于特征库的制作和updateserver处对原始库处理的逻辑均在云瞻，两处直接不再直接提供API接口交互，云瞻直接提供设备的获取接口。相关接口如下：

使用方	接口方法	接口名	
沙箱设备	GET	/signature/library/ishadow/download	

情报应用框架

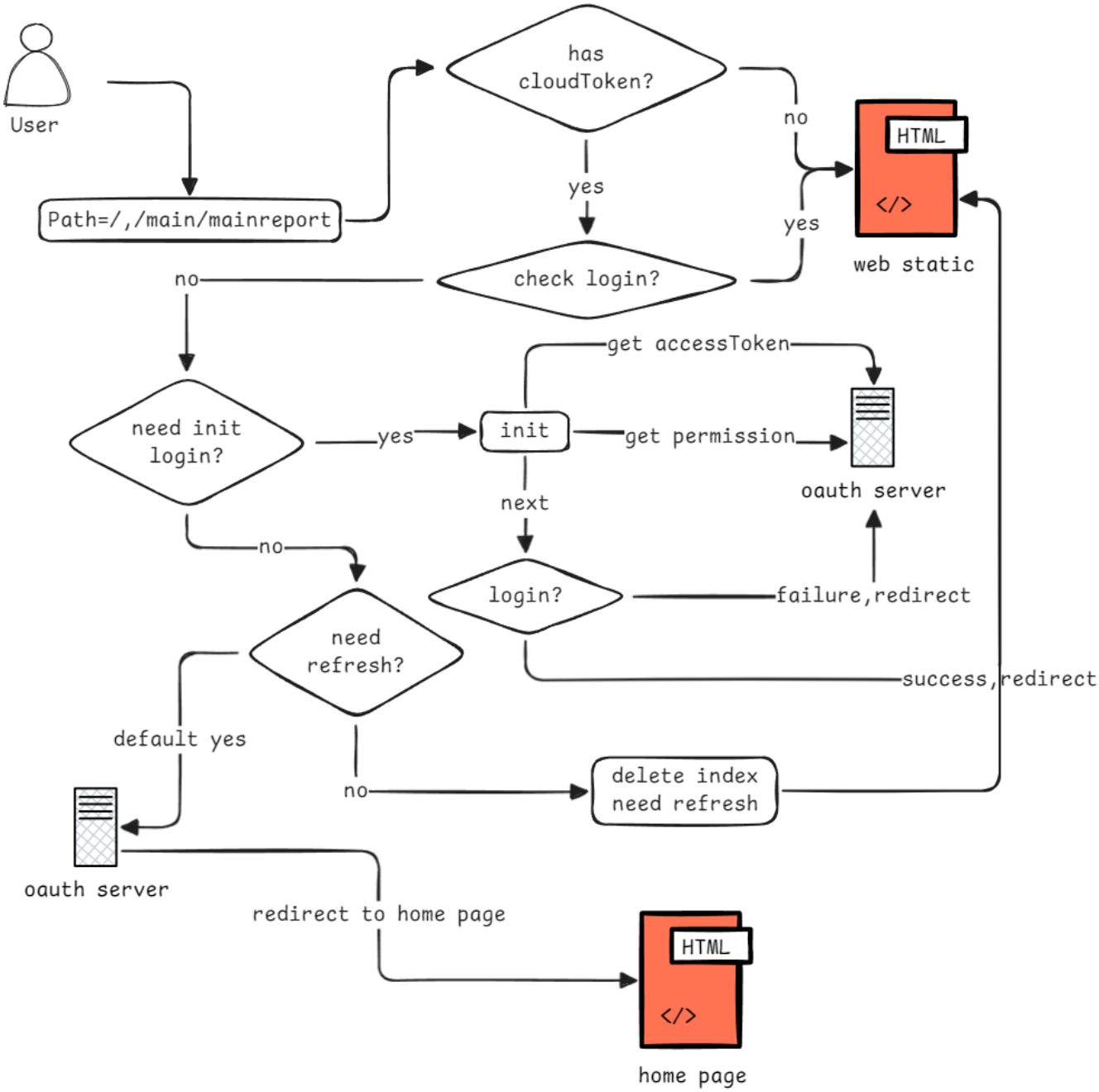
权限控制框架



> TIP 的权限控制目前有以下几种类型：
>
> 1. 页面路由权限校验
>   1. IndexCheck 云瞻首页页面权限校验
>   2. IndexManageCheck 云瞻管理页面权限校验
> 2. 登录状态信息校验
>   1. LoginCheck 登录状态信息校验
> 3. 接口权限校验
>   1. WebCheck Web 权限校验
>   2. ManageCheck Manage 权限校验
>   3. ApiCheck Api 权限校验
>   4. DeviceCheck Device 权限校验（目前留空，可扩展）

页面路由权限校验

云瞻首页页面权限校验

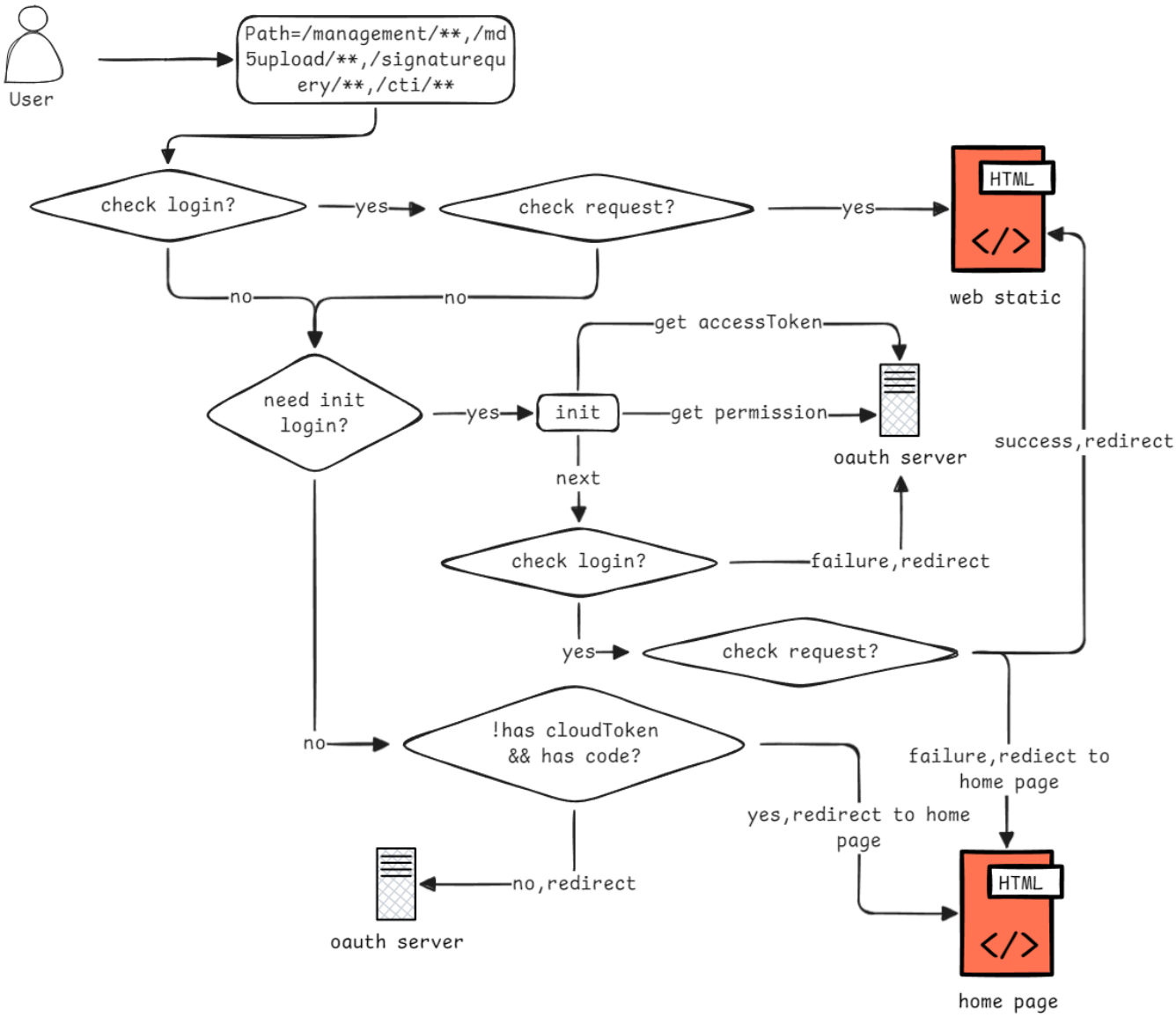


对于上述流程图的补充说明

1. 该流程针对页面路由为 `Path=/,/main/mainreport` （即云瞻首页）的请求
2. Has cloudToken 的含义
    1. 当从 Oauth 服务登录成功跳转回来后，正常情况下 cookie 中会放入对应的该字段和值，所以先通过是否有用该字段来判断是否已登录（值和登录状态不一定合法，这一点需要注意）
    2. 同时，cloudToken 也可能在 oauth server 已经过期，所以需要尝试登录
    3. 首页允许所有用户（包括匿名用户）的访问，当用户未登录，直接定向至前端文件中即可
3. Check login 过程
    1. 首先从 cookie 中取出 sessionToken 来获取 TIP 登录的相关信息，若不存在则认为 TIP 未登录，然后校验 cookie 中的 cloudToken 与 username 与 TIP 维护的是否一致，一致则认为登录状态正常
    2. PS. SessionToken 是 TIP 的当前登录会话的唯一标识，需要通过该值来获取一些登录后的权限，从 TIP 的角度看，可以草草的将 cloudToken 当作 oauth server 的登录会话标识（实际上并不是，方便理解）

4. Need init login 的含义
   在 TIP 本地校验不通过的情况下，同时判断 cookie 中是否存在 cloudToken 与 query param 中是否存在 code 字段，若同时存在，则认为该请求是从 oauth server 登录后重定向回来的，后续在 TIP 尝试登录（即 init 操作）
5. Init 操作
   1. 通过 code 去 oauth 获取 accessToken，然后通过 accessToken 去 oauth 获取 peimission id 列表，用来加载 TIP 的权限信息
   2. 若成功则直接定向至前端文件中
   3. 失败（code 不对，或 oauth 权限过期等）则跳转至 oauth server 登录页面，让用户重新登录
6. Need refresh 的含义
   1. 存在的意义
      当用户在 oauth 已登录的情况，通过刷新来自动完成 TIP 权限会话信息的加载，避免用户的重复跳转登录，优化用户体验
   2. 刷新条件
      在 has cloudToken（可能在 oauth 已经登录了）但是 check login 失败（TIP 权限未加载或信息不对应）且参数里没有 code（不是从 oauth 登录跳转回来的） 的情况下
   3. 执行操作
      1. 从 cookie 取出 index_need_refresh，若不存在则为 true
      2. 若为 false 则删除该字段，正常重定向至前端文件；若为 true 则将值置为 false，然后访问 oauth 进行登录尝试，不跳转页面，若成功则重定向回来的 query param 中带 code，重新进行 TIP 登录，若失败，则直接重定向为云瞻首页

云瞻管理页面权限校验



对于上述流程图的补充说明（大部分在上述 IndexCheck 已介绍）

1. 该流程针对页面路由为 `Path=/management/**,/md 5 upload/**,/signaturequery/**,/cti/**` （即云瞻管理页面）的请求
2. PS. 一些需要注意到点
   注意，code 存于 query params 中

   在 oauth 进行登录后，重定向回来时会多一个 code 的请求参数，通过该参数值我们可以去 oauth 获取到登录用户的相关信息\
   包括用户 ID，用户名，其所拥有的角色 ID

   需要额外注意的是，当用户在 oauth 中进行登录之后跳转回 tip，用户手动删除 cookie 中的 cloudToken\
   此时 tip 的登录信息丢失，但是跳转去 oauth 进行登陆时，oauth 的会话信息还存在，会带 code 参数重定向回来

   但是，但是，但是，oauth 并不会判断当前请求的 cookie 中是否丢失 cloudToken，重定向回来的时候不会去判断并重新写入该值

   Oauth 并不通过 cloudToken 判断是否已经登录，但是 tip 判断用户是否在 oauth 进行登录需要通过 cloudToken\
   （同时，cloudToken 为 oauth 写入，不能在其他服务写入）

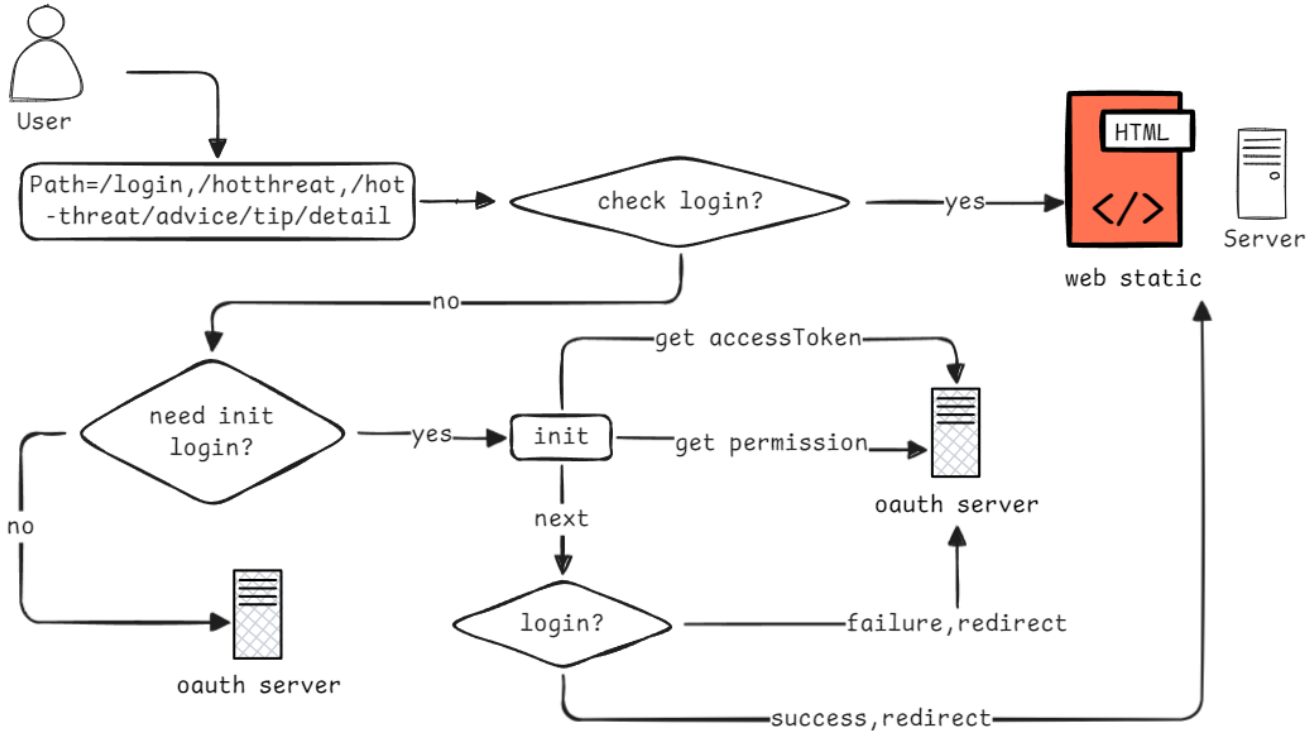
目前，我所采用的方法是，从 oauth 登录重定向回来的带 code 的请求，若只包含 code 不包含 cloudToken，则重定向回 tip 主页\
当用户试图再次进入例如/management 等需要登录权限的页面时，会进行上面的两次跳转后，重定向回主页\
直至该用户在 oauth 的登录会话过期（例如清理浏览器缓存或删除 user. Hillstonenet. Com. Cn 的 cookie 信息），则用户可以重新进行正常登录

除此之外，用户也可以通过手动输入/logout 的方法进行手动登出，之后也可正常登录

可能的优化方法是，出现上述情况时，服务内部调用/logout 请求后，重定向到 oauth 登录页面让用户进行重新登录\
但是考虑到，用户进行手动删除 cookie 的危险操作，重定向回主页并且出现无法登录的问题，也应该符合其操作的后果

登录状态信息校验

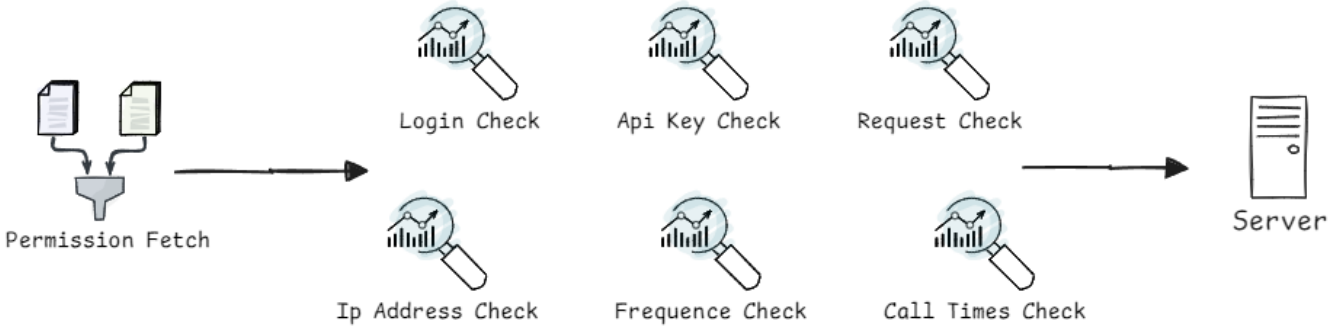
> 该校验地位比较特殊（过程还是比较清晰的），即可能包含页面请求，也可能包含接口请求，所以单独列举



对于上述流程图的补充说明

1. 该流程针对页面路由为 `Path=/login,/hotthreat,/hot-threat/advice/tip/detail` （只需要登录状态的请求）的请求\
   其中 `/login,/hotthreat` 为页面请求， `/hot-threat/advice/tip/detail` 为接口请求
2. 请求结果可能为页面也可能为接口返回的数据

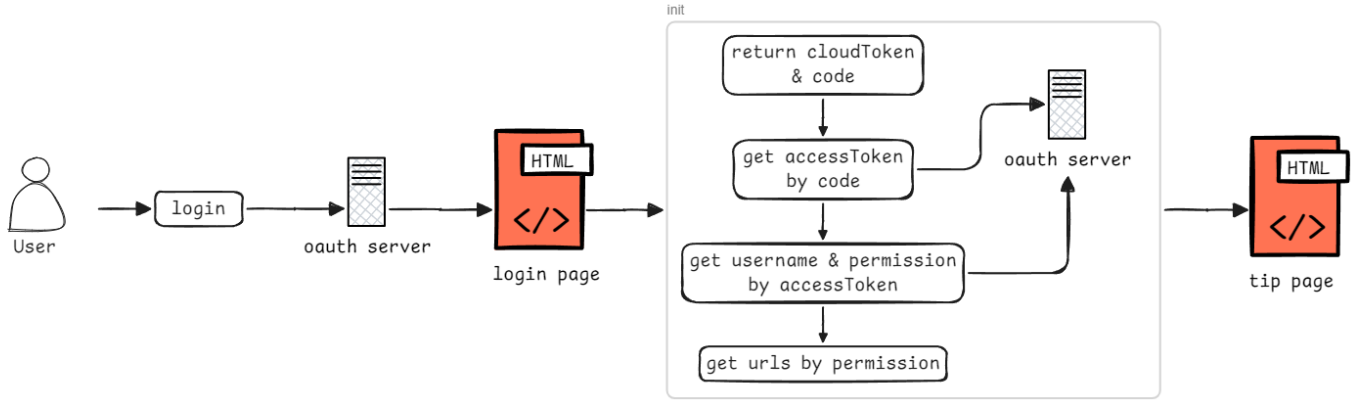
接口权限校验



接口权限的校验，抽象出来总共分为上述 7 步，对于不同的场景，具体的实现会有区别

1. 权限信息的获取，可能包括
    1. 其用户名
    2. 授权开始及结束时间
    3. 授权角色
    4. ApiKey
    5. 所拥有权限下所有资源组下的接口及其数量限制
    6. 资源组接口的请求数量限制
    7. 数量限制计算规则，资源组限制还是单个接口限制
    8. 允许请求的 Ip 地址
    9. 请求频率相关信息
2. 登录校验
   与上述页面登录的校验过程一致
3. Api Key 校验
   Api 接口请求规范需要添加对应的 Api Key 至请求 header 中，该步骤校验该 Key 是否合法（及是否存在）
4. 请求校验
   查询该用户是否拥有其请求接口的权限
5. Ip 地址校验
   若设置了该选项，则只允许设置的 Ip 地址（范围）的请求
6. 频率校验
   接口的频率限制，主要针对 ApiCheck 请求和 WebCheck 请求
7. 请求次数校验
   根据设置的校验规则，计算其访问次数是否超过限制

Manage 权限校验



上述权限的获取依赖于以下数据

hsm_db.t_oauth_client_permission

client to permission id

hsm_db.t_oauth_user_client_permission

user to permission id

tip_db.t_ti_role

role id == permission id

tip_db.t_ti_permission

role id to urls

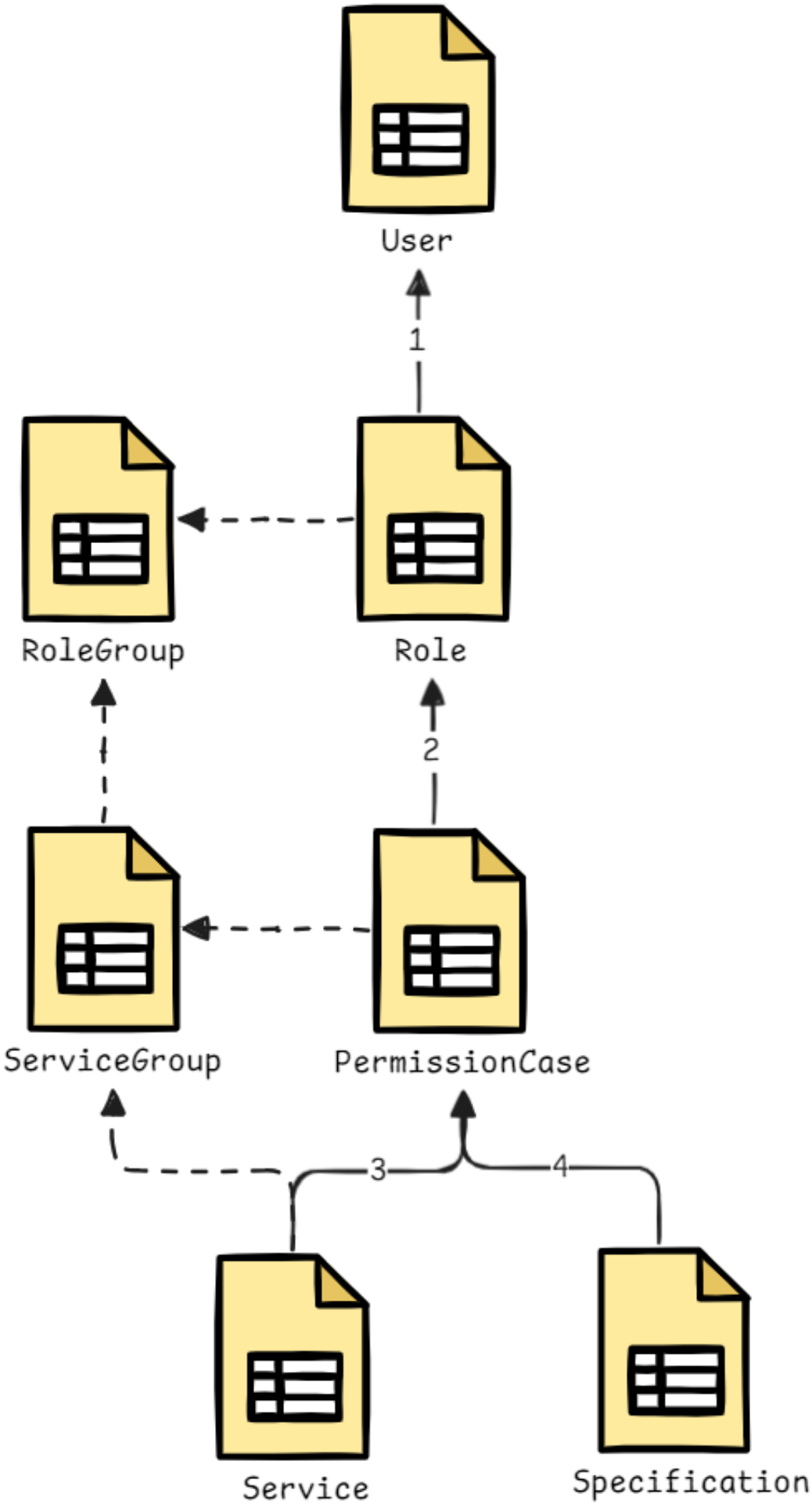
对于上述流程图的补充说明

1. 在 oauth 页面登录后会返回 code，可通过 code 去 oauth server 获取 accessToken，再通过 accessToken 获取该角色对应的 username 和 permission id 列表
2. TIP 权限加载分别根据 username 获取相关用户信息，根据 permission id 列表加载其所有可访问的 url 信息
3. 对于 Manage 所有的接口无次数和频率的限制

目前 TIP Manage 可控制的权限角色如下

id	description
2	查看运营统计数据
3	情报管理
4	特征库管理
5	av 测试特征库
6	威胁特征库对外查询
7	CTI 平台管理
8	CTI 平台 Schema 读权限
9	CTI 平台 Schema 写权限
10	CTI 平台 Data 读权限
11	CTI 平台 Data 写权限

API 权限


User
1
RoleGroup
Role
2
ServiceGroup
PermissionCase
3
4
Service
Specification

Api 接口请求

1. 每个用户注册成功后都会拥有一个 Api Key
2. Api 接口通过在 Header 中添加 X-Auth-Token 来进行请求

权限加载的流程说明

1. 说明
    1. User：基础相关信息
    2. Role：角色信息，属于某个角色组（Role Group）
    3. PermissionCase：资源配置方案，属于某个资源组（Service Group）
       资源组属于某个角色组
    4. Service：具体的 url 信息，属于某个资源组（Service Group）
    5. Specification：资源配置规格，限制数量和频率
       独立，只关联资源配置方案
2. 用户关联角色，即线 1
    1. 依赖于角色组
    2. 每个角色组最多选择一个角色进行关联
    3. 即每个用户可以最多关联**等同于角色组数量**的角色
3. 角色关联资源配置方案，即先线 2
    1. 依赖于角色组和资源组，角色组和资源组为一对多的关系，不可交叉
    2. 每个资源组最多选择一个资源配置方案进行关联
    3. 即每个角色可以最多关联**等同于该角色所在角色组下资源组数量**的资源配置方案
4. 资源配置方案可选的 url，即线 3
    1. 依赖于资源组
    2. 即每个资源配置方案可以最多关联资源组下的所有的 url
5. 资源配置方案的规格，即线 4
   直接关联即可

以下为目前 TIP 的角色组和资源组展示

Role Group 展示

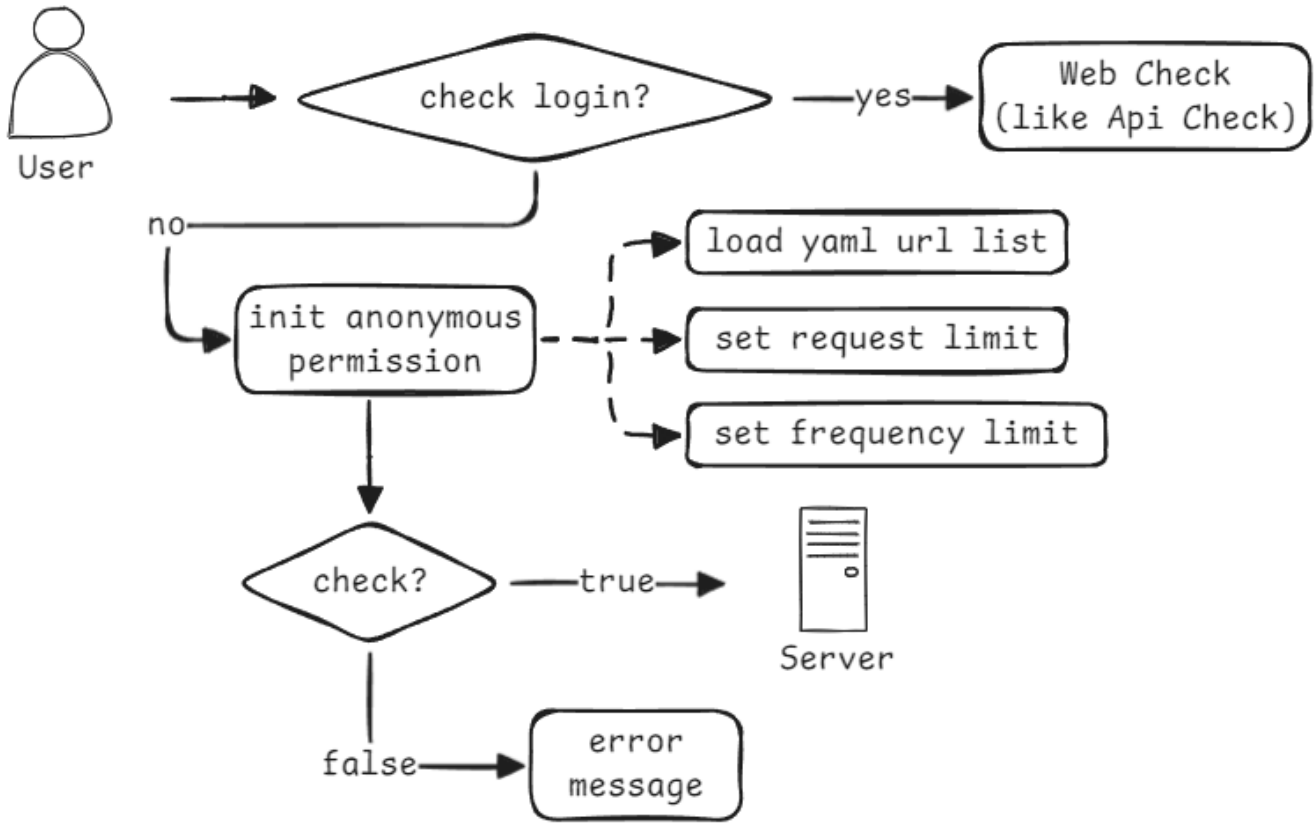
id	group_name	parent_id	description
1	USER	NULL	用户角色组
2	OPERATOR	NULL	运维角色组

Service Group 展示

id	group_name	role_group_id	parent_id	description
1	api	1	NULL	api 类型服务资源
2	web	1	NULL	web 类型服务资源
3	sandbox	2	NULL	沙箱查询服务资源
4	isource	2	NULL	智源查询服务资源
5	management	2	NULL	后台运维资源
6	av	2	NULL	av 特征库服务资源
7	c 2 library	2	NULL	c 2 库下载资源
9	web-unlimited	1	NULL	web 类型不计数资源
11	extra-library	2	NULL	特征库无需分类的额外的资源
13	ipr-library	2	NULL	ip 信誉库下载资源
15	vulnerability	2	NULL	漏洞知识库资源组
17	attck-library	2	NULL	ATTCK 离线库 api 资源
19	dga-model	2	NULL	dga 模型相关资源组
21	vulnerability-library	2	NULL	漏洞知识库下载资源组
23	reputation	1	NULL	设备信誉查询资源组
25	quota	1	NULL	设备信誉配额查询资源组
26	iot	1	NULL	iot 交互相关资源组

Web 权限校验

> 登录后的整体校验过程与基本与上述 ApiCheck 一致，区别在于匿名访问的接口限制，且增加了频率限制和图片验证



对于上述流程图的补充说明

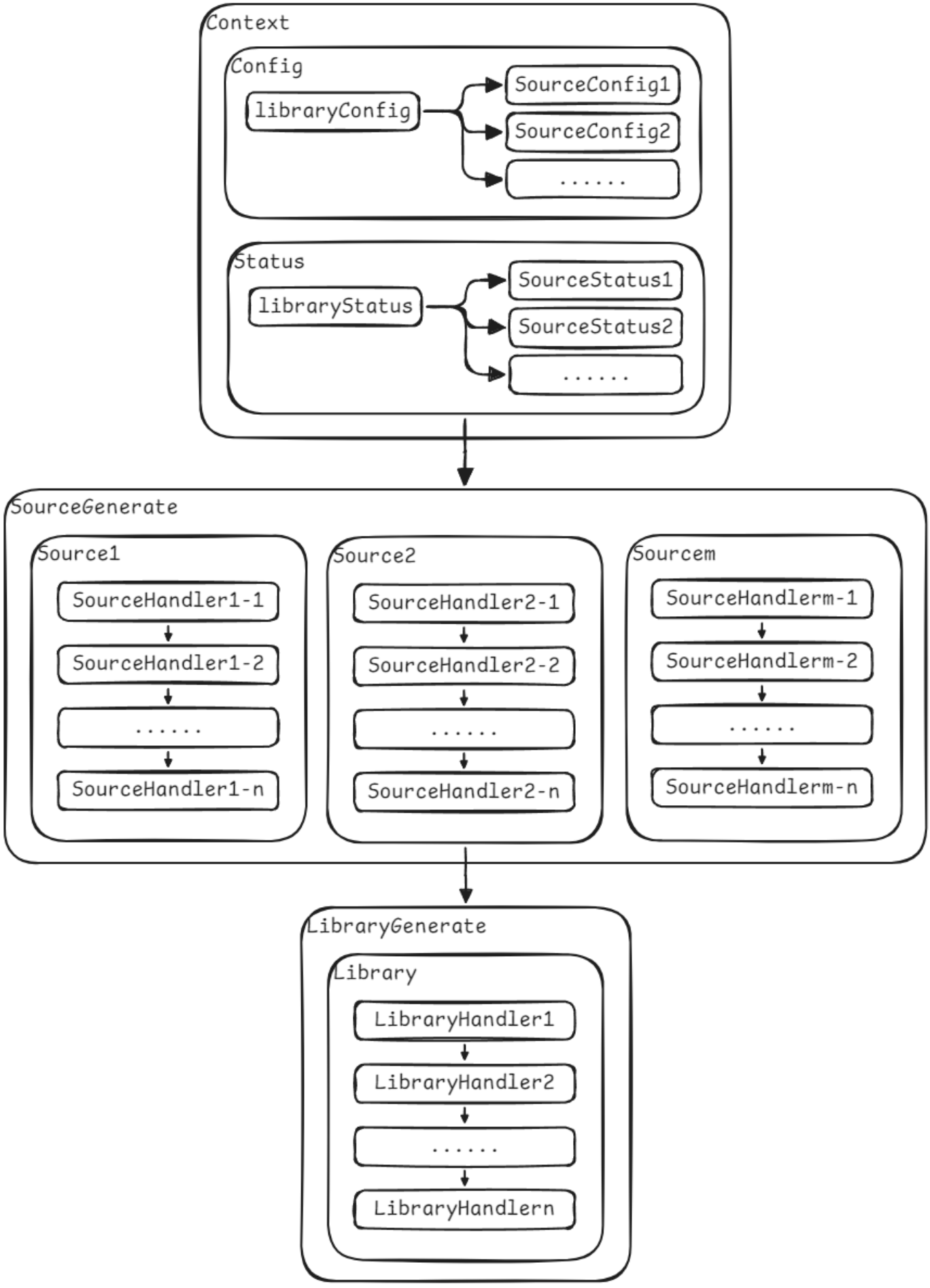
1. 该流程针对接口路径符合 `Path=/ti/report/**`（即查询威胁情报相关） 的请求
2. 在频率超限时，可通过页面的图片验证来解除，并重新开始计数

Device 权限校验

暂时未作任何限制，可后续扩展设备端请求权限校验

特征库制作框架

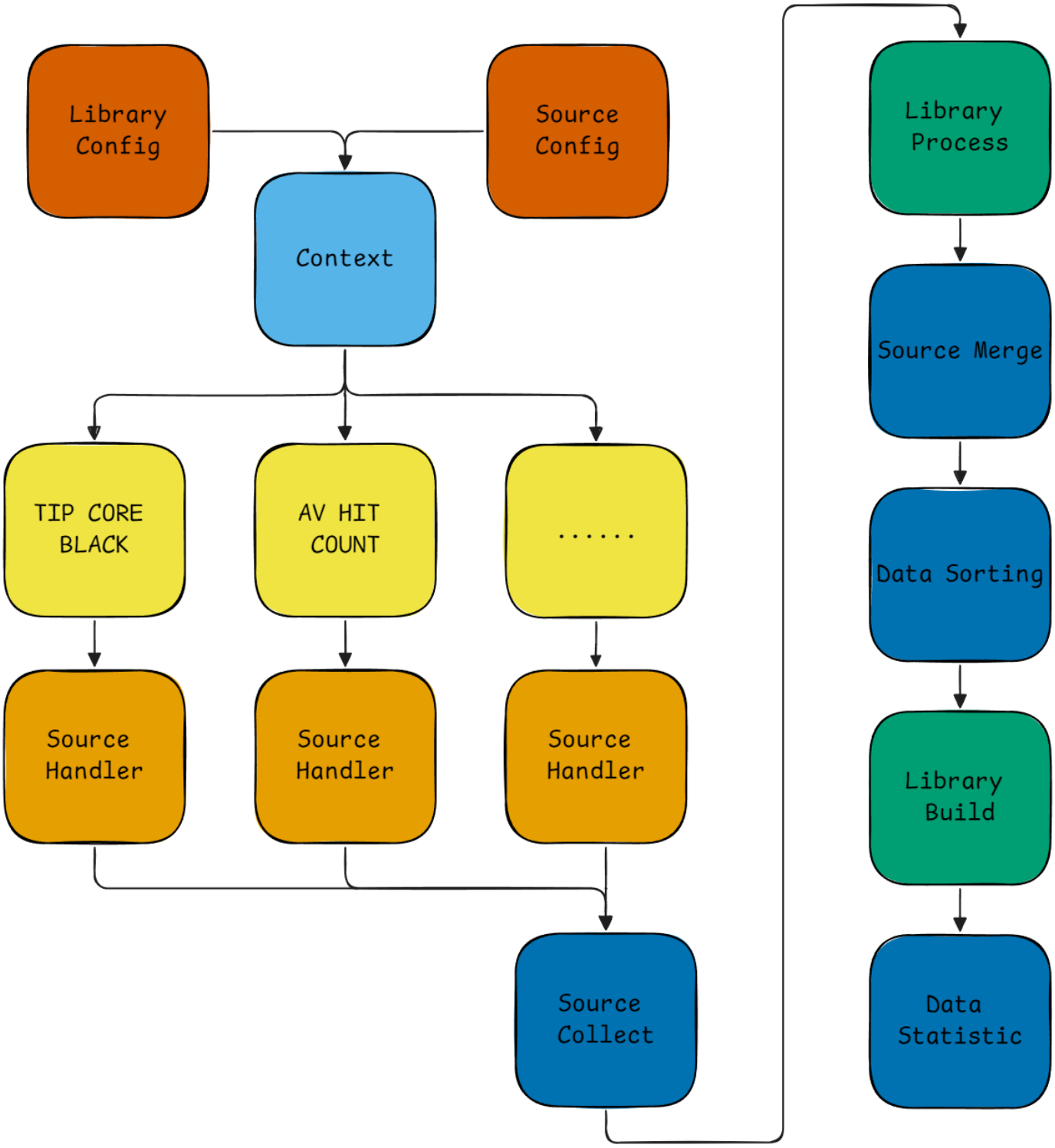
框架总览


Context
Config
libraryConfig
SourceConfig1
SourceConfig2
.....
Status
libraryStatus
SourceStatus1
SourceStatus2
.....
SourceGenerate
Source1
SourceHandler1-1
SourceHandler1-2
.....
SourceHandler1-n
Source2
SourceHandler2-1
SourceHandler2-2
.....
SourceHandler2-n
Sourcem
SourceHandlerm-1
SourceHandlerm-2
.....
SourceHandlerm-n
LibraryGenerate
Library
LibraryHandler1
LibraryHandler2
.....
LibraryHandlern

整体框架：

1. 第一步，获取该特征库的所有配置和其关联数据源的所有配置
   配置内容主要包括一些基本信息，和需要执行的 Handler
    1. 该框架的核心内容就是这些可配置的 Handler，分为 SourceHandler 和 LibraryHandler
    2. 每个特征库执行的 LibraryHandler 和其下关联数据源执行的 SourceHandler 没有数量限制，且都可使用动态表单进行配置，即 SourceHandler 的代码是可以复用的，多个特征库库如果使用相同的数据源数据，只需在动态表单里添加参数，使其配置不同即可，极大的增强了代码的复用性和可扩展性
2. 第二步，异步执行该特征库下所有的数据源处理操作，按照其配置的 SourceHandler 顺序执行，并在执行完成后更新数据源生产状态，所有数据源皆执行完成后才会进入下一步
3. 第三步，按照特征库配置的 LibraryHandler 顺序执行

制作框架应用



所有的流程都包含两部分，数据源和特征库制作

先抽象出特征库制作的流程

1. 数据源数据收集 Source Collect
2. 各数据源数据过滤（IOC 类型过滤等），格式统一 Library Process
3. 数据合并 Source Merge
4. 数据排序 Data Sorting
5. 转换为该特征库需要的出库格式，打包上传 Library Build
6. 数据统计（可选）Data Statistics

框架将特征库的每一步操作都切割为了一个 Library Handler

除了 Library Process 和 Library Build 需要具体特征库具体处理外

其余部分都可直接使用公共方法，只需要在文件处理过程中注意相关的文件名和路径要求即可

同时考虑到特征库数据误漏报等问题，也已经实现关于误报和漏报的 Library Handler，只需添加至 Source Merge 和 Data Sorting 之间即可

对于后续可能需要添加的任何过滤、筛选、添加操作都可在 Library Handler 上进行添加动态配置或者添加新的 Library Handler 实现更多更复杂的功能

类比一下，数据源也有着同样的这些步骤，最后根据当前的具体情况，实现做了简化，但框架本身支持

对于同一数据源我们做到尽可能的代码复用和极高的扩展性，所以将每个数据源的处理代码单独实现，并根据需要添加对应的配置，为了更快的支持新功能的添加，以及优化用户体验，前端采用动态表单，依据后端的 Json Schema 来进行配置，

用以适配该数据源在不同特征库下的不同筛选条件

因为数据源的处理操作较为简单，所以对于单个数据源只需实现一个 Source Handler 即可，当然，对于复杂的数据源，我们也可添加多个 Source Handler 进行选择搭配

威胁情报整合分析

概要描述

情报整合分析是云瞻业务的基础与核心。通过情报整合分析，云瞻将不同来源的IOC情报通过统一的数据格式转换、多源整合、数据拦截过滤，最终写入数据存储，以提供情报查询、特征库制作等上层业务。

功能设计

总体架构

总体逻辑示意图如下：

情报源接入

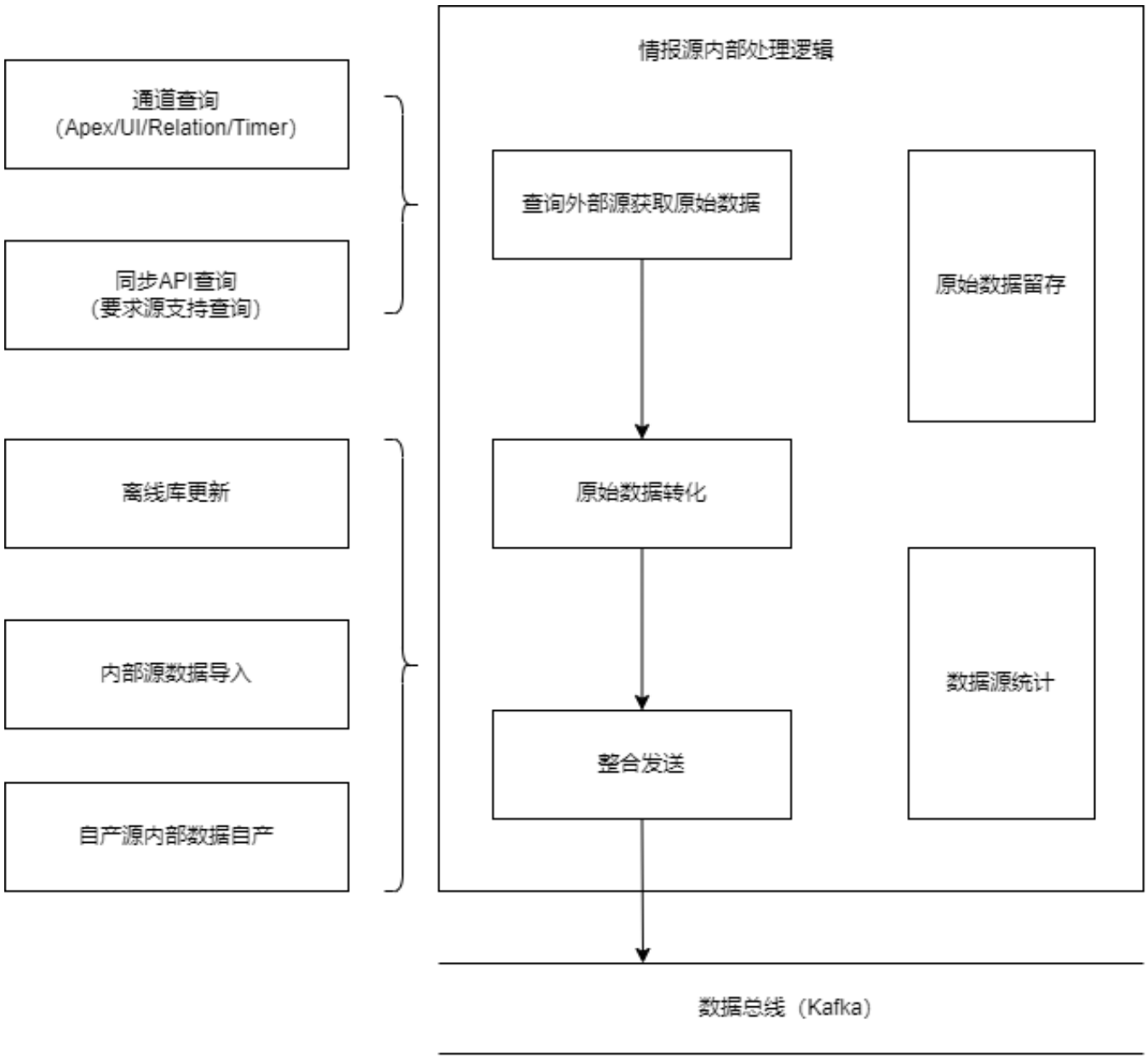
云瞻把情报提供方定义为具体的情报源，情报源有以下属性：

属性	说明	备注
名称	情报源名称	
id	根据名称生成ID，用做情报源标识	
token	随机生成token	
是否支持查询	表示该源是否支持API查询	一般外部购买的情报源，如virustotal、Tencent等提供API接口，则支持
支持类型	该情报源支持的IOC情报类型	
支持类型字段基础分值	用于后续多源整合判定	
描述	情报源描述信息	

数据的输入作为情报源统一处理，有助于保持整合逻辑框架的一致性，目前云瞻接入的情报源有：

情报源名称	说明	
Virustotal	Virustotal情报，合作购入	
Tencent	腾讯情报，合作购入	
Valac	友商情报上送	
Enrich	IOC情报富化	
Mobius	威胁日志检出的可回馈平台的IOC数据源	
aliyun-IP	阿里云IP地址服务	已废弃
Scylla	自产情报源	
NetStar	NetStar情报，合作购入	
360	360情报源，合作购入	
ManualReview	人工审核判定入库源	
ETpro	ETpro情报，合作购入	
Ues	Ues上送引擎检测文件IOC	
Sandbox	沙箱情报源	
Aiwen-IP	埃文地理位置情报，合作购入	
Antiy	Antiy情报，合作购入	
WhiteList	白名单情报源	

在数据源内部，一般采用以下逻辑结构：



根据每个源的特性，支持的源数据获取触发方式不同，大致可分为以下几种：

- 通道查询：对于支持查询的源，整合过程、数据老化等其他业务会通过通道送来数据查询需求，触发源数据采集；
- 同步API查询：同通道查询，触发方式为实时的接口调用；
- 离线库更新：对于部分提供离线情报库的源，定时获取下个周期的离线库，与前一个周期的离线库进行对比，对变化数据送去更新参与整合；
- 内部源数据导入：对于部分内部维护的情报源，数据来源为运维接口的数据上送，上送约定格式数据后，数据源将数据送去整合；
- 自产源内部数据自产：特指通过一定逻辑，处理设备上送信息的数据源，持续处理设备上送数据，转换后送入数据整合。

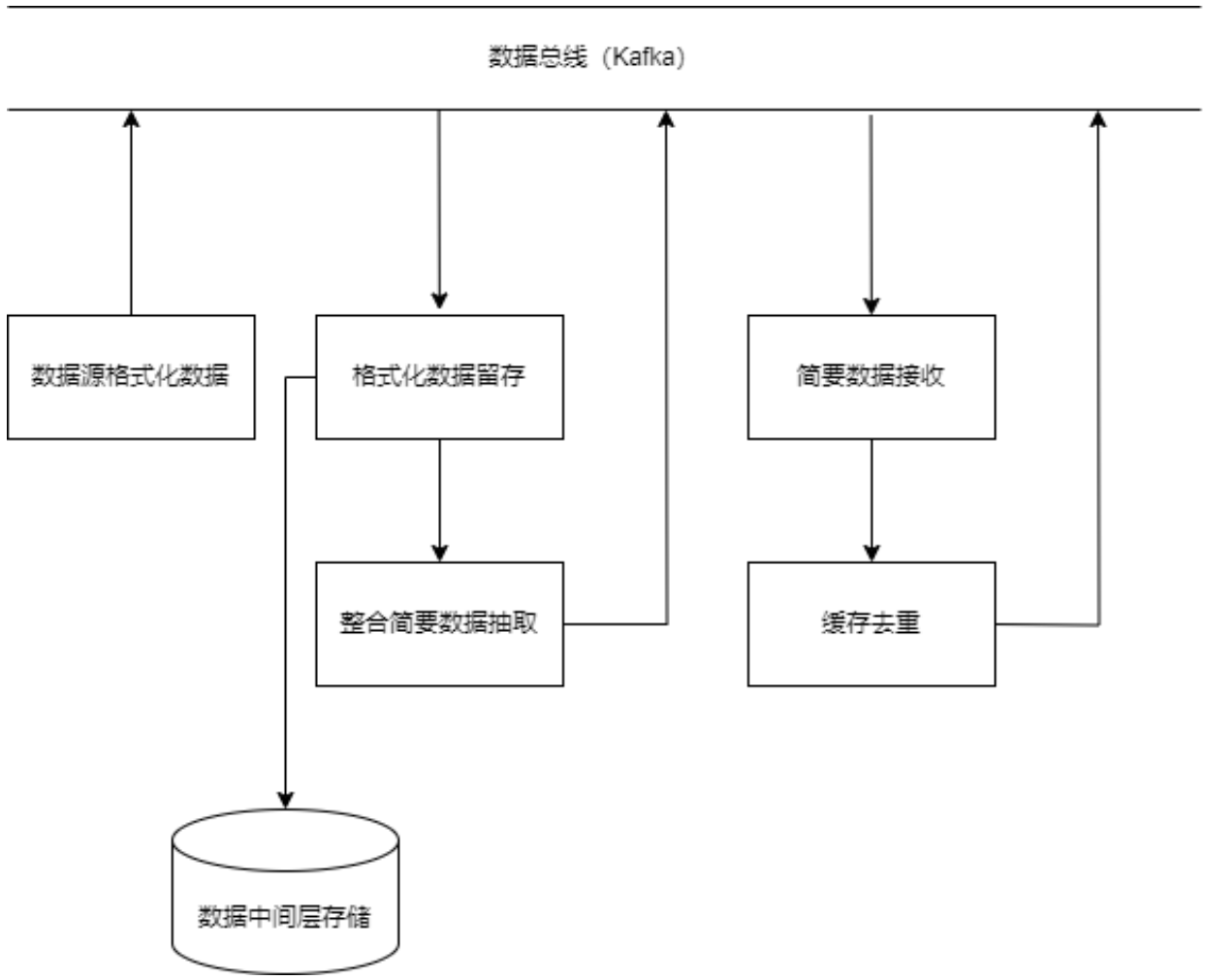
数据源内部处理模块主要有以下几部分：

- 查询外部源获取数据：对于部分友商源，触发源数据获取后，需要通过不同方式获取到数据源的源数据研判信息；
- 原始数据转换：需要将数据源的原始不同格式数据进行转换，统一处理为整合流程可接受的格式；

- 原始数据留存：为后续回溯，需要留存最原始的数据源数据；
- 数据源统计：对于支持API查询类数据源，记录其API访问情况；
- 整合发送：将转换为统一格式的数据送入数据整合的接收消息总线（Kafka）中。

数据中间层

数据中间层对接数据源的数据上送，通过一系列过程为后续整合准备数据：



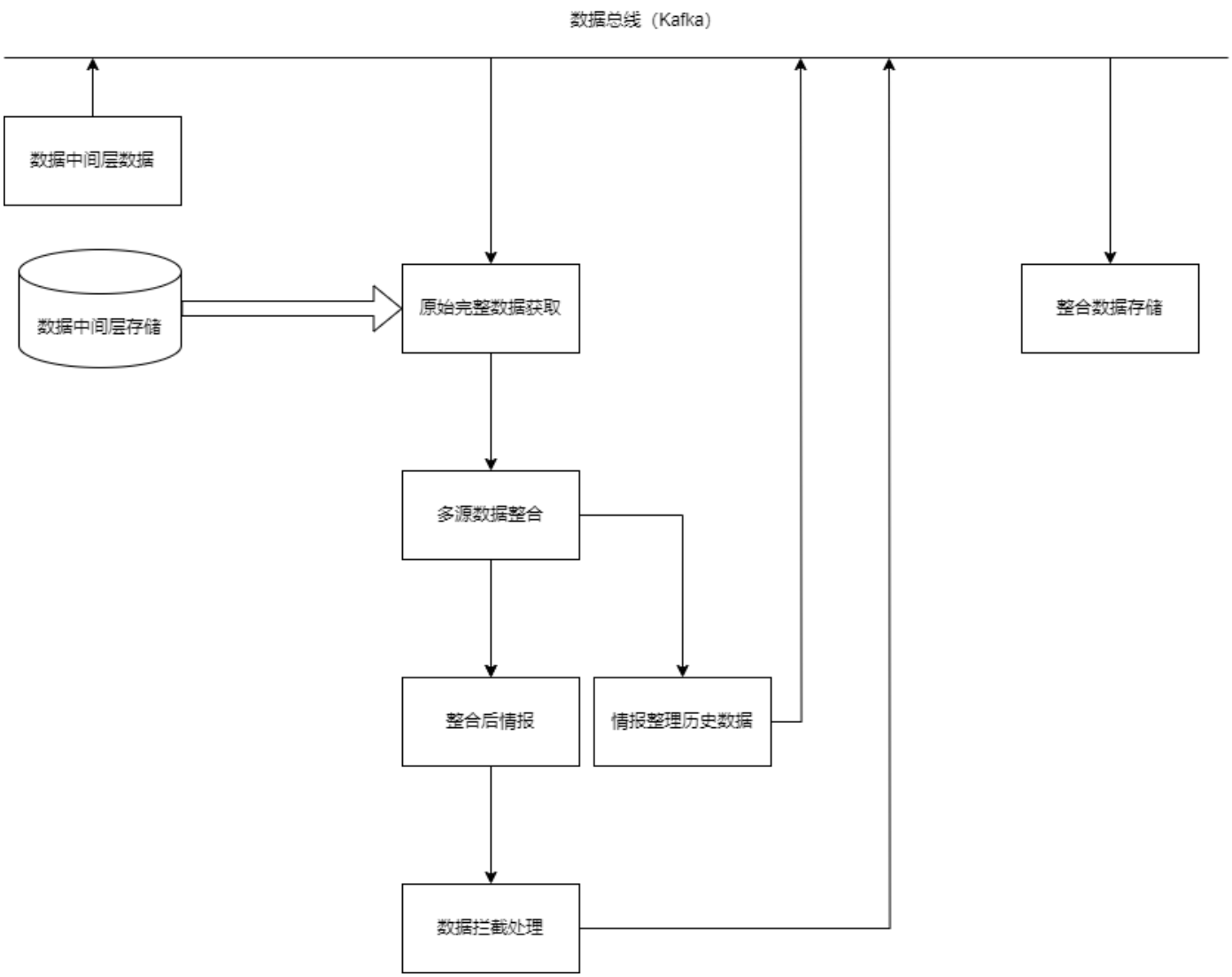
数据中间层与数据总线交互密切，主要有以下过程：

- 格式化数据留存：为减少数据总线数据传递压力，在数据中间层对接收的完整数据源数据进行存储；
- 数据中间层存储：数据中间层使用HBase进行存储，按列区分不同的数据源信息；
- 整合简要数据抽取：作为后续数据整合的信息，在原始数据留存后，仅仅抽取待整合IOC情报的内容合类型即可，通过数据总线向后传递；
- 简要数据接收：在数据中间层中接受先前发出的简要信息，此处分开发送和接受，以应对一些业务场景下的数据汇入；
- 缓存去重：接受到简要数据后，通过缓存对数据进行一定程度上的去重，达到削峰限流的目的。

数据通过缓存去重后，仍送入数据总线，参与后续的整合。

数据整合

数据整合是核心，是一个多源判定的过程，数据流向如下：



在数据中间层的处理中，完整数据被抽取为简要信息送往数据整合。所以在数据整合开始，需要通过简要信息获取到原始的完整数据，并将其进行整理，符合要求的源数据送往后续判定。

通过多源 整合算法，最终产生最终的整合情报，以及情报整合过程中的历史信息。两部分信息仍被送往数据总线，对接后续不同业务的存储涉及。

多源数据整合算法

数据多源整合整体上是一个根据优先级评分取最大值的过程。

数据字段分类

我们把IOC情报的字段整体分为三类，适配不同的处理逻辑：

- 基础属性：情报中具有明确唯一值的字段，如恶意程度、地理位置等；
- 标签：标签字段处理逻辑较为特殊，单独进行讨论；
- 关联关系：某种类型IOC情报与其他类型或本类型情报的关联信息。

基础属性整合

基础属性由于具有唯一确定值，采用根据评分取最大值的逻辑定值。

在数据源接入时，对于一个数据源的支持IOC类型，会根据对该来源的可信度，给出IOC对于基础属性的基础分值。整合时，综合基础分值以及当前IOC的置信度，计算两者乘积，作为最终的该源得分，参与多源对比。我们选取分值最高源的该字段属性，作为该字段整合后最终取值。同时记录证据信息作为整合历史数据。

以恶意程度为例，给出部分源的基础分值情况：

> result confidence = base confidence * source confidence

IP/Domain	Etpro	Tencent	VT	Antiy	白名单	人工审核	360	Valac	Mobius	Enrich
基础可信度	91	80	50	70	95	100	93	92	94	30
高(90)	8190	7200	4500	6300	8550	9000	8370	8280	8460	NA
中(50)	4590	4000	NA	NA	NA	NA	4650	4600	4700	NA
低(10)	910	800	NA	NA	NA	NA	930	920	940	300
未知(1)	91	80	NA	NA	NA	NA	93	92	94	30

> 人工审核 > 白名单 > Mobius高 > 360高 > Valac高 > ETPRO高 > Tencent高 >
>
> Antiy > Mobius中 > 360中 > Valac中 > ETPRO中 > VT > Tencent中 >
>
> Mobius低 > 360低 > Valac低 > ETPRO低 > Tencent低 > Enrich

File	Tencent	VT	Antiy	白名单	人工审核	360	Valac	Enrich	Ues	Sandbox
基础可信度	30	60	70	90	100	50	0（暂不考虑file类型）	80	82	85
高(90)	2700	5400	6300	8550	9000	4500	NA	7200	7380	7650
中(50)	1500	NA	NA	NA	NA	2500	NA	4000	NA	4250
低(10)	300	NA	NA	NA	NA	500	NA	800	NA	850
未知(1)	30	60	70	90	100	50	NA	80	NA	85

> 人工审核 > 白名单 > Sandbox(high) > Ues > Enrich(High) > Antiy > VT > 360(high) >
>
> Sandbox(middle) > Enrich(middle) > Tencent(high) > 360(Middle) > Tencent(middle) >
>
> Sandbox(low) > Enrich(low) > 360(low) > Tencent(low)

URL	Etpro	VT	Antiy	白名单	人工审核(暂不支持)	360
基础可信度	90	50	70	95	100	30
高(90)	8100	4500	6300	8550		2700
中(50)	4500	NA	NA	NA		1500
低(10)	900	NA	NA	NA		300
未知(1)	90	NA	NA	NA		30

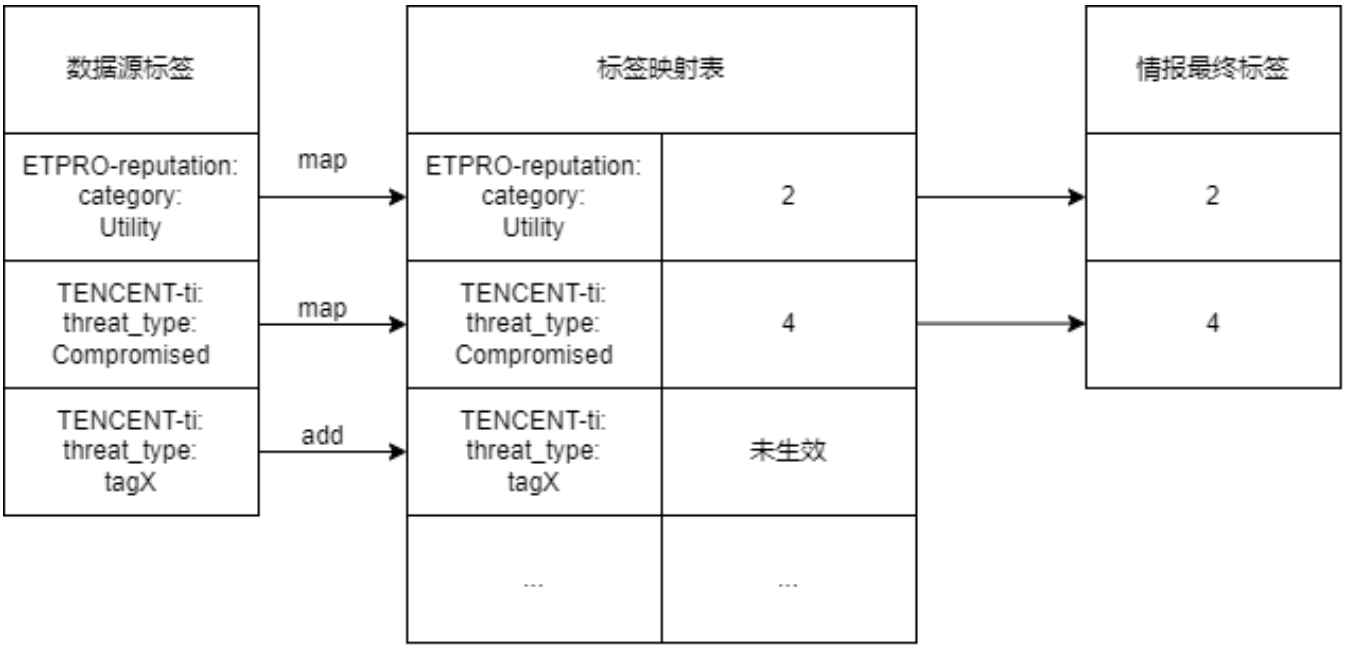
> 白名单 > Etpro(high) > antiy > vt = Etpro(middle) > 360(middle) > etpro(low) > 360(low)

其中，基础可信度即为数据源接入时赋予的某字段的基础分值；高、中、低和未知为数据源给到具体IOC的可信度判断，不同程度有对应分数；基础分值和置信度分值相乘即为整合时最终分值。

标签整合

标签整合的主要工作是转换数据源原始标签为云瞻内部标签，再进行去重合并剔除非法标签。

在标签系统中，云瞻引入了标签映射表，即在标签整合时使用：



根据映射表处理后的标签，会综合考虑情报整合后的恶意程度，去除两者明显相悖的数据，如整合为白名单，则去除恶意标签。

标签整合时，会记录不同标签分别由哪些数据源提供，作为整合历史数据的一部分。

关联关系整合

关联关系整合方式为多源合并，不同源提供的同一类型关联关系数据，合并后按照对应逻辑进行去重，作为最终整合后的关联关系。

数据老化

情报的生命周期管理是情报管理的重要部分。情报整合入库可以视为生命该情报生命周期的开始，数据的老化则意味着情报此生命周期的结束。及时、可靠的数据老化管理有助于保持IOC情报的准确性，云瞻目前设计了两种情报老化机制：

- 被动老化
- 主动老化

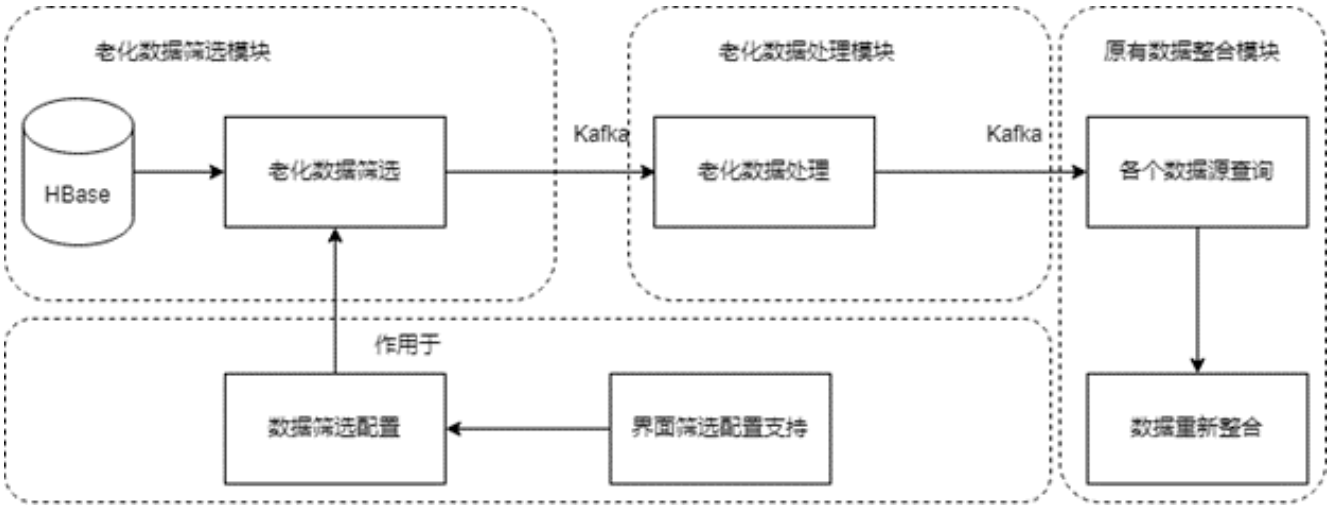
被动老化

被动老化是基于触发式的数据重走整合流程的过程。

当IOC情报被访问时，如接口查询，会送入被动老化流程。首先根据IOC情报在核心库中的更新时间，与对应设置的老化阈值时间做对比，如果触发阈值，则送入完整的数据整合流程。

主动老化

被动老化的方式具有随机性，部分长久不会被访问的IOC则无法进行被动触发。因此需要主动老化的方式进行老化补充：



通过相关配置，利用剩余的购入情报源的查询能力，对数据中间层存储的完整数据进行定时定量扫描，触发对应源原始数据采集， 再送入情报整合流程，达到数据主动老化更新的目的。

情报整合记录

在前面整合过程中已经提到，伴随不同类型字段的整合，会形成一份完整的整合历史记录，主要包括：

- 各个基础属性值，以及对应选取源；
- 各个标签的数据来源。

完整的整合记录信息，会送入数据总线，供不同业务进行消费。

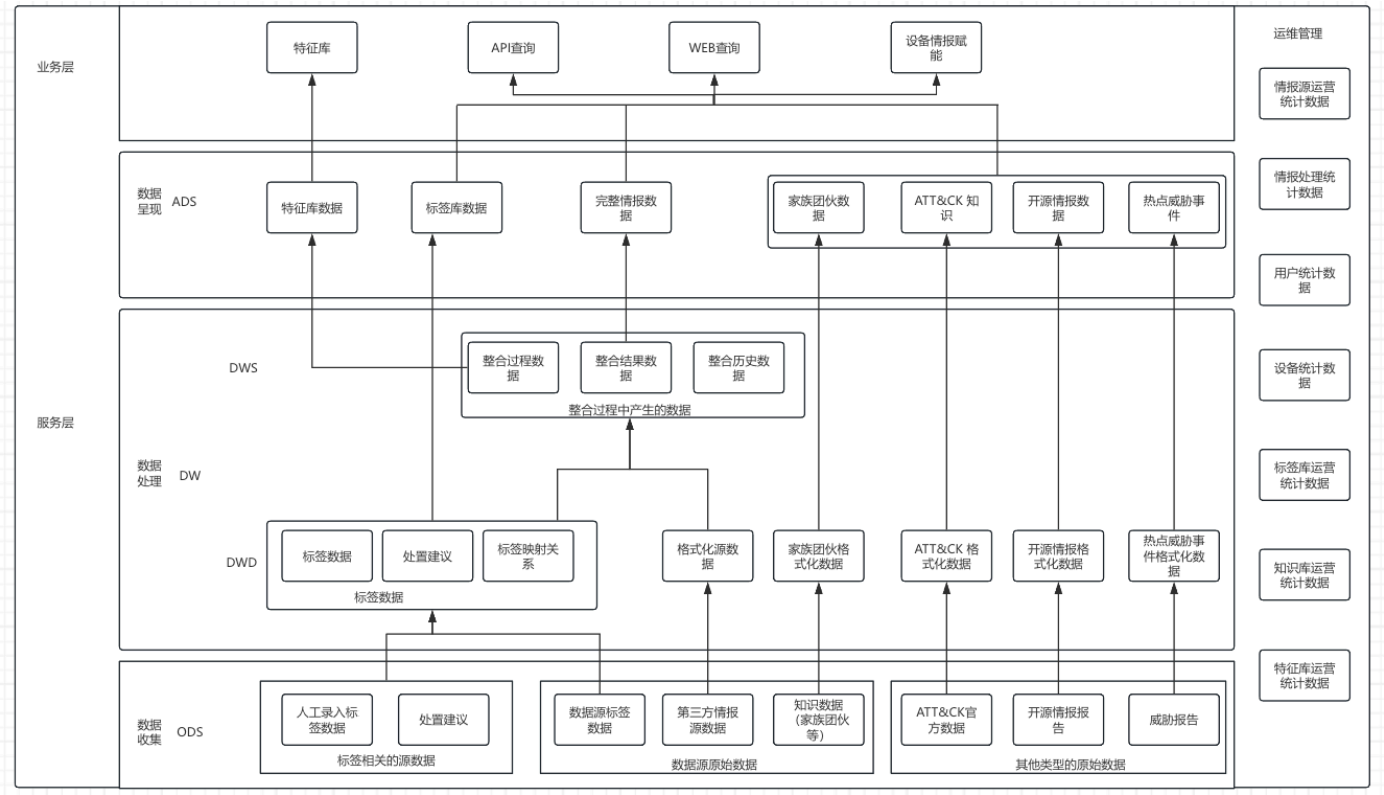
整合数据入库前拦截处理

整合后的数据在送入数据总线提供存储消费前，会进行合法性过滤处理，由于是经由不同拦截器依次处理，我们又称为拦截处理。

云瞻对不同的IOC类型设置了不同的拦截器，通过配置项统一组织实现。IOC数据使用流的形式经由配置的拦截器链，最终完整通过的才准许落库。

威胁情报存储

在构建一个威胁情报存储系统时，我们面临的主要挑战是如何高效地处理、存储和检索大量的安全数据，同时确保这些数据的安全性、可靠性和实时性。威胁情报数据的多样性和复杂性要求我们采用灵活且强大的技术解决方案，以支持从数据收集、分析、处理、对外提供的全过程。数据存储的设计依赖于数据的使用方式，下图中描述了威胁情报平台中各种数据的流向：



下面的章节会对图中比较重要的存储进行详细解释。

威胁情报核心存储

威胁情报核心存储的内容是完成整情报内容，其中包括多样的情报类型，如ip/domain/url/file/tag/whois等，情报之间复杂的数据关系，如rdns，domoan关联ip，domain关联file，domain的同级域名等，基于这样的数据环境需要我们的存储设计满足如下需求：

- **快速数据检索能力**：能够迅速从大量数据中提取关键信息。
- **复杂查询操作**：支持高级查询以分析复杂的数据关系和模式。
- **威胁信息可扩展**：需要灵活适应不断变化的威胁信息和新的数据源。
- **系统可扩展**：随着数据量的增加，系统需保持高效的处理能力，能进行横向扩展

  故，我们采用ArangoDB进行数据存储。

ArangoDB 的以下能力能够满足我们情报的存储要求

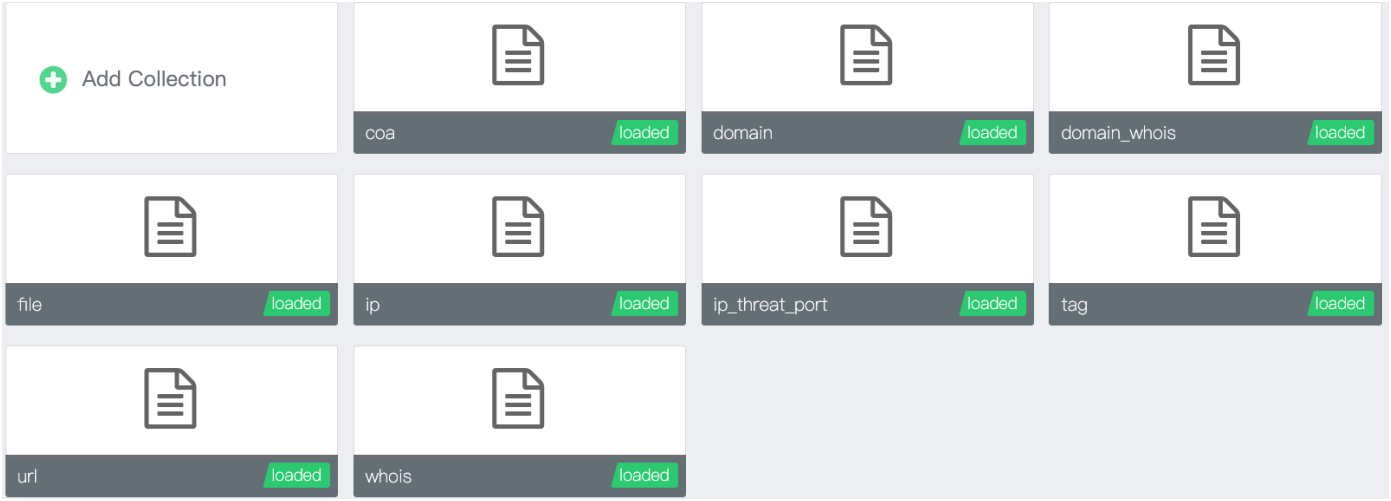
- **多模型支持**：ArangoDB 支持文档、图形和键值存储，可以有效管理多种数据类型，包括IP地址、域名、URLs、文件哈希值等。我们选用的是文档型集合作为情报实体的存储。
- **统一的查询语言**：使用 AQL（ArangoDB Query Language），可以在单一查询中处理多种数据类型，简化数据操作和分析
- **关系建模与分析**：可以轻松构建和查询数据之间的复杂关系，帮助分析师理解和预测威胁行为
- **高效的索引机制**：ArangoDB 提供多种索引选项，包括全文索引、地理空间索引等，以支持快速数据检索和复杂查询。
- **优化的查询执行器**：ArangoDB 的查询执行器针对大数据量优化，确保即使在复杂查询下也能保持良好的性能。
- **水平扩展能力**：ArangoDB 支持集群模式，可以通过增加更多的节点来水平扩展，处理更大的数据量。

- **灵活的数据模型**：随着新的威胁类型和数据源的加入，ArangoDB 的灵活数据模型允许快速适应和扩展。

目前，威胁情报平台使用点集合(Vertex Collections)来存储情报内容中的实体，使用边集合(Edge Collections)来存储实体之间的关系。点集合和边集合的设计如下：

点集合：

现有情报的点集合如下：



ArangoDB中存储的数据结构如下，以ip的点集合为例：

```
{
  "_id": "ip/8c9b24b37693686d2b19445ed7b8591f8bec958598fa4587530871144a043662",
  "_rev": "_g5d6WFe--B",
  "_key": "8c9b24b37693686d2b19445ed7b8591f8bec958598fa4587530871144a043662",
  "object": {
    "ip_address": "104.21.3.107",
    "status": 0,
    "id": "ip-b539ffdb62899a3b39ac21141a027f72",
    "create_time": 1697227825011,
    "update_time": 1697227825011,
    "xxxx":"xxxxxx", // 其他的属性字段和对应的值
    "....":"....."
  }
}
```

_id:文档的全局唯一标识符，它由集合名称和 _key的值组合而成，格式为 collection/key。

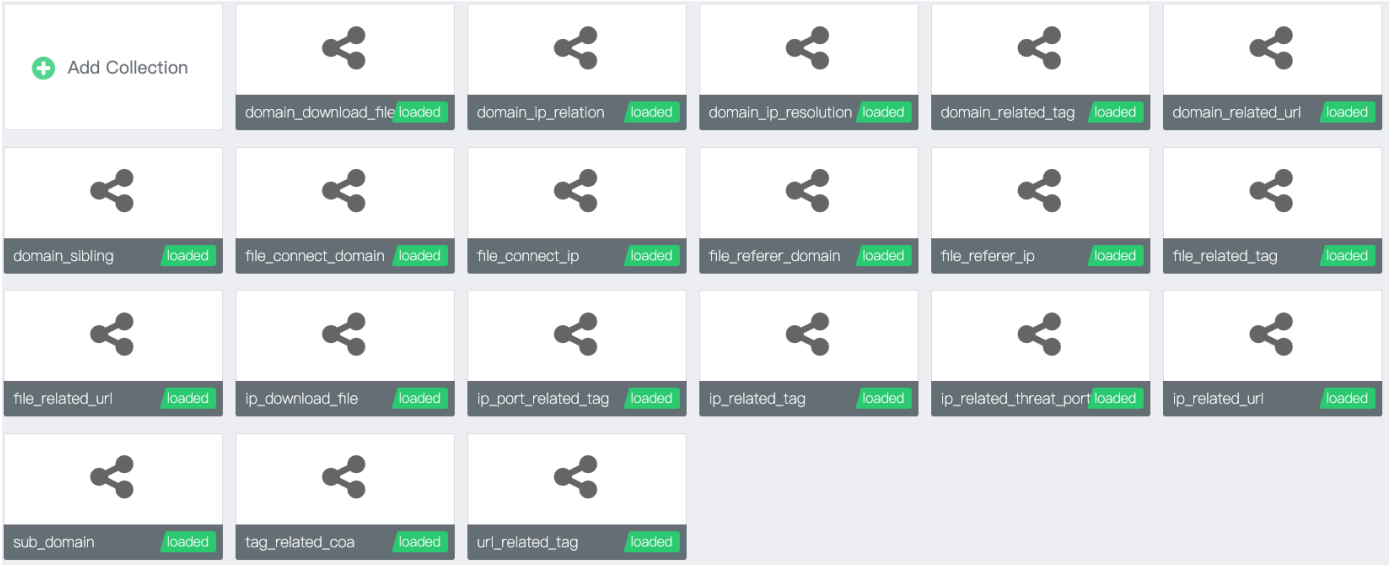
_key:文档在其所属集合中的唯一标识符。这个值是可以自定义的，也可以让 ArangoDB 自动生成。

_rev:文档的修订版本号，每次文档被更新时，ArangoDB 都会自动更新这个版本号。这个属性主要用于处理并发更新和冲突解决，确保数据的一致性。

ip的各种属性数据是以json的形式进行存储的，当情报内容的字段增加不会影响历史数据，也不需要对数据库进行修改。

边集合：

现有情报的边集合如下：



ArangoDB中存储的数据结构如下，以domain_ip_resolution的边集合为例：

```
{
  "_id": "domain_ip_relation/292682067",
  "_rev": "_iDM8CoC---",
  "_key": "292682067",
  "_from": "domain/550480f1c4d375b5204c4b60bd2eb5e3aceb38b160d4ef0c8bacfba3cc60394a",
  "_to": "ip/40c7353e729629e38f857f712dcdb4816bec1b7e290248e897b568f91b064063",
  "object": {
    "is_current": true,
    "timestamp": 1719391882689
  }
}
```

集合的名字"domain_ip_resolution"用以表明这个边集合是哪些情报实体之间的关系。

在基础的信息中 _id, _key, _rev 与点集合一致，相对于点集合多了一个 _from 和 _to 属性。

_from 和 _to：定义边的起点和终点，分别指向域名和 IP 地址的文档中的全局唯一标识符（ _id ）。

object数据中存储了这边的属性，也是以json的形式进行存储，可以兼容字段的增加和删除。

高性能查询的存储

情报核心库中的情报内容是威胁情报平台处理后的完整数据，其中实体之间的关系非常多且复杂。对于用户API查询的场景，只查询部分数据和关系、大批量、高并发的特点，而ArangoDB中的数据是完整的，且在关系复杂的情况下ArangoDB的查询性能达不到上千的QPS，故我们需要一个能够存储海量数据且能够高并发的数据库进行支撑。

存储设计需要满足的需求：

- **海量数据存储**：情报的数量目前是2.5亿，且是一直增加的，需要数据库支持海量数据的存储。
- **字段扩充**：情报内容在逐渐丰富的过程或API查询的提供的数据中会有字段的增加。
- **高并发的能力**：API查询的场景需要上千的QPS

故我们选取了Hbase作为数据库进行数据存储，Hbase以下能力能够满足我们的存储要求：

- **列存储**：HBase 是基于列的存储，适合读写大量的非结构化数据，对于字段扩充来说可以增加对应的列。
- **水平扩展**：HBase 通过简单地增加更多的节点来扩展，支持自动和透明的分片，可进行海量数据存储。
- **实时读写**：提供对数据的实时随机访问能力，适合需要大量实时查询的应用。

建表规则：

目前威胁情报平台是根据情报实体进行Hbase分表设计，表划分和创建原则如下：

1. 不同实体分开存储，例如不同种类 IOC 分表存储；
2. 存储面向数据快速查询，允许数据冗余，例如本设计中不同类型 File Hash 分表存储；
3. 尽力消除关联，由于 HBase 不存在连表查询，对关联查询支持较弱，在存储设计考虑上，尽量化关联为实体节点字段；
4. 建表必须预分区；
5. 沿用目前习惯，表名以及列族名称小写。

存储设计：

1. 以ioc的值为rowkey
2. 字段名称全部小写，多个单词之间使用下划线拼接；
3. 字段名称参考目前 arangodb 中设计，要求在 1-16 字符之间，过长考虑使用缩写；
4. 数据存储时直接序列化对象为字节数组，提升存储性能

字段类型区分：

1. 基础属性：基础属性指最简单的 k-v 形式的字段，如 status、result 等。
2. 标签属性：标签列固定为 cf:tag，由于标签信息易变性以及可穷举性，仅存储标签 ID 列表，详细信息后续通过缓存进行关联查询
3. 关联关系：关联关系的内容与ArangoDB中的保持一致，多个关联的实体以数组的方式存储到一列中。

具体的表结构以ti_ip表示例如下：

列名	类型	含义
ip_address	String	ip地址
ip_type	String	ip的类型 ipv4/ipv6
status	Integer	情报状态 0｜1｜2｜3｜4
result	Integer	情报结果 0｜1｜2｜3
asn	String	自治系统信息
history_domain	List	历史域名（关联关系）
tag	List	标签id列表 [80,91]，具体信息根据id去标签表查询
……		其他字段，可扩展

情报源数据格式化存储

威胁情报平台在进行数据采集时会收到多方的数据，每种数据的格式不尽相同，为了让数据能够更好的进行数据的兼容和集成以及后续的自动化处理，需要把数据进行统一格式进行存储。

建表规则：

1. 数据库选用Hbase，以应对海量数据以及数据源的可扩展
2. 不同实体分开存储，例如不同种类 IOC 分表存储；
3. 沿用目前习惯，表名以及列族名称小写。

存储设计：

1. 以ioc为rowkey，保证ioc的唯一性，每个ioc的数据只更新不存储历史
2. 每个ioc的固定属性(不属于任何数据源)存储在单独的列
3. 每种情报源数据的列是固定的，每个ioc的所有情报源的数据都在同一行
4. 情报源的数据内容不进行列拆分，保证数据列紧凑

情报源字段定义：

字段	类型	含义
element_value	String	ioc的值
result	Integer	ioc的整合结果 0｜1｜2｜3
result_tags	List	ioc的整合标签列表 例如：[10,11]
result_version	String	结果的版本 1.0.0
<情报源>_id	String	情报源的id
<情报源>_status	Integer	情报源数据的状态
<情报源>_timestamp	Long	情报源数据写入的时间
<情报源>_data	String	情报源格式化之后的数据内容

威胁情报平台依据实体类型将表分为以下：

实体类型	表名
IP	ti_source_intelligence_ip
DOMAIN	ti_source_intelligence_domain
URL	ti_source_intelligence_url
FILE	ti_source_intelligence_file_md5
	ti_source_intelligence_file_sha1
	ti_source_intelligence_file_sha256

此处以domain类型的ti_source_intelligence_domain表举例展示表结构：

字段	值
element_value	www.hillstonent.com
result	1
result_tags	[1,2]
result_version	1.0.0
qihoo_id	26rUQhZAYL6oExrFY7Ly0w==
qihoo_status	1
qihoo_timestamp	1724680866539
qihoo_data	{ "needSend": null, "type": "domain", "status": 1, "result": 1, "result_confidence": 3, "domain_name": "www.hillstonent.com", "registrar_name": "hillstonent.com", "download_files": [], "connect_to_files": [], "referer_files": [], "related_urls": [], "tags": [{ "namespace": "TENCENT-ti", "key": "threat_type", "value": "白名单" }] }

情报源原始数据存储

威胁情报平台在进行数据采集时会收到多方的数据，对于每个情报来源的原始数据，为了分析过去的安全事件和威胁活动以及进行威胁事件的证据回溯，需要将从情报源收集到的原始数据进行存储。

建表规则：

1. 数据库选用Hbase
2. 每个情报源一张表
3. 沿用目前习惯，表名以及列族名称小写。

存储设计：

1. 存储历史数据，以{ioc}-{timestamp}作为rowkey，存储历史数据
2. 情报内容不进行解析和格式化，存储原始数据
3. 同一张表中多种类型的数据使用不同的列进行区分

现存储的情报源和对应的Hbase表如下：

情报源	表名
virustotal	thirdparty-data-table
tencent	ti-tencent-data-table
qihoo	ti-qihoo-data-table
sandbox	ti-sandbox-data-table

此处以tencnet情报源的ti-tencent-data-table 表举例展示表结构：

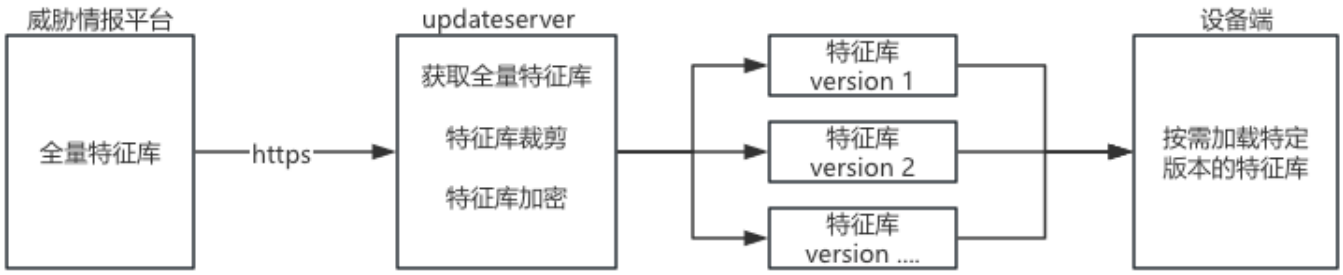
表中字段定义如下：

字段	类型	含义
type	String	ioc类型 ip | domain | url | file
result	object	情报源原始数据，一般为json字符串

字段	值
type	domain
result	{"return_code":0,"return_msg":"success","ver":"3.y.com-17246808075360","result":"white","intelligences":[],"tags":[{"threat_type":"\xE7\x99\xBD\xE5\x90\x8D\xE5\x8D\x95","stamp":["\xE7\x99\xBD\xE5\x90\x8D\xE5\x8D\x95"]}],"threat_level":0,"family":[],"campaign":{"name":"","alias":[],"links":"","attack_method":[],"source_area":"","desc":"","industry":[]},"ttps":[],"first_seen":"2019-11-01 19:09:38","last_seen":"2024-08-08 21:30:22","rank":{"umbrella_rank":0,"tranco_rank":0},"basic":{"registrant_organization":"godaddy.com, llc","registrar_name":"sendsafelycom","registrar_email":"abuse@godaddy.com","registrar_address":"","registrar_phone":"","create_time":"2011-10-13 00:00:00 UTC","expire_time":"2024-10-25 00:00:00 UTC","update_time":"2023-10-26"},"pdns":[{"rvs":"34.232.126.4","time":"2023-09-04 00:00:00"},{"rvs":"52.200.220.50","time":"2023-09-04 00:00:00"}],"icp":{"record":false,"body_name":"","icp_num":"","review_time":"","site_manager":"","site_name":"","unit_nature":""},"context":{"articles":[],"black_md5_contain_domain":[],"black_md5_download_from_domain":[],"black_md5_visit_domain":[],"black_url_of_domain":[]}}

特征库存储

情报特征库是依据威胁情报平台的数据，增加一定的筛选和判定逻辑之后输出给设备的特有情报。特征库加载到设备上方式如下图所示：



故，在威胁情报平台上特征库以文件的形式存储在hdfs上，并以接口的方式提供给updateserver的。

各种特征库的存储路径如下：

特征库	路径
ISOURCE	/tip/signature/library/isource-black-new/{时间戳}.zip /tip/signature/library/isource-black-old/{时间戳}.zip
ISHADOW	/tip/library/ishadow/{版本号}/ishadow.tar.gz
C2	/tip/library/blacklist/integration/blacklist-{版本号}-{时间戳}.zip /tip/library/whitelist/integration/blacklist-{版本号}-{时间戳}.zip
AV	/tip/signature/library/av/{时间戳}.zip
IPR	/tip/library/ipr/archive/{时间戳}.zip
SANDBOX	

分析整合记录存储

多源情报在整合成最终情报的过程中会产生大量的整合记录，存储的历史整合信息为数据分析提供基础，安全团队可以利用这些数据进行趋势分析和模式识别；在发生安全事件后，历史情报整合数据可以进行事件回溯，分析数据来源、误漏报来源。威胁情报平台中存储的分析整合记录主要包含ioc每次整合的结果、整合的标签列表、各个属性的值以及这个值是从哪来的。

建表规则：

1. 数据库选用Hbase，整合记录的数据量非常大，每天的记录条数达百万级
2. 每种ioc类型一张表
3. 沿用目前习惯，表名以及列族名称小写，字段多单词之间使用下划线分割。

存储设计：

1. 存储历史数据，以{ioc}-{timestamp}作为rowkey
2. 整合记录中只记录ioc属性的来源和结果，不记录关联关系

现存储的整合记录的Hbase表如下：

情报源	表名
ip	ti_analysis_intelligence_ip_record
domain	ti_analysis_intelligence_domain_record
url	ti_analysis_intelligence_url_record
file	ti_analysis_intelligence_file_record

表中存储字段如下：

字段	类型	描述
value	String	ioc的值
type	String	ioc的类型 ip｜domian｜file｜url
result	Integer	整合最终的结果
result_source	String	最终整合结果的来源 如 virustotal
source_list	List	参与整合的情报源名称
original_result	Integer	数据源原始整合的result
original_result_source	String	数据源原始整合结果的来源
attributes_list	List	每个属性的值以及来源
timestamp	Long	整合时间戳

AttributeSource 表示每个属性的值以及来源，是有固定字段的，通过AttributeSource的列表形式能够应对情报字段的扩充和减少。AttributeSource 的结构如下：

字段	类型	描述
attributeName	String	ioc的属性名称 如 asn，topPrivateDomain 等
attributeResult	Object	ioc的属性值 ，不同的属性值类型不一样
attributeSource	String	此属性值的来源 如 virustotal

多源情报接入

情报源是指TIP获取威胁情报的来源和渠道，从情报源获取的原始数据在进行处理以及格式转化后，就可以进入TIP的整合流程，成为TIP的威胁情报数据。威胁情报平台的核心数据源可以分为可查询情报源和上送数据源，目前的情报源在下表所示：

情报源	接入方式	备注
UES	数据上送云端，云瞻通过kafka消费	智铠上送情报
Etpro	提供文件下载，云瞻定时任务处理	Etoro静态库
Sanbox	云瞻通过Kakfa消费接入	沙箱上送
Enrich	运维通过接口上送	富化数据上送
Valac	运维通过接口上送	友商数据源
Aiwen	提供文件下载，云瞻定时任务处理	Ip地理库
Mobius	监听kafka的威胁事件topic，云瞻处理后接入	可信威胁事件
腾讯	云瞻主动采集	腾讯数据源
奇虎	云瞻主动采集	360数据源
VirusTotal	云瞻主动采集	VT数据源
NetStart	云瞻主动采集	NetStar数据源

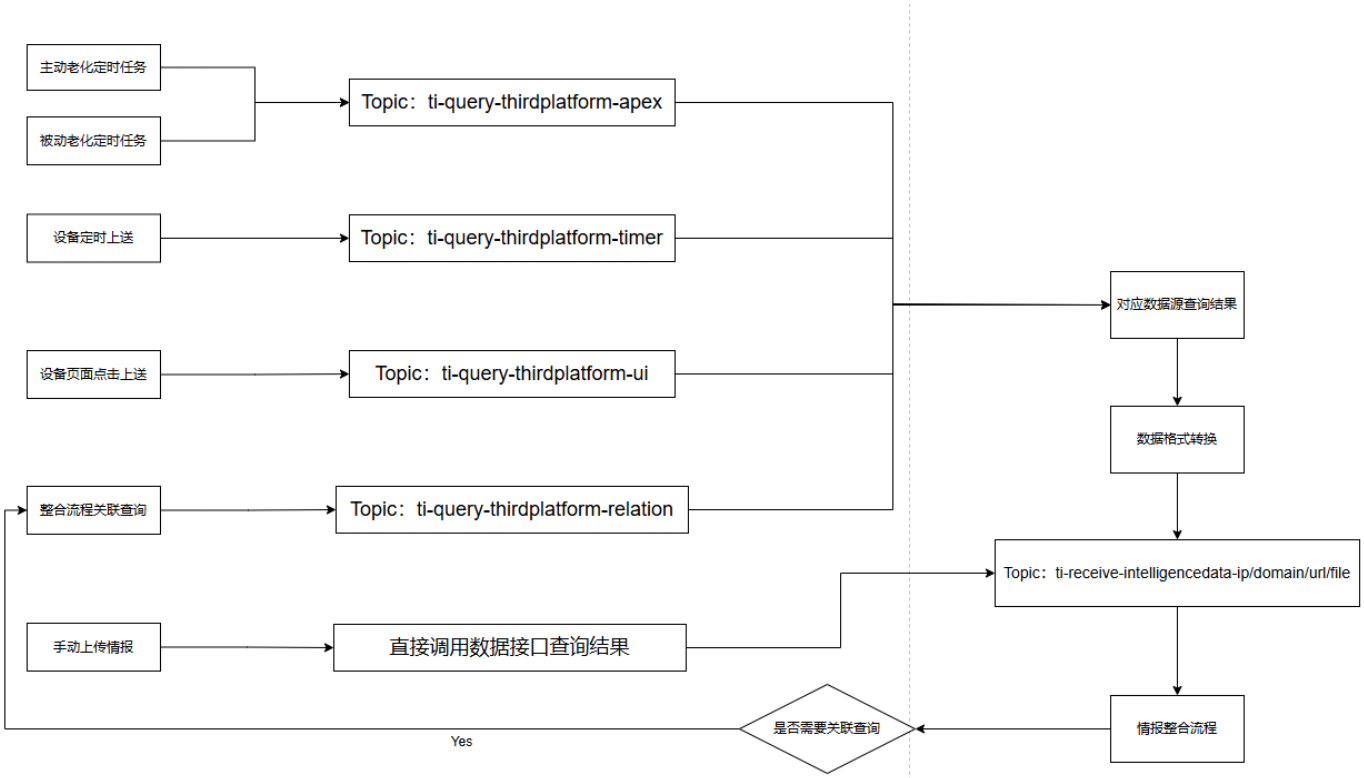
可查询情报源接入

可查询情报源是指支持云瞻发起主动查询并返回结果的数据源，但这并不代表此类数据源会主动给威胁情报平台提供数据。威胁情报平台向这些情报源发起查询之后将结果进行转换，转换成云瞻通用的数据格式之后将数据发送到整合流程进行整合，使其变成云瞻的自有情报，我们可以把这个过程叫做数据采集。

此类情报源在接入TIP之后大多不会主动发送数据，云瞻在查询这些数据后会结果发送到整合流程，整合之后该数据才会成为进入云瞻的情报库，查询的触发方式有以下几种：

- 数据源老化：原本存在于云瞻情报库内的情报，在经过一段时间后需要重新查询商业源，以保持情报库的时效性。在老化的流程中，数据源的老化分为主动老化以及被动老化，每种老化方法的触发方式如下
    - 主动老化：云瞻后台会定时启动主动老化任务，筛选出上次整合时间超出阈值时间的IOC
    - 被动老化：云瞻的页面查询情报之后会触发情报的被动老化
- 手动上传情报：在云瞻后台手动上传情报时，TIP会直接同步查询数据源
- 设备定时以及UI查询情报：设备会在定时以及用户手动点击查询威胁情报时上送数据给云瞻的kafka，云瞻后台会监听数据并同步查询相关数据源
- 情报整合时的关联查询：云瞻在整合流程会查询一个IOC的关联情报，查询关联情报时会同时查询相应的数据源

每种触发方式的数据流动如下图，模块之间的数据使用kakfa来进行传递：



在每个kafka通道的监听中，收取的查询对象都为`QueryObject`：

```
public class QueryObject {
    private String threatEleType;
    private String threatEleValue;
    private List<String> sourceList;
    private Integer queryType;
    private QueryMode queryMode;
}
```

每个监听通道接受`QueryObject`之后，会去查询相关的数据源。由于每个数据源的返回结果并不相同，云瞻会在后续处理中将其转换成威胁情报平台通用的格式，其中`Ip`对应的格式为`IntelligenceData<SourceIntelligenceIp>`，其余类型同理。同时，在后续的整合过程中，source-framework模块会校验进入整合流程的数据源是否合法（数据源真是存在且处于开启状态）。

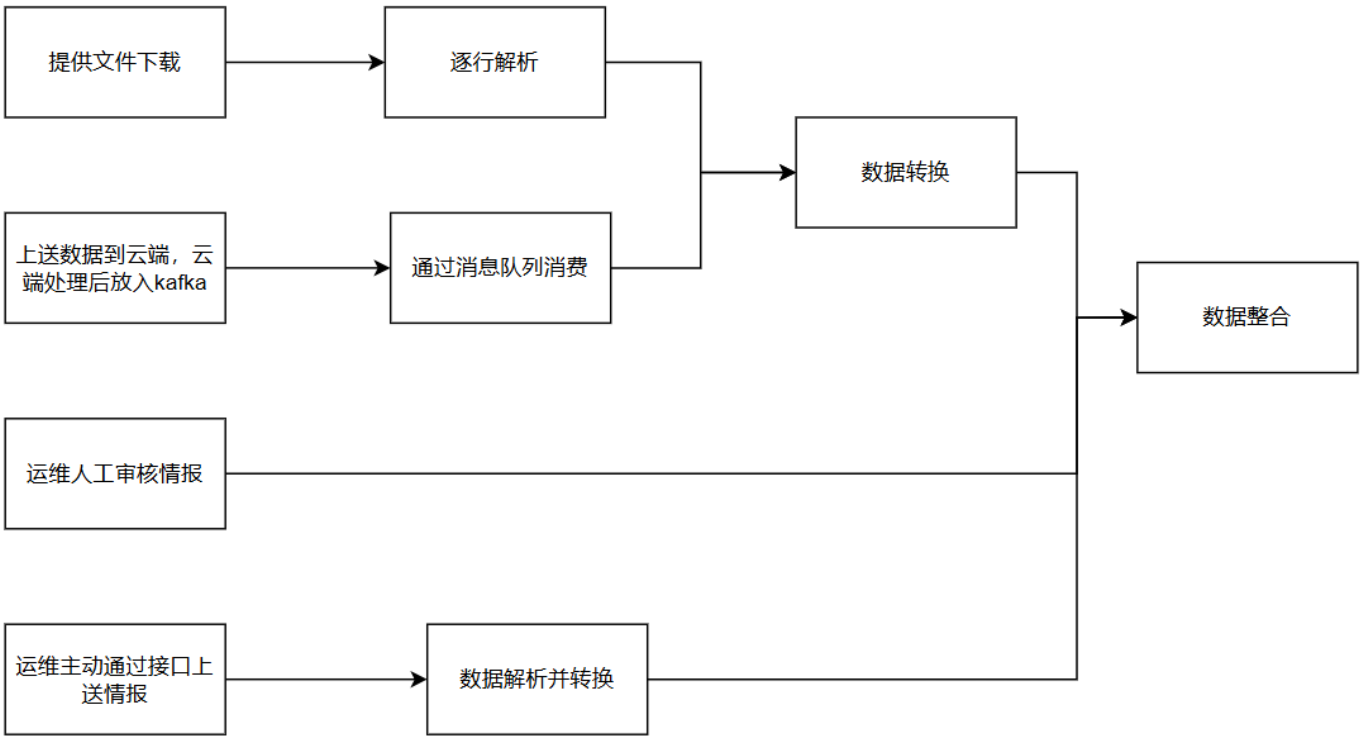
可查询情报源大部分是与TIP合作的第三方商业源，有以下几种：

数据源	是否商用	查询链接
奇虎	是	https://api.ti.360.net/v2/evaluate
腾讯	是	https://xti.qq.com/api/v3/ti
VirusTotal	是	https://www.virustotal.com/api/v3
NetStar	否	http://incompass3.netstar-inc.com
上表中各数据源的查询链接的具体使用方式可以查看RD系统的相关文档，不再在此处赘述。		

上送数据接入

上送数据源是指会以文件、消息队列等形式主动向威胁情报平台提供情报的数据源。此类数据源的接入形式通常分为以下几种：

- 数据源提供文件下载，云瞻下载后逐行对数据进行解析，将数据转化后发送到整合流程
- 数据源通过WebSokcet通道将情报数据上送到云端，云端处理后存储到kafka，云瞻通过消费kafka内的数据并转化后送入整合流程
- 运维人员完成对待审核列表内情报的审核，该情报会进入整合流程



下面是各上送数据源接入威胁情报的途径：

- UES：通过kafka通道device_avro-ues-virus_event
- Etpro：定时任务下载文件处理
- Sandbox：通过kafka通道ti-sandbox-report
- Enrich：运维通过页面上传文件
- Valac：运维通过页面上传文件
- Aiwen：定时任务下载文件处理
- Mobius：通过kafka通道device_avro-event-threat_event

开源情报接入

开源情报是指网上以开源形式出现的非商业数据源，威胁情报会下载这些数据源并进行保存。由于开源情报数据的不可靠性，威胁情报平台在保存后并不会将其送入整合流程进入情报库，而是在情报查询时展示，仅供用户进行参考。

开源情报分为两类：

- 手动开源情报：需要运维手动上传后进行处理
- 自动开源情报：通常以定时任务的形式展现，下载网上对应的情报后存入数据库

开源情报功能支持在页面管理手动开源情报和自动开源情报信息，相关的任务配置保存在Mysql中，情报源配置页面位于**情报管理**-->**开源情报管理**-->**情报源管理**。开源情报的具体情报数据存放在Hbase中，`RowKey`为对应的IOC，仅保存该IOC在各数据源的结果以及更新时间。

开源情报管理的具体细节可以查看6.3节。

安全运营

白名单管理

概要描述

云瞻维护了不同IOC类型的白名单，在IOC整合分析、特征库输出等上层业务中会使用这部分白名单数据。

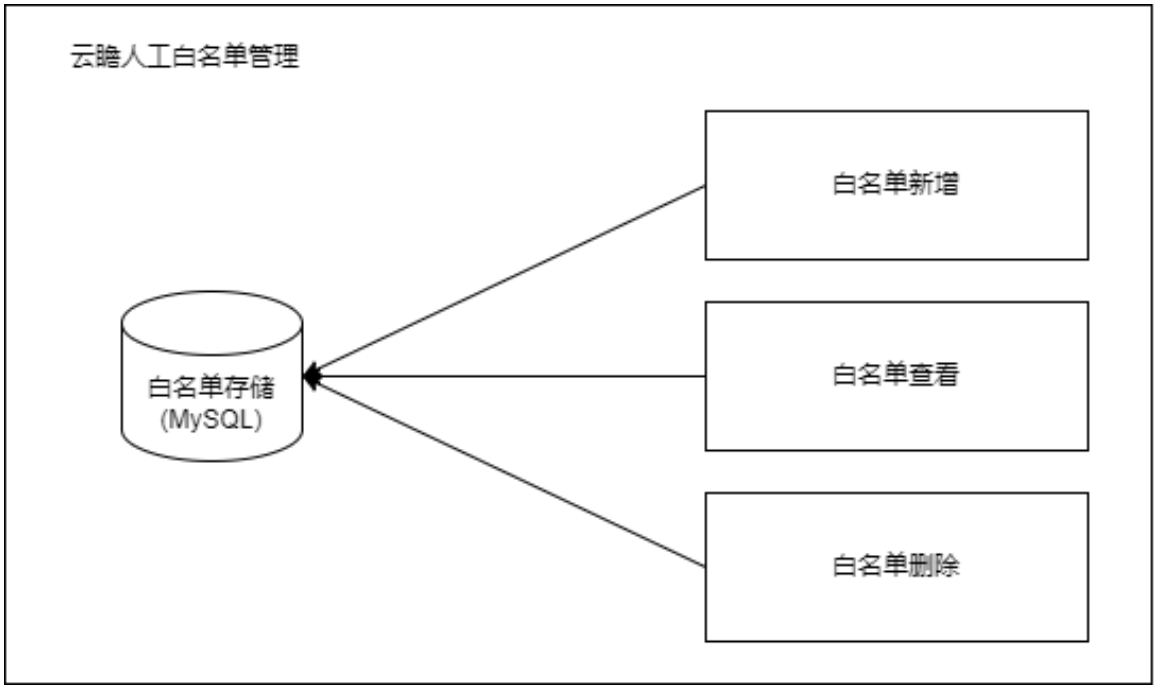
目前，云瞻支持的白名单类型包括IP、域名和文件哈希。

这些白名单数据主要来源于两个方面：一种是由相关人员手动维护的白名单，虽然数量相对较少，但可信度较高；另一种是来自友商的数据，定期更新，我们在某些业务场景中直接应用这部分数据，目前主要是腾讯源提供。

综合所有IOC类型，云瞻目前可用的白名单分为三部分：人工白名单、域名固化白名单和腾讯源白名单。下面分别介绍不同部分内容。

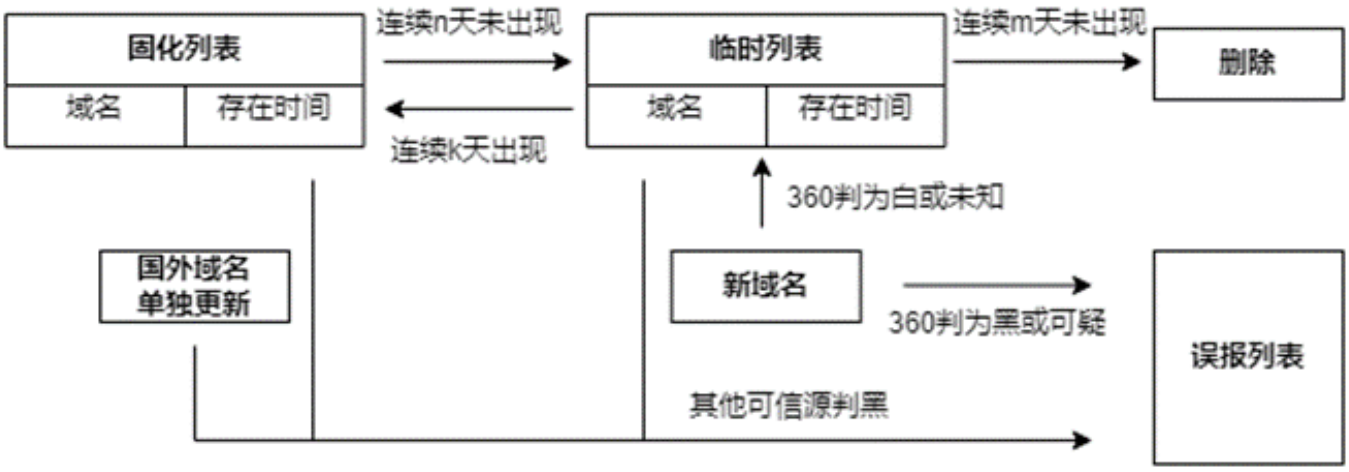
人工白名单

人工白名单指相关人员手动维护的白名单，云瞻管理页面提供白名单数据的展示、新增、删除（批量）等功能。目前支持IP、域名和文件哈希。

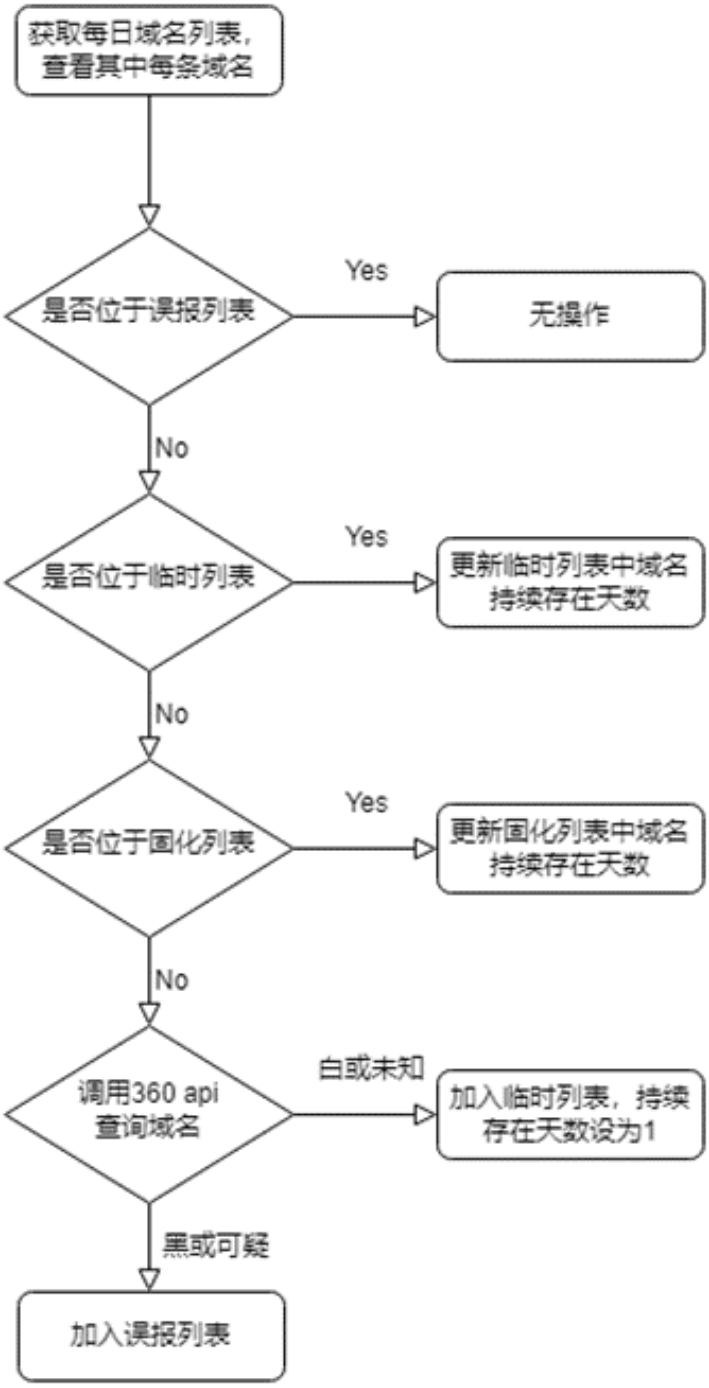


固化白名单

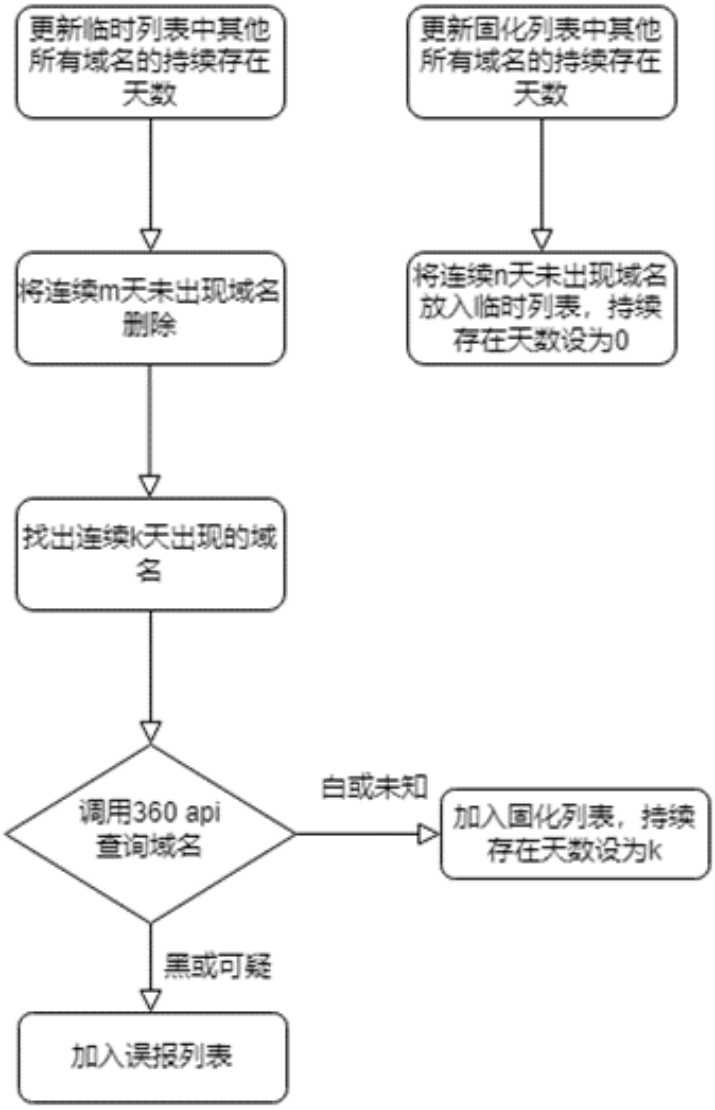
腾讯白名单的误报率较高，为减少误报，威胁情报平台设计了一个固化列表、临时列表、误报列表循环的机制，具体的循环逻辑如下：



云瞻后台会定时去下载腾讯白名单以及Umbrella源的数据，并以下面的流程图来进行固化操作：



每个列表内的数据都会根据域名的存在时间进行持续性的更行，以保证固化库的时效性，更新的逻辑如下：

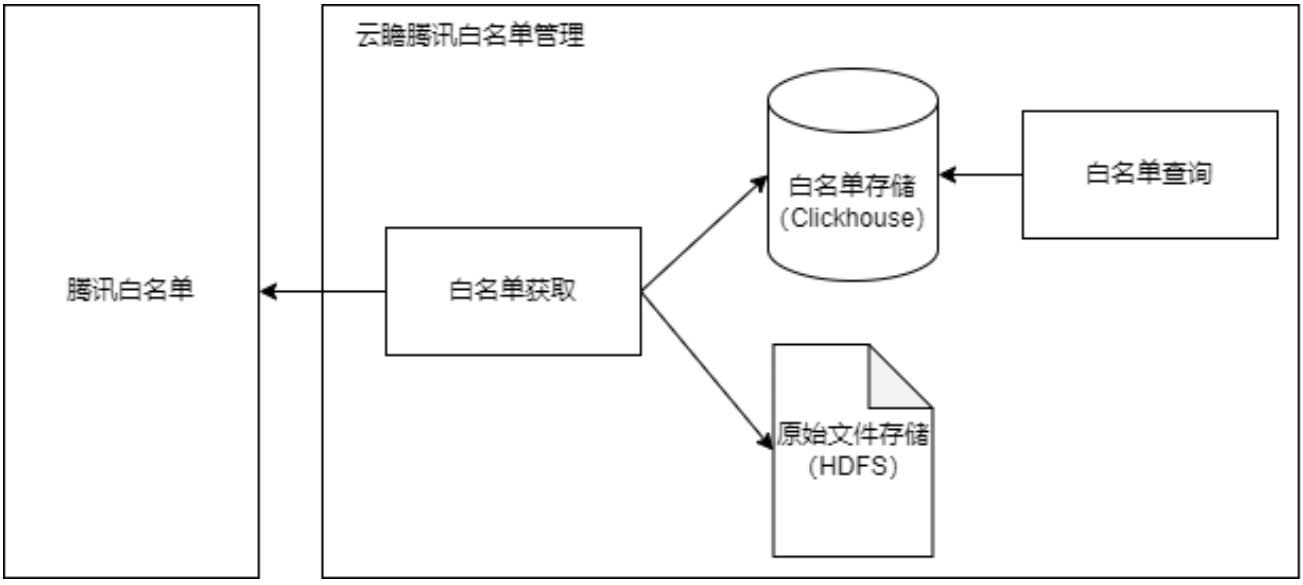


综上，我们可以总结出固化白名单的制作流程：

- 若今日腾讯白名单中域名位于误报列表，不做处理；
- 若今日腾讯白名单中域名位于固化列表，更新该域名在固化列表中的连续出现天数。固化列表中的域名连续n天不出现，将其放入临时域名列表；
- 若今日腾讯白名单中域名位于临时列表，更新该域名在临时列表中的连续存在天数。若临时列表中的域名连续m天未出现，将其删除。若连续k天出现，将其放入固化列表；
- 若今日腾讯白名单中域名为新域名，查询Hbase结果表，在有效期内判为白放入临时列表，判为黑放入误报列表，不再有效期内走老化流程
- 国外域名列表根据其来源单独更新。
  最初的固化列表、临时列表、误报列表由运营给出，之后根据每日下载的腾讯白名单以及Umbrella域名数据进行更新。固化库中阈值m、n、k值可以在页面中设定，具体位置在**情报管理**-->**Domain白名单管理**-->**山石白名单**。固化白名单会在制作C2白名单中被使用。

腾讯源白名单

在与腾讯方合作时，购买了其白名单数据授权。



腾讯白名单以文件形式提供IP和域名数据， 每日提供一份全量库文件。我们使用其提供的相关SDK进行数据获取。

云瞻在获取到腾讯白名单后，会把原始文件保存至HDFS，提供给其他业务下载使用；此外，云瞻会将每天的白名单数据同步写入Clickhouse中，提供云瞻管理界面的腾讯白名单查询功能。

腾讯白名单格式如下：

列（从左至右）	含义	备注
第一列	IP地址或者域名	
第二列	标志位，可选0或者1	对于IP白名单，此列默认为0；对于域名白名单，1-进行泛匹配（比如：数据是com，可以匹配所有为a.b.c.com后缀的域名），0-进行全匹配（比如：数据是baidu.com，只能匹配域名为baidu.com域名）

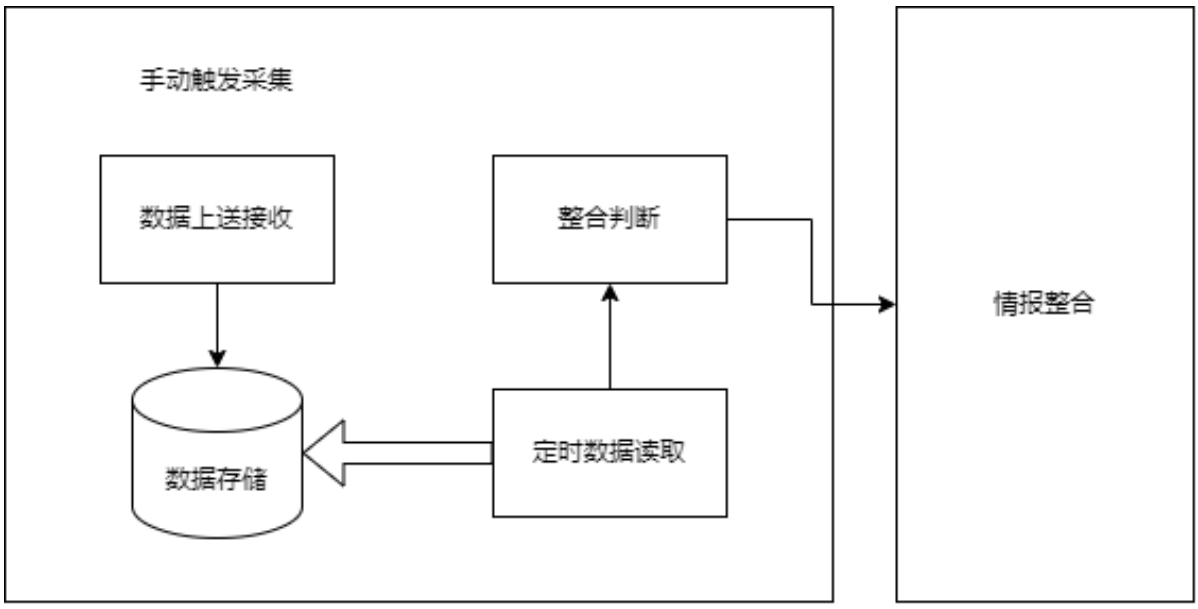
人工生产情报

手动触发采集

云瞻在情报丰富阶段，需要支持通过文件上传的形式，大批量导入IOC情报参与整合，进入核心库。

为了满足上述需求，设计了手动触发采集的功能：通过在管理平台上传IOC情报文件，读取文件存储进数据库，后续使用定时任务定期读取数据库，触发对应IOC参与整合。

总体逻辑示意图如下：



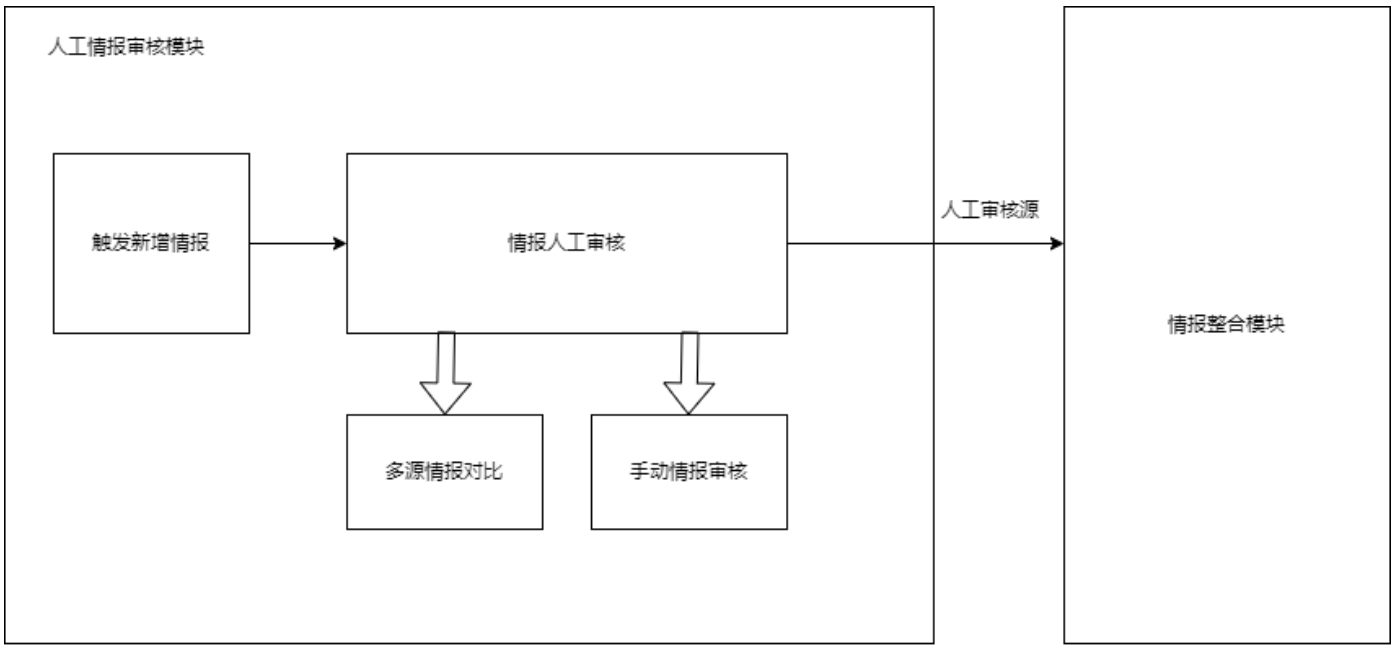
- 数据上送接收：云瞻在管理平台提供上传界面，接收收集到的IOC情报文件；
- 数据存储：接收文件后，将文件数据读取后存储进数据库（HBase）；
- 定时数据读取：采用定时读取的形式，从数据中读取数据进行整合判断；
- 整合判断：上送数据需要经过研判后才会送入情报整合模块，此处判断逻辑为，查询腾讯源，当腾讯源返回数据时，再进行整合；
- 手动采集数据整合：以被动老化的形式（非源身份），将情报数据送入整合入库流程。

人工审核情报

为了提高情报研判的及时性和准确性，云瞻引入了情报的人工审核机制。

新增人工审核源，将人工判定后的IOC情报，以人工审核源的身份，参与情报整合。

总体逻辑示意图如下：



- 触发新增情报：在云瞻管理平台界面手动或各业务处，生产需要经由人工判断的情报，可选关联查询外部情报源等，提交人工审核；

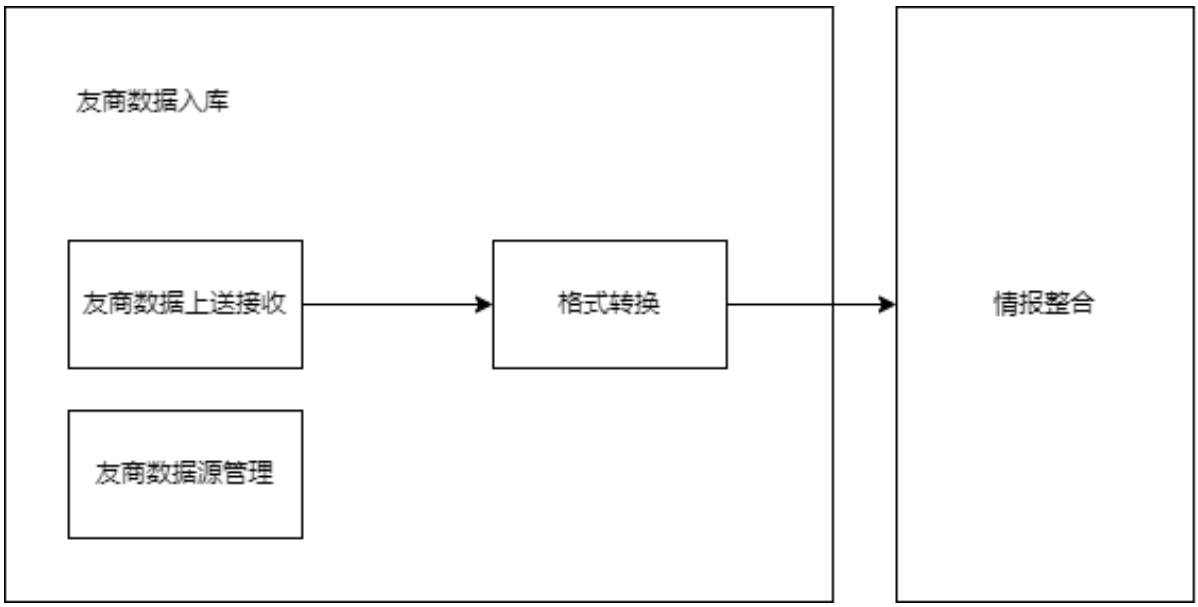
- 多源情报对比：为辅助人工研判，提供云瞻对于某IOC情报收录信息的查询，可以在管理平台快速查看商业源对于某情报的判定结果，支持商业源原始数据的下载；
- 手动情报审核：为辅助人工研判，快速了解该IOC情报在经过情报整合后的结果，可选进行“预整合”处理，了解情报现状；
- 情报人工审核：设置IOC情报字段审核值，以绝对优先级的人工审核源，参与情报整合，达到情报数据人工判断的效果。

友商数据上传

为了丰富云瞻收录情报的数量以及质量，运营同学收集到一些外部情报数据。云瞻提供友商数据上传模块支持外部数据快速导入并送入情报整合，同时可以补充云瞻制作的特征库中的数据能力。

友商数据参与整合时，仍是以一个数据源（Valac源）的身份；通过区分标签类型，适配后续数据来源扩展。

总体逻辑示意图如下：

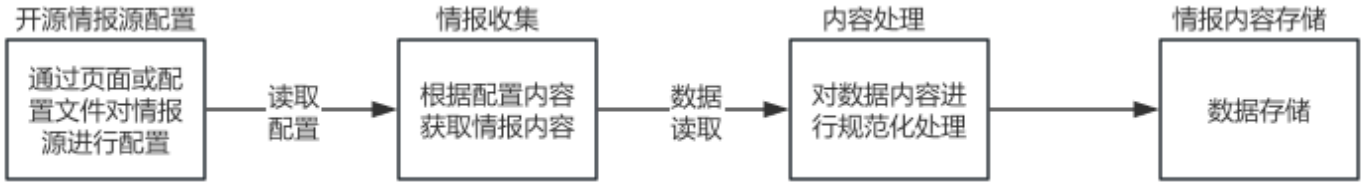


- 友商数据上送接收：云瞻管理平台提供界面，支持友商数据上传，适配不同源、不同IOC情报类型；
- 友商数据源管理：支持新增源内部的区分，适配不同友商上送的区别；
- 格式转换：接收到上送数据后，根据选择内容转换标签等结构，送往情报整合

开源情报管理

网络上的开源情报相对于第三方厂商的提供的专有情报可以提供更广泛的威胁视角，且成本更低，是专有情报非常上的补充，故威胁情报平台要提供对开源情报的收集处理能力。

开源情报管理的主要目的是能够通过配置开源情报的信息，将开源情报的内容收集下来，并通过一定的数据处理之后将开源情报的内容进行存储，为其他服务提供情报能力。故，开源情报管理的整体设计框图如下：



下面对上述流程中的模块进行详细说明：

开源情报源配置

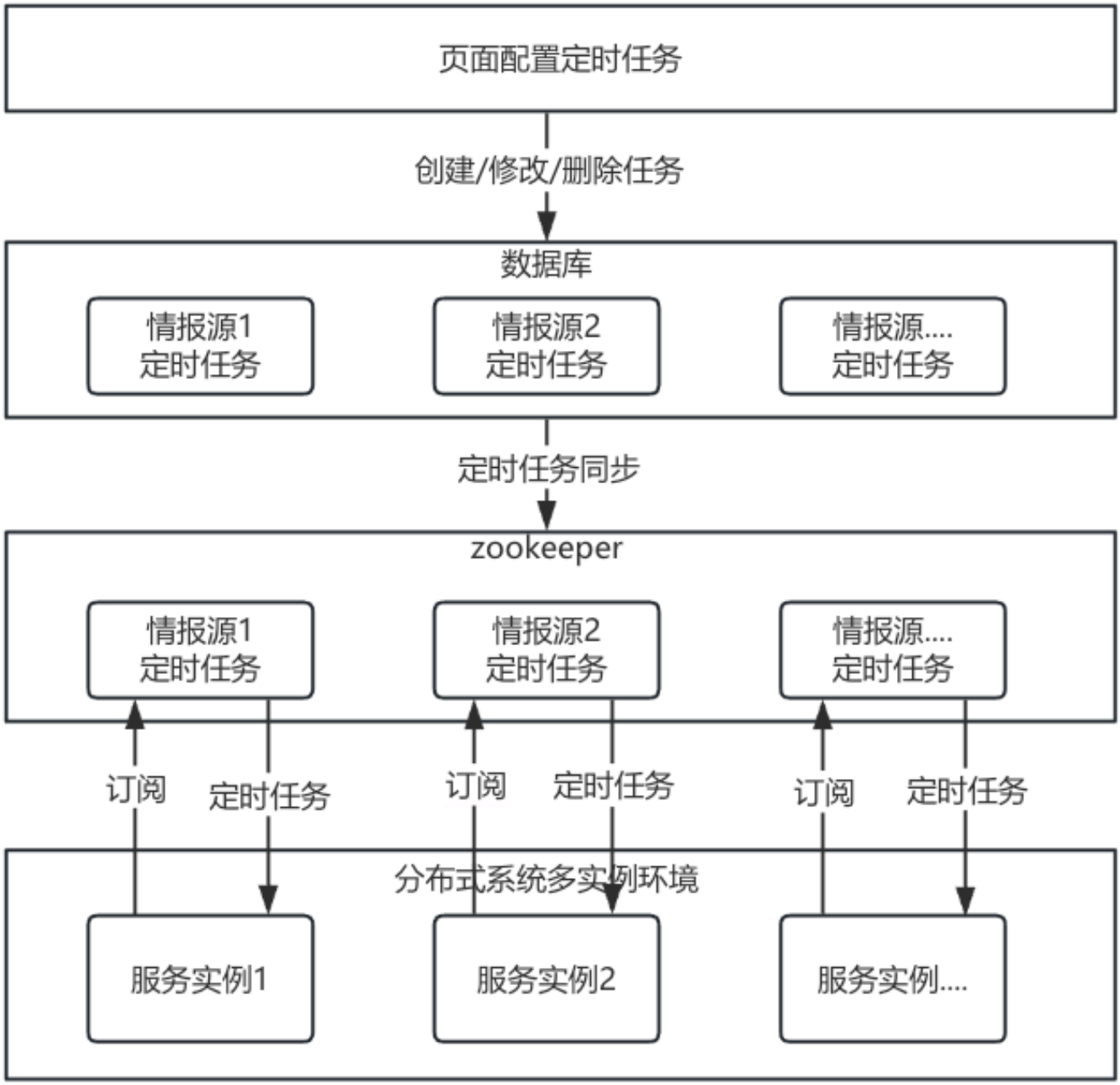
开源情报管理功能需要支持多种情报源，且支持情报源扩展，故需要使用配置的方式对多个情报源进行区分，情报源的配置信息如下：

配置字段	含义
source_name	情报源名称，情报源唯一标识
source-type	情报源类型：0-预定义，1-自动，2-手动
source_stattus	情报源状态：0-未启用，1-启用
source_describe	情报源描述信息
source_file_download_url	预定义和自动情报源指定下载文件的 url
source_file_hdfs_url	情报源离线文件在 hdfs 的存放路径
source_scheduler	情报源配置的定时任务信息 自动情报源通过页面指定 手动情报源不需要指定 预定义情报源通过代码定义
source_result	情报源的设定结果 自动情报源通过页面指定 手动情报源不需要指定 预定义情报源通过代码定义
source_tag	情报源的设定标签

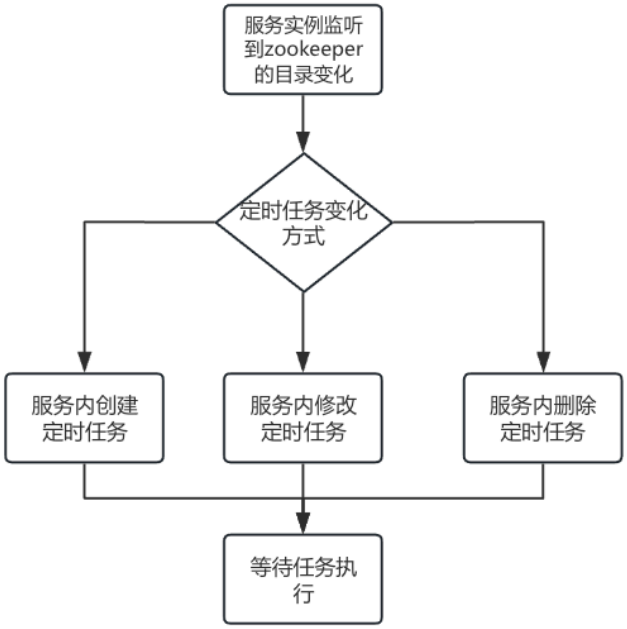
情报收集

情报收集的过程是根据配置文件中source_scheduler的定时任务信息，启动定时任务去下载source_file_download_url的链接，然后将下载内容存储到source_file_hdfs_url指定的hdfs文件地址。

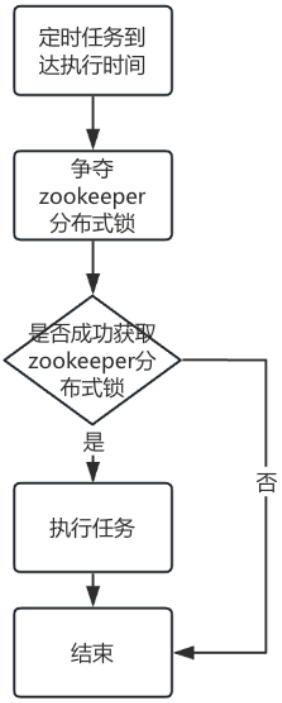
整个情报收集流程是比较清晰简单的，但source_scheduler的定时任务信息是需要可动态配置，且配置完成后即可生效，故在此功能中设计了一个在分布式系统下的定时任务动态调整的方案，整体的方案设计如下：



从页面上进行情报源的定时任务配置，将定时任务写入数据库，并将定时任务同步到zookeeper的目录中。服务实例通过监听zookeeper的目录能够获取到定时任务的信息，下面对服务实例执行定时任务的流程进行详细说明：



每个服务实例内操作定时任务信息流程



每个服务实例执行定时任务流程

1. 每个服务实例通过监听zookeeper的目录变化，获取定时任务信息是创建/修改/删除，根据信息创建、修改或删除服务内的定时任务。
2. 同步步骤1的操作，多个服务实例内会有相同的定时任务在同一时间执行，为了保证相同的情报源定时任务只执行一次，使用zookeeper的分布式锁来保证只执行一次。
3. 文件保存 下载的文件作为临时文件存放到 `/tmp/opensource/<source_name>/<timestamp>- <filename>` 来进行后续处理。其中`<source_name>`从配置文件中取出，`<timestamp>`取文件上传时的时间戳，`<filename>`取下载文件的原始名字数据源。 另外每天下载的文件作为历史文件存储到 hdfs 中，存储路径见 source_file_hdfs_url。

内容处理

文件内容比较：当此次下载的文件与前一次定时任务下载成功的文件完全一样时，不进行查询库和 ioc 时间追溯库的更新，对于监控信息则直接返回状态是成功，导入数量是 0，信息提示 与上一次下载的文件完全一致。 当此次下载的文件与前一次定时任务下载的文件不完全一样时则进行后续的处理。

文件内容补全：因为自动化情报源下载的文件都是单独一列 ioc 的文件，故要将开源情报的内容补充 完整。最终补全之后的文件格式为<ioc, source_name, tag, timestamp>。

ioc：从文件中读取 source_name：从配置中读取

tag：从配置中读取

timestamp：当天时间零点的时间戳。

文件存储

因为 hbase 的列可扩充，在录入相同 rowkey 的相同属性时可进行覆盖，且通过 rowkey 查询速度较快，故使用 hbase 表作为开源情报的查询表，表结构设计如下：

TableName：ti_open_source_table

Rowkey: 以 ioc 的值作为 rowkey，保持该 ioc 的最新信息，使用 rowkey 查询提 高查询速度。

Namespace: cloud-ti。

Table 设计: 列族数量 1，不需要压缩。

ColumnFamily 列族设计：默认使用 cf，不需要多版本控制。

Qualifier 列设计如下：

列名	类型	含义
ioc	String	ioc的值
<source_name>_result	Integer	情报源设定结果0｜1｜2｜3
<source_name>_timestamp	Long	更新的时间戳

情报分析管理

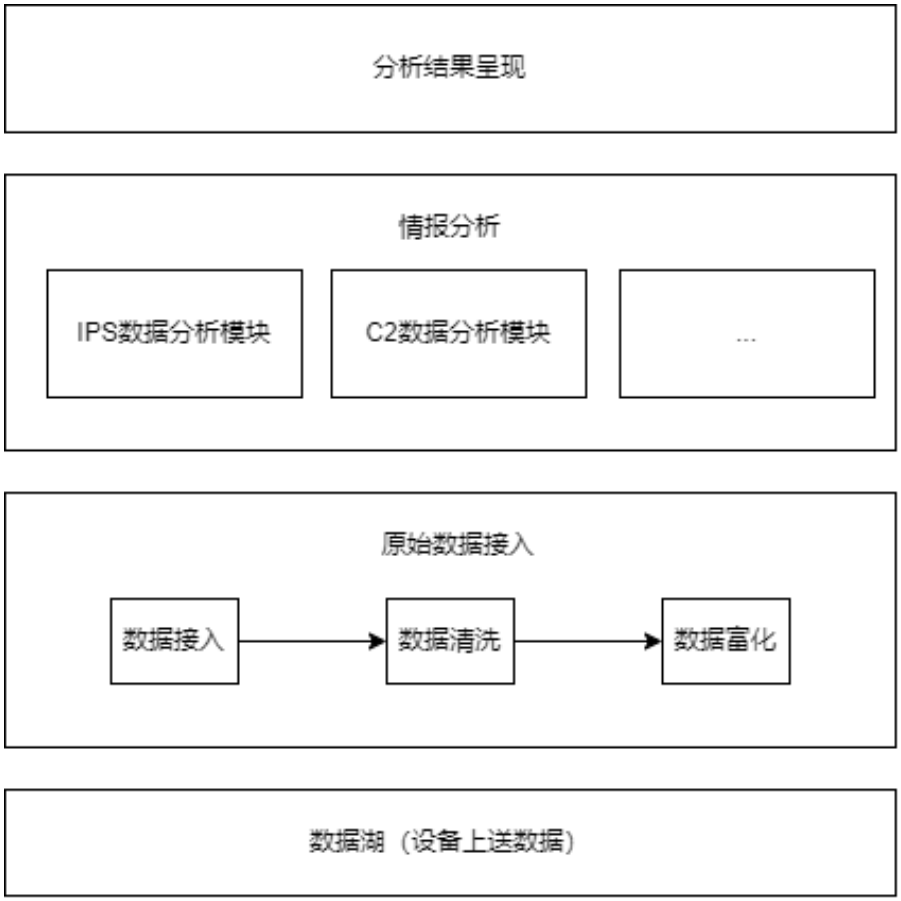
概要描述

现网设备会上送一系列数据至云平台，原始数据存储在数据湖中。设备上送数据由不同引擎控制上传，在一定程度上可以反映现网中的威胁动态，引擎相关同学希望可以通过原始的上送数据，获取到一些威胁信息、或者有利于增强引擎能力的信息，云瞻借助其云产品身份，与云景、数据湖沟通便利，为不同团队提供设备上送数据的分析处理能力，即情报分析管理模块。

功能设计

总体架构

总体架构图如下：



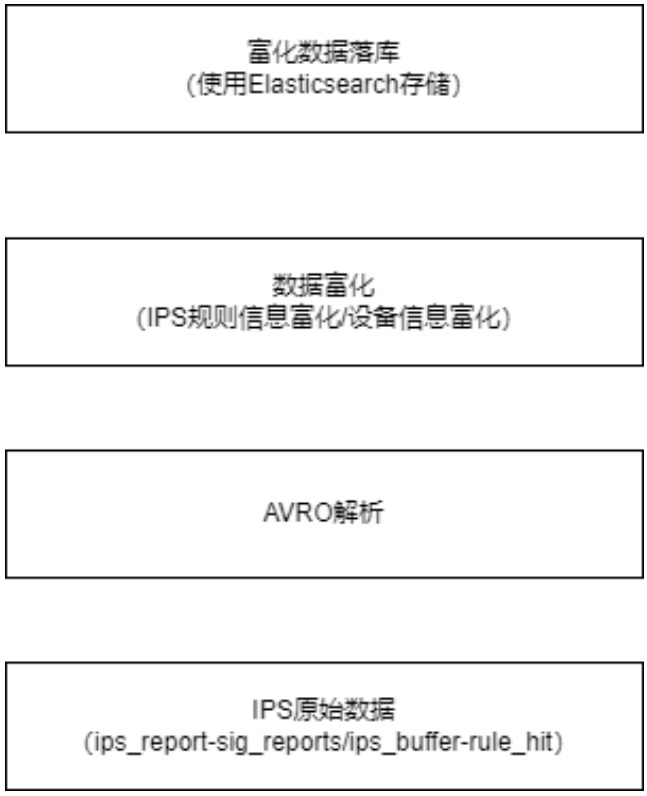
- 云瞻根据相关团队需要，从数据湖中获取对应类型的上送数据；
- 对原始数据进行解析、清洗、富化等基础操作；
- 根据不同分析场景需求，按照既定的可固话的分析流程进行原始数据分析产出；
- 根据产出的数据提供不同的呈现方式。

需要说明的是，分析处理逻辑要求明确、具体可实现，需求相关团队经过探索、验证后，再接入云瞻该模块。相关步骤，如清洗富化等不同分析场景并不完全一致，按需进行。

目前云瞻接入的分析流程有以下几项：

- IPS数据分析
- C2数据分析
- 威胁事件分析-可信规则管理
- AV数据分析

IPS分析



根据IPS规则团队分析需求，云瞻情报分析模块接入IPS引擎上送原始数据进行分析：

- ips_report-sig_reports
- ips_buffer-rule_hit

两部分数据处理逻辑一致：

- 对数据内容中IPS规则信息进行富化；
- 对上送数据的设备信息进行富化；
- 存储进Elasticsearch提供上层展示支持。

对于处理完成的数据，在云瞻管理平台界面，提供以下固化分析场景：

- 聚合统计时间周期内设备上送IPS Buffer数据中的IPS规则ID，计算对应的命中数，按照命中数降序排列展示，并支持攻击buffer数据展示；
- 聚合统计时间周期内设备上送IPS report数据中的IPS规则ID，计算设备的规则命中比例；
- 聚合统计时间周期内设备上送IPS report数据中的设备SN，计算不同设备命中规则的整体情况；
- 支持运营同学根据上述结果对IPS规则进行误报标记；
- 支持生成IPS上送数据的统计周报。

C2数据分析

处理设备上送的C2（botnet）命中数据进行分析：

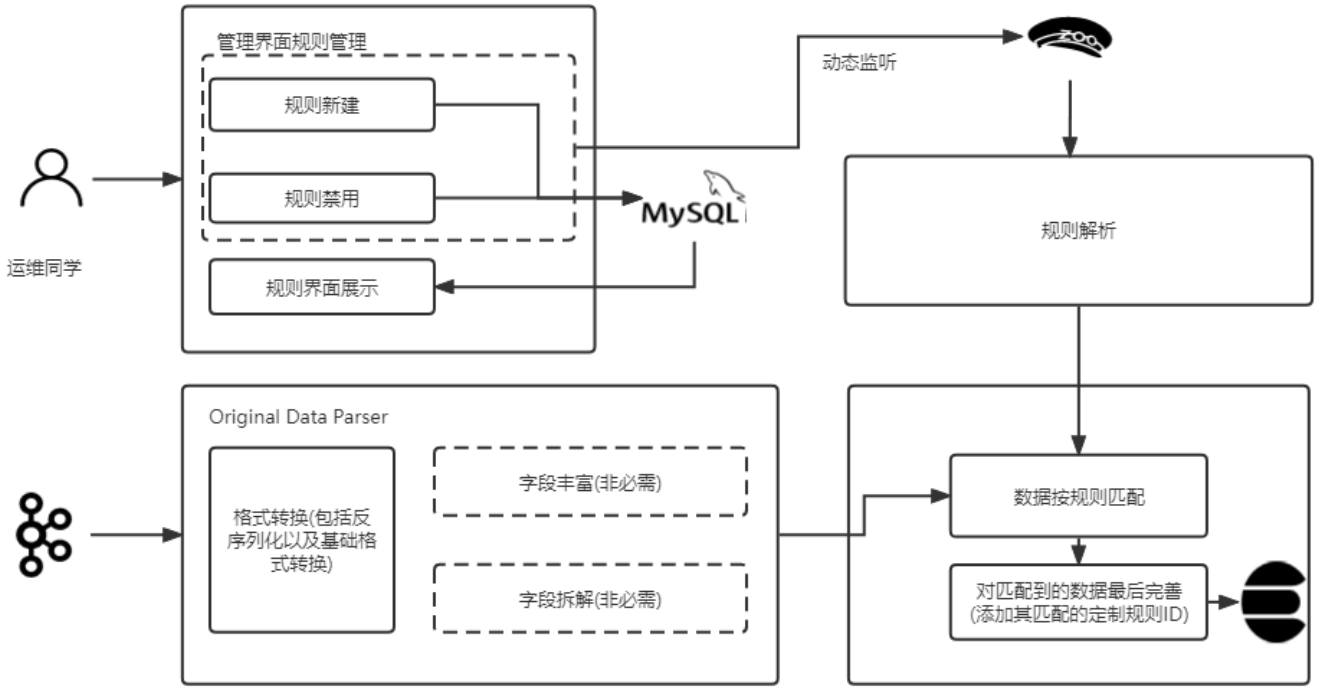
- botnet_report-c2_black
- botnet_report-c2_grey

此处分析流程较为简单，解析原始数据后存储进clickhouse提供查询：

- 统计每周命中IP top10；
- 统计每周命中域名 top10;
- IOC高频命中展示；
- c2 灰、黑数据命中图表展示。

威胁事件分析-可信规则管理

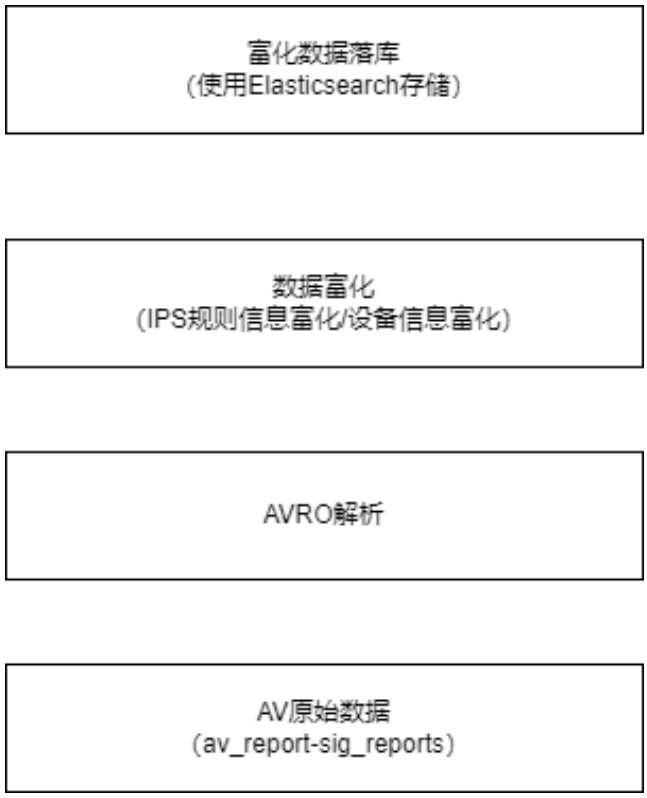
针对设备上送的event-threat_event数据，仅针对其中IPS引擎数据，云瞻固化了可信规则管理流程：



流程处理分为以下几步：

- 规则管理：运维同学根据可信的IPS规则信息，或者其他威胁事件字段，制定可信规则；
- 原始数据解析：云瞻接收原始的threat-event数据，进行解析、富化，用于可信规则匹配；
- 可信规则匹配：将处理后的原始数据，与当前全部可信规则进行匹配，如果与可信规则设置项信息一致，则视为命中，将数据落库；
- 命中数据分析：从命中数据中提取源目的IP等IOC情报，参与云瞻情报整合。

AV数据分析



处理设备上送的AV原始数据：

- av_report-sig_reports

数据处理逻辑：

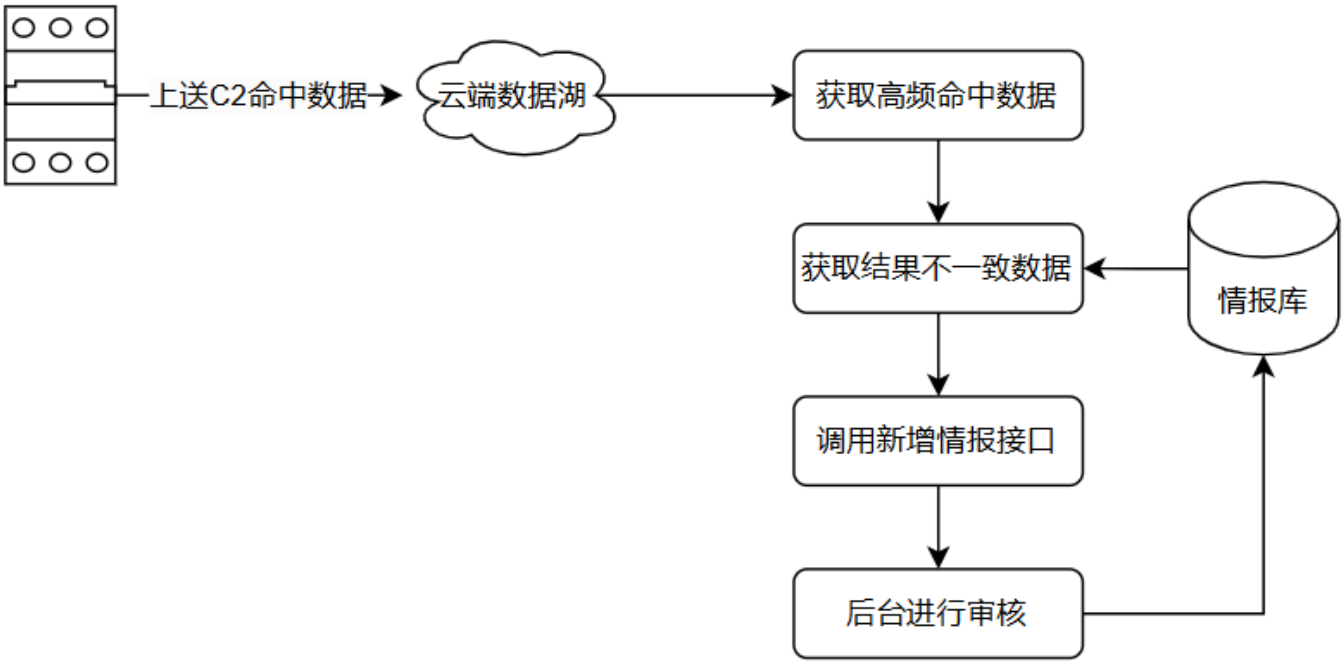
- 对上送数据的设备信息进行富化；
- 存储进Elasticsearch提供上层展示支持。

在云瞻管理平台界面展示：

- 最近一周设备命中MD5 top10

ioc高频命中分析

设备会收集命中的C2黑名单数据并上传到云端数据湖，威胁情报平台监听并消费相应kafka通道内的数据，将其保存在云瞻本地的数据库中。由于误报以及时效性等问题的存在，云瞻情报库与C2特征库可能会对部分情报产生结果上的冲突。针对这种情况，云瞻后台会定时获取C2库中高频命中的IOC，从中筛选出与情报中心结果不一致的条目，并且发送给运维进行整合。

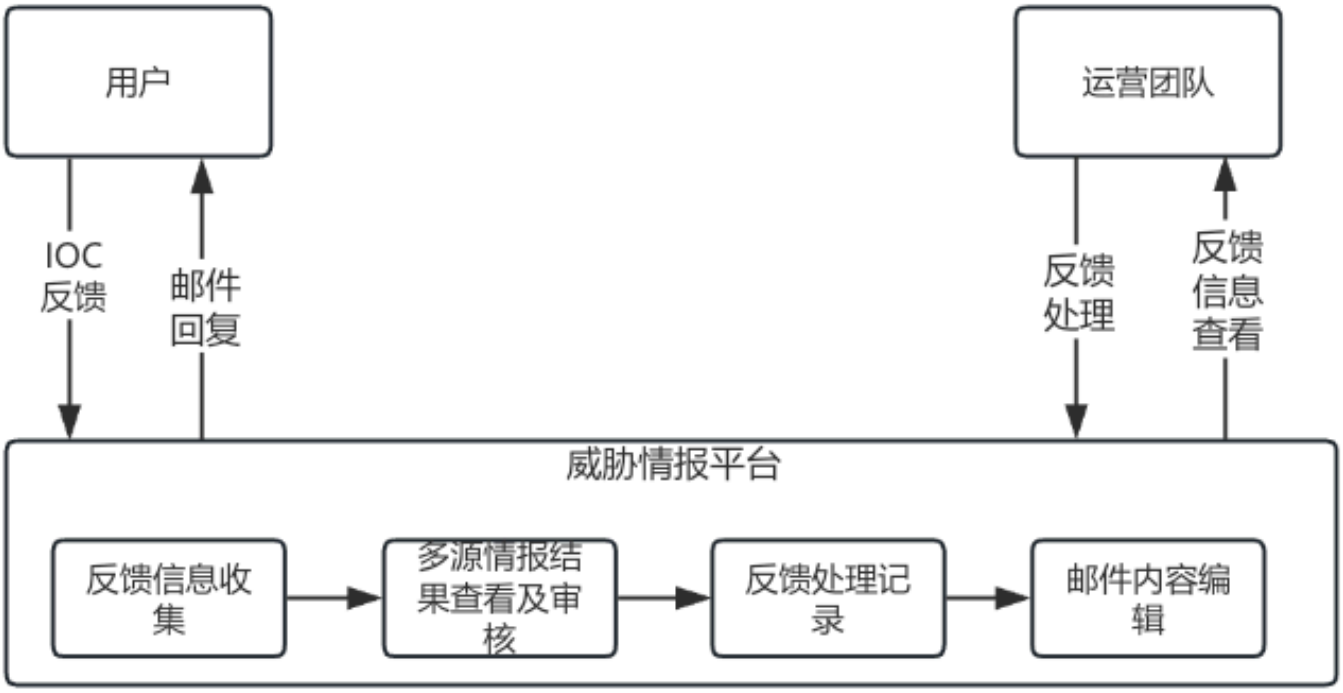


整个流程的运行逻辑具体为：

1. 云瞻定时获取过去四个小时内高频命中的IOC
2. 对比核心情报库获取结果不一致的情报
3. 调用新增情报接口，查询该情报在各数据源上的结果并进入待审核队列
4. 运维完成审核，情报进入整合流程
   在上述的流程中，高频的标准可以在后台页面进行配置，位于后台管理页面中的**情报管理-->云景IOC分析管理-->IOC命中结果不一致情况**，可配置项包括命中数阈值以及IOC数量。在调用新增情报接口之后，后台会将情报数据先存入Hbase，之后定时将情报数据取出后查询各数据源的结果，之后存入**新增情报库**中等待运营审核。

情报反馈管理

威胁情报反馈机制有助于评估情报质量、及时发现误报漏报情况，避免造成更大的恶劣影响；另一方面也尽可能多的为后台分析收集证据信息。故，威胁情报平台需要支持情报反馈的渠道，整体的设计框图如下：

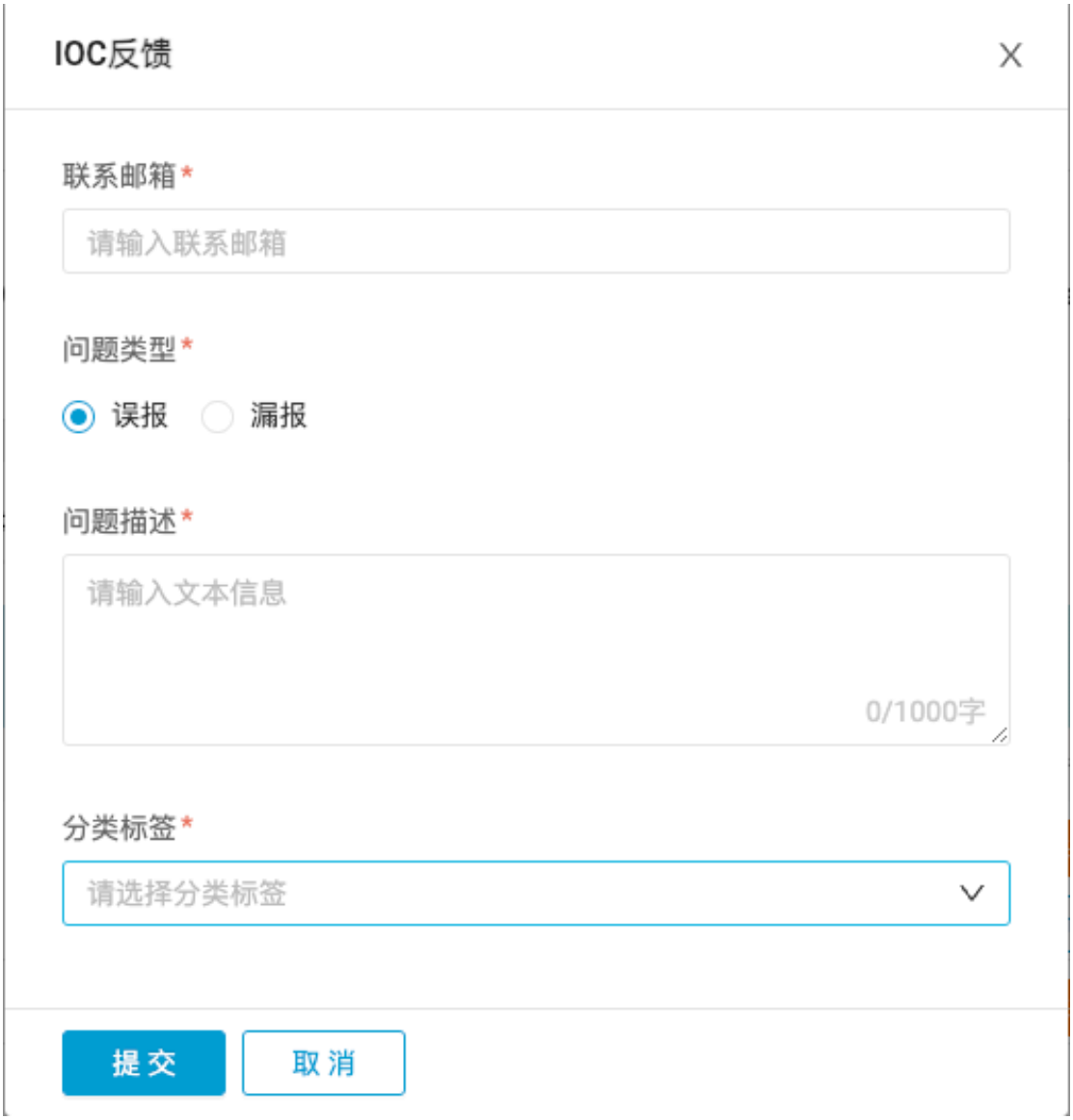


1. 在包括威胁情报平台查询页面、设备查询跳转页面在内的IOC查询结果页面中，新增IOC反馈（Feedback）按钮，点击IOC反馈按钮后，弹出填写框。用户必填<联系邮箱>、<问题类型>、<问题描述>、<分类标签>的信息等。
2. 对于用户填写的反馈信息，威胁情报平台将反馈信息存储到数据库中。
3. 运营团队在威胁情报平台的后台管理页面查看并处理反馈的IOC
4. 对运营团队处理IOC的记录进行邮件生成，并回复给用户。

对于以上四点进行详细说明如下：

用户反馈信息收集

用户反馈的信息如下：



以上是用户反馈的信息，对于以上信息使用mysql表进行存储，存储结构设计入下：

表名：t_ti_feed_back_info

表描述：用户IOC反馈的记录表

字段名	类型	描述
id	int	自增id，主键
value	varchar（255）	反馈ioc的值
type	varchar（6）	反馈ioc的类型
user_name	varchar（50）	用户名，只有web页面反馈时才有
sn	varchar（50）	设备sn，只有设备页面反馈时才有
classification_tag	int	分类标签id
problem_type	int	问题类型 0 误报 1 漏报
details	text	反馈的描述信息
email	varchar（50）	反馈人邮箱
feeb_back_time	bigint	反馈时间
handler_status	int	处理状态 0：未处理，1：处理
email_status	int	邮件发送状态 0：未发送，1：发送

多源结果查看和审核

运营团队在对用户反馈的ioc进行处理时需要参考多情报源的结果，需要提供多源情报的结果查询能力，此能力在多源情报管理-->人工审核的功能里已经实现，故此处直接将多源情 报管理中查看多源情报结果和人工审核的功能复用到反馈处理的页面

反馈处理记录

反馈信息的处理内容主要包括以下几点：

1. 反馈类型的处理：

   问题类型是误报：误报，非误报 两种结果

   问题类型是漏报：漏报，非漏报，不能确定三种结果

1. 反馈标签的处理：

   反馈标签合理，反馈标签补充，反馈标签可信度无法确认 三种结果

   当处理的结果为反馈标签补充时，则后续的反馈标签补充字段为必填。

1. 反馈标签处理：字符串形式记录
2. 多方判定：字符串形式记录
3. 相关报告：字符串形式记录
4. 补充：字符串形式记录

处理的内容记录到mysql数据库中，存储结构设计如下：

字段名	类型	描述
id	int	自增id，主键
problem_type	int	问题类型 0 误报 1 漏报
result	int	类型处理结果 0: 非误报 ,1:误报， 2：非漏报，3：漏报，4：无法确定
tag_result	int	标签处理结果 0：反馈结果合理 1：反馈标签补充，2：反馈标签可信无 法确认
tag_comments	varchar(255)	标签补充
muti_decision	varchar(255)	多方判定
report	varchar(255)	相关判定
comments	varchar(255)	补充信息
value	varchar（255）	ioc的值
type	varchar（50）	ioc的类型
handler	varchar（50）	处理人
process_id	int	关联的用户反馈表的记录id，t_ti_feed_back_info的id

邮件内容编辑和发送

将运营团队填写的处理结果，标签补充，多方判定等信息，填写到邮件模版中，通过pushmessage模块的邮件发送服务，将处理之后的信息邮件到用户。

标签管理

概要描述

云瞻维护了IOC的标签系统，用于管理IOC和标签之间的关系，一个IOC可以关联多个标签，用以体现IOC本身的一些概要信息。

在IOC情报整合流程中，IOC会被标记上若干标签。通过标签，可以实现不同场景下对IOC的筛选、分类以及查询。

云瞻的标签系统包括以下几个部分：

- 标签表：存储云瞻情报标签的完整信息，包含名称、描述、分类、恶意程度等；
- 标签映射表：存储标签的映射关系，通过映射关系，可以将外部来源标签转化成云瞻内部标签；
- 标签处置建议（COA）表：存储标签对应的处置建议信息，通过处置建议ID与标签进行关联。

功能设计

标签

标签作为一个实体集合，存储在云瞻的数据库中。同时云瞻在管理界面提供标签的管理支持，如新增、筛选等功能。

一个标签有以下字段信息：

字段	含义	备注
id	标签ID	唯一标识云瞻标签库的一个标签，用于其他业务以及别的团队合作时标签数据的识别信息。
namespace	命名空间	
key	分类	
hs_value_en	标签英文名	
hs_value_cn	标签中文名	
severity	危险级别	可选严重、高、中、低
type	类型	分类标签、普通标签、家族标签、团伙标签、场景标签
visible	用户可见性	0为可见,1 为不可见
description_cn	中文描述	
description_en	英文描述	
coa_id	处置建议id	默认为1,表示无处置建议
create_time	创建时间	
update_time	更新时间	

需要说明的是：

- 命名空间和分类是业务自定，可选值确定，需要有明确含义；
- 在一些场景下，命名空间、分类和标签英文名的三元组合也可以代表一个标签；
- 处置建议ID代表着一条处置建议信息，与标签分开存储，同一个处置建议可以与多个标签关联。

除了上述用户可见属性外，标签还存在两个内部属性：

- 热点标签：用于云瞻查询界面标签展示。云瞻界面展示IOC标签时，会根据标签的类型以及热点信息，区分展示；
- 分类标签优先级：此属性仅分类标签存在，用于在智源特征库制作时标签选取。

标签映射关系

标签映射关系用于把数据源标签信息转化为山石标签。标签映射关系核心字段如下：

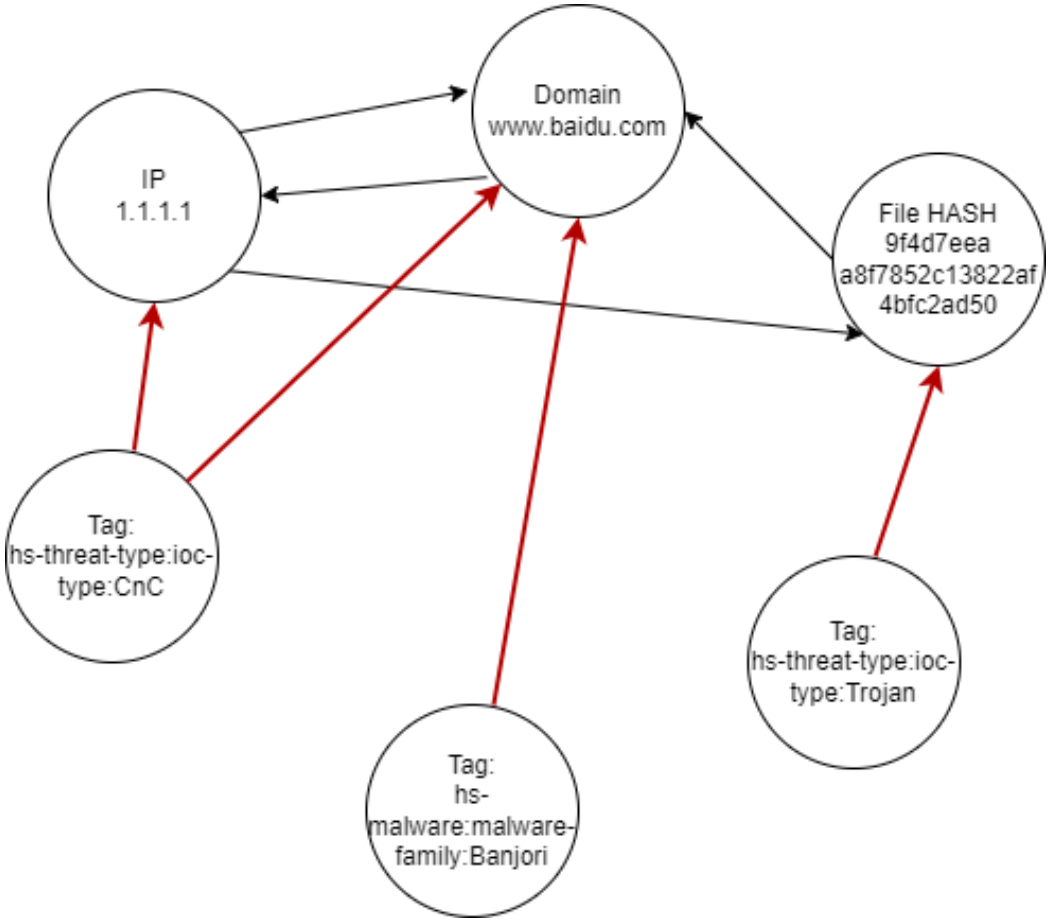
字段	含义	备注
source	数据来源	
key	分类	
value	被匹配字段	
tag_id	映射标签ID	
valid	生效状态	

- 数据来源和分类：用于标识数据源标签的来源信息，根据数据源来制定，如腾讯源的数据，数据开源标记为"TENCENT-ti"；
- 被匹配字段：即数据源方提供的标签名称；
- 映射标签：即云瞻标签ID；
- 生效状态：由于数据源提供的标签不可控，数量较多，对于已知的明确映射关系才会标记为生效。

情报与标签

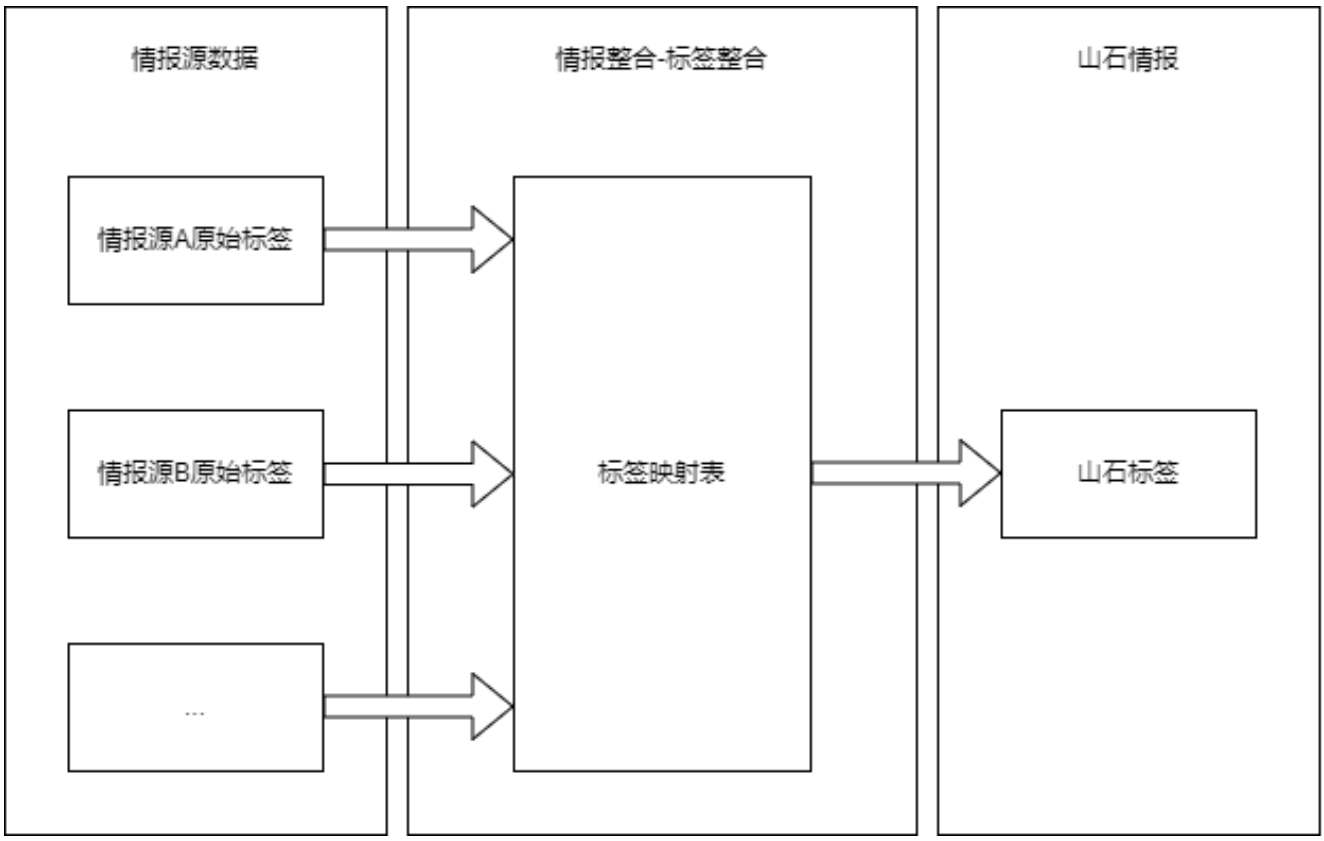
情报与标签关系

情报标签设计为威胁情报库中的威胁情报对象，和其他的情报对象，如IP、域名等为同一级别，在图数据库中实现为数据顶点，通过图的边与IP、域名等情报对象关联。相关示意图如下：



情报标签映射处理

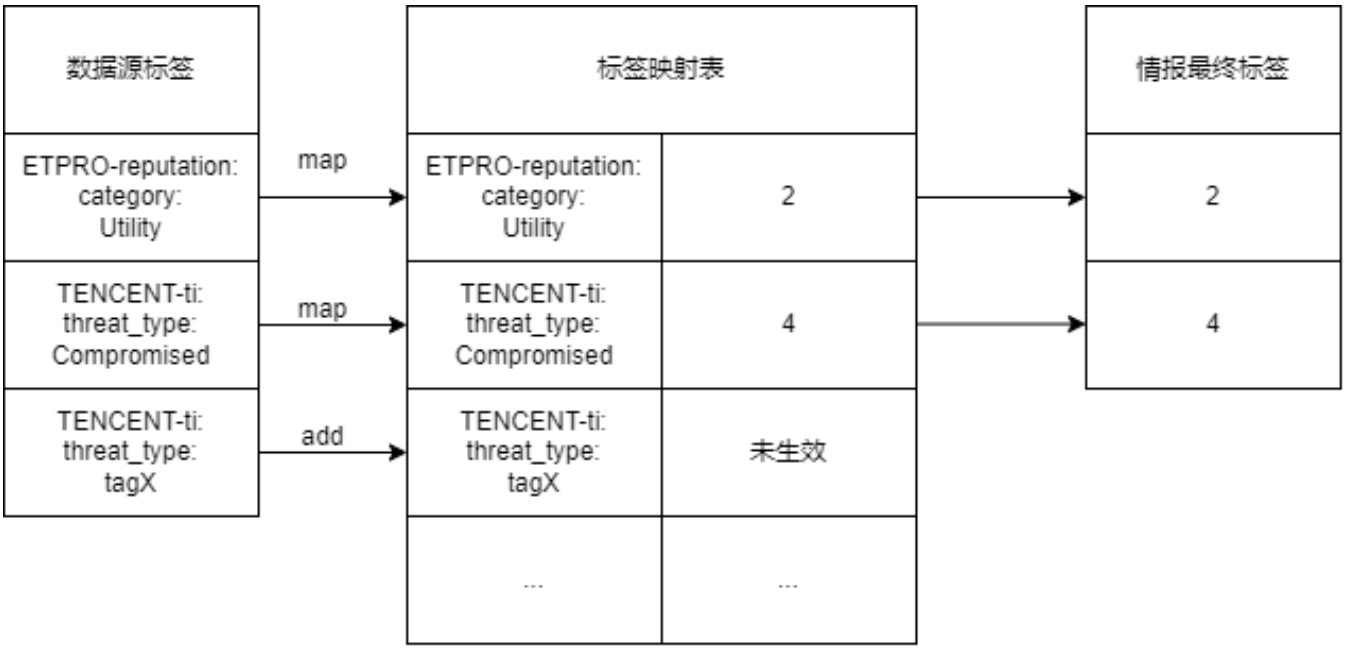
在情报整合时，会根据标签映射关系完成数据源标签到云瞻标签的映射：



相关信息解释如下：

- 情报源数据：代表参与情报整合的数据源数据，具有统一格式；
- 山石情报：代表情报整合后的情报信息，包含标签数据。

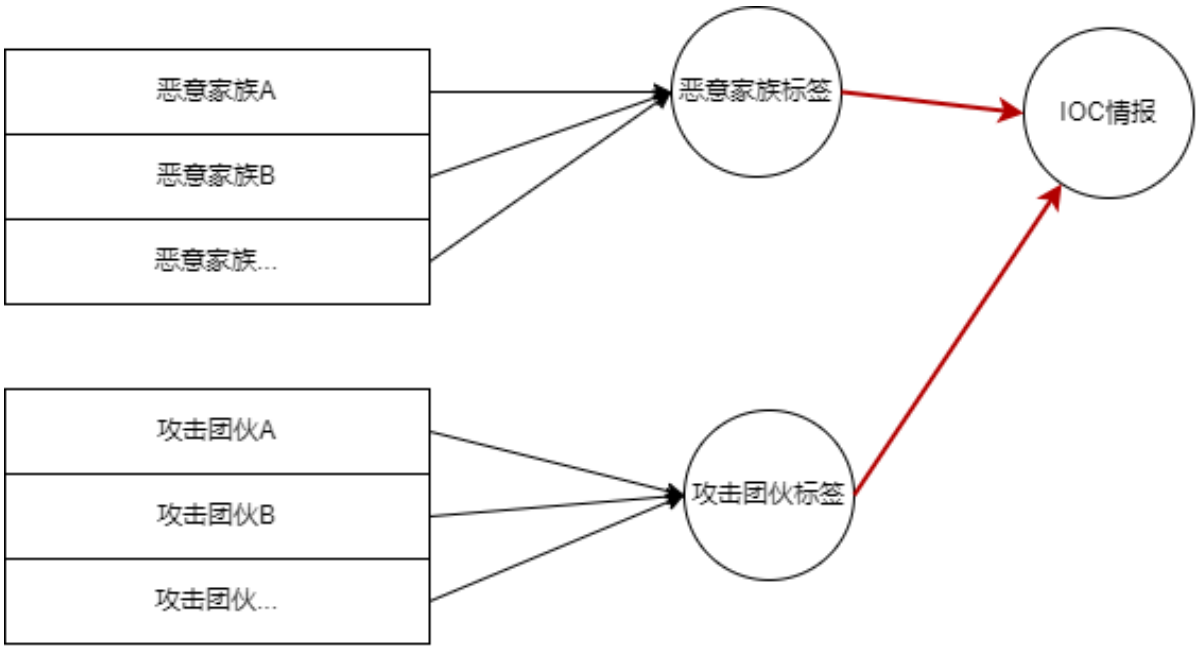
标签映射流程如下：



- 数据源标签信息实际为标签映射的key，根据此key，在标签映射表中寻找对应的value，即为云瞻标签库标签ID；
- 对于标签映射表中不存在的key，整合分析流程中会将其添加进标签映射表，标记为未生效，此时没有value信息；
- 运营人员持续维护标签映射表，处理未生效映射关系，将其与具体标签关联，置为生效状态。

知识库与标签

云瞻维护的恶意家族与攻击团伙知识，通过标签作为桥梁，与情报进行关联。



云瞻的恶意家族和攻击团伙在新增时，会首先对应增加同名的恶意家族标签（标签类型为家族标签）和攻击团伙标签（标签类型为团伙标签），在恶意家族和攻击团伙中保存对应标签ID，形成关联。

如果某IOC情报关联某个恶意家族或者攻击团伙，则在云瞻核心库存储时，存储该IOC情报与对应的恶意家族标签和攻击团伙标签的关联，根据标签间接与原始的恶意家族和攻击团伙关联。

特征库与标签

前面提到，云瞻业务可以根据标签，来对情报进行筛选，特征库制作即为一个典型示例：根据特征库所需情报类型，凭借标签筛选，关联获取到云瞻核心库中的对应情报信息。

知识库管理

ATT&CK

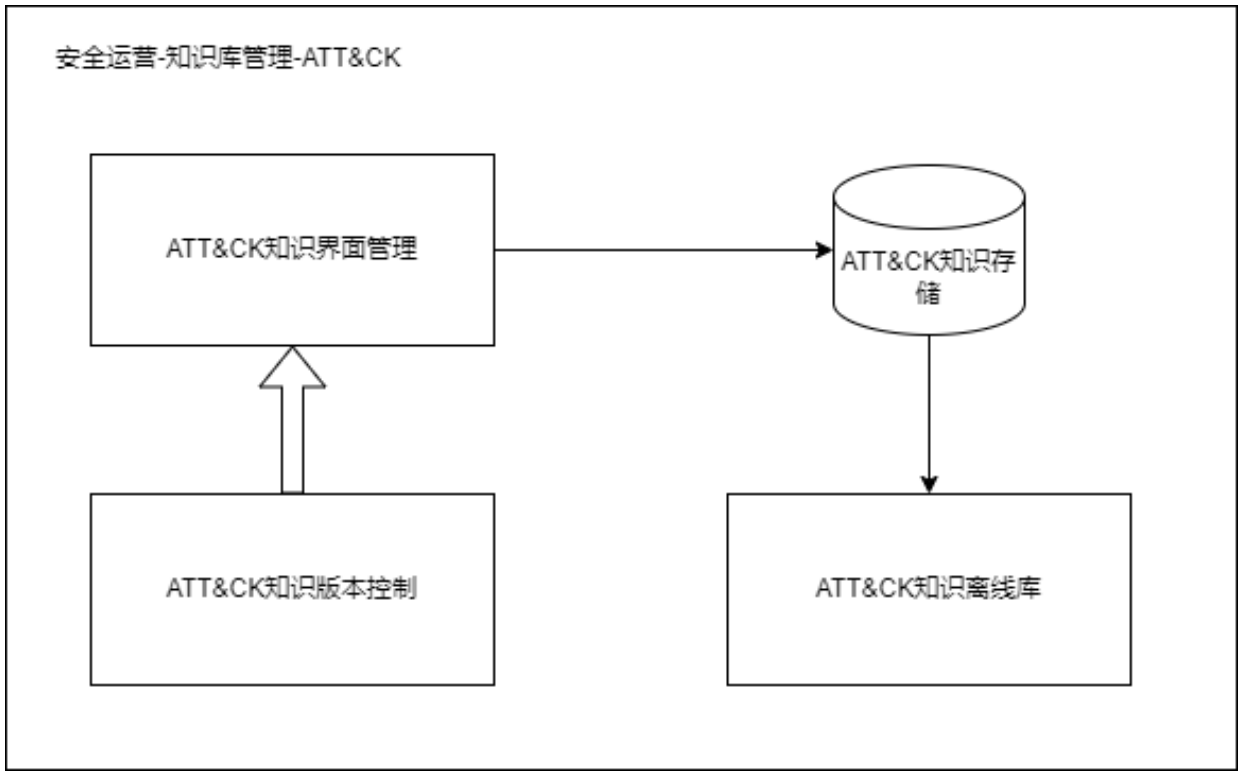
概要描述

MItre ATT&CK(Adversarial Tactics,Techniques,and Common Knowledge) 是一个攻击行为知识库和模型，主要应用与评估攻防能力覆盖、APT情报分析、攻击链还原、威胁狩猎及攻击模拟等场景。应用ATT&CK框架，可以加速对威胁的检测、分析和响应过程。

由于公司检测类设备，如智源、IPS、沙箱等均需要支持ATT&CK知识库并将威胁事件与ATT&CK的技术点进行映射，为了维护公司统一的ATT&CK知识库，云瞻实现了ATT&CK的统一管理，并提供标准的离线知识库给各个产品线使用。

功能设计

总体架构



云瞻对于ATT&CK知识库的维护，主要分为以下几个部分：

- 版本控制：构建ATT&CK官方版本向山石内部的映射，便于不同产品线的统一知识输出；
- 界面管理：主要负责ATT&CK知识的更新以及维护；
- ATT&CK离线库：根据存储构建各个产品线使用的知识库。

版本控制

云瞻统一维护的ATT&CK知识库采用 A.B 格式版本，其中A版本与官方版本对应，从版本1开始依次递增；B版本为小版本，当对当前A版本进行修改时，如汉化数据补充、数据内容新增时，对应更新B版本。

界面知识管理

对于官方的知识，云瞻暂时只处理以下几项：

- 技术（Techniques）
- 战术（Tactics）以及战术链（chain）
- 处置建议（Mitigations）

同时云瞻支持展示ATT&CK攻击矩阵。

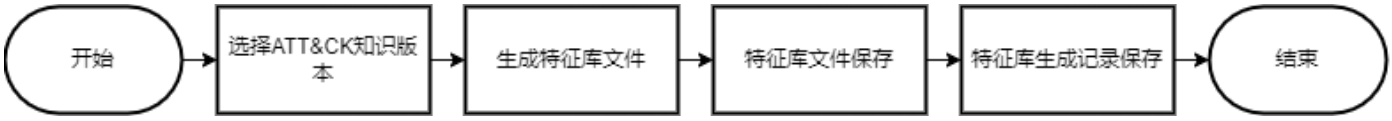
云瞻管理平台支持查看、编辑技术、战术和处置建议信息；支持新增知识版本并全量上传数据。

ATT&CK离线库

为满足产品线加载离线ATT&CK知识，云瞻输出ATT&CK离线库。

制作流程

云瞻提供界面，选择对应的ATT&CK知识版本，对应生成其离线库。



版本控制

离线库采用A.B.C的格式：

- A：对应该版本离线库使用的ATT&CK知识的A版本；
- B：对应该版本离线库使用的ATT&CK知识的B版本；
- C：离线库小版本。

离线库下发

ATT&CK离线库的下方形式与特征库下发形式一致，云瞻提供API接口，由使用方主动获取。

对不同的使用团队做出区分：

使用方	接口方法	接口名	
StoneOS团队	GET	/api/library/attck	
智源团队	GET	/api/library/attck/isource	

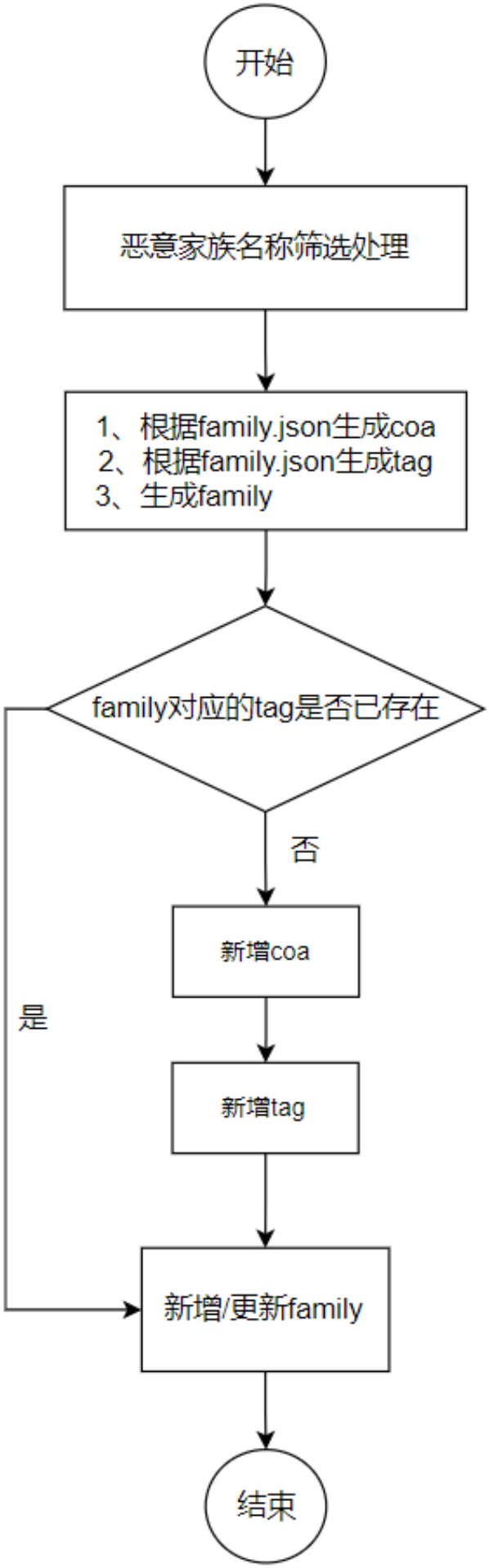
需要特别说明的是，对于沙箱设备，云瞻直接提供可用的ATT&CK离线库包，即云瞻为沙箱设备获取ATT&CK离线库的updateserver。此时由于特征库的制作和updateserver处对原始库处理的逻辑均在云瞻，两处直接不再直接提供API接口交互，云瞻直接提供设备的获取接口。相关接口如下：

使用方	接口方法	接口名	
沙箱设备	GET	/signature/attck/download	

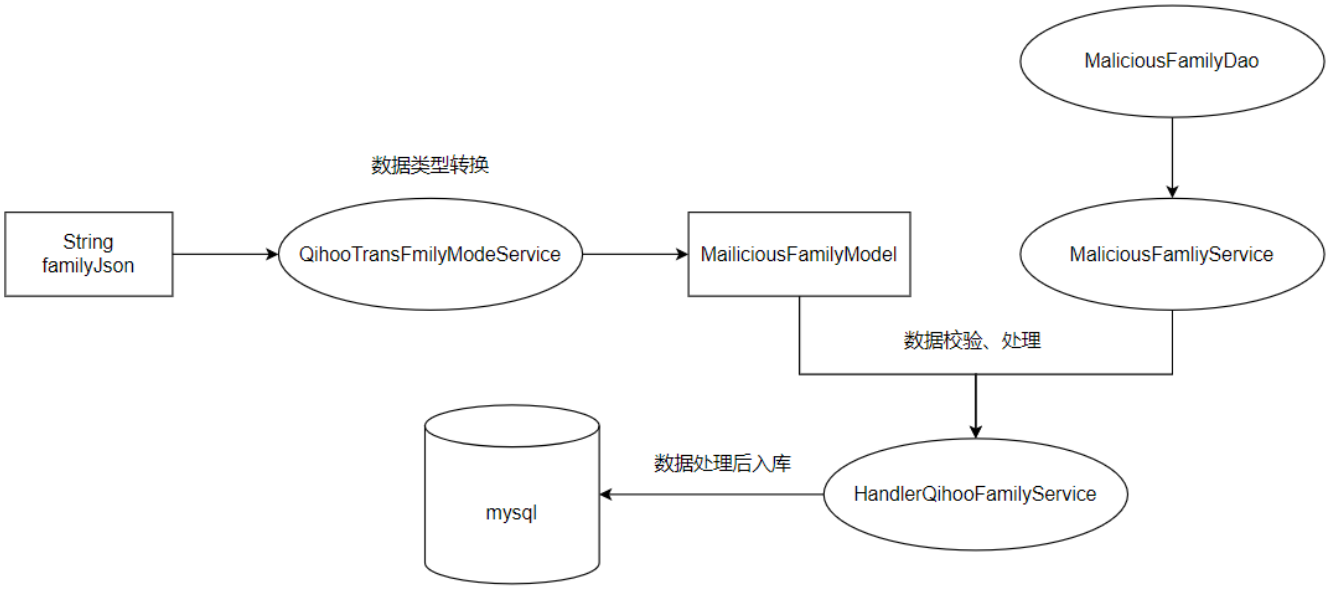
恶意家族&攻击团伙

威胁情报平台支持维护恶意家族&攻击团伙知识库，支持将360离线库中的恶意家族&攻击团伙信息（family.json/group.json)导入tip平台的恶意家族&攻击团伙知识库中，将恶意家族&攻击团伙信息中的响应处置建议入到处置建议知识库(coa库)中，再通过tag进行关联（tag中包含coa的id）

TIP平台导入360家族/团伙情报数据，360的恶意家族/攻击团伙情报数据每日更新，采用diff昨天与今天的文件产生的增量，对数据进行更新，以恶意家族业务进行举例，流程如下：


开始
恶意家族名称筛选处理
1、根据family.json生成coa
2、根据family.json生成tag
3、生成family
family对应的tag是否已存在
否
是
新增coa
新增tag
新增/更新family
结束

数据处理流程如下；



1. 获取360恶意家族数据并对数据进行解析，提取恶意家族信息(family)、处置建议信息(coa)
2. 处置建议信息入coa库
3. 而已家族信息入tag库(family tag)
4. 将tag 与coa进行关联(表之间用id进行关联)

漏洞

漏洞库的实施分成三个步骤：

1. Step 1：构建对内的线上漏洞知识库运营平台，围绕安全运营人员的漏洞维护工作展开，搭建漏洞知识库基础并不断维护、补充；√目前已实现
2. Step 2：拥有漏洞库输出到防护单品的能力，将公司能够防护的、紧急的、热点的漏洞下发到安全设备上，配合设备完成本地安全运维；
3. Step 3：建设对外安全漏洞平台平台，支持用户通过漏洞名称、CVEID、CNVDID、攻击方式等进行漏洞查询。

威胁情报运维中心的知识库管理新增漏洞知识库，目前已支持的功能：

1. 支持运维人员文件批量上传漏洞、在线增、删、改、查漏洞。
2. 支持全量导出、选中部分导出以及将检索后的内容导出三种方式，导出的格式为 CSV 文件

特征库管理

> 基于上述介绍的特征库框架下

1. 对特征库及其数据源的配置进行增删改，同时保存操作记录
2. 根据已完成的配置来触发生成特征库
3. 特征库及其相关联的数据源生成记录的展示
4. 特征库数据的统计信息，目前已支持 IOC 子类型、数据源、分类标签的数量统计

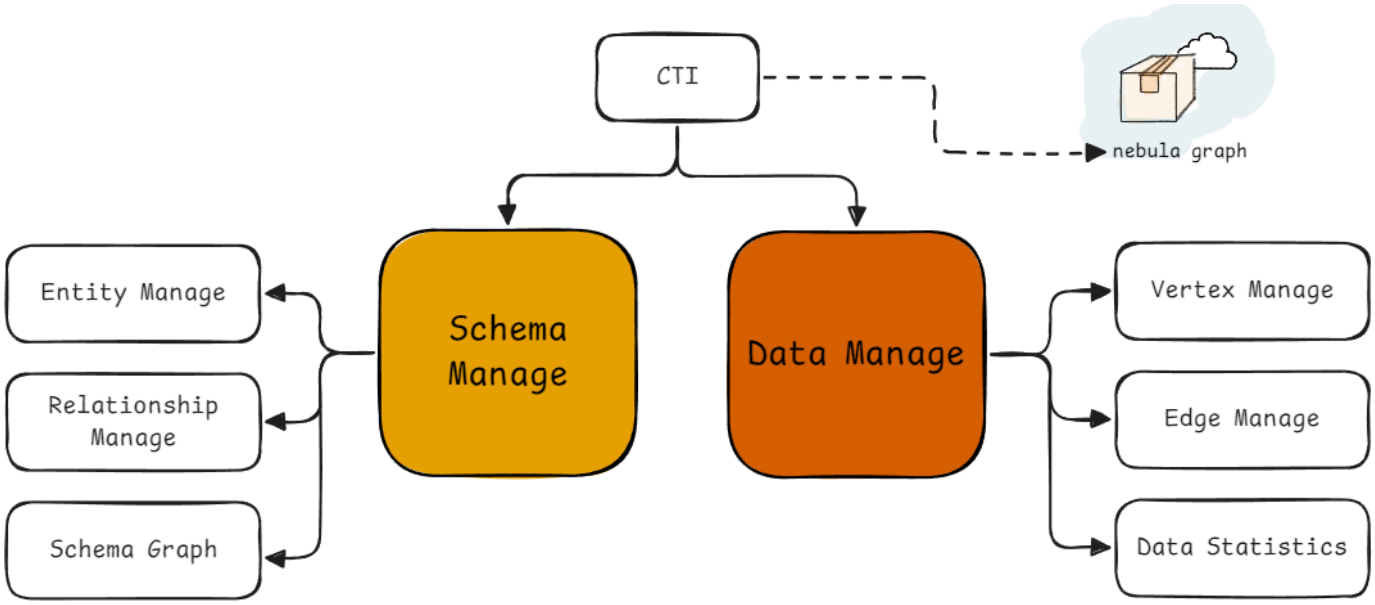
CTI 平台

为解决以下问题

1. 威胁组织管理缺失，无法体系性对最新威胁组织进行跟踪、狩猎和管理。
2. 无报告管理平台，缺少对公开情报的管理。
3. IOC 缺少威胁组织关联，家族关联，恶意软件关联等，不利于后期 IOC 归属、溯源和样本内部扩线等

本项目是基于图数据库的威胁情报运维管理平台，安全人员前期完成对威胁知识库的中的威胁实体、关系的定义，和整理当下急需的安全场景需求。由开发人员借助图数据库以及对安全需求理解完成对威胁实体的管理开发工作。该平台有以下三点特征：

1. 基于图数据库管理威胁知识，非简单静态存储。
2. 弥补了当下内部 IOC 管理功能、报告管理缺失等问题。
3. 解决了内部 Hillstone-CTI 数据安全性和性能瓶颈问题。
4. 推进新技术研究院攻击知识图谱设计建设工作。



当前能力

运维功能：

1. 对图数据库中的实体和关系以及其下的属性进行增删改查
    1. 实体包括名称、层级、描述以及属性信息
    2. 关系包括名称、描述以及属性信息
    3. 属性（实体和关系都有）包括名称、类型、是否允许为 NULL 以及备注信息
2. 展示整个数据库的 Schema 图谱，可以清晰的梳理各个实体之间的关系
3. 对图数据库中的点和边以及其下的属性进行增删改查
4. 对图书库中各个实体和关系下点和边的数量统计
5. 扩展部分实体的单独展示列表

对外能力：

1. 丰富目前云瞻的一些威胁情报描述信息
2. 提供报告查询

产品目标

1. 目前平台内容都是些描述性信息，缺少重要的 TKG 和 AKG 的知识图谱
2. 希望能以一些信息为跳板，查询到某个攻击行为匹配的攻击链路，对恶意工具的下一阶段做一些预测

目前已有的数据

1. MalwareTools 恶意软件工具

   以工具名称为标识，主要包含别名、恶意软件类别/类型、中英描述、软件 ID 等相关信息
2. Attck 入侵者战术、技术和共有知识库

   以 ATT&CK 技术 ID 为标识，主要包含版本号、技术名称和展示 ID 等相关信息
3. Report 威胁报告

   以报告名称为标识，主要包含中英名称、中英摘要、报告类型和等级以及报告原文等相关信息
4. Org 厂商-机构

   以厂商名称为标识，包含别名和公司类型
5. ThreatActor 威胁组织

   以威胁组织家族名称为标识，主要包含别名、ATTCK 组织 ID、内部组织 ID、描述、行业、类别等相关信息
6. Country 国家地区信息

   包含中英简称、英文名以及字母代码
7. Vulnerability 漏洞信息

   以 CVE-ID 为标识，主要包含中英漏洞名称、来源、危害等级、中英描述、涉及产品等相关信息
8. Operation 安全行动

   以名称为标识，包含事件相关时间、描述、行动威胁等级以及处理状态等相关信息
9. Ioc

   以 Ioc 值为标识，包含类型等信息

信息关联

1. Operation -> MalwareTools、Ioc、ThreatActor、Vulnerability、Country、Report
2. Report -> Operation、Org、Attck、Ioc、MalwareTools、ThreatActor、Vulnerability、Country
3. MalwareTools -> Report
4. Ioc -> Operation、MalwareTools、ThreatActor
5. ThreatActor -> MalwareTools、Country、Report

上述是以目前单独做管理的实体出发所延申的关联，实际上只要两个实体之间有联系，无论是直接还是间接，亦或是谁指向谁，都可以通过相关检索条件查询所需要的信息

用户能够通过任何一个已知的信息，如 Ioc、Report 等查询出所相关联的所有信息，并根据其所重点关注的部分如 ThreatActor 等，再次进行二次查询

目前能考虑支持的场景

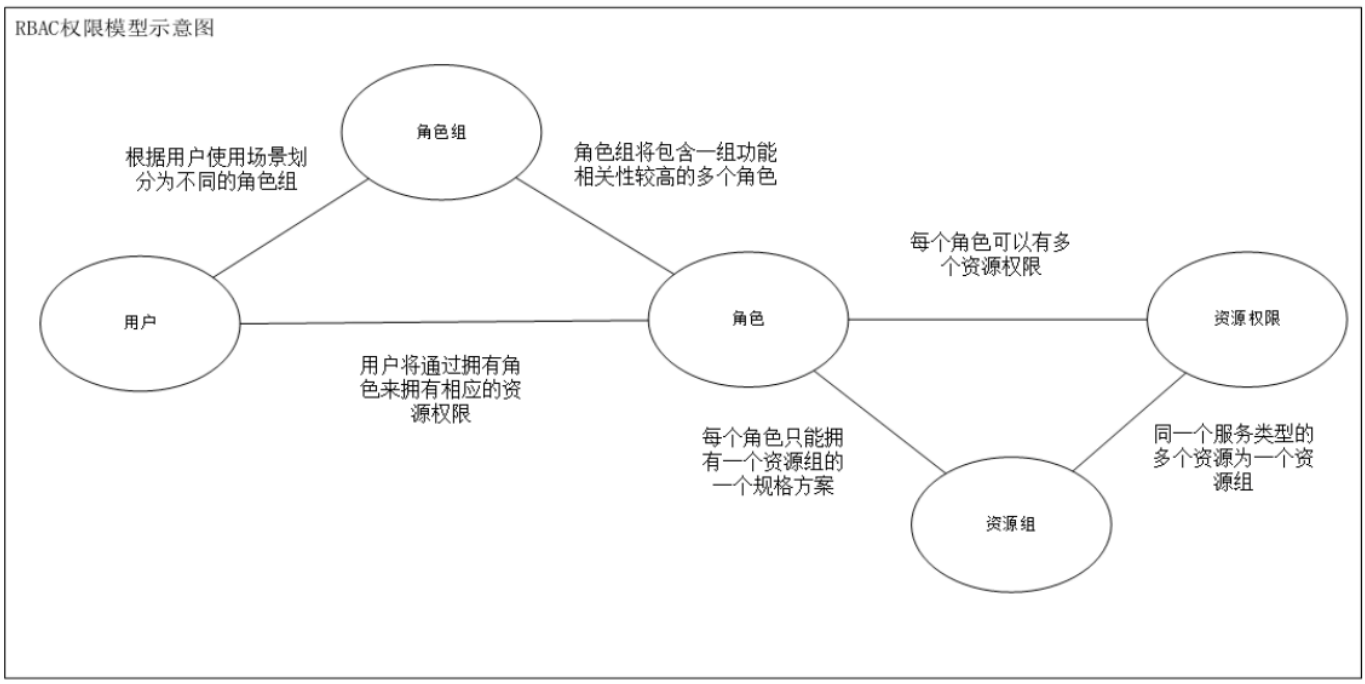
1. 根据指定 Ioc 查询指定时间内与之关联的 Report，并按照时间排序
2. 云瞻首页查询 Ioc 同时显示相关信息
3. Ioc、漏洞、标签相关数据，与目前云瞻已有的库进行关联，或者数据导入

运维管理

用户管理

用户管理主要是对用户权限的管理，什么样的用户应该分配什么样的权限，这是一个重要且需要思考的问题。此框架采用RBAC（Role-Based Access Control，基于角色的访问控制）模型，有着简化权限管理，灵活的角色与权限关系，提高安全性，易于扩展和维护的好处。

采用的RBAC模型图如下所示：



概念描述如下：

- **资源**：用户可操作的所有接口都认为是资源
- **角色**：根据不同的操作对象类型，设置不同角色。不同的角色可以访问特定类型的资源，每个用户可以拥有多个角色。
- **角色分组**：角色组用来管理实现同一组功能的角色。每个角色组中的角色定义是互斥的，即用户只能被授予同一个角色组中的至多一个角色。
- **资源组**：一组特定的能够实现完整功能的资源及其规格组合构成一个资源组，不同资源组之间的资源不存在交集。当为角色绑定资源时，每个角色可以配置多个资源组，但只能配置一个资源组中一个资源配置方案。

用户管理

管理员可以查看用户信息、为用户进行授权、设置授权时间。并且仅能通过赋予角色的方式对用户进行服务权限的授予，如需要为特定用户添加定制化的资源获取权限，需要首先由管理员设置新的角色，然后再为用户绑定该角色来实现。

角色授权限制：

1. 角色授权操作仅对管理员及TAC人员开放；
2. 为用户赋予任意两个角色不能属于相同角色组。

角色管理

角色管理功能允许管理员在设置角色时设置角色名称，选定角色组，添加零个或多个资源方案，然后添加描述信息。不同的角色组拥有不同的资源方案，每个角色只能属于一个角色组，所以当选定了角色组之后，角色所拥有的资源方案也就确定了。

当前角色组有`USER`和`OPERATOR`两个，它们对应的资源方案如下表所示：

USER	OPERATOR
api资源方案	sandbox资源方案
web资源方案	isource资源方案
web-unlimited资源方案	management资源方案
reputation资源方案	av资源方案
quota资源方案	c2library资源方案
iot资源方案	extra-library资源方案
	ipr-library资源方案
	vulnerability资源方案
	attck-library资源方案
	dga-model资源方案
	vulnerability-library资源方案

对于新用户，设置默认`ROLE_DEFAULT`角色，以保证注册用户可以直接使用最低级别的查询服务；如需删除角色，需要先解除所有与该角色相关的用户绑定。

资源配置管理

在设置资源配置方案时，需要指定资源配置方案名称，配置资源规格，并指定所属资源组，在指定所属资源组后，页面上的`URL`部分将只会显示当前资源组中的资源（`URL`接口），然后选择其中的资源即可。

目前为止，资源组一共有`6`个，分别是`api`，`web`，`web-unlimited`，`reputation`，`quota`，`iot`，一个资源（接口）只属于一个资源组，当资源组选定之后，能选择的资源范围也就确定了。

规格管理

管理员可通过规格管理来查看和设置规格。访问规格要求如下：

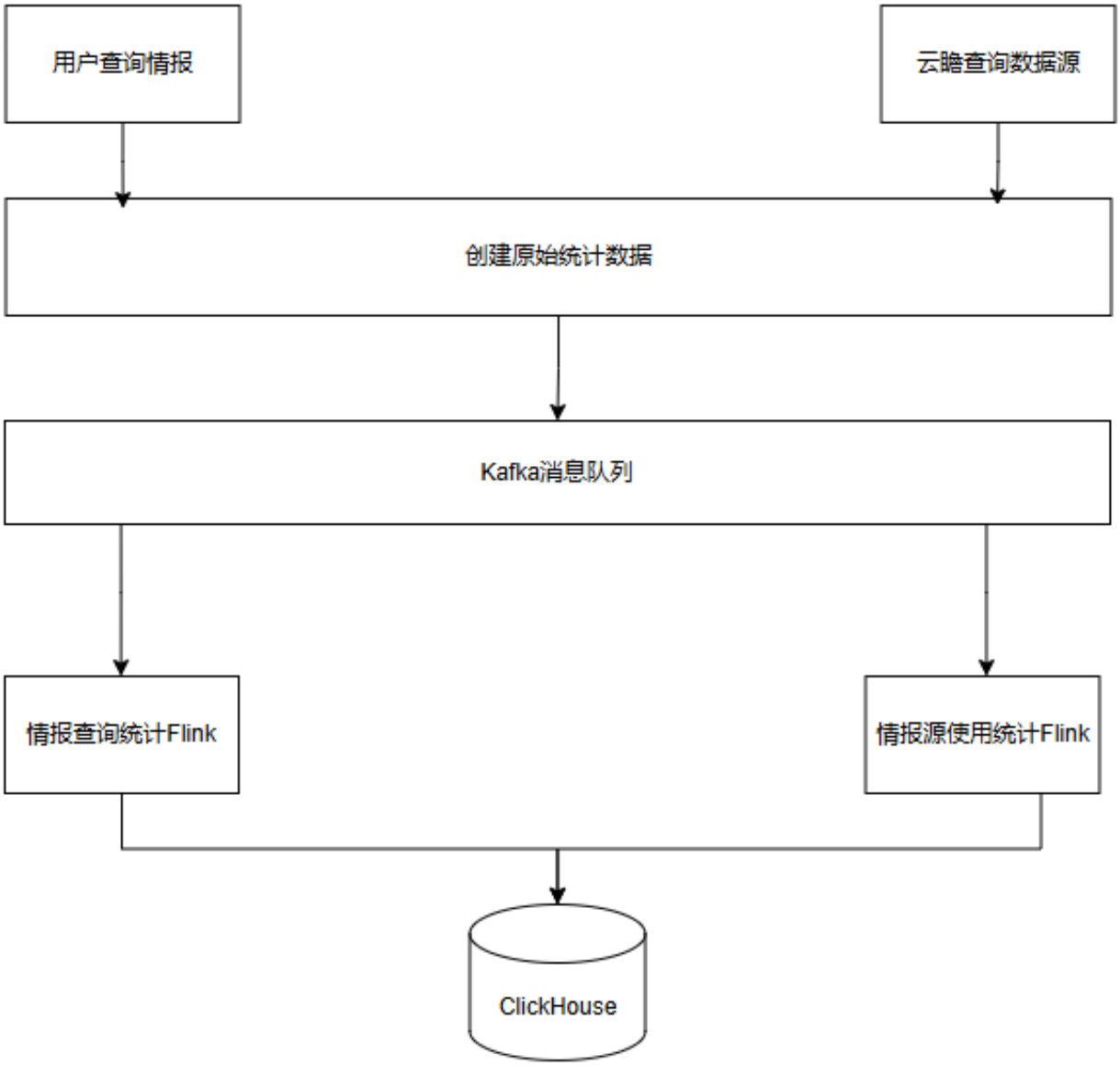
规格名称	授权配额/天	频率限制	说明
API体验版	1.5K	6/分钟	用户默认访问规格
在线UI规格	1K	6/分钟	在线页面访问限制
企业付费版-低端	20K	无	低端版本
企业付费版-中端	50K	无	中端版本
企业付费版-高端	100K	无	高端版本

情报运营数据统计

在威胁情报的运行当中，会涉及到许多的情报的生产、情报源的接入以及情报的查询，为更好让运营感知到整个平台的运行状态，云瞻会在后台对情报的查询情况、情报源的使用情况以及情报的生产情况进行统计，并在后台页面中以图表方式显示。

情报源使用情况以及情报查询统计

情报源的使用情况统计以及情报的查询统计数据都使用了如下的流程进行处理，各个模块在每个使用的节点进行埋点然后向kafka发送数据，之后使用flink进行处理，最后以批量形式存入ClickHouse数据库。



情报源使用情况统计

商用的第三方情报源（奇虎、VT、腾讯）通常会有每天查询次数的限制，云瞻对其在以下几个方面进行了统计

- 各情报源的查询次数
- 各个查询方式（主动老化、被动老化、直接查询、设备定时上送等方式）所消耗的查询次数比例
- 每个情报类型（IP、DOMAIN、URL以及FILE）所消耗的查询次数
- 查询结果的统计，包括命中数量、未命中数量、超时数量、错误数量以及超限数量
  可以在后台页面中**情报服务**页面查看统计情况。

用户查询情报状况统计

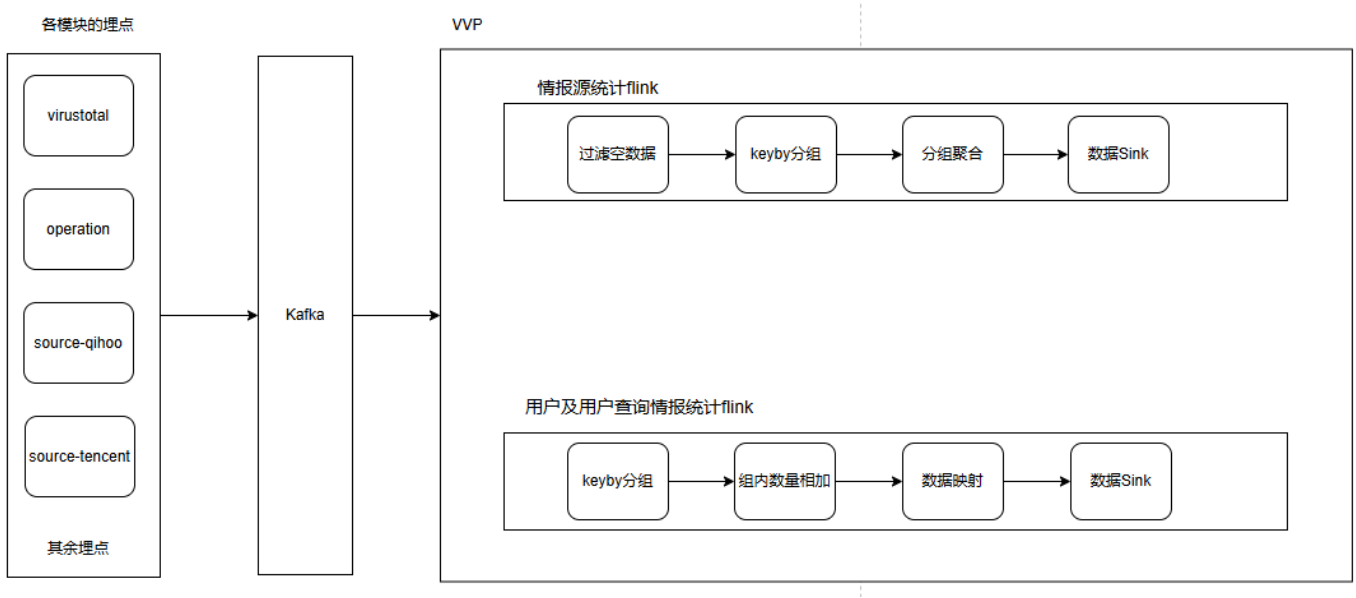
情报的查询统计指的是用户的查询信息统计，其中包括：

- 用户的查询详情
- 用户的查询次数以及其查询类型分布
- 各情报类型中查询最频繁的几个情报
- 查询接口的调用分布

- 设备的查询统计
  可以在后台页面中**情报源统计**-->**具体情报源页面**查看统计情况。

统计Flink运行流程

威胁情报平台会在各个模块中使用到情报源查询以及用户查询数据的地方进行埋点，创建原始的统计记录数据并将其发送到kafka，同时云瞻会在vvp建立两个flink任务，用于处理之后的流程。两个统计Flink任务的部署如下：



数据存储

考虑到表中有大量的列，且每次查询都要做一次少数列的聚合，云瞻使用ClickHouse来进行存储，来保持查询速度。以统计用户查询情报的数据表为例，其字段如下：

字段	备注
sn	设备sn
ip_upload_count	定时上送ip数量
domain_upload_count	定时上送domain数量
url_upload_count	定时上送url数量
file_upload_count	定时上送file数量
total_count	定时上送总数量
de_ip_upload_count	定时上送去重后ip数量
de_url_upload_count	定时上送去重后domain数量
de_domain_upload_count	定时上送去重后url数量
de_file_upload_count	定时上送去重后file数量
de_total_count	定时上送去重后总数量
u_ip_upload_count	UI上送ip数量
u_domain_upload_count	UI上送domain数量
u_url_upload_count	UI上送url数量
u_file_upload_count	UI上送file数量
u_total_count	UI上送总数量
a_ip_upload_count	管理员上送ip数量
a_domain_upload_count	管理员上送domain数量
a_url_upload_count	管理员上送url数量
a_file_upload_count	管理员上送file数量
a_total_count	管理员上送总数量
time	

情报的生产统计

威胁情报平台对情报进行了以下几个层面的统计：

- 各情报类型中空情报与非空情报的空数量
- 各个判定结果（黑、白、灰、未知）在所有情报中的分布状况
- 情报的分类标签分布占比

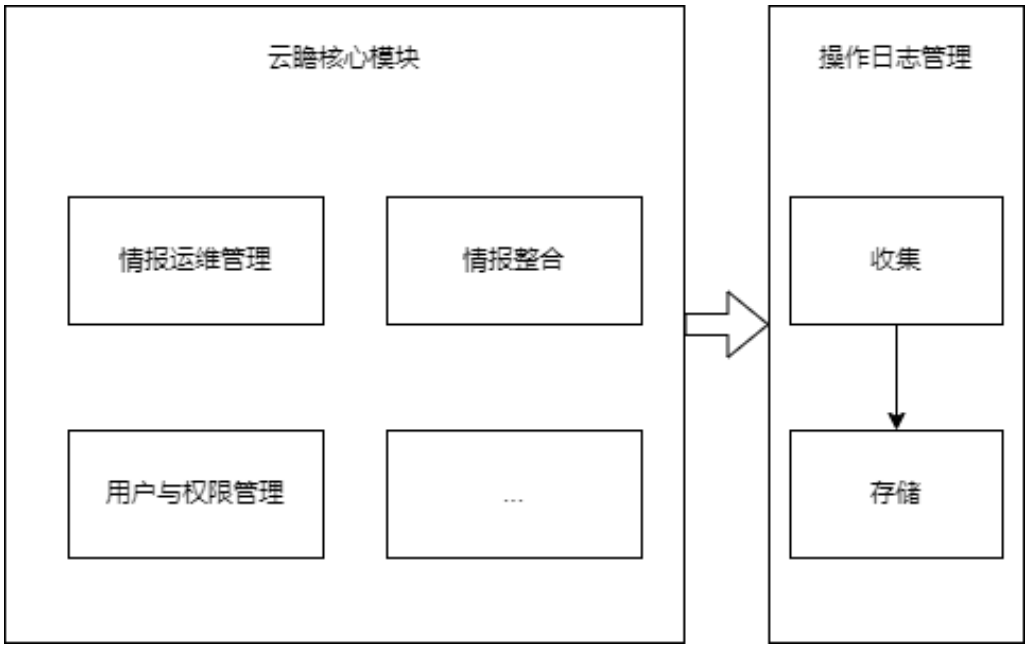
威胁平台的情报库采用Arango进行存储，由于性能原因，以上查询在Arango的查询速度较慢，为使前端能够迅速返回结果并且减小Arango压力，云瞻采用定时任务的方式每两小时去查询一次Arango，并将查询结果存放在Redis中，后台页面会首先去查询Redis，只有在Redis中没有数据时，才会去查询Arango。

操作日志管理

概要描述

操作日志（Operation Log）是记录系统或程序在运行过程中所发生事件和操作的日志信息，该信息持续留存。对于云瞻来说，部分业务操作，需要有明确的操作记录留存，便于后续问题诊断与故障排除，同时可以对系统状况进行监控，回溯问题现场。对于一些风险较高的管理操作，尤其是在涉及敏感数据和系统权限的情况下，可以根据操作日志进行用户行为追踪。

功能设计



云瞻使用云平台公共的操作日志中间件，在需要记录的业务逻辑处，增加相关配置。操作日志存储进Elasticsearch中，并在云瞻管理平台界面提供操作日志查询。